aws

Guia do usuário

Amazon Relational Database Service



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

Amazon Relational Database Service: Guia do usuário

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

As marcas comerciais e imagens comerciais da Amazon não podem ser usadas no contexto de nenhum produto ou serviço que não seja da Amazon, nem de qualquer maneira que possa gerar confusão entre os clientes ou que deprecie ou desprestigie a Amazon. Todas as outras marcas comerciais que não pertencem à Amazon pertencem a seus respectivos proprietários, que podem ou não ser afiliados, conectados ou patrocinados pela Amazon.

Table of Contents

O que é o Amazon RDS? 1
Vantagens do Amazon RDS 1
Comparação de responsabilidades 2
Modelo de responsabilidade compartilhada do Amazon RDS 3
Instâncias de banco de dados 4
Arquitetura da aplicação do Amazon RDS: exemplo 5
Mecanismos de banco de dados 6
Classes da instância de banco de dados 7
Armazenamento da instância de banco de dados 8
As instâncias de banco de dados em uma Amazon Virtual Private Cloud (Amazon VPC) 9
Regiões da AWS e zonas de disponibilidade 9
Zonas de disponibilidade 10
Implantações multi-AZ 10
Controle de acesso com grupos de segurança. 12
Monitoramento do Amazon RDS 14
Interfaces de usuário para o Amazon RDS 15
AWS Management Console 15
Interface da linha de comando 16
APIs do Amazon RDS 16
Como você é cobrado pelo Amazon RDS 17
Nível gratuito da AWS no Amazon RDS 17
Próximas etapas 18
Conceitos básicos 18
Tópicos específicos para mecanismos de banco de dados 18
Instâncias de banco de dados 19
Classes da instância de banco de dados 22
Tipos de classe de instância de banco de dados 22
Mecanismos de banco de dados compatíveis 32
Determinar o suporte para classes de instância de bancos de dados nas Regiões da AWS 90
Alterar a classe de instância de banco de dados 97
Configurar o processador no RDS para Oracle 97
Especificações de hardware 121
Armazenamento da instância de banco de dados 169

Tipos de armazenamento ... 169
Armazenamento de IOPS provisionadas ... 171
Armazenamento de uso geral ... 176
Comparar tipos de armazenamento SSD ... 180
Armazenamento magnético (herdado, não recomendado) ... 184
Volume de log dedicado (DLV) ... 184
Monitoramento da performance de armazenamento ... 185
Fatores que afetam a performance de armazenamento ... 186
Regiões, zonas de disponibilidade e zonas locais ... 190
AWSRegiões de ... 191
Zonas de disponibilidade ... 196
Zonas Locais ... 197
Recursos do Amazon RDS compatíveis por região e mecanismo ... 199
Convenções de tabela ... 200
Referência rápida do recurso ... 200
Implantações azul/verde ... 203
Backups automatizados entre regiões ... 204
Réplicas de leitura entre regiões ... 206
Fluxos de atividades do banco de dados ... 208
Modo de pilha dupla ... 216
Exportar snapshots para o S3 ... 240
Autenticação do banco de dados do IAM ... 255
Autenticação de Kerberos ... 260
Clusters de banco de dados multi-AZ ... 275
Insights de Performance ... 285
RDS Custom ... 286
Amazon RDS Proxy ... 300
Integração do Secrets Manager ... 319
Integrações ETL zero ... 320
Atributos nativos do mecanismo ... 326
Faturamento de instâncias de banco de dados para o Amazon RDS ... 327
Instâncias de banco de dados sob demanda ... 329
Instâncias de bancos de dados reservadas ... 331
Configuração ... 346
Inscrever-se para uma Conta da AWS ... 346
Criar um usuário com acesso administrativo ... 347

Conceder acesso programático ... 348
Determinar requisitos ... 350
Fornecer acesso à sua instância de banco de dados ... 353
Conceitos básicos ... 356
Criar e conectar uma instância de banco de dados MariaDB ... 357
Pré-requisitos ... 358
Criar uma instância do EC2 ... 359
Criar uma instância de banco de dados do MariaDB ... 365
(Opcional) Criar VPC, instância do EC2 e instância do MariaDB usando o AWS CloudFormation ... 370
Conectar-se a uma instância de banco de dados MariaDB ... 372
Excluir a instância do EC2 e a instância de banco de dados ... 376
(Opcional) Excluir a instância do EC2 e a instância de banco de dados criada com o CloudFormation ... 376
(Opcional) Conectar a instância de banco de dados a uma função do Lambda ... 377
Criar uma instância de banco de dados do Microsoft SQL Server e conectar-se a ela ... 378
Pré-requisitos ... 379
Criar uma instância do EC2 ... 380
Criar uma instância de banco de dados do SQL Server ... 385
(Opcional) Criar VPC, instância do EC2 e instância do SQL Server usando o AWS CloudFormation ... 391
Conectar-se à instância de banco de dados do SQL Server ... 393
Explorar sua instância de banco de dados de exemplo ... 396
Excluir a instância do EC2 e a instância de banco de dados ... 397
(Opcional) Excluir a instância do EC2 e a instância de banco de dados criada com o CloudFormation ... 398
(Opcional) Conectar a instância de banco de dados a uma função do Lambda ... 399
Criar e conectar a uma instância de banco de dados MySQL ... 400
Pré-requisitos ... 401
Criar uma instância do EC2 ... 402
Criar uma instância de banco de dados do MySQL ... 408
(Opcional) Criar VPC, instância do EC2 e instância do MySQL usando o AWS CloudFormation ... 414
Conectar-se a uma instância de banco de dados MySQL ... 416
Excluir a instância do EC2 e a instância de banco de dados ... 420

(Opcional) Excluir a instância do EC2 e a instância de banco de dados criada com o CloudFormation ... 421
(Opcional) Conectar a instância de banco de dados a uma função do Lambda ... 422
Criar uma instância de banco de dados Oracle e conectar-se a ela ... 423
Pré-requisitos ... 424
Etapa 1: Criar uma instância do EC2 ... 425
Etapa 2: Criar uma instância de banco de dados Oracle ... 431
(Opcional) Criar VPC, instância do EC2 e instância de banco de dados do Oracle usando o AWS CloudFormation ... 436
Etapa 3: Conectar seu cliente SQL a uma instância de banco de dados Oracle. ... 438
Etapa 4: Excluir a instância do EC2 e a instância de banco de dados ... 442
(Opcional) Excluir a instância do EC2 e a instância de banco de dados criada com o CloudFormation ... 443
(Opcional) Conectar a instância de banco de dados a uma função do Lambda ... 443
Criar uma instância de banco de dados PostgreSQL e conectar-se a ela ... 444
Pré-requisitos ... 445
Criar uma instância do EC2 ... 446
Criar uma instância de banco de dados do PostgreSQL ... 452
(Opcional) Criar VPC, instância do EC2 e instância do PostgreSQL usando o AWS CloudFormation ... 457
Conectar-se à instância de banco de dados PostgreSQL ... 459
Excluir a instância do EC2 e a instância de banco de dados ... 463
(Opcional) Excluir a instância do EC2 e a instância de banco de dados criada com o CloudFormation ... 464
(Opcional) Conectar a instância de banco de dados a uma função do Lambda ... 465
Tutorial: crie um servidor Web e uma instância de banco de dados do Amazon RDS ... 466
Iniciar uma instância do EC2 para se conectar à instância de banco de dados ... 468
Criar uma instância de banco de dados ... 474
Instalar um servidor Web ... 495
Tutorial: Criar uma função do Lambda para acessar a instância de banco de dados do Amazon RDS ... 507
Pré-requisitos ... 508
Criar uma instância de banco de dados do Amazon RDS ... 508
Criar a função do Lambda e o proxy ... 510
Criar um perfil de execução de função ... 511
Criar o pacote de implantação do Lambda ... 512

Atualizar a função do Lambda ... 515
Teste sua função do Lambda no console ... 516
Criar uma fila do Amazon SQS ... 517
Crie um mapeamento da origem do evento para invocar sua função do Lambda ... 518
Teste e monitore sua configuração ... 519
Limpe os recursos ... 520
Tutoriais e código de exemplo ... 522
Tutoriais neste guia ... 522
Tutoriais em outros guias daAWS ... 523
Portal de conteúdo de workshops e laboratório da AWS para Amazon RDS PostgreSQL ... 524
Portal de conteúdo de workshops e laboratório da AWS para Amazon RDS MySQL ... 525
Tutoriais e código de exemplo no GitHub ... 525
Como trabalhar com AWS SDKs ... 525
Práticas recomendadas do Amazon RDS ... 527
Diretrizes operacionais básicas do Amazon RDS ... 527
Recomendações de RAM para a instância de banco de dados ... 529
Mantenha as versões do mecanismo de banco de dados atualizadas ... 529
Drivers de banco de dados da AWS ... 530
Usar o Monitoramento avançado ... 530
Uso de métricas do ... 530
Visualização de métricas de performance ... 531
Avaliação de métricas de performance ... 534
Ajuste das consultas ... 536
Práticas recomendadas para trabalhar com o MySQL ... 537
Tamanho da tabela ... 537
Número de tabelas ... 538
Mecanismo de armazenamento ... 539
Práticas recomendadas para trabalhar com o MariaDB ... 540
Tamanho da tabela ... 540
Número de tabelas ... 541
Mecanismo de armazenamento ... 541
Práticas recomendadas para trabalhar com o Oracle ... 542
Práticas recomendadas para trabalhar com PostgreSQL ... 542
Carregamento de dados em uma instância de banco de dados PostgreSQL ... 542
Trabalhar com o recurso autovacuum do PostgreSQL ... 543
Amazon RDS for PostgreSQL vídeo de práticas recomendadas ... 545

Práticas recomendadas para trabalhar com o SQL Server ... 545
Amazon RDS for SQL Server vídeo de práticas recomendadas ... 546
Trabalhar com grupos de parâmetros de banco de dados ... 546
Automatização da criação de instâncias de banco de dados ... 547
Vídeo de novos recursos ... 548
Acesso programático ao Amazon RDS ... 549
Console-to-Code ... 550
Configurar uma instância de banco de dados ... 551
Criar uma instância de banco de dados ... 552
Pré-requisitos ... 552
Criar uma instância de banco de dados ... 560
Configurações disponíveis ... 568
Criar recursos com o AWS CloudFormation ... 613
RDS e modelos do AWS CloudFormation ... 613
Saiba mais sobre o AWS CloudFormation ... 613
Conexão a uma instância de banco de dados ... 614
Insira as informações de conexão ... 614
Cenários para acessar uma instância de banco de dados ... 618
Conectar-se a instâncias de banco de dados com os drivers da AWS ... 619
Conectar a uma instância de banco de dados executando um mecanismo de banco de dados específico ... 621
Gerenciar conexões com o RDS Proxy ... 621
Opções de autenticação do banco de dados ... 621
Conexões criptografadas ... 622
Trabalhar com grupos de opções ... 623
Visão geral de grupos de opções ... 623
Criar um grupo de opções ... 626
Copiar um grupo de opções ... 628
Adicionar uma opção a um grupo de opções ... 629
Listar as opções e as configurações de opções para um grupo de opções ... 636
Modificar uma configuração de opção ... 637
Remover uma opção de um grupo de opções ... 641
Excluir um grupo de opções ... 643
Grupos de parâmetros ... 647
Visão geral dos grupos de parâmetros ... 647
Grupos de parâmetros do banco de dados ... 652

Grupos de parâmetros de cluster de banco de dados ... 670
Comparação de grupos de parâmetros de banco de dados ... 684
Especificação de parâmetros de banco de dados ... 685
Criar um cache do ElastiCache pelo Amazon RDS ... 693
Visão geral da criação do cache do ElastiCache com as configurações da instância de banco de dados do RDS ... 693
Criar um cache do ElastiCache com as configurações de uma instância de banco de dados do RDS ... 695
Migração automática de bancos de dados do EC2 ... 698
Visão geral ... 698
Pré-requisitos ... 699
Limitações ... 700
Criar recursos do IAM ... 701
Configurar a migração de dados ... 709
Gerenciamento de migrações ... 711
Monitoramento ... 713
Tutorial: criar uma instância de banco de dados do MySQL com um grupo de parâmetros personalizado e um novo grupo de opções ... 715
Pré-requisitos ... 716
Criar um grupo de parâmetros personalizado ... 716
Modificar os valores dos parâmetros ... 717
Criar um grupo de opções ... 718
Adicionar opção ... 718
Criar uma instância de banco de dados ... 719
Gerenciar uma instância de banco de dados ... 721
Interromper uma instância de banco de dados ... 722
Casos de uso ... 723
Regiões, classes e mecanismos de banco de dados compatíveis ... 724
Suporte a multi-AZ ... 724
Como funciona ... 725
Limitações ... 726
Grupos de parâmetros e de opções ... 727
Endereços IP públicos ... 727
Interromper uma instância de banco de dados ... 727
Iniciar uma instância de banco de dados ... 729
Reinicializar uma instância de banco de dados ... 731

Casos de uso para reinicializar uma instância de banco de dados ... 731
Como a reinicialização funciona ... 732
Reinicialização em multi-AZ ... 733
Considerações ... 733
Pré-requisitos ... 734
Reinicializar uma instância de banco de dados: etapas básicas ... 734
Conectar uma instância do EC2 ... 737
Visão geral ... 738
Conectar uma instância do EC2 ... 743
Visualizar recursos computacionais conectados ... 746
Conectar a uma instância de banco de dados executando um mecanismo de banco de dados específico ... 747
Conectar uma função do Lambda ... 748
Visão geral ... 750
Conectar uma função do Lambda ... 761
Visualizar recursos computacionais conectados ... 763
Modificação de uma instância de banco de dados ... 765
Programação de modificações ... 767
Configurações disponíveis ... 768
Manutenção de uma instância de banco de dados ... 816
Visão geral das atualizações de manutenção da instância de banco de dados ... 816
Visualização de manutenção pendente ... 818
Aplicação de atualizações ... 821
Manutenção para implantações multi-AZ ... 824
A janela de manutenção do ... 825
Ajustar a janela de manutenção de uma instância de banco de dados ... 827
Atualizações do sistema operacional ... 829
Atualizar a versão do mecanismo ... 834
Atualizar manualmente a versão do mecanismo ... 835
Atualizar automaticamente a versão do mecanismo espelho ... 837
Renomear uma instância de banco de dados ... 843
Renomear para substituir uma instância de banco de dados existente ... 844
Trabalhar com réplicas de leitura de instância de banco de dados ... 847
Visão geral ... 849
Diferenças entre mecanismos de banco de dados ... 853
Como criar uma réplica de leitura ... 857

Promover uma réplica de leitura ... 861
Monitoramento da replicação de leitura ... 867
Réplicas de leitura entre regiões ... 870
Marcar recursos do RDS ... 885
Por que usar tags do RDS? ... 886
Como as tags do RDS funcionam ... 887
Práticas recomendadas ... 890
Copiar tags para snapshots de banco de dados ... 891
Adicionar e excluir tags no Amazon RDS ... 892
Tutorial: especificar quais instâncias de banco de dados devem ser interrompidas usando tags ... 896
ARNs no Amazon RDS ... 901
Criação de um ARN ... 901
Obter um ARN existente ... 909
Trabalhar com armazenamento ... 912
Aumento da capacidade de armazenamento da instância de banco de dados ... 912
Gerenciar a capacidade automaticamente com a escalabilidade automática de armazenamento do ... 915
Atualizando o sistema de arquivos de armazenamento ... 924
Modificar as configurações de IOPS provisionadas ... 926
Alterações de armazenamento com alto consumo de E/S ... 929
Modificar configurações de uso geral (gp3) ... 930
Usando um volume de registro dedicado (DLV) ... 933
Excluir uma instância de banco de dados ... 940
Pré-requisitos para excluir uma instância de banco de dados ... 940
Considerações ao excluir uma instância de banco de dados ... 940
Excluir uma instância de banco de dados ... 942
Tutorial: Gerenciar uma instância de banco de dados do MySQL ... 945
Introdução ... 945
Pré-requisitos ... 946
Adicionar tags à instância de banco de dados ... 946
Aumentar armazenamento de banco de dados ... 946
Criar réplicas de leitura ... 947
Atualizar tags ... 948
Configurar e gerenciar uma implantação multi-AZ ... 952
Implantações de instâncias de banco de dados multi-AZ ... 954

Converter uma instância de banco de dados em uma implantação multi-AZ 956
Fazer failover de uma instância de banco de dados multi-AZ 958
Implantações de clusters de banco de dados multi-AZ 964
Disponibilidade de classe de instância para clusters de banco de dados multi-AZ 965
Arquitetura de cluster de banco de dados multi-AZ 966
Grupos de parâmetros para clusters de banco de dados multi-AZ 968
RDS Proxy com clusters de banco de dados multi-AZ 968
Atraso de réplica e clusters de banco de dados multi-AZ 968
Snapshots de cluster de banco de dados multi-AZ 971
Criar um cluster de banco de dados multi-AZ 973
Conectar a um cluster de banco de dados multi-AZ 1006
Conectar um recurso de computação da AWS e um cluster de banco de dados multi-AZ .. 1013
Modificar um cluster de banco de dados multi-AZ 1042
Atualização de um cluster de banco de dados multi-AZ 1066
Renomear um cluster de banco de dados multi-AZ 1069
Reinicializar um cluster de banco de dados multi-AZ 1072
Fazer o failover de um cluster de banco de dados multi-AZ 1074
Replicação lógica do PostgreSQL com clusters de banco de dados multi-AZ 1078
Trabalhar com réplicas de leitura de cluster de banco de dados multi-AZ 1083
Configurar a replicação externa de clusters de banco de dados multi-AZ 1096
Excluir um cluster de banco de dados multi-AZ 1098
Limitações de clusters de banco de dados multi-AZ 1101
Suporte estendido do RDS 1103
Visão geral do Suporte estendido do RDS 1104
Cobranças do Suporte estendido do RDS 1105
Evitar cobranças do Suporte estendido do RDS 1106
Versões com o Suporte estendido do RDS 1106
Nomenclatura da versão do Suporte estendido do RDS 1106
Responsabilidades com o Suporte estendido do RDS 1108
Responsabilidades do Amazon RDS 1108
Suas responsabilidades 1108
Criação de uma instância de banco de dados ou um cluster de banco de dados multi-AZ 1109
Comportamento do Suporte estendido do RDS 1109
Considerações para o Suporte estendido do RDS 1110
Criar uma instância de banco de dados ou um cluster de banco de dados multi-AZ com o Suporte estendido do RDS 1111

Visualizar a inscrição no Suporte estendido do RDS 1112
Visualizar datas de suporte 1115
Restauração de uma instância de banco de dados ou um cluster de banco de dados multi-AZ 1118
Comportamento do Suporte estendido do RDS 1119
Considerações para o Suporte estendido do RDS 1119
Restaurar uma instância de banco de dados ou um cluster de banco de dados multi-AZ com o Suporte estendido do RDS 1120
Usar implantações azul/verde para atualizações de banco de dados 1123
Visão geral das implantações azul/verde do Amazon RDS 1124
Disponibilidade de região e versão 1125
Benefícios 1125
Fluxo de trabalho 1126
Autorizar acesso 1130
Limitações e considerações 1131
Práticas recomendadas 1139
Criar uma implantação azul/verde 1144
Preparação para uma implantação azul/verde 1145
Especificando alterações 1147
Carregamento lento e inicialização do armazenamento 1150
Criar uma implantação azul/verde 1152
Configurações disponíveis 1154
Visualizar uma implantação azul/verde 1160
Alternar uma implantação azul/verde 1164
Tempo limite de transição 1165
Barreiras de proteção de transição 1165
Ações de transição 1166
Práticas recomendadas de transição 1167
Verificar as métricas do CloudWatch antes da transição 1169
Monitorar o atraso da réplica antes da transição 1169
Realizar a transição de uma implantação azul/verde 1170
Após a transição 1173
Excluir uma implantação azul/verde 1175
Backup, restauração e exportação de dados 1178
Introdução aos backups 1179
Armazenamento de backup 1179

Gerenciar backups automatizados 1181
Janela de backup 1181
Backup retention period (Período de retenção de backup) 1184
Ativar backups automáticos 1185
Reter backups automatizados 1187
Excluir backups automatizados retidos 1190
Mecanismos de armazenamento MySQL sem suporte 1193
Mecanismos de armazenamento MariaDB sem suporte 1194
Backups automatizados entre regiões 1196
Gerenciar backups manuais 1213
Criar um snapshot de banco de dados para uma instância de banco de dados de uma única zona de disponibilidade 1214
Criar um snapshot de cluster de banco de dados Multi-AZ 1217
Excluir um snapshot de banco de dados 1219
Restaurar uma instância de banco de dados 1222
Restauração a partir de um snapshot 1223
Considerações 1225
Recuperação para um ponto no tempo 1230
Restaurar um cluster de banco de dados multi-AZ para um horário especificado 1236
Restaurar a partir de um snapshot para cluster de banco de dados multi-AZ 1240
Restauração de um snapshot de cluster de banco de dados multi-AZ para uma instância de banco de dados 1244
Tutorial: restaurar uma instância de banco de dados de um snapshot de banco de dados . 1247
Copiar um snapshot de banco de dados 1252
Copiar um snapshot de banco de dados 1252
Limitações 1263
Considerações 1264
Compartilhar um snapshot do de banco de dados 1274
Compartilhar um snapshot 1276
Compartilhamento de snapshots públicos 1279
Compartilhamento de snapshots criptografados 1282
Interromper o compartilhamento de snapshots 1286
Exportação de dados de snapshot de banco de dados para o Amazon S3 1288
Visão geral da exportação de dados de snapshot 1289
Configurar o acesso a um bucket do S3 1290
exportar um snapshot de banco de dados 1296

Disponibilidade de região e versão ... 1299
Limitações ... 1300
Monitorar exportações de snapshots ... 1301
Cancelar uma exportação de snapshot ... 1304
Mensagens de falha ... 1305
Solucionar problemas de erros de permissões do PostgreSQL ... 1307
Convenções de nomenclatura de arquivos ... 1307
Conversão de dados ... 1309
Usar o AWS Backup ... 1319
Métricas de monitoramento em uma instância de banco de dados ... 1320
Plano de monitoramento ... 1320
Linha de base de performance ... 1321
Orientações de performance ... 1321
Ferramentas de monitoramento ... 1323
Ferramentas de monitoramento automatizadas ... 1323
Ferramentas de monitoramento manual ... 1325
Visualizar o status do cluster ... 1327
Visualizar o status de uma instância de banco de dados Amazon RDS em um ... 1328
Recomendações do Amazon RDS ... 1336
Visualização de recomendações ... 1338
Aplicar recomendações ... 1346
Ignorar recomendações ... 1352
Modificar recomendações ignoradas para ativas ... 1354
Referência de recomendações ... 1355
Visualizar métricas no console do Amazon RDS ... 1390
Visualizar o painel do Insights de Performance ... 1391
Escolher a nova visualização de monitoramento na guia Monitoramento ... 1392
Escolher a nova visualização de monitoramento na página do Insights de Performance ... 1394
Criar um painel personalizado ... 1396
Escolher o painel predefinido ... 1399
Monitorando RDS com CloudWatch ... 1401
Visão geral do Amazon RDS e do Amazon CloudWatch ... 1402
Visualizar métricas do CloudWatch do ... 1404
Exportar as métricas do Performance Insights para o CloudWatch ... 1410
Criar alarmes do CloudWatch ... 1416

Tutorial: criar um alarme do CloudWatch para atraso de réplica de cluster de banco de dados ... 1416
Monitoramento com o Database Insights ... 1424
Preços ... 1424
Suporte para mecanismo, região e classe de instância ... 1424
Ativação do modo Avançado ... 1430
Ativação do modo Standard ... 1434
Monitorar consultas lentas ... 1440
Considerações ... 1442
Monitoramento de carga de banco de dados com o Performance Insights ... 1443
Visão geral do Performance Insights ... 1444
Ativar e desativar o Performance Insights ... 1460
Performance Schema para o MariaDB ou o MySQL ... 1465
Políticas do Performance Insights ... 1471
Análise de métricas usando o painel do Performance Insights ... 1484
Visualizar as recomendações proativas do Performance Insights ... 1534
Recuperar métricas com a API do Performance Insights ... 1537
Registrar em log as chamadas do Performance Insights usando o AWS CloudTrail ... 1562
Enpoints da VPC (AWS PrivateLink) ... 1564
Analisar a performance com o DevOps Guru para RDS ... 1568
Benefícios do DevOps Guru para RDS ... 1568
Como funciona o DevOps Guru para RDS ... 1570
Configurar o DevOps Guru para RDS ... 1571
Monitorar o SO com o monitoramento avançado ... 1580
Visão geral do monitoramento avançado ... 1580
Configurar e habilitar o monitoramento avançado ... 1582
Como visualizar métricas do SO no console do RDS ... 1588
Visualizar métricas do SO usando CloudWatch Logs ... 1592
Referência de métricas do RDS ... 1594
Métricas do CloudWatch para RDS ... 1594
Dimensões do CloudWatch para o RDS ... 1618
Métricas do CloudWatch para Performance Insights ... 1619
Métricas de contadores do Performance Insights ... 1622
Estatísticas SQL para Performance Insights ... 1658
Métricas do sistema operacional em monitoramento avançado ... 1675
Monitorar eventos, logs e transmissões de atividades de bancos de dados ... 1692

Visualizar logs, eventos e transmissões no console do Amazon RDS 1693
Monitorar eventos do RDS 1697
Visão geral dos eventos para Amazon RDS 1697
Visualizar eventos do Amazon RDS 1699
Trabalhar com a notificação de eventos do Amazon RDS 1703
Criar uma regra que é acionada em um evento do Amazon RDS 1729
Categorias de eventos e mensagens de eventos do Amazon RDS 1735
Monitorar logs do RDS 1788
Como visualizar e listar arquivos de log do banco de dados 1788
Como baixar um arquivo de log de banco de dados 1789
Como observar um arquivo de log de banco de dados 1791
Publicação no CloudWatch Logs 1793
Leitura do conteúdo de arquivos de log usando REST 1796
Arquivos de log do banco de dados do Db2 1798
Arquivos de log do banco de dados MariaDB 1803
Arquivos de log do banco de dados do Microsoft SQL Server 1819
Arquivos de log do banco de dados MySQL 1825
Arquivos de log do banco de dados Oracle 1843
Arquivos de log do banco de dados do PostgreSQL 1854
Monitorar chamadas de API do RDS no CloudTrail 1869
Integração do CloudTrail com o Amazon RDS 1869
Entradas do arquivo de log do Amazon RDS 1870
Monitorar o RDS com o recurso Database Activity Streams 1874
Visão geral 1874
Configurar a auditoria unificada do Oracle 1881
Configurar auditoria no SQL Server 1883
Iniciar um stream de atividade de banco de dados 1884
Modificar um fluxo de atividades de banco de dados 1887
Obter o status do fluxo de atividades 1890
Interromper um fluxo de atividade de banco de dados 1892
Monitorar os fluxos de atividades 1893
Exemplos de políticas do IAM para fluxos de atividades 1937
Monitorar ameaças com o GuardDuty RDS Protection 1940
Amazon RDS Custom 1942
Desafio da personalização do banco de dados 1942
Modelo de gerenciamento e benefícios do RDS Custom 1944

Modelo de responsabilidade compartilhada no RDS Custom 1945
Perímetro de suporte e configurações sem suporte no RDS Custom 1947
Principais benefícios do RDS Custom 1947
Arquitetura do RDS Custom 1948
VPC 1949
Automação e monitoramento do RDS Custom 1950
Amazon S3 1955
AWS CloudTrail 1956
Segurança do RDS Custom 1957
Como o RDS Custom gerencia com segurança as tarefas em seu nome 1957
Certificados SSL 1958
Proteger o bucket do Amazon S3 contra o problema do substituto confuso 1958
Alternar as credenciais do RDS Custom para Oracle 1959
Trabalhar com o RDS Custom for Oracle 1965
Fluxo de trabalho do RDS Custom for Oracle 1965
Arquitetura de banco de dados para o Amazon RDS Custom para Oracle 1971
Disponibilidade de recursos e suporte do RDS Custom para Oracle 1973
Requisitos e limitações do RDS Custom for Oracle 1976
Configurar seu ambiente RDS Custom for Oracle 1980
Trabalhar com CEVs do RDS Custom for Oracle 2001
Configurar uma instância de banco de dados do RDS Custom para Oracle 2033
Gerenciar uma instância de banco de dados do RDS Custom for Oracle 2054
Gerenciar uma implantação multi-AZ para o RDS Custom para Oracle 2073
Trabalhar com réplicas do RDS Custom para Oracle 2093
Fazer o backup e a restauração de uma instância de banco de dados do RDS Custom for Oracle 2104
Trabalhar com grupos de opções no RDS Custom para Oracle 2116
Migrar para o RDS Custom para Oracle 2125
Fazer upgrade de uma instância de banco de dados para o RDS Custom for Oracle 2126
Solucionar problemas do RDS Custom para Oracle 2140
Problemas conhecidos do RDS Custom para Oracle 2165
Trabalhar com o RDS Custom for SQL Server 2169
Fluxo de trabalho do RDS Custom for SQL Server 2169
Requisitos e limitações do RDS Custom for SQL Server 2172
Configurar seu ambiente do RDS Custom for SQL Server 2269
Modelo BYOM (traga sua própria mídia) com o RDS Custom para SQL Server 2295

Trabalhar com CEVs para RDS Custom para SQL Server ... 2297
Criar uma instância de banco de dados do RDS Custom for SQL Server e conectar-se a ela ... 2321
Gerenciar uma instância de banco de dados do RDS Custom for SQL Server ... 2334
Trabalhar com o Microsoft Active Directory e o RDS Custom para SQL Server ... 2350
Gerenciar uma implantação multi-AZ para o RDS Custom para SQL Server ... 2378
Fazer backup e restauração de uma instância de banco de dados do RDS Custom for SQL Server ... 2395
Copiar um snapshot de banco de dados do RDS Custom para SQL Server ... 2413
Migrar um banco de dados on-premises para o RDS Custom for SQL Server ... 2425
Fazer atualização de uma instância de banco de dados do RDS Custom para SQL Server 2429
Solucionar problemas do Amazon RDS Custom para SQL Server ... 2433
Amazon RDS em AWS Outposts ... 2476
Pré-requisitos ... 2477
Suporte para recursos do Amazon RDS ... 2478
Classes de instância de banco de dados compatíveis ... 2486
Endereços IP de propriedade do cliente ... 2489
Usar CoIPs ... 2489
Limitações ... 2491
Implantações multi-AZ ... 2493
Trabalhar com o modelo de responsabilidade compartilhada ... 2493
Melhorar a disponibilidade ... 2494
Pré-requisitos ... 2494
Trabalhar com operações de API para permissões do Amazon EC2 ... 2496
Criar a instância de banco de dados para RDS on Outposts ... 2497
Criar réplicas de leitura do RDS no Outposts ... 2508
Considerações para restaurar instâncias de banco de dados ... 2512
Amazon RDS Proxy ... 2513
Disponibilidade de região e versão ... 2514
Cotas e limitações ... 2514
Limitações do RDS para MariaDB ... 2516
Limitações do RDS para SQL Server ... 2517
Limitações do MySQL ... 2518
Limitações do PostgreSQL ... 2519
Planejar onde usar o RDS Proxy ... 2520
Conceitos e terminologia do RDS Proxy ... 2521

Visão geral dos conceitos do RDS Proxy ... 2522
Agrupamento de conexões ... 2523
Segurança ... 2524
Failover ... 2526
Transações ... 2527
Conceitos básicos do RDS Proxy ... 2528
Configurar uma rede de proxy ... 2528
Configurar de credenciais de banco de dados ... 2531
Configuração de políticas do IAM ... 2535
Criar um proxy ... 2538
Visualizar um proxy ... 2548
Conectar-se por meio do RDS Proxy ... 2550
Gerenciar um RDS Proxy ... 2554
Modificar um RDS Proxy ... 2554
Adicionar um usuário de banco de dados ... 2561
Considerações sobre a conexão do RDS Proxy ... 2563
Evitar a fixação de um RDS Proxy ... 2567
Excluir um RDS Proxy ... 2573
Como trabalhar com endpoints do RDS Proxy ... 2574
Visão geral dos endpoints de proxy ... 2575
Limitações de endpoints de proxy ... 2575
Endpoints de proxy para clusters de banco de dados multi-AZ ... 2576
Acesso aos bancos de dados do Aurora e do RDS entre VPCs ... 2578
Criação de um endpoint de proxy ... 2579
Visualização dos endpoints de proxy ... 2583
Modificação de um endpoint de proxy ... 2584
Exclusão de um endpoint de proxy ... 2585
Monitorar o proxy do RDS com o CloudWatch ... 2587
Trabalhar com eventos do RDS Proxy ... 2595
Eventos do RDS Proxy ... 2596
Solução de problemas do RDS Proxy ... 2599
Verificar a conectividade para um proxy ... 2599
Problemas e soluções comuns de ... 2601
Solução de problemas do RDS para MySQL ... 2602
Solução de problemas do RDS para PostgreSQL ... 2604
Usar o proxy do RDS com o AWS CloudFormation ... 2610

Integrações ETL zero ... 2612
Benefícios ... 2614
Principais conceitos ... 2614
Limitações ... 2615
Limitações gerais ... 2615
Limitações do RDS para MySQL ... 2616
Limitações do Amazon Redshift ... 2616
Limitações referentes a lakehouses do Amazon SageMaker ... 2617
Cotas ... 2617
Regiões do compatíveis ... 2617
Conceitos básicos das integrações ETL zero ... 2618
Etapa 1: Criar um grupo de parâmetros de de banco de dados personalizado ... 2618
Etapa 2: Selecionar ou criar um banco de dados de origem ... 2619
Etapa 3: Criar um data warehouse de destino do Amazon Redshift ... 2619
Próximas etapas ... 2621
Criar Integrações ETL zero com o Amazon Redshift ... 2621
Pré-requisitos ... 2622
Permissões obrigatórias ... 2622
Criar integrações ETL zero ... 2625
Criptografar integrações ... 2629
Próximas etapas ... 2631
Criar Integrações ETL zero com um lakehouse do Amazon SageMaker ... 2632
Pré-requisitos ... 2632
Permissões obrigatórias ... 2632
Criar Integrações ETL zero com um lakehouse do Amazon SageMaker ... 2635
Criptografar integrações ... 2640
Próximas etapas ... 2642
Filtragem de dados para integrações ETL zero ... 2642
Formato de um filtro de dados ... 2643
Lógica de filtros ... 2645
Precedência do filtro ... 2645
Exemplos ... 2646
Adicionar filtros de dados ... 2647
Remover filtros de dados ... 2649
Adicionar e consultar dados ... 2649
Criação de um banco de dados de destino no Amazon Redshift ... 2650

Adicionar dados ao banco de dados de origem ... 2650
Consultar os dados do Amazon RDS no Amazon Redshift ... 2651
Diferenças dos tipos de dados ... 2652
Visualizar e monitorar integrações ETL zero ... 2657
Visualizar integrações ... 2657
Monitorar usando tabelas do sistema ... 2659
Monitoramento com o EventBridge ... 2660
Modificar integrações ETL zero ... 2660
Excluir integrações ETL zero ... 2661
Solução de problemas em integrações ETL zero ... 2662
Não consigo criar uma integração ETL zero. ... 2663
Minha integração está travada em um estado de `Syncing`. ... 2664
Minhas tabelas não estão sendo replicadas para o Amazon Redshift. ... 2664
Uma ou mais das minhas tabelas do Amazon Redshift exigem ressincronização. ... 2664
Db2 no Amazon RDS ... 2669
Visão geral do Db2 ... 2670
Recursos do Db2 ... 2671
Versões do Db2 ... 2674
Licenciamento do Db2 ... 2679
Classes de instância do Db2 ... 2697
Perfis padrão do Db2 ... 2699
Parâmetros do Db2 ... 2700
Agrupamentos EBCDIC ... 2707
Fuso horário local do Db2 ... 2708
Pré-requisitos da instância de banco de dados ... 2711
Conta de administrador ... 2711
Considerações adicionais ... 2712
Vários bancos de dados do Db2 ... 2713
Conectar-se à instância de banco de dados do Db2 ... 2715
Encontrar o endpoint ... 2715
IBM Db2 CLP ... 2717
IBM CLPPlus ... 2722
DBeaver ... 2725
IBM Db2 Data Management Console ... 2729
Considerações sobre os grupos de segurança ... 2739
Proteger as conexões do Db2 ... 2740

Criptografia com SSL/TLS ... 2740
Utilizar a autenticação Kerberos ... 2747
Administrar a instância de banco de dados do RDS para Db2 ... 2763
Tarefas do sistema ... 2765
Tarefas de banco de dados ... 2778
Integração com o S3 ... 2802
Criar uma política do IAM ... 2802
Criar um perfil do IAM e associar a política do IAM ... 2805
Adicionar o perfil do IAM à instância de banco de dados ... 2807
Migrar dados para o RDS para Db2 ... 2810
Migrar dados com serviços da AWS ... 2810
Migrar dados com ferramentas nativas do Db2 ... 2822
Federação ... 2836
Federação homogênea ... 2836
Federação heterogênea ... 2841
Trabalhando com réplicas do Db2 ... 2846
Réplicas em espera ... 2846
Ativações do banco de dados ... 2846
Configurações de HADR ... 2847
Arquivar retenção de logs ... 2847
Interrupções durante a replicação do Db2 ... 2847
Requisitos e considerações sobre réplicas do Db2 ... 2848
Preparar-se para criar uma réplica do Db2 ... 2850
Criação de uma réplica do Db2 em espera ... 2851
Como trabalhar com backups de réplicas do Db2 ... 2853
Solução de problemas de replicação do Db2 ... 2855
Opções para instâncias de banco de dados do RDS para Db2 ... 2857
Registro em log de auditoria do Db2 ... 2858
Procedimentos armazenados externos ... 2874
Procedimentos armazenados externos baseados em Java ... 2874
Problemas conhecidos e limitações ... 2883
Limitação de autenticação ... 2883
Rotinas não protegidas ... 2883
Espaços de tabela de armazenamento não automáticos durante a migração ... 2883
Configuração do parâmetro db2_compatibility_vector ... 2884
Migrar bancos de dados que contêm pacotes INVALID ... 2884

Procedimentos armazenados do RDS para Db2 ... 2886
Considerações para procedimentos armazenados ... 2895
Conceder e revogar privilégios ... 2896
Políticas de auditoria ... 2910
Grupos de buffer ... 2916
Gerenciar bancos de dados ... 2922
Acesso ao armazenamento ... 2945
Tablespaces ... 2948
Funções definidas pelo usuário do RDS para Db2 ... 2959
rdsadmin.get_task_status ... 2959
rdsadmin.list_databases ... 2964
Solução de problemas ... 2966
Erro de conexão de banco de dados ... 2966
Erro de E/S de arquivo ... 2966
Erros de procedimentos armazenados ... 2970
MariaDB no Amazon RDS ... 2982
Suporte ao recurso MariaDB ... 2984
Versões principais do MariaDB ... 2984
Mecanismos de armazenamento compatíveis ... 2993
Aquecimento de cache ... 2995
Recursos não compatíveis ... 2996
Versões do MariaDB ... 2998
Versões secundárias do MariaDB compatíveis ... 2998
Versões principais do MariaDB compatíveis ... 3002
Ambiente de visualização do banco de dados ... 3003
MariaDB versão 11.4 no ambiente de versão prévia do banco de dados. ... 3006
Versões obsoletas do MariaDB ... 3006
Conexão a uma instância de banco de dados executando o MariaDB ... 3007
Insira as informações de conexão ... 3008
Conectar pelo cliente de linha de comandos ... 3012
Conectar-se a drivers da AWS ... 3012
Solução de problemas ... 3013
Proteger as conexões do MariaDB ... 3015
Segurança do MariaDB ... 3015
Plug-in de validação de senhas ... 3017
Criptografia com SSL/TLS ... 3018

Uso de novos certificados SSL/TLS ... 3023
Melhorar a performance das consultas com leituras otimizadas do RDS ... 3028
Visão geral ... 3028
Casos de uso ... 3029
Práticas recomendadas ... 3030
O uso do ... 3031
Monitoramento ... 3031
Limitações ... 3032
Melhorar a performance das gravações com gravações otimizadas pelo RDS para MariaDB . 3033
Visão geral ... 3033
Usando com um banco de dados ... 3035
Habilitação de ativação em um banco de dados existente ... 3039
Limitações ... 3040
Atualizações do mecanismo de banco de dados do MariaDB ... 3041
Considerações ... 3042
Encontrar destinos de atualização válidos ... 3043
Números de versão do MariaDB ... 3044
Números de versão do RDS ... 3046
Atualizações da versão principal ... 3047
Atualizar uma instância de banco de dados MariaDB ... 3048
Atualizações automáticas da versão secundária ... 3048
Atualizar com tempo de inatividade reduzido ... 3051
Atualizar uma versão do mecanismo de snapshot de banco de dados do MariaDB ... 3056
Opções de atualização para versões de mecanismo incompatíveis ... 3058
Importar dados para uma instância de banco de dados do RDS para MariaDB ... 3060
Considerações sobre importação de dados ... 3066
Importar dados de um banco de dados externo ... 3072
Importar dados com tempo de inatividade reduzido ... 3076
Importar dados de qualquer fonte ... 3098
Replicação do MariaDB ... 3105
Réplicas de leitura do MariaDB ... 3106
Configurar a replicação baseada em GTID com uma instância de origem externa ... 3122
Configurar a replicação da posição do arquivo de log binário com uma instância de origem externa ... 3126
Opções do MariaDB ... 3132
Suporte ao plugin de auditoria do MariaDB ... 3132

Parâmetros para MariaDB 3140
Visualizar parâmetros do MariaDB 3140
Parâmetros do MySQL que não estão disponíveis 3142
Migrar dados de um snapshot de banco de dados do MySQL para uma instância de banco de dados do MariaDB 3145
Executar a migração 3145
Incompatibilidades entre o MariaDB e o MySQL 3147
Referência do MariaDB no SQL do Amazon RDS 3149
mysql.rds_replica_status 3149
mysql.rds_set_external_master_gtid 3151
mysql.rds_kill_query_id 3154
Fuso horário local 3156
Limitações e problemas conhecidos do MariaDB 3159
Limites de tamanho de arquivo 3159
Palavra reservada InnoDB 3161
Portas personalizadas 3161
Insights de Performance 3161
Microsoft SQL Server no Amazon RDS 3162
Tarefas comuns de gerenciamento 3164
Limitações 3167
Suporte a classes de instância de banco de dados 3170
Segurança 3175
Programas de conformidade 3176
HIPAA 3176
Suporte a SSL 3178
Suporte à versão 3178
Gerenciamento de versão 3180
Patches e versões de mecanismos de banco de dados 3181
Programação de suspensão 3181
Suporte a recursos 3182
Recursos do SQL Server 2022 3183
Recursos do SQL Server 2019 3183
Recursos do SQL Server 2017 3184
Recursos do SQL Server 2016 3185
Recursos do SQL Server 2014 3185
Fim do suporte ao SQL Server 2012 no Amazon RDS 3185

Fim do suporte ao SQL Server 2008 R2 no Amazon RDS 3186
Suporte a CDC 3186
Recursos não compatíveis e recursos com suporte limitado 3187
Implantações multi-AZ 3188
Uso de TDE 3189
Funções e procedimentos armazenados 3189
Fuso horário local 3196
Fusos horários compatíveis 3197
Licenciamento do SQL Server no Amazon RDS 3209
Restaurar instâncias de banco de dados com licenças encerradas 3209
SQL Server Developer Edition 3210
Conexão a uma instância de banco de dados executando SQL Server 3211
Antes de se conectar 3211
Localizando o endpoint e o número da porta da instância de banco de dados 3212
Conexão à instância de banco de dados com SSMS 3214
Conectar-se à sua instância de banco de dados com o SQL Workbench/J 3217
Considerações sobre os grupos de segurança 3219
Solução de problemas 3220
Trabalhar com o Active Directory com o RDS para SQL Server 3222
Trabalhar com um Active Directory autogerenciado com uma instância de banco de dados do SQL Server 3223
Trabalhar com o AWS Managed Active Directory com o RDS para SQL Server 3244
Atualizar aplicações para novos certificados SSL/TLS 3261
Determinar se alguma aplicação está se conectando à sua instância de banco de dados Microsoft SQL Server usando SSL 3262
Determinar se um cliente requer verificação de certificado para se conectar 3262
Atualizar o armazenamento confiável de aplicações 3265
Atualizações do mecanismo de banco de dados do SQL Server 3266
Atualizações da versão principal 3267
Considerações sobre atualização 3269
Testar uma atualização 3273
Atualizar de uma instância de banco de dados SQL Server 3274
Fazer upgrade de instâncias de banco de dados defasadas antes do fim do suporte 3274
Importar e exportar bancos de dados do SQL Server 3275
Limitações e recomendações 3278
Configuração 3279

Usar o backup e restauração nativos ... 3285
Compactar arquivos de backup ... 3303
Solução de problemas ... 3303
Importar e exportar dados do SQL Server usando outros métodos ... 3308
Usar o BCP do Linux ... 3323
Réplicas de leitura do SQL Server ... 3329
Configurar réplicas de leitura do SQL Server ... 3329
Limitações de réplica de leitura com o SQL Server ... 3330
Considerações opcionais ... 3331
Sincronizar usuários e objetos do banco de dados ... 3333
Solução de problemas ... 3335
Multi-AZ paraRDS for SQL Server ... 3336
Adicionar Multi-AZ a uma instância de banco de dados do SQL Server ... 3337
Remover multi-AZ de uma instância de banco de dados do SQL Server ... 3338
Limitações, observações e recomendações ... 3338
Determinar a localização do secundário ... 3343
Migrar para AGs Always On ... 3343
Recursos adicionais do SQL Server ... 3345
Usar política de senha com uma instância de banco de dados do SQL Server ... 3346
Integração do Amazon S3 ... 3354
Usar Database Mail ... 3376
Suporte ao armazenamento de instâncias para tempdb ... 3392
Uso de eventos estendidos ... 3396
Acesso aos backups de logs de transações ... 3400
Opções para SQL Server ... 3446
Listas de opções disponíveis para versões e edições do SQL Server ... 3449
Servidores vinculados ao Oracle OLEDB ... 3451
Servidores vinculados ao Teradata ODBC ... 3464
Backup e restauração nativos ... 3474
Transparent Data Encryption ... 3479
Auditoria do SQL Server ... 3492
SQL Server Analysis Services ... 3503
SQL Server Integration Services ... 3533
SQL Server Reporting Services ... 3557
Microsoft Distributed Transaction Coordinator ... 3579
Tarefas comuns do DBA ... 3597

Acessar o banco de dados tempdb ... 3599
Análise da workload do banco de dados com o Orientador de Otimização do Mecanismo de Banco de dados ... 3603
Alterar o db_owner para a conta rdsa do banco de dados ... 3608
Gerenciar agrupamentos e conjuntos de caracteres ... 3609
Criar um usuário de banco de dados ... 3618
Determinar um modelo de recuperação ... 3619
Determinar o último tempo de failover ... 3620
Solucionar problemas de falhas de PITR devido a descontinuidades no LSN ... 3622
Negar ou permitir a visualização de nomes de bancos de dados ... 3622
Desativar inserções rápidas ... 3623
Eliminar um banco de dados do SQL Server ... 3624
Renomear um banco de dados multi-AZ ... 3625
Redefinir a associação ao perfil db_owner para o usuário mestre ... 3626
Restaurar instâncias de banco de dados com licenças encerradas ... 3626
Transição de um banco de dados de OFFLINE para ONLINE ... 3627
Uso de CDC ... 3628
Uso do SQL Server Agent ... 3631
Trabalhos com logs do SQL Server ... 3639
Trabalhar com arquivos de rastreamento e despejo ... 3640
MySQL no Amazon RDS ... 3642
Compatibilidade dos recursos do MySQL ... 3645
Versões principais do MySQL ... 3645
Mecanismos de armazenamento compatíveis ... 3648
Usar memcached e outras opções ... 3648
Aquecimento de cache do InnoDB ... 3649
Alterações de linguagem inclusiva para MySQL 8.4 ... 3650
Recursos não compatíveis ... 3653
Versões do MySQL ... 3655
Versões secundárias do MySQL compatíveis ... 3655
Versões principais do MySQL compatíveis ... 3659
Versões do Suporte estendido do RDS para RDS para MySQL ... 3660
Ambiente de visualização do banco de dados ... 3662
MySQL versão 8.4 no Ambiente de Pré-visualização do Banco de Dados ... 3666
MySQL versão 8.3 no Ambiente de Pré-visualização do Banco de Dados ... 3666
O MySQL versão 8.2 no Ambiente de Pré-visualização do Banco de Dados ... 3667

PostgreSQL versão 12.0 no ambiente de visualização de banco de dados ... 3667
Versões obsoletas do MySQL ... 3667
Conexão a uma instância de banco de dados executando MySQL ... 3668
Insira as informações de conexão ... 3669
Instalação do cliente de linha de comando ... 3673
Conectar pelo cliente de linha de comandos ... 3673
Conexão pelo MySQL Workbench ... 3674
Conectar-se a drivers da AWS ... 3676
Solução de problemas ... 3677
Proteger as conexões do MySQL ... 3679
Validação de senhas ... 3682
Criptografia com SSL/TLS ... 3683
Uso de novos certificados SSL/TLS ... 3687
Usar a autenticação Kerberos do MySQL ... 3693
Melhorar a performance das consultas com leituras otimizadas do RDS ... 3709
Visão geral ... 3709
Casos de uso ... 3710
Práticas recomendadas ... 3711
O uso do ... 3712
Monitoramento ... 3713
Limitações ... 3713
Melhorar a performance das gravações com gravações otimizadas pelo RDS para MySQL ... 3715
Visão geral ... 3033
Usando com um banco de dados ... 3717
Habilitação de ativação em um banco de dados existente ... 3721
Limitações ... 3722
Atualizações do mecanismo de banco de dados do MySQL ... 3723
Considerações ... 3725
Encontrar destinos de atualização válidos ... 3725
Números de versão do MySQL ... 3726
Números de versão do RDS ... 3728
Atualizações da versão principal ... 3729
Testar uma atualização ... 3735
Atualizar uma instância de banco de dados MySQL ... 3737
Atualizações automáticas da versão secundária ... 3737
Atualizar com tempo de inatividade reduzido ... 3740

Atualizar uma versão do mecanismo de snapshot de banco de dados do MySQL 3745
Opções de atualização para versões de mecanismo incompatíveis 3747
Importar dados para uma instância de banco de dados do RDS para MySQL. 3750
Considerações sobre importação de dados 3757
Restauração de um backup em uma instância de banco de dados MySQL 3763
Importar dados de um banco de dados externo 3776
Importar dados com tempo de inatividade reduzido 3781
Importar dados de qualquer fonte 3803
Replicação mysql 3810
Réplicas de leitura do MySQL 3811
Replicação baseada em GTID 3829
Configurar a replicação da posição do arquivo de log binário com uma instância de origem externa 3838
Configurar a replicação de várias fontes 3843
Configurar clusters ativos-ativos 3851
Casos de uso 3851
Limitações e considerações sobre cluster ativo-ativo 3852
Preparar um cluster ativo-ativo entre VPCs 3856
Configurações de parâmetros obrigatórias para clusters ativos-ativos 3858
Converter uma instância de banco de dados em um cluster ativo-ativo 3861
Configurar um novo cluster ativo-ativo 3867
Adicionar uma instância de banco de dados em um cluster ativo-ativo 3874
Monitorar clusters ativos-ativos 3878
Interromper a replicação de grupo em uma instância de banco de dados em um cluster ativo-ativo 3878
Renomear uma instância de banco de dados em um cluster ativo-ativo 3879
Remover uma instância de banco de dados de um cluster ativo-ativo 3880
Exportar dados de uma instância de banco de dados MySQL 3882
Preparar um banco de dados MySQL externo 3882
Preparar a instância de banco de dados MySQL 3884
Copiar o banco de dados 3885
Concluir a exportação 3887
Opções para MySQL 3889
Plug-in de auditoria do MariaDB 3890
memcached 3898
Parâmetros do MySQL 3904

Tarefas comuns de DBA para MySQL ... 3907
Noções básicas sobre usuários predefinidos ... 3907
Modelo de privilégios baseados em funções ... 3908
Privilégios dinâmicos ... 3911
Encerrar uma sessão ou consulta ... 3915
Ignorar o erro de replicação atual ... 3916
Melhorar o tempo de recuperação de falhas ... 3918
Como gerenciar o histórico de status global ... 3921
Configuração do tamanho do pool de buffer e a capacidade de log redo ... 3923
Fuso horário local ... 3925
Problemas conhecidos e limitações ... 3928
Palavra reservada InnoDB ... 3928
Comportamento de armazenamento cheio ... 3928
Tamanho do grupo de buffers do InnoDB inconsistente ... 3929
A otimização de mesclagem de índice retorna resultados errados ... 3930
Exceções de parâmetros do MySQL para instâncias de bancos de dados do Amazon RDS ... 3931
Limites de tamanho de arquivo do MySQL no Amazon RDS ... 3932
Não há suporte ao plugin Keyring do MySQL ... 3935
Portas personalizadas ... 3935
Limitações de procedimentos armazenados do MySQL ... 3935
Replicação baseada em GTID com uma instância de origem externa ... 3935
Plug-in de autenticação padrão do MySQL ... 3936
Substituir innodb_buffer_pool_size ... 3936
Atualização do MySQL 5.7 para o MySQL 8.4 ... 3937
Compressão de páginas do InnoDB ... 3937
Procedimentos armazenados do RDS para MySQL ... 3938
Coletar e manter o histórico de status global ... 3939
Configurar, iniciar e interromper a replicação de logs binários (binlogs) ... 3942
Encerrar uma sessão ou consulta ... 3987
Gerenciar clusters ativos-ativos ... 3989
Gerenciar a replicação de várias fontes ... 3994
Replicar transações com GTIDs ... 4017
Alternar os logs de consulta ... 4020
Definir e mostrar a configuração de logs binários ... 4022
Aquecer o armazenamento em cache do InnoDB ... 4027

Oracle no Amazon RDS 4029
Visão geral do Oracle 4030
Recursos da Oracle 4031
Oracle versões 4035
Licenciamento do Oracle 4041
Usuários e privilégios do Oracle 4046
Classes de instância da Oracle 4047
Arquitetura de banco de dados Oracle 4056
Parâmetros da Oracle 4057
Conjuntos de caracteres Oracle 4060
Limitações da Oracle 4064
Conectar-se a uma instância de banco de dados Oracle 4067
Encontrar o endpoint 4067
SQL Developer 4069
SQL*Plus 4072
Considerações sobre os grupos de segurança 4073
Processos de servidores dedicados e compartilhados 4074
Solução de problemas 4074
Modificar parâmetros do sqlnet.ora para Oracle 4076
Proteção das conexões da Oracle 4082
Criptografia com SSL 4082
Uso de novos certificados SSL/TLS 4083
Criptografia com SSL 4087
Configurar a autenticação Kerberos 4088
Configurar o acesso UTL_HTTP 4108
Trabalhar com CDBs 4121
Visão geral dos CDBs 4121
Configurar um CDB 4127
Fazer backup e restaurar um CDB 4134
Converter um não CDB em CDB 4135
Converter locatário único em multilocatário 4138
Adicionar um banco de dados de locatários do RDS for Oracle à sua instância de CDB 4142
Modificando um banco de dados de locatários do RDS para Oracle 4145
Excluindo um banco de dados de locatários do RDS for Oracle do seu CDB 4147
Visualizando detalhes do banco de dados do inquilino 4149
Fazer upgrade de um CDB 4154

Administração da sua instância de banco de dados Oracle 4155
Tarefas do sistema 4170
Tarefas de banco de dados 4197
Tarefas de log 4228
Tarefas do RMAN 4242
Tarefas do Oracle Scheduler 4277
Diagnosticar problemas 4287
Outras tarefas 4297
Configurar recursos avançados do RDS para Oracle 4313
Configurar o armazenamento de instância 4313
Ativar o HugePages 4327
Ativar tipos de dados estendidos 4330
Importar dados para o Oracle 4334
Importar usando o Oracle SQL Developer 4335
Migrar usando espaços de tabela transportáveis da Oracle 4335
Importar usando o Oracle Data Pump 4353
Importar usando Oracle Export/Import 4371
Importar usando o Oracle SQL*Loader 4372
Migrar com visualizações materializadas do Oracle 4374
Trabalhar com réplicas do Oracle 4377
Visão geral das réplicas do Oracle 4377
Requisitos e considerações sobre réplicas do Oracle 4380
Preparar-se para criar uma réplica do Oracle 4384
Criar uma réplica montada do Oracle 4386
Modificar o modo da réplica 4387
Como trabalhar com backups de réplicas do Oracle 4389
Executar uma alternância do Oracle Data Guard 4392
Solução de problemas de réplicas do Oracle 4400
Opções para Oracle 4402
Visão geral das opções do Oracle DB 4402
Integração do Amazon S3 4405
Oracle Application Express (APEX) 4433
Integração do Amazon EFS 4458
Java Virtual Machine (JVM) 4476
Enterprise Manager 4481
Label Security 4507

Localizador ... 4511
Native Network Encryption (NNE) ... 4516
OLAP ... 4532
Secure Sockets Layer (SSL) ... 4536
Espacial ... 4548
SQLT ... 4553
Statspack ... 4563
Fuso horário ... 4567
Atualização automática do arquivo de fuso horário ... 4573
Transparent Data Encryption (TDE) ... 4584
UTL_MAIL ... 4589
XML DB ... 4593
Atualizar o mecanismo de banco de dados Oracle ... 4594
Visão geral das atualizações do Oracle ... 4594
Atualizações da versão principal ... 4602
Atualizações de versões secundárias ... 4603
Considerações sobre atualização ... 4610
Testar uma atualização ... 4614
Atualizar uma instância de banco de dados do RDS para Oracle ... 4615
Como atualizar para um snapshot de banco de dados Oracle ... 4617
Ferramentas e software de terceiros para Oracle ... 4620
Usar o Oracle GoldenGate ... 4621
Usar o Oracle Repository Creation Utility ... 4640
Configurando o CMAN ... 4648
Instalar um Siebel Database no Oracle no Amazon RDS ... 4651
Versões do mecanismo Oracle Database ... 4656
PostgreSQL no Amazon RDS ... 4657
Tarefas comuns de gerenciamento ... 4659
Trabalhar com o ambiente de visualização de banco de dados ... 4664
Atributos não compatíveis com o ambiente de visualização de banco de dados ... 4665
PostgreSQL versão 17 no Ambiente de Pré-visualização do Banco de Dados ... 4665
Criar uma nova instância de banco de dados no ambiente de visualização de banco de dados ... 4665
Versões do PostgreSQL ... 4668
Versões descontinuadas do PostgreSQL ... 4668
Processo de lançamento do RDS para PostgreSQL ... 4670

Vantagens do processo de lançamento incremental do RDS para PostgreSQL. 4670
Gerenciamento de atualizações de versão 4670
Versões de extensões do PostgreSQL 4672
Restringir a instalação de extensões do PostgreSQL 4672
Extensões confiáveis do PostgreSQL 4674
Recursos do PostgreSQL 4676
Tipos de dados personalizados e enumerações 4677
Gatilhos de evento 4677
Páginas grandes 4678
Executar replicação lógica 4679
Disco de RAM para o stats_temp_directory 4686
Tablespaces 4687
Agrupamentos para EBCDIC e outras migrações de mainframe 4688
Gerenciar sincronização de slots lógicos 4694
Conectar-se a uma instância PostgreSQL 4696
Instalar o cliente psql 4697
Insira as informações de conexão 4697
Usar pgAdmin para se conectar a uma instância de banco de dados RDS for PostgreSQL 4700
Usar psql para se conectar à sua instância de banco de dados RDS for PostgreSQL 4702
Conectar-se ao RDS para PostgreSQL com o driver JDBC da AWS 4704
Conectar-se ao RDS para PostgreSQL com o driver Python da AWS 4704
Solucionar problemas de conexões com a sua instância RDS for PostgreSQL 4704
Proteger conexões com SSL/TLS 4707
Usar o SSL com uma instância de banco de dados PostgreSQL 4707
Atualização de aplicações para uso de novos certificados SSL/TLS 4715
Usar a autenticação Kerberos 4720
Disponibilidade de região e versão 4721
Visão geral da autenticação Kerberos 4721
Configuração 4722
Gerenciar uma instância de banco de dados do RDS para PostgreSQL em um domínio do Active Directory 4737
Conectar-se com a autenticação Kerberos 4738
Usar um servidor DNS personalizado para acesso à rede de saída 4742
Ativar resolução DNS personalizada 4742
Desativar resolução DNS personalizada 4742
Configuração de um servidor DNS personalizado 4742

Atualizações do mecanismo de banco de dados do PostgreSQL 4745
Considerações 4747
Encontrar destinos de atualização válidos 4748
Números de versão do PostgreSQL 4749
Números de versão do RDS 4750
Escolher uma atualização de versão principal 4750
Como atualizar a versão principal 4766
Atualizações automáticas da versão secundária 4775
Atualizar extensões do PostgreSQL 4777
Atualizar uma versão do mecanismo de snapshot de banco de dados PostgreSQL 4779
Trabalhar com réplicas de leitura do RDS para PostgreSQL 4782
Decodificação lógica em uma réplica de leitura 4782
Limitações de réplicas de leitura com o PostgreSQL 4786
Configuração de réplica de leitura com o PostgreSQL 4787
Usar réplicas de leitura em cascata 4790
Criar réplicas de leitura em cascata entre regiões 4791
Como a replicação funciona para diferentes versões do RDS para PostgreSQL 4793
Monitoração e ajuste do processo de replicação 4798
Solução de problemas da réplica de leitura do RDS para PostgreSQL 4801
Melhorar a performance das consultas com leituras otimizadas do RDS 4802
Visão geral das leituras otimizadas pelo RDS no PostgreSQL 4802
Casos de uso 4803
Práticas recomendadas 4804
O uso do 4804
Monitoramento 4805
Limitações 4806
Importação de dados no PostgreSQL 4807
Importar um banco de dados PostgreSQL de uma instância do Amazon EC2 4810
Uso do comando \copy para importar dados em uma tabela em uma instância de banco de dados PostgreSQL 4812
Como importar dados do Amazon S3 para o RDS para PostgreSQL 4814
Transportar bancos de dados PostgreSQL entre instâncias de banco de dados 4834
Exportação de dados do PostgreSQL para Amazon S3 4844
Instalar a extensão 4845
Visão geral da exportação para o S3 4846
Especificar o caminho do arquivo do Amazon S3 para o qual exportar 4847

Configurar o acesso a um bucket do Amazon S3 4848
Exportar dados de consulta usando a função aws_s3.query_export_to_s3 4853
Referência de funções 4856
Solução de problemas de acesso ao Amazon S3 4861
Invocar uma função Lambda do RDS para PostgreSQL 4862
Etapa 1: configurar conexões de saída 4863
Etapa 2: configurar o IAM para o cluster e o Lambda 4864
Etapa 3: instalar a extensão 4866
Etapa 4: usar as funções Lambda auxiliares 4867
Etapa 5: invocar uma função Lambda 4868
Etapa 6: Conceder permissões a usuários 4869
Exemplos: Chamar funções do Lambda 4870
Mensagens de erro da função Lambda 4872
Função do Lambda e referência de parâmetros 4874
Tarefas comuns de DBA do RDS para PostgreSQL 4879
Agrupamentos compatíveis com RDS para PostgreSQL 4880
Noções básicas de perfis e permissões do PostgreSQL 4881
Tratamento de conexões inativas no PostgreSQL 4897
Como trabalhar com o autovacuum do PostgreSQL 4900
Mecanismos de registro em log 4948
Gerenciar arquivos temporários com o PostgreSQL 4949
Uso de pgBadger para análise de logs com o PostgreSQL 4956
Usar o PGSnapper para monitorar o PostgreSQL 4956
Trabalhar com parâmetros 4956
Ajustar com eventos de espera do RDS para PostgreSQL 4978
Conceitos essenciais para o ajuste do RDS para PostgreSQL 4979
Eventos de espera do RDS para PostgreSQL 4984
Client:ClientRead 4986
Client:ClientWrite 4990
CPU 4992
IO:BufFileRead and IO:BufFileWrite 4999
IO:DataFileRead 5008
IO:WALWrite 5016
IPC: eventos de espera paralelos 5020
Lock:advisory 5026
Lock:extend 5029

Lock:Relation ... 5032
Lock:transactionid ... 5035
Lock:tuple ... 5038
LWLock:BufferMapping (LWLock:buffer_mapping) ... 5043
LWLock:BufferIO (IPC:BufferIO) ... 5045
LWLock:buffer_content (BufferContent) ... 5048
LWLock:lock_manager (LWLock:lockmanager) ... 5050
LWLock:pg_stat_statements ... 5056
Tempo limite:PgSleep ... 5060
Timeout:VacuumDelay ... 5061
Ajustar o RDS para PostgreSQL com insights proativos do Amazon DevOps Guru ... 5064
O banco de dados está inativo há muito tempo na conexão da transação ... 5064
Usar extensões do PostgreSQL ... 5068
Usar funções da extensão orafce ... 5069
Usar a compatibilidade com extensões delegadas do Amazon RDS para PostgreSQL ... 5071
Gerenciar partições com a extensão pg_partman ... 5085
Usar pgAudit para registrar a atividade do banco de dados ... 5092
Agendar manutenção com a extensão pg_cron ... 5106
Usar pglogical para sincronizar dados ... 5116
Usar pgactive para comportar a replicação ativa-ativa ... 5131
Reduzir o inchaço com a extensão pg_repack ... 5204
Atualizar e usar PLV8 ... 5210
Usar PL/Rust para escrever funções na linguagem Rust ... 5213
Gerenciar dados espaciais com PostGIS ... 5218
Wrappers de dados externos compatíveis no Amazon RDS para PostgreSQL ... 5228
Usar a extensão log_fdw ... 5228
Usar postgres_fdw para acessar dados externos ... 5231
Trabalhar com um banco de dados MySQL ... 5232
Trabalhar com um banco de dados Oracle ... 5236
Trabalhar com um banco de dados do SQL Server ... 5240
Trabalhar com Trusted Language Extensions para PostgreSQL ... 5244
Terminologia ... 5245
Requisitos para usar o Trusted Language Extensions ... 5246
Configurar o Trusted Language Extensions ... 5249
Visão geral do Trusted Language Extensions ... 5253
Criar extensões TLE ... 5255

Descartar suas extensões TLE de um banco de dados ... 5260
Desinstalar o Trusted Language Extensions ... 5261
Usar ganchos do PostgreSQL com suas extensões TLE ... 5262
Usar tipos de dados personalizados no Trusted Language Extensions ... 5269
Referência de funções para Trusted Language Extensions ... 5269
Referência de ganchos para Trusted Language Extensions ... 5283
Exemplos de código ... 5286
Ações ... 5297
CreateDBInstance ... 5298
CreateDBParameterGroup ... 5314
CreateDBSnapshot ... 5320
DeleteDBInstance ... 5329
DeleteDBParameterGroup ... 5338
DescribeAccountAttributes ... 5344
DescribeDBEngineVersions ... 5348
DescribeDBInstances ... 5356
DescribeDBParameterGroups ... 5366
DescribeDBParameters ... 5374
DescribeDBSnapshots ... 5384
DescribeOrderableDBInstanceOptions ... 5391
GenerateRDSAuthToken ... 5399
ModifyDBInstance ... 5401
ModifyDBParameterGroup ... 5407
RebootDBInstance ... 5413
Cenários ... 5416
Começar a usar instâncias de banco de dados ... 5416
Exemplos sem servidor ... 5514
Como se conectar a um banco de dados do Amazon RDS em uma função do Lambda ... 5514
Exemplos entre serviços ... 5531
Crie um rastreador de itens de trabalho do Aurora Sem Servidor ... 5531
Segurança ... 5536
Autenticação de banco de dados ... 5538
Autenticação com senha ... 5539
Autenticação do banco de dados do IAM ... 5539
Autenticação de Kerberos ... 5540
Gerenciamento de senhas com RDS e o Secrets Manager ... 5542

Limitações ... 5542
Visão geral ... 5543
Benefícios ... 5544
Permissões ... 5545
Impor o gerenciamento pelo RDS ... 5546
Gerenciar senha para uma instância de banco de dados ... 5547
Gerenciar senha para um banco de dados de locatários ... 5551
Gerenciar senha para um cluster de banco de dados multi-AZ ... 5555
Alternar segredos para uma instância de banco de dados ... 5559
Alternar segredos para clusters de banco de dados multi-AZ ... 5561
Visualizar detalhes do segredo ... 5563
Visualizar detalhes do segredo para clusters de banco de dados multi-AZ ... 5566
Visualizar detalhes do segredo para bancos de dados de locatários ... 5570
Disponibilidade de região e versão ... 5573
Proteção de dados ... 5573
Criptografia de dados ... 5574
Privacidade do tráfego entre redes ... 5607
Gerenciamento de identidade e acesso ... 5609
Público ... 5609
Autenticar com identidades ... 5610
Gerenciamento do acesso usando políticas ... 5614
Como o Amazon RDS funciona com o IAM ... 5616
Exemplos de políticas baseadas em identidade ... 5624
Políticas gerenciadas pela AWS ... 5647
Atualizações da política ... 5655
Prevenção contra o ataque do “substituto confuso” em todos os serviços ... 5675
Autenticação do banco de dados do IAM ... 5678
Solução de problemas ... 5731
Registro em log e monitoramento ... 5733
Validação de conformidade ... 5736
Resiliência ... 5737
Backup e restauração ... 5737
Replicação ... 5737
Failover ... 5738
Segurança da infraestrutura ... 5739
Grupos de segurança ... 5739

Public accessibility 5739
VPC endpoints (AWS PrivateLink) 5741
Considerações 1565
Disponibilidade 1565
Criar um VPC endpoint de interface 1565
Criar uma política de VPC endpoint 1565
Práticas recomendadas de segurança 5744
Controlar acesso com grupos de segurança 5745
Visão geral dos grupos de segurança de VPC 5746
Cenário de grupos de segurança 5747
Criar um grupo de segurança de VPC 5748
Associação a uma instância de banco de dados 5749
Privilégios da conta de usuário mestre 5749
Funções vinculadas ao serviço 5754
Permissões de função vinculada ao serviço do Amazon RDS 5754
Permissões de funções vinculadas a serviços para o Amazon RDS Custom 5757
Permissões de perfis vinculados ao serviço para o Amazon RDS Beta 5759
Perfil vinculado ao serviço da Pré-visualização do Amazon RDS 5760
Uso do Amazon RDS com a Amazon VPC 5762
Trabalhar com uma instância de banco de dados em uma VPC 5762
Atualizar a VPC para uma instância de banco de dados 5782
Cenários para acessar uma instância de banco de dados em uma VPC 5783
Tutorial: Criar uma VPC para usar com uma instância de banco de dados (somente IPv4) 5790
Tutorial: Criar uma VPC para uso com uma instância de banco de dados (modo de pilha dupla) 5798
Mover uma instância de banco de dados para uma VPC 5810
Cotas e restrições 5813
Cotas no Amazon RDS 5813
Restrições de nomenclatura no Amazon RDS 5820
Número máximo de conexões de banco de dados 5821
Limites de tamanho de arquivo no Amazon RDS 5824
Solução de problemas 5825
Não é possível conectar-se à instância de banco de dados do 5825
Teste de conexão com a instância de banco de dados 5828
Solução de problemas de autenticação da conexão 5829
Problemas de segurança 5829

Mensagem de erro "Falha ao recuperar atributos da conta, certas funções do console podem ser prejudicadas." ... 5829
Solução de problemas de estado de rede incompatível ... 5830
Causas ... 5830
Resolução ... 5830
Redefinir a senha de proprietário da instância de banco de dados ... 5832
Interrupção ou reinicialização da instância de banco de dados ... 5832
Alterações de parâmetros que não entram em vigor ... 5834
Instância de banco de dados sem espaço de armazenamento ... 5834
Instâncias de banco de dados disponíveis insuficientes ... 5836
Problemas de memória liberável do RDS ... 5837
Problemas com o MySQL e o MariaDB ... 5837
Máximo de conexões MySQL e MariaDB ... 5838
Diagnosticar e resolver o status de parâmetros incompatíveis para um limite de memória . 5838
Diagnosticar e resolver atrasos entre réplicas de leitura ... 5841
Diagnosticar e resolver uma falha de replicação de leitura do MySQL ou MariaDB ... 5843
Criar triggers com o registro de logs binários habilitado requer o privilégio SUPER ... 5845
Diagnosticar e resolver falhas de restauração pontual ... 5847
Erro de replicação interrompida ... 5847
Falha na criação da réplica de leitura ou interrupção da replicação com o erro fatal 1236 .. 5848
A replicação da réplica de leitura falha ao inicializar a estrutura de metadados ... 5849
Não é possível definir o período de retenção de backup como 0 ... 5849
Referência da API do Amazon RDS ... 5850
Uso da API de consulta ... 5850
Parâmetros de consulta ... 5850
Autenticação de solicitação de consulta ... 5851
Solução de problemas de aplicações ... 5851
Recuperação de erros ... 5851
Dicas de solução de problemas ... 5852
Histórico do documento ... 5853
Atualizações anteriores ... 6040
Glossário da AWS ... 6075

O que é o Amazon Relational Database Service (Amazon RDS)?

O Amazon Relational Database Service (Amazon RDS) é um serviço da Web que facilita a configuração, a operação e escalabilidade de um banco de dados relacional na Nuvem AWS. Ele fornece capacidade econômica e redimensionável para um banco de dados relacional padrão do setor e gerencia tarefas comuns de administração de banco de dados.

> **Note**
>
> Este guia abrange mecanismos de banco de dados do Amazon RDS diferentes do Amazon Aurora. Para obter informações sobre o Amazon Aurora, consulte o Guia do usuário do Amazon Aurora.

Caso esteja começando a usar os produtos e serviços AWS, passe a saber mais com os seguintes recursos:

- Para obter uma visão geral de todos os produtos AWS, consulte O que é computação em nuvem?
- A Amazon Web Services fornece vários serviços de banco de dados. Para saber mais sobre a variedade de opções de bancos de dados disponíveis na AWS, consulte Escolher um serviço de banco de dados na AWS e Executar bancos de dados na AWS.

Vantagens do Amazon RDS

O Amazon RDS é um serviço de banco de dados gerenciado. Ele é responsável pela maioria das tarefas de gerenciamento. Ao eliminar tarefas manuais tediosas, o Amazon RDS libera você para se concentrar na aplicação e nos usuários.

O Amazon RDS oferece as seguintes principais vantagens em relação às implantações de banco de dados que não são totalmente gerenciadas:

- É possível usar os mecanismos de banco de dados já conhecidos: IBM Db2, MariaDB, Microsoft SQL Server, MySQL, Oracle Database e PostgreSQL.
- O Amazon RDS gerencia backups, patches de software, detecção automática de falhas e recuperação.

- Você pode ativar backups automatizados ou pode criar manualmente seus próprios snapshots de backup. Você pode usar esses backups para restaurar um banco de dados. O processo de restauração do Amazon RDS funciona de maneira confiável e eficiente.
- É possível ter alta disponibilidade com uma instância de banco de dados primária e uma instância de banco de dados secundária síncrona que pode ser usada para failover em caso de problemas. Também é possível usar réplicas de leitura para aumentar a escalabilidade de leitura.
- Além da segurança no seu pacote de banco de dados, é possível controlar o acesso usando o AWS Identity and Access Management (IAM) para definir usuários e permissões. Você também pode ajudar a proteger seus bancos de dados colocando-os em uma nuvem privada virtual (VPC).

Comparação de responsabilidades com o Amazon EC2 e implantações on-premises

Recomendamos o Amazon RDS como opção padrão para a maioria das implantações de banco de dados relacional. As seguintes alternativas têm a desvantagem de fazer com que você passe mais tempo gerenciando software e hardware:

Implantação on-premises

Quando você compra um servidor on-premises, recebe CPU, memória, armazenamento e IOPS, todos no mesmo pacote. Você assume total responsabilidade pelo servidor, sistema operacional e software de banco de dados.

Amazon EC2

O Amazon Elastic Compute Cloud (Amazon EC2) oferece uma capacidade de computação escalável na Nuvem AWS. Ao contrário de um servidor on-premises, CPU, memória, armazenamento e IOPS são separados para que você possa escalá-los de forma independente. A AWS gerencia as camadas de hardware, o que elimina parte do trabalho de gerenciar um servidor de banco de dados on-premises.

A desvantagem de executar um banco de dados no Amazon EC2 é a maior propensão a erros do usuário. Por exemplo, quando você atualiza o sistema operacional ou o software do banco de dados manualmente, pode acabar causando acidentalmente tempo de inatividade da aplicação. Você pode perder horas verificando todas as alterações para identificar e corrigir um problema.

A tabela a seguir compara os modelos de gerenciamento para bancos de dados on-premises, o Amazon EC2 e o Amazon RDS.

Recurso	Gerenciamento on-premises	Gerenciamento do Amazon EC2	Gerenciamento do Amazon RDS
Otimização de aplicações	Cliente	Cliente	Cliente
Escalabilidade	Cliente	Cliente	AWS
Alta disponibilidade	Cliente	Cliente	AWS
Backups de banco de dados	Cliente	Cliente	AWS
Aplicação de patches de softwares para banco de dados	Cliente	Cliente	AWS
Instalação de softwares para banco de dados	Cliente	Cliente	AWS
Aplicação de patches de sistema operacional (SO)	Cliente	Cliente	AWS
Instalação do sistema operacional	Cliente	Cliente	AWS
Manutenção do servidor	Cliente	AWS	AWS
Ciclo de vida do hardware	Cliente	AWS	AWS
Energia, rede e desaquecimento	Cliente	AWS	AWS

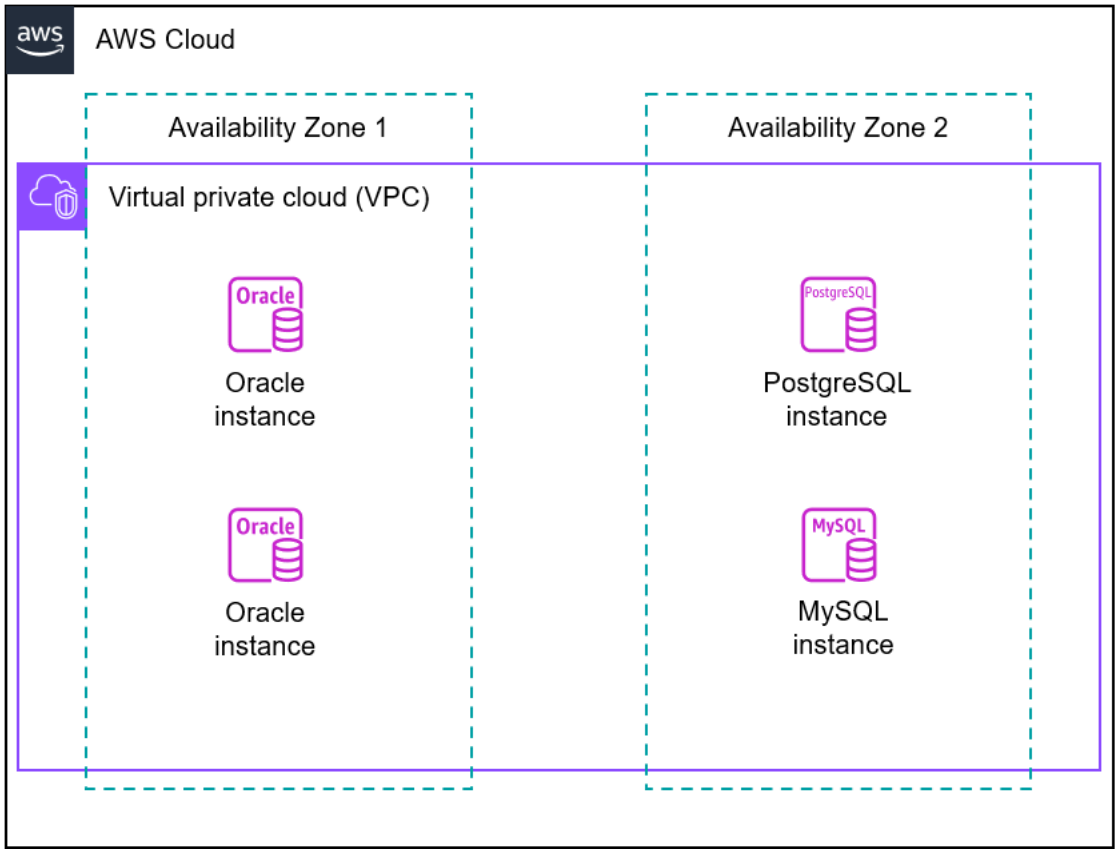
Modelo de responsabilidade compartilhada do Amazon RDS

O Amazon RDS é responsável por hospedar os componentes de software e a infraestrutura de instâncias de banco de dados e clusters de banco de dados. Você é responsável pelo ajuste

das consultas, que é o processo de ajustar as consultas SQL para melhorar a performance. A performance de consulta é altamente dependente do design do banco de dados, do tamanho dos dados, da distribuição dos dados, da workload da aplicação e dos padrões de consulta, que podem variar muito. Monitoramento e ajuste são processos altamente individualizados que você controla para seus bancos de dados do RDS. É possível usar o recurso Insights de Performance do Amazon RDS e outras ferramentas para identificar consultas problemáticas.

Instâncias de banco de dados do Amazon RDS

Uma instância de banco de dados é um ambiente isolado de banco de dados na Nuvem AWS. O bloco de construção básico do Amazon RDS é a instância do banco de dados. Sua instância de banco de dados pode conter um ou mais bancos de dados criados pelo usuário. O diagrama a seguir mostra uma nuvem privada virtual (VPC) que contém duas zonas de disponibilidade, com cada AZ contendo duas instâncias de banco de dados.



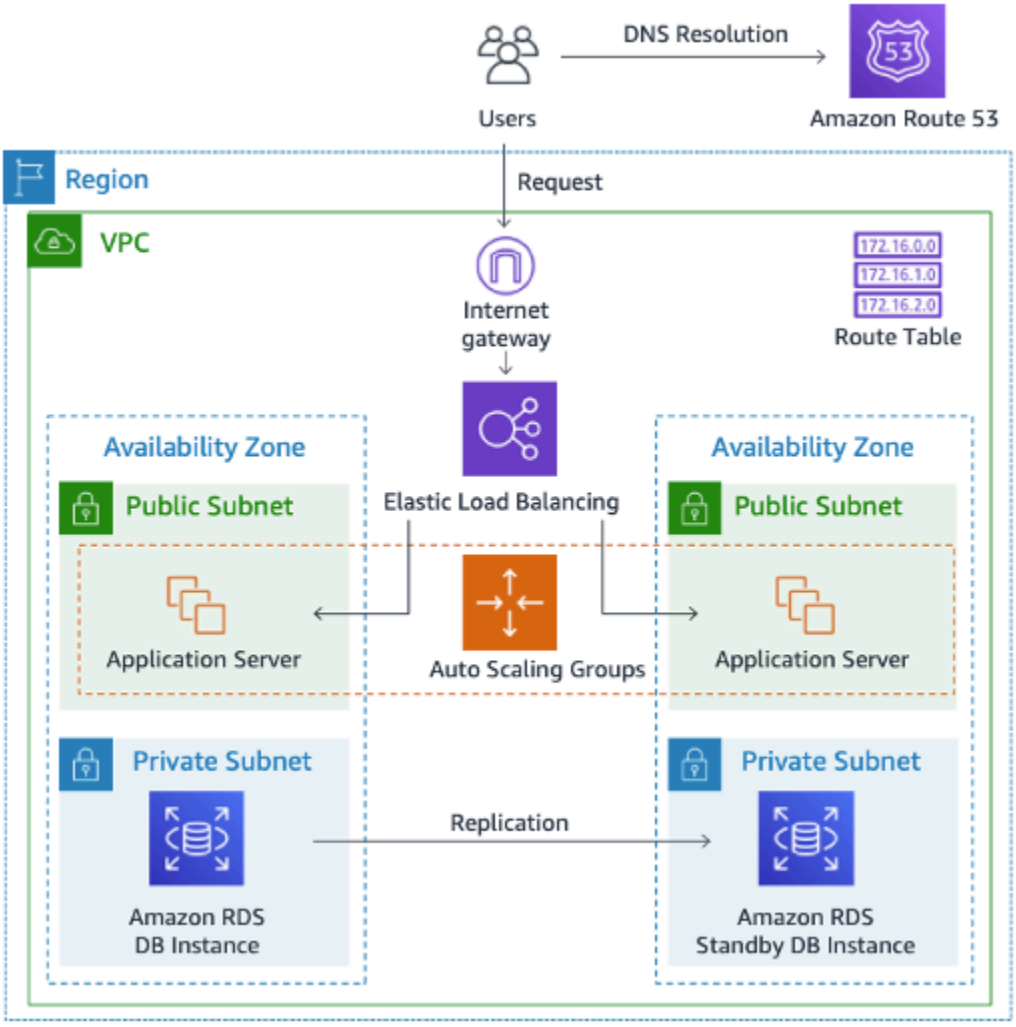
É possível acessar as instâncias de banco de dados usando as mesmas ferramentas e aplicações usadas com uma instância de banco de dados independente. Crie e modifique uma instância de banco de dados usando a AWS Command Line Interface (AWS CLI), a API do Amazon RDS ou o AWS Management Console.

Tópicos

- Arquitetura da aplicação do Amazon RDS: exemplo
- Mecanismos de banco de dados
- Classes da instância de banco de dados
- Armazenamento da instância de banco de dados
- As instâncias de banco de dados em uma Amazon Virtual Private Cloud (Amazon VPC)

Arquitetura da aplicação do Amazon RDS: exemplo

A imagem a seguir mostra um caso de uso típico de um site dinâmico que usa instâncias de banco de dados do Amazon RDS para armazenamento de banco de dados:



Os principais componentes da arquitetura anterior são os seguintes:

Elastic Load Balancing

A AWS direciona o tráfego do usuário por meio do Elastic Load Balancing. Um load balancer distribui cargas de trabalho para vários recursos computacionais, como servidores virtuais. Neste exemplo de caso de uso, o Elastic Load Balancer encaminha solicitações do cliente para servidores de aplicações.

Servidores de aplicações

Os servidores de aplicações interagem com instâncias de banco de dados do RDS. Na AWS, um servidor de aplicações geralmente é hospedado em instâncias do EC2, que oferecem capacidade de computação escalável. Os servidores de aplicações residem em sub-redes públicas com diferentes zonas de disponibilidade (AZs) dentro da mesma nuvem privada virtual (VPC).

Instâncias de banco de dados do RDS

Os servidores de aplicações do EC2 interagem com instâncias de banco de dados do RDS. As instâncias de banco de dados residem em sub-redes privadas em diferentes zonas de disponibilidade (AZs) dentro da mesma nuvem privada virtual (VPC). Como as sub-redes são privadas, nenhuma solicitação da internet é permitida.

A instância de banco de dados primária é replicada para outra instância de banco de dados, chamada de réplica de leitura. Ambas as instâncias de banco de dados estão em sub-redes privadas dentro da VPC, o que significa que os usuários da internet não podem acessá-las diretamente.

Mecanismos de banco de dados

Um mecanismo de banco de dados é o software de banco de dados relacional específico que é executado na sua instância de banco de dados. O Amazon RDS é compatível com os seguintes mecanismos de banco de dados:

- IBM Db2

  Para obter mais informações, consulte Amazon RDS para Db2.
- MariaDB

  Para obter mais informações, consulte Amazon RDS para MariaDB.
- Microsoft SQL Server

  Para obter mais informações, consulte Amazon RDS para Microsoft SQL Server.

- MySQL

  Para obter mais informações, consulte Amazon RDS para MySQL.
- Oracle Database

  Para obter mais informações, consulte Amazon RDS para Oracle.
- PostgreSQL

  Para obter mais informações, consulte Amazon RDS para PostgreSQL.

Cada mecanismo de banco de dados tem seus próprios recursos compatíveis, e cada versão de um mecanismo de banco de dados pode incluir recursos específicos. O suporte para recursos do Amazon RDS varia entre Regiões da AWS e versões específicas de cada mecanismo de banco de dados. Para conferir o suporte a recursos em diferentes versões do mecanismo e regiões, consulte Recursos compatíveis com o Amazon RDS por Região da AWS e com o mecanismo de banco de dados.

Além disso, cada mecanismo de banco de dados tem um conjunto de parâmetros em um grupo de parâmetros de banco de dados que controlam o comportamento dos bancos de dados que ele gerencia. Para ter mais informações sobre parameter groups, consulte Grupos de parâmetros para Amazon RDS.

Classes da instância de banco de dados

A classe de instância de banco de dados determina a capacidade de computação e de memória de uma instância de banco de dados. Uma classe de instância de banco de dados consiste no tipo de classe e no tamanho de instância de banco de dados. O Amazon RDS é compatível com os seguintes tipos de classe de instância, em que o asterisco (*) representa a geração, o atributo opcional e o tamanho:

- Uso geral: db.m*
- Memória otimizada: db.z*, db.x*, db.r*
- Otimizada para computação: db.c*
- Desempenho expansível: db.t*

Cada classe de instância oferece diferentes recursos de computação, memória e armazenamento. Por exemplo, db.m7g é um tipo de classe de instância de banco de dados de uso geral de 7ª

geração com a tecnologia de processadores Graviton3 da AWS. Ao criar uma instância de banco de dados, você especifica uma classe de instância de banco de dados, como db.m7g.2xlarge, em que 2xlarge é o tamanho. Para ter mais informações sobre as especificações de hardware das diferentes classes de instância, consulte Especificações de hardware para classes de instância de banco de dados .

É possível selecionar a instância de banco de dados mais adequada às suas necessidades. Se os seus requisitos mudarem com o passar do tempo, você poderá alterar a classe de instância de banco de dados. Por exemplo, você pode aumentar a escala verticalmente de sua instância db.m7g.2xlarge para db.m7g.4xlarge. Para obter mais informações, consulte Classes de instâncias de banco de dados .

> **Note**
>
> Para obter informações sobre a definição de preço de classes de instâncias de banco de dados, consulte a seção Definição de preço da página de produto do Amazon RDS.

Armazenamento da instância de banco de dados

O Amazon EBS fornece volumes de armazenamento em bloco duráveis que podem ser anexados a uma instância em execução. O armazenamento de instâncias de banco de dados é fornecido nos seguintes tipos:

- Finalidade geral (SSD)

  Esse tipo de armazenamento econômico é ideal para uma série de workloads em execução em instâncias de banco de dados de médio porte. O armazenamento de uso geral é mais adequado para ambientes de desenvolvimento e teste.

- IOPS provisionadas (PIOPS)

  Esse tipo de armazenamento foi projetado para atender às necessidades de workloads de uso intenso de E/S, em especial workloads de banco de dados que exigem baixa latência de E/S e throughput de E/S consistente. O armazenamento de IOPS provisionadas é mais adequado para ambientes de produção.

- Magnético

O Amazon RDS comporta armazenamento magnético para compatibilidade com versões anteriores. Recomendamos que você use SSD de uso geral ou SSD de IOPS provisionadas para toda e qualquer nova necessidade de armazenamento.

Os tipos de armazenamento diferem em características de performance e preços. É possível adaptar o custo e a performance de armazenamento às necessidades do banco de dados.

Cada instância de banco de dados tem requisitos mínimos e máximos de armazenamento, de acordo com o tipo de armazenamento e o mecanismo de banco de dados compatíveis. É importante ter armazenamento suficiente para que os bancos de dados tenham espaço para se expandir. Além disso, armazenamento suficiente garante que os recursos do mecanismo de banco de dados tenha espaço para escrever conteúdo ou entradas de log. Para obter mais informações, consulte Armazenamento de instâncias de banco de dados do Amazon RDS.

As instâncias de banco de dados em uma Amazon Virtual Private Cloud (Amazon VPC)

É possível executar uma instância de banco de dados em uma nuvem privada virtual (VPC) usando o serviço Amazon Virtual Private Cloud (Amazon VPC). Ao usar uma VPC, você tem controle sobre o ambiente de rede virtual. É possível escolher seu próprio intervalo de endereços IP, criar sub-redes e configurar o roteamento e listas de controle de acesso.

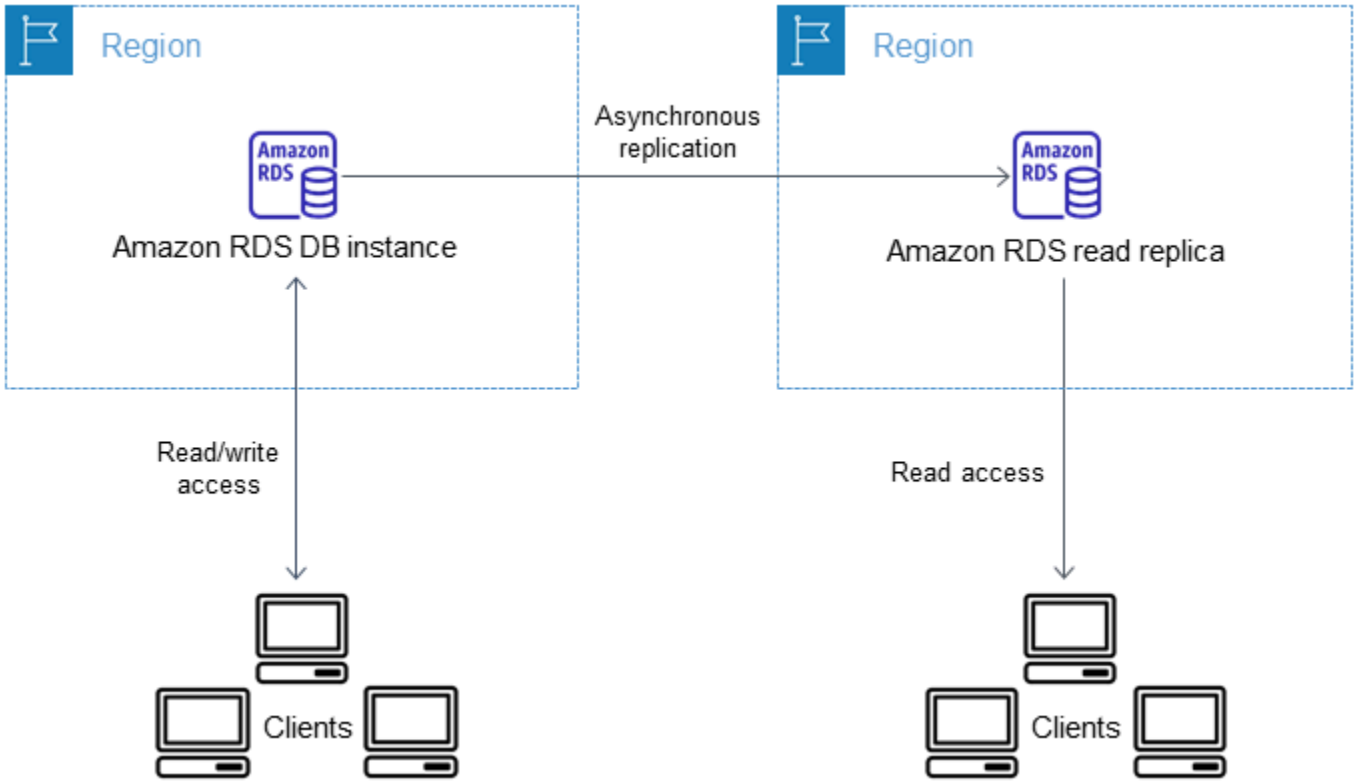
A funcionalidade básica do Amazon RDS é a mesma, não importa se ela está sendo executada ou não em uma VPC. O Amazon RDS gerencia backups, patches de software, detecção automática de falhas e recuperação. Não há custos adicionais para executar a instância de banco de dados em uma VPC. Para obter mais informações sobre como usar a Amazon VPC com o RDS, consulte Amazon VPC e Amazon RDS.

O Amazon RDS usa o Network Time Protocol (NTP) para sincronizar o tempo em instâncias de banco de dados.

Regiões da AWS e zonas de disponibilidade

Os recursos de computação em nuvem da Amazon estão alojados em instalações de datacenter altamente disponíveis em diferentes áreas do mundo (por exemplo, América do Norte, Europa ou Ásia). Cada localização de data center é chamada de Região da AWS. Com o Amazon RDS, é possível criar as instâncias de banco de dados em várias regiões.

O cenário a seguir mostra uma instância de banco de dados do RDS em uma região que se replica de forma assíncrona para uma instância de banco de dados em espera em uma região diferente. Se uma região ficar indisponível, a instância na outra região ainda estará disponível.



Zonas de disponibilidade

Cada região da AWS contém vários locais distintos, chamados de zonas de disponibilidade, ou AZs. Cada zona de disponibilidade é projetada para ser isolada de falhas em outras zonas de disponibilidade. Cada uma é projetada para fornecer conectividade de rede de baixa latência e custo reduzido para outras zonas de disponibilidade na mesma região da AWS. Ao iniciar as instâncias de banco de dados em zonas de disponibilidade separadas, é possível proteger suas aplicações de falhas de um único local. Para obter mais informações, consulte Regiões, zonas de disponibilidade e Local Zones.

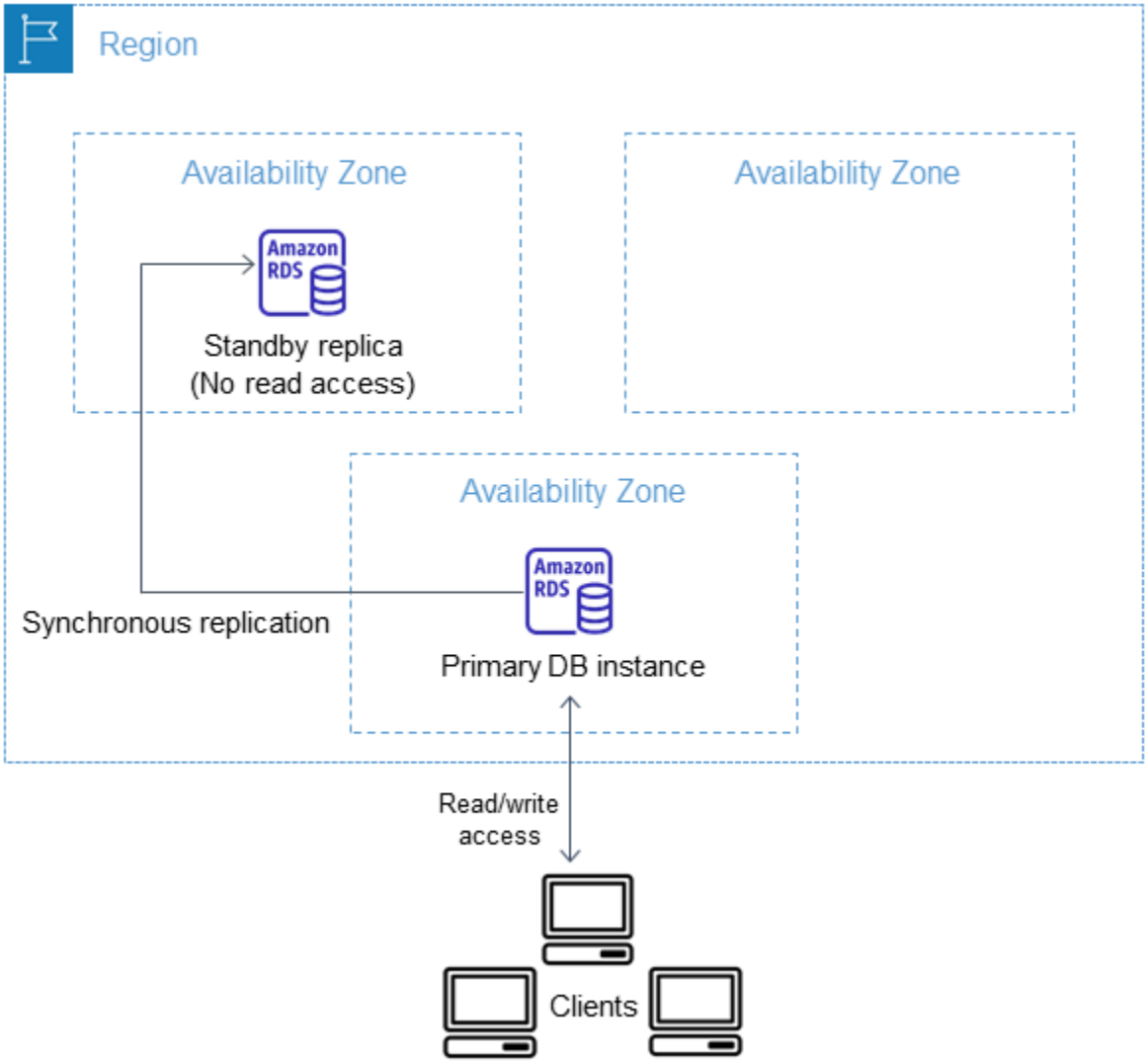
Implantações multi-AZ

Você pode executar sua instância de banco de dados em várias zonas de disponibilidade, uma opção chamada de implantação multi-AZ. Quando você escolhe essa opção, a Amazon automaticamente provisiona e mantém uma ou mais instâncias de banco de dados secundárias em espera em uma AZ diferente. A instância de banco de dados primária é replicada ao longo das zonas de disponibilidade em todas as instâncias de banco de dados secundárias.

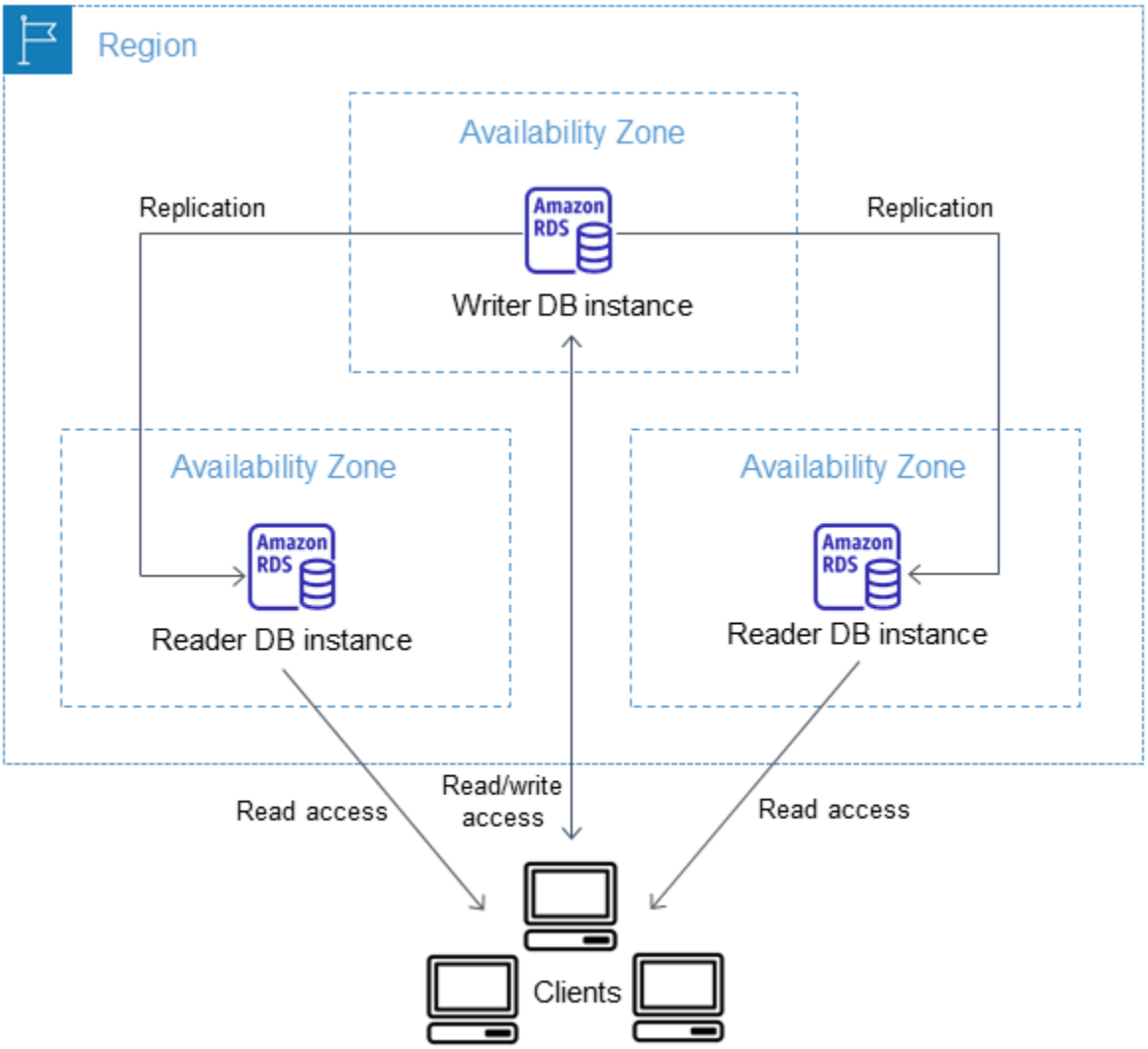
Uma implantação multi-AZ oferece as seguintes vantagens:

- Oferece redundância de dados e suporte a failover.
- Elimina congelamentos de E/S.
- Minimiza os picos de latência durante backups do sistema.
- Fornece tráfego de leitura em instâncias de banco de dados secundárias (somente implantação de clusters de banco de dados multi-AZ).

O diagrama a seguir exibe uma implantação de instância de banco de dados multi-AZ, em que o Amazon RDS provisiona e mantém automaticamente uma réplica em espera síncrona em outra zona de disponibilidade. O banco de dados de réplica não oferece tráfego de leitura.



O diagrama a seguir exibe uma implantação de cluster de banco de dados multi-AZ, que tem uma instância de banco de dados do gravador e duas instâncias de banco de dados do leitor em três zonas de disponibilidade diferentes na mesma Região da AWS. Todas as três instâncias de banco de dados podem fornecer tráfego de leitura.

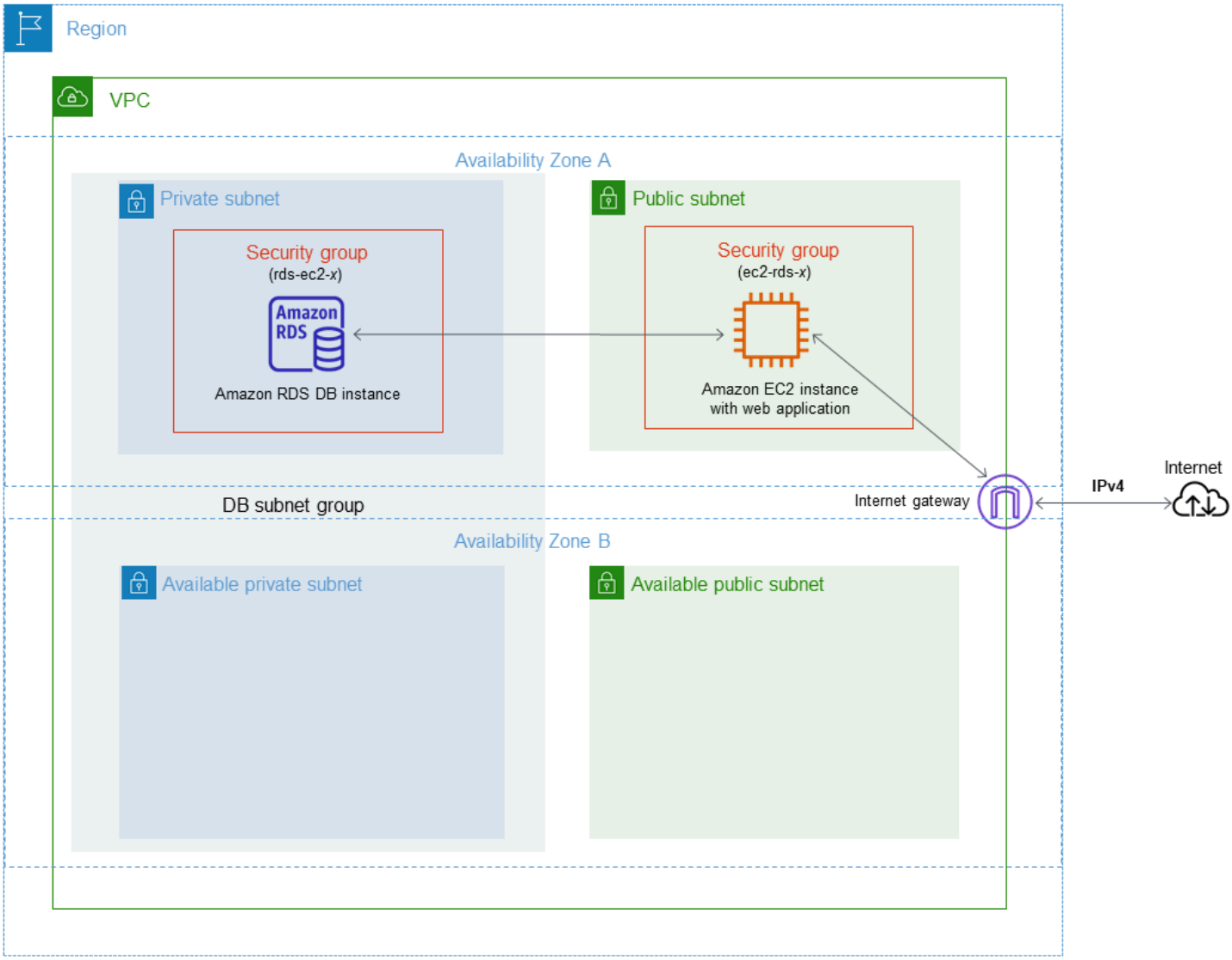


Para obter mais informações, consulte Configurar e gerenciar uma implantação multi-AZ para o Amazon RDS.

Controle de acesso com grupos de segurança.

Um grupo de segurança controla o acesso a uma instância de banco de dados por meio da permissão de acesso aos intervalos de endereços IP ou instâncias do Amazon EC2 que você especificar. É possível aplicar um grupo de segurança a uma ou mais instâncias de banco de dados.

Um uso comum de uma instância de banco de dados em uma VPC é compartilhar dados com um servidor de aplicações na mesma VPC. O exemplo a seguir usa o grupo de segurança `ec2-rds-x` de VPC para definir as regras de entrada que utilizam os endereços IP da aplicação cliente como a origem. O servidor de aplicações pertence a esse grupo de segurança. Um segundo grupo de segurança (chamado `rds-ec2-x`) especifica `ec2-rds-x` como a origem e é anexado a uma instância de banco de dados do RDS. De acordo com as regras do grupo de segurança, as aplicações cliente não podem acessar diretamente a instância de banco de dados, mas a instância do EC2 pode acessar a instância de banco de dados.



Para obter mais informações sobre grupo de seguranças, consulte Segurança no Amazon RDS.

Monitoramento do Amazon RDS

O monitoramento é uma parte importante para manter a confiabilidade, a disponibilidade e a performance do Amazon RDS e suas outras soluções da AWS. A AWS fornece várias ferramentas de monitoramento para observar o Amazon RDS, gerar relatórios quando algo estiver errado e realizar ações automáticas quando for apropriado.

É possível controlar a performance e a integridade das instâncias de banco de dados usando várias ferramentas automatizadas e manuais:

Recomendações e status da instância de banco de dados do Amazon RDS

Visualize os detalhes sobre o status atual de sua instância utilizando o console do Amazon RDS, a AWS CLI ou a API do RDS. Também é possível reagir a recomendações automatizadas para recursos de banco de dados, como instâncias de banco de dados, réplicas de leitura e grupo de parâmetros de banco de dados. Para obter mais informações, consulte Recomendações do Amazon RDS.

Métricas do Amazon CloudWatch para o Amazon RDS

Você pode usar o serviço gratuito do Amazon CloudWatch para monitorar a performance e a integridade de uma instância. Os gráficos de performance do CloudWatch são exibidos no console do Amazon RDS. O Amazon RDS envia métricas automaticamente ao CloudWatch a cada minuto para cada banco de dados ativo. Você não recebe cobranças adicionais para métricas do Amazon RDS no CloudWatch.

Usando alarmes do Amazon CloudWatch, é possível observar uma única métrica do Amazon RDS em um período específico. Depois, você pode realizar uma ou mais ações com base no valor da métrica em relação a um limite definido. Para obter mais informações, consulte Monitorar métricas do Amazon RDS com o Amazon CloudWatch.

Insights de Performance do Amazon RDS e monitoramento do sistema operacional

O Insights de Performance avalia a carga no banco de dados e determina quando e onde tomar medidas. Para obter mais informações, consulte Monitorar a carga de banco de dados com o Performance Insights no Amazon RDS. O Monitoramento avançado do Amazon RDS observa as métricas em tempo real para o sistema operacional. Para obter mais informações, consulte Monitorar métricas do SO com o monitoramento avançado.

Serviços da AWS integrados

O Amazon RDS é integrado ao Amazon EventBridge, ao Amazon CloudWatch Logs e ao Amazon DevOps Guru. Para obter mais informações, consulte Métricas de monitoramento em uma instância do Amazon RDS.

Interfaces de usuário para o Amazon RDS

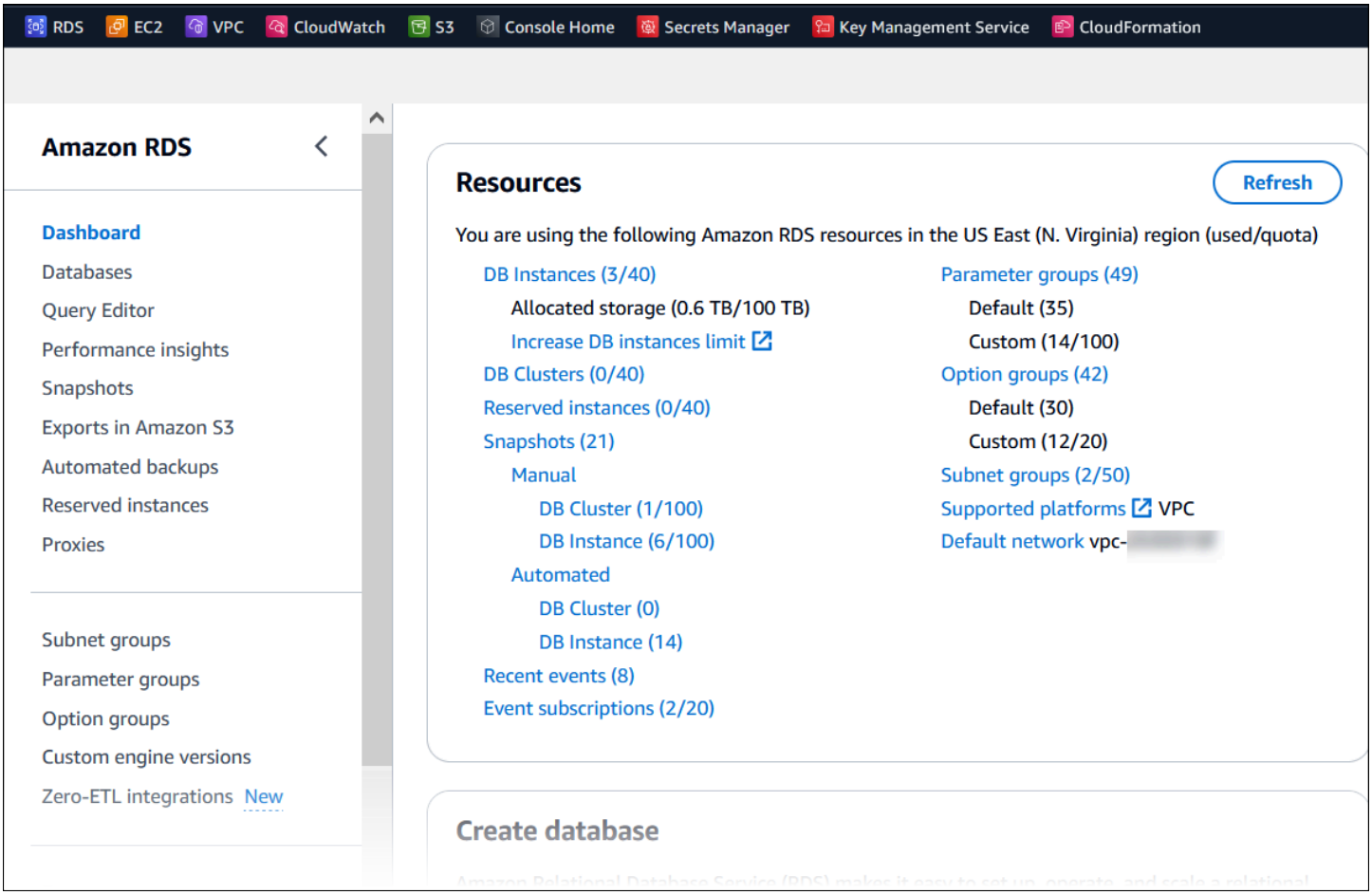
É possível interagir com o Amazon RDS de várias maneiras.

Tópicos

- AWS Management Console
- Interface da linha de comando
- APIs do Amazon RDS

AWS Management Console

O AWS Management Console é uma interface de usuário simples e baseada na Web. Você pode gerenciar suas instâncias de banco de dados a partir do console sem necessidade de programação. Para acessar o console do Amazon RDS, faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.



Interface da linha de comando

Você pode usar a AWS Command Line Interface (AWS CLI) para acessar interativamente a API do Amazon RDS. Para instalar a AWS CLI, consulte Instalação da AWSinterface da linha de comando. Para começar a usar a AWS CLI para o RDS, consulte a referência da AWS Command Line Interface para o Amazon RDS.

APIs do Amazon RDS

Se você for desenvolvedor, poderá acessar o Amazon RDS de forma programática usando APIs. Para obter mais informações, consulte Referência da API do Amazon RDS.

Para desenvolvimento de aplicativos, recomendamos usar um dos Kits de desenvolvimento de software (SDKs) da AWS. Os SDKs AWS contêm detalhes de nível baixo, como autenticação, lógica de nova tentativa e gerenciamento de erros, para que você possa se concentrar na lógica da aplicação. AWS Os SDKs estão disponíveis para uma ampla variedade de linguagens. Para obter mais informações, consulte Ferramentas para a Amazon Web Services .

AWSA também fornece bibliotecas, código de exemplo, tutoriais e outros recursos para ajudar você a começar de maneira mais fácil. Para obter mais informações, consulte Código de exemplo e bibliotecas.

Como você é cobrado pelo Amazon RDS

Quando você usa o Amazon RDS, você pode optar por usar instâncias de banco de dados sob demanda ou instâncias de banco de dados reservadas. Para obter mais informações, consulte Faturamento da instância de banco de dados para Amazon RDS.

Para obter informações sobre a definição de preço do Amazon RDS, consulte a página de produto do Amazon RDS.

Nível gratuito da AWS no Amazon RDS

Você pode usar o Nível gratuito da AWS no Amazon RDS com os seguintes mecanismos e classes de instância de banco de dados:

- Tipos de mecanismo: MariaDB, MySQL, PostgreSQL ou SQL Server Express Edition.
- Classes da instância de banco de dados
  - t3.micro: para todos os tipos de mecanismo.
  - t4g.micro: para todos os tipos de mecanismo, exceto SQL Server Express Edition.

Não é possível usar os seguintes recursos, funcionalidades e ações com o Nível gratuito da AWS no Amazon RDS:

- Opções de mecanismo do Amazon RDS Custom
- Instâncias de bancos de dados reservadas
- Outras opções de implantação além da single-AZ
- Migrar snapshots
- Editor de consultas
- Criar réplicas de leitura do Aurora

Essas limitações estão sujeitas a alterações à medida que o programa Nível gratuito da AWS evolui. Para ter mais informações sobre o Nível gratuito da AWS, consulte Explore AWS services with Nível

gratuito da AWS. Para ter mais informações sobre o nível gratuito do Amazon RDS, consulte Nível gratuito do Amazon RDS.

Se você já era cliente do Nível gratuito da AWS antes de 17 de julho de 2025, pode continuar usando o nível gratuito do Amazon RDS até o término dos 12 meses de uso gratuito. Durante esse período, você continua tendo direito ao seguinte:

- 750 horas por mês de uso das classes de instância db.t3.micro e db.t4g.micro single-AZ que executam MySQL, MariaDB ou PostgreSQL no Amazon RDS;
- 750 horas por mês de uso de uma classe de instância db.t3.micro que executa o SQL Server Express Edition no Amazon RDS.

Depois de 17 de julho de 2025, todos os recursos do Nível gratuito da AWS que você criar na AWS usarão a nova oferta do Nível gratuito da AWS.

Próximas etapas

Na seção anterior, você conheceu os componentes de infraestrutura básicos que o RDS oferece. O que você deve fazer em seguida?

Conceitos básicos

Criar uma instância de banco de dados usando as instruções em Conceitos básicos do Amazon RDS.

Tópicos específicos para mecanismos de banco de dados

Você pode rever informações específicas para um determinado mecanismo de banco de dados nas seções a seguir:

- Amazon RDS para Db2
- Amazon RDS para MariaDB
- Amazon RDS para Microsoft SQL Server
- Amazon RDS para MySQL
- Amazon RDS para Oracle
- Amazon RDS para PostgreSQL

Instâncias de banco de dados do Amazon RDS

Uma instância de banco de dados é um ambiente de banco de dados isolado em execução na nuvem. Ela é o bloco de criação básico do Amazon RDS. Uma instância de banco de dados pode conter vários bancos de dados criados por usuários e pode ser acessada usando as mesmas ferramentas e aplicativos do cliente que você pode usar para acessar uma instância de banco de dados autônoma. É simples criar e modificar instâncias de banco de dados com o uso das ferramentas da linha de comando da AWS, as operações da API do Amazon RDS ou o AWS Management Console.

> **Note**
>
> O Amazon RDS oferece suporte a acesso a bancos de dados que usam qualquer aplicação de cliente SQL padrão. O Amazon RDS não permite o acesso direto ao host, exceto com o RDS Custom.

É possível ter até 40 instâncias de banco de dados do Amazon RDS, com as seguintes limitações:

- 10 de cada edição do SQL Server (Enterprise, Standard, Web e Express) no modelo "licença incluída"
- 10 para Oracle no modelo "licença incluída"
- 40 para Db2 no modelo de licenciamento “traga a sua própria licença” (BYOL)
- 40 MySQL, MariaDB ou PostgreSQL
- 40 Oracle no modelo "bring-your-own-license" (BYOL)

> **Note**
>
> Se o aplicativo exige mais instâncias de banco de dados, é possível solicitar instâncias de banco de dados adicionais usando este formulário.

Cada instância de banco de dados tem um DB instance identifier. Este nome fornecido por cliente identifica de forma exclusiva a instância de banco de dados quando interage com os comandos da AWS CLI e da API do Amazon RDS. O identificador da instância do banco de dados deve ser exclusivo para aquele cliente em uma região da AWS.

O identificador de instância de banco de dados é usado como parte do nome do host DNS alocado à instância pelo RDS. Por exemplo, se você especificar db1 como o identificador de instância de banco de dados, o RDS alocará automaticamente um endpoint de DNS para a instância. Um exemplo de endpoint é *db1*.*abcdefghijkl*.*us-east-1*.rds.amazonaws.com, onde *db1* é o ID da instância.

No endpoint de exemplo *db1*.*abcdefghijkl*.*us-east-1*.rds.amazonaws.com, a string *abcdefghijkl* é um identificador exclusivo para uma combinação específica de Região da AWS e Conta da AWS. O identificador *abcdefghijkl* no exemplo é gerado internamente pelo RDS e não muda para a combinação especificada de região e conta. Assim, todas as instâncias de banco de dados nessa região compartilham o mesmo identificador fixo. Considere as seguintes características do identificador fixo:

- Se você renomear a instância de banco de dados, o endpoint será diferente, mas o identificador fixo será o mesmo. Por exemplo, se você renomear *db1* para *renamed-db1*, o novo endpoint da instância será *renamed-db1*.*abcdefghijkl*.*us-east-1*.rds.amazonaws.com.
- Se você excluir e recriar uma instância de banco de dados com o mesmo identificador de instância de banco de dados, o endpoint será o mesmo.
- Se você usar a mesma conta para criar uma instância de banco de dados em uma região diferente, o identificador gerado internamente será diferente porque a região é diferente, como em *db2*.*mnopqrstuvwx*.*us-west-1*.rds.amazonaws.com.

Cada instância de banco de dados oferece suporte a um mecanismo de banco de dados. No momento, o Amazon RDS comporta os mecanismos de banco de dados Db2, MySQL, MariaDB, PostgreSQL, Oracle, Microsoft SQL Server e Amazon Aurora.

Ao criar uma instância de banco de dados, alguns mecanismos de banco de dados exigem que um nome de banco de dados seja especificado. Uma instância de banco de dados pode hospedar vários bancos de dados ou um único banco de dados Oracle com vários esquemas. O valor do nome do banco de dados depende do mecanismo do banco de dados:

- Para o mecanismo do Db2, o nome do banco de dados é o nome de um banco hospedado na instância de banco de dados. Esse campo é opcional. Você pode criar um banco de dados posteriormente chamando o procedimento armazenado rdsadmin.create_database. Para ter mais informações, consulte the section called “Criação de um banco de dados”.
- Para os mecanismos de banco de dados MySQL e MariaDB, o nome do banco de dados é o nome de um banco de dados hospedado na sua instância de banco de dados. Os bancos de dados

hospedados pela mesma instância de banco de dados devem ter um nome exclusivo naquela instância.

- Para o mecanismo de banco de dados Oracle, o nome do banco de dados é usado para configurar o valor de ORACLE_SID, que deve ser fornecido quando se conectar à instância do Oracle RDS.
- Para o mecanismo de banco de dados do Microsoft SQL Server, o nome do banco de dados não é um parâmetro compatível.
- Para o mecanismo PostgreSQL, o nome do banco de dados é o nome de um banco de dados hospedado na sua instância de banco de dados. Um nome de banco de dados não é necessário ao uma instância de banco de dados. Os bancos de dados hospedados pela mesma instância de banco de dados devem ter um nome exclusivo naquela instância.

O Amazon RDS cria uma conta de usuário principal para a sua instância de banco de dados como parte do processo de criação. Este usuário principal tem permissões para criar bancos de dados e realizar operações de criação, exclusão, seleção, atualização e inserção em tabelas criadas pelo usuário principal. É necessário definir a senha de usuário principal ao criar uma instância de banco de dados, mas essa senha pode ser alterada a qualquer momento usando a AWS CLI, as operações da API do Amazon RDS ou o AWS Management Console. Você também pode usar comandos SQL padrão para alterar a senha de usuário principal e gerenciar usuários.

> **Note**
>
> Este guia abrange mecanismos de banco de dados do Amazon RDS não Aurora. Para obter informações sobre o Amazon Aurora, consulte o Guia do usuário do Amazon Aurora.

Classes de instâncias de banco de dados

A classe de instância de banco de dados determina a capacidade computacional e de memória de uma instância de banco de dados do Amazon RDS . A classe de instância de banco de dados da qual você precisa depende dos requisitos de memória e potência de processamento.

Uma classe de instância de banco de dados consiste no tipo de classe e no tamanho de instância de banco de dados. Por exemplo, db.r6g é um tipo de classe de instância de banco de dados otimizado para memória por processadores AWS Graviton2. No tipo de classe instância db.r6g, db.r6g.2xlarge é uma classe de instância de banco de dados. O tamanho dessa classe é 2xlarge.

Para ter mais informações sobre a definição de preço da classe de instância, consulte Definição de preço do Amazon RDS.

Para ter mais informações sobre tipos de classe de instância de banco de dados, mecanismos de banco de dados aceitos, Regiões da AWS compatíveis, alteração de classe de instância de banco de dados, configuração de processador para RDS para Oracle ou especificações de hardware referentes a classes de instância de banco de dados, consulte as seções a seguir.

Tópicos

- Tipos de classe de instância de banco de dados
- Mecanismos de banco de dados compatíveis para classes de instância de banco de dados
- Determinar o suporte para classes de instância de bancos de dados nas Regiões da AWS
- Alterar a classe de instância de banco de dados
- Configurar o processador para uma classe de instância de banco de dados no RDS para Oracle
- Especificações de hardware para classes de instância de banco de dados

Tipos de classe de instância de banco de dados

O Amazon RDS é compatível com as classes de instância de banco de dados para os seguintes casos de uso:

- Uso geral
- Otimizada para memória
- Otimizada para computação

- Performance expansível
- Optimized Reads

Para ter mais informações sobre os tipos de instância do Amazon EC2, consulte Tipos de instância na documentação do Amazon EC2.

Tipos de classe de instância de uso geral

As seguintes classes de instância de banco de dados de uso geral estão disponíveis:

- db.m8g: classes de instância de banco de dados de uso geral com processadores Graviton4 da AWS. Essas classes de instância fornecem computação, memória e redes balanceadas para uma grande variedade de workloads de uso geral. Em comparação com as instâncias de M7g da sétima geração baseadas em Graviton3 da AWS, essas novas classes oferecem tamanhos de instância maiores com até 3 vezes mais vCPUs e memória.

  Você pode modificar uma instância de bancos de dados para usar uma das classes de instância de banco de dados com processadores Graviton3 da AWS. Para fazer isso, conclua as mesmas etapas de qualquer outra modificação da instância de banco de dados.
- db.m7i: classes de instância de uso geral com processadores Intel Xeon escaláveis de quarta geração. As instâncias db.m7i são certificadas pela SAP e ideais para dar suporte a aplicações empresariais que precisam de tamanhos maiores de instância ou de alto uso contínuo de CPU. Essas classes de instância fornecem computação, memória e redes balanceadas para uma grande variedade de workloads de uso geral. Esse tipo de classe de instância oferece largura de banda EBS de até 40.000 Mbps e largura de banda da rede de até 50 Gbps. As instâncias db.m7i oferecem uma relação de preço/performance até 15% melhor em comparação com as instâncias db.m6i.
- db.m7g: classes de instância de banco de dados de uso geral com processadores Graviton3 da AWS. Essas classes de instância fornecem computação, memória e redes balanceadas para uma grande variedade de workloads de uso geral.

  Você pode modificar uma instância de bancos de dados para usar uma das classes de instância de banco de dados com processadores Graviton3 da AWS. Para fazer isso, conclua as mesmas etapas de qualquer outra modificação da instância de banco de dados.
- db.m6g: classes de instância de banco de dados de uso geral com processadores Graviton2 da AWS. Essas instâncias fornecem computação, memória e redes balanceadas para uma grande variedade de workloads de uso geral. As classes de instância db.m6gd têm armazenamento local

em nível de bloco SSD baseado em NVME para aplicações que precisam de armazenamento local de alta velocidade e baixa latência.

Você pode modificar uma instância de bancos de dados para usar uma das classes de instância de banco de dados com processadores Graviton2 da AWS. Para fazer isso, conclua as mesmas etapas de qualquer outra modificação da instância de banco de dados.

- db.m6id: classes de instância de uso geral com processadores Intel Xeon escaláveis de terceira geração. Essas instâncias são certificadas pela SAP e ideais para workloads (por exemplo, servidores de back-end) que comportam aplicações empresariais, servidores de jogos, frotas de armazenamento em cache e ambientes de desenvolvimento de aplicações. As classes de instância db.m6id e db.m6idn oferecem até 7,6 TB de armazenamento SSD local baseado em NVMe, enquanto a db.m6in oferece armazenamento somente EBS. As classes db.m6in e db.m6idn oferecem até 200 Gbps de largura de banda da rede.
- db.m5: classes de instância de banco de dados de uso geral que fornecem um equilíbrio de computação, memória e recursos de rede. São uma boa escolha para muitas aplicações. A classe de instância db.m5d oferece armazenamento SSD baseado em NVMe que esteja fisicamente conectado ao servidor host. As classes de instância db.m5 fornecem mais capacidade de computação do que as classes de instância anteriores, db.m4. Elas são desenvolvidas pelo Nitro System da AWS, uma combinação de hardware dedicado e hipervisor leve.
- db.m4: classes de instância de banco de dados de uso geral que fornecem maior capacidade computacional do que as classes de instância anteriores db.m3.

  Para os mecanismos de banco de dados do RDS para Oracle, o Amazon RDS não permite mais as classes de instância de banco de dados db.m4. Se você já havia criado anteriormente as instâncias de banco de dados db.m4 do RDS para Oracle, o Amazon RDS atualiza automaticamente essas instâncias de banco de dados para classes de instância de banco de dados db.m5 equivalentes.

  Para os mecanismos de bancos de dados do RDS para MariaDB, do RDS para MySQL, do RDS para SQL Server e do RDS para PostgreSQL, o Amazon RDS iniciará o processo de fim de suporte para essa classe de instância de banco de dados usando a programação a seguir. Com relação a todas as instâncias de bancos de dados do RDS que usam essa classe de instância, recomendamos fazer upgrade para uma classe de instância de banco de dados de geração mais recente o quanto antes.

Ação ou recomendação	Data
A partir dessa data, o Amazon RDS começou a fazer upgrade automaticamente das instâncias usando db.m4 para a classe de instância db.m5 da geração mais recente. Não é mais possível criar instâncias de banco de dados usando a classe de instância db.m4.	1.º de junho de 2024
O Amazon RDS encerra o suporte para db.m4.	31 de dezembro de 2024

- db.m3: classes de instância de banco de dados de uso geral que fornecem maior capacidade computacional do que as classes de instância anteriores db.m1.

  Para os mecanismos de bancos de dados do RDS para MariaDB, do RDS para MySQL e do RDS para PostgreSQL, o Amazon RDS iniciará o processo de fim de vida útil das classes de bancos de dados db.m3 usando a seguinte programação, que inclui recomendações de atualização. Para todas as instâncias de bancos de dados do RDS que usam classes de instância de banco de dados db.m3, recomendamos realizar a atualização para uma classe de instância de banco de dados de geração posterior o quanto antes.

Ação ou recomendação	Datas
Não é mais possível criar instâncias de bancos de dados do RDS que usem as classes de instâncias de banco de dados db.m3.	Agora
O Amazon RDS iniciou atualizações automáticas de instâncias de bancos de dados do RDS que utilizam classes de instância de banco de dados db.m3 para classes de instância de banco de dados db.m5 equivalentes.	1° de fevereiro de 2023

Tipos de classe de instância otimizada para memória

A família Z otimizada para memória comporta as seguintes classes de instância:

- db.z1d Classes de – instância otimizadas para aplicações com uso intenso de memória. Essas classes de instância oferecem alta capacidade computacional e grandes quantidades de memória. As instâncias z1d de alta frequência oferecem uma frequência contínua de até 4,0 GHz em todos os núcleos.

A família X otimizada para memória comporta as seguintes classes de instância:

- db.x2g: classes de instância otimizada para aplicações com uso intensivo de memória e baseadas em processadores Graviton2 da AWS. Essas classes de instância oferecem custo baixo por GiB de memória.

  Você pode modificar uma instância de bancos de dados para usar uma das classes de instância de banco de dados com processadores Graviton2 da AWS. Para fazer isso, conclua as mesmas etapas de qualquer outra modificação da instância de banco de dados.
- db.x2i: classes de instância otimizada para aplicações com uso intensivo de memória. Os tipos de classe de instância db.x2iedn e db.x2idn utilizam processadores Intel Xeon escaláveis de terceira geração (Ice Lake). Elas incluem até 3,8 TB de armazenamento SSD NVMe local, até 100 Gbps de largura de banda de rede e até 4 TiB (db.x2iden) ou 2 TiB (db.x2idn) de memória. O tipo db.x2iezn utiliza processadores Intel Xeon escaláveis de segunda geração (Cascade Lake) com frequência turbo de até 4,5 GHz e até 1,5 TiB de memória.
- db.x1 – classes de instância otimizadas para aplicações com uso intensivo de memória. Essas classes de instância oferecem um dos menores preços por GiB de RAM entre as classes de instância de banco de dados e até 1.952 GiB de memória de instância baseada em DRAM. O tipo de classe de instância db.x1e oferece até 3.904 GiB de memória de instância baseada em DRAM.

A família R otimizada para memória é compatível com os seguintes tipos de classe de instância:

- db.r8g: classes de instância com processadores Graviton4 da AWS. Essas classes de instância são ideais para executar workloads com uso intensivo de memória em bancos de dados de código aberto, como MySQL e PostgreSQL. Essas instâncias oferecem tamanhos de instância maiores com até 3 vezes mais vCPUs e memória do que as instâncias db.r7g da sétima geração baseadas em Graviton3 da AWS.

  Você pode modificar uma instância de bancos de dados para usar uma das classes de instância de banco de dados com processadores Graviton3 da AWS. Para fazer isso, conclua as mesmas etapas de qualquer outra modificação da instância de banco de dados.

- db.r7g: classes de instância com processadores Graviton3 da AWS. Essas classes de instância são ideais para executar workloads com uso intensivo de memória em bancos de dados de código aberto, como MySQL e PostgreSQL.

  Você pode modificar uma instância de bancos de dados para usar uma das classes de instância de banco de dados com processadores Graviton3 da AWS. Para fazer isso, conclua as mesmas etapas de qualquer outra modificação da instância de banco de dados.
- db.r7i: classes de instância com processadores Intel Xeon escaláveis de quarta geração. Essas classes de instância são certificadas pela SAP e ideais para executar workloads com uso intenso de memória em bancos de dados de código aberto, como MySQL e PostgreSQL. Você pode modificar uma instância de bancos de dados para usar uma das classes de instância de banco de dados com processadores Intel Xeon escaláveis de quarta geração. Para fazer isso, conclua as mesmas etapas de qualquer outra modificação da instância de banco de dados.
- db.r6g: classes de instância com processadores Graviton2 da AWS. Essas classes de instância são ideais para executar workloads com uso intensivo de memória em bancos de dados de código aberto, como MySQL e PostgreSQL. O tipo db.r6gd oferece armazenamento ao nível do bloco SSD baseado em NVME local para aplicações que precisam de armazenamento local de alta velocidade e baixa latência.

  Você pode modificar uma instância de bancos de dados para usar uma das classes de instância de banco de dados com processadores Graviton2 da AWS. Para fazer isso, conclua as mesmas etapas de qualquer outra modificação da instância de banco de dados.
- db.r6id: classes de instância com processadores Intel Xeon escaláveis de terceira geração. Essas classes de instância são certificadas pela SAP e ideais para executar workloads com uso intenso de memória em bancos de dados de código aberto, como MySQL e PostgreSQL. As classes de instância db.r6id, db.r6in e db.r6idn têm uma proporção de memória/vCPU de 8:1 e memória máxima de 1 TiB. As classes db.r6id e db.r6idn oferecem até 7,6 TB de armazenamento SSD baseado em NVMe de conexão direta, enquanto a db.r6in oferece armazenamento somente EBS. As classes db.r6idn and db.r6in oferecem até 200 Gbps de largura de banda da rede.
- db.r5b – classes de instância otimizadas para memória para aplicativos com uso intensivo de taxa de transferência. Com a tecnologia do Nitro System da AWS, as instâncias db.r5b fornecem até 60 Gbps de largura de banda e 260.000 IOPS de performance do EBS. Essa é a performance de armazenamento em blocos mais rápido no EC2.
- db.r5d: classes de instância otimizada para baixa latência, performance de E/S aleatória muito alta e alta taxa de transferência de leitura sequencial.

- db.r5: classes de instância otimizada para aplicações com uso intenso de memória. Essas classes de instância oferecem performance aprimorada de rede . Elas são desenvolvidas pelo Nitro System da AWS, uma combinação de hardware dedicado e hipervisor leve.
- db.r4: classes de instância que fornecem rede aprimorada em relação às classes de instância db.r3 anteriores.

  Para o mecanismo de banco de dados do RDS para Oracle, o Amazon RDS iniciará o processo de fim de vida útil das classes de instância de banco de dados db.r4 usando a seguinte programação, que inclui recomendações de atualização. Para as instâncias de banco de dados do RDS para Oracle que usam classes de instância db.r4, recomendamos realizar a atualização para uma instância de geração posterior o quanto antes.

Ação ou recomendação	Datas
Não é mais possível criar instâncias de banco de dados do RDS para Oracle que usem as classes de instâncias de banco de dados db.r4.	Agora
O Amazon RDS iniciou atualizações automáticas de instâncias de banco de dados do RDS para Oracle que usam classes de instância de banco de dados db.r4 para classes de instância de banco de dados db.r5 equivalentes.	17 de abril de 2023

Para os mecanismos de bancos de dados do RDS para MariaDB, do RDS para MySQL, do RDS para SQL Server e do RDS para PostgreSQL, o Amazon RDS iniciará o processo de fim de suporte para essa classe de instância de banco de dados usando a programação a seguir. Com relação a todas as instâncias de bancos de dados do RDS que usam essa classe de instância, recomendamos fazer upgrade para uma classe de instância de banco de dados de geração mais recente o quanto antes.

Ação ou recomendação	Datas
A partir dessa data, o Amazon RDS começou a fazer upgrade automaticamente das instâncias	1.º de junho de 2024

Ação ou recomendação	Datas
usando db.r4 para a classe de instância db.r5 da geração mais recente. Não é mais possível criar instâncias de banco de dados usando a classe de instância db.m4.	
O Amazon RDS encerra o suporte para db.r4.	31 de dezembro de 2024

- db.r3 – classes de instância que fornecem otimização de memória.

  Para os mecanismos de bancos de dados do RDS para MariaDB, do RDS para MySQL e do RDS para PostgreSQL, o Amazon RDS iniciou o processo de fim de vida útil de classes de instância de banco de dados db.r3 utilizando a programação a seguir, que inclui recomendações de atualização. Para todas as instâncias de bancos de dados do RDS que usam classes de instância de banco de dados db.r3, recomendamos realizar a atualização para uma classe de instância de banco de dados de geração posterior o quanto antes.

Ação ou recomendação	Datas
Não é mais possível criar instâncias de bancos de dados do RDS que usem as classes de instâncias de banco de dados db.r3.	Agora
O Amazon RDS iniciou atualizações automáticas de instâncias de bancos de dados do RDS que utilizam classes de instância de banco de dados db.r3 para classes de instância de banco de dados db.r5 equivalentes.	1° de fevereiro de 2023

Tipo de classe de instância otimizada para computação

Os seguintes tipos de classe de instância otimizada para computação estão disponíveis:

- db.c6gd: classes de instâncias ideais para executar workloads avançadas que fazem uso intenso de computação. Desenvolvidas com processadores AWS Graviton2, essas classes de instância oferecem armazenamento ao nível do bloco local de SSD baseado em NVMe para aplicações que precisam de armazenamento local de alta velocidade e baixa latência.

> **Note**
>
> As classes de instância c6gd são compatíveis somente com implantações de cluster de banco de dados Multi-AZ. Elas são a única classe de instância compatível com clusters de banco de dados multi-AZ que oferecem o tamanho `medium` de instância. Para obter mais informações, consulte the section called “Implantações de clusters de banco de dados multi-AZ”.

Tipos de classe de instância de performance expansível

Os seguintes tipos de classe de instância de banco de dados de performance expansível estão disponíveis:

- db.t4g: classes de instância de uso geral com processadores Graviton2 da AWS baseados em ARM. Essas classes de instância oferecem uma relação de performance e preço melhor do que as classes de instância de banco de dados de intermitência anteriores para um amplo conjunto de workloads de uso geral. As instâncias db.t4g do Amazon RDS são configuradas para o modo Ilimitado. Isso significa que elas podem se expandir além da linha de base em uma janela de 24 horas por um custo adicional.

  Você pode modificar uma instância de bancos de dados para usar uma das classes de instância de banco de dados com processadores Graviton2 da AWS. Para fazer isso, conclua as mesmas etapas de qualquer outra modificação da instância de banco de dados.
- db.t3: classes de instância que fornecem um nível básico de performance, com capacidade de intermitência para o uso total da CPU. As instâncias db.t3 são configuradas para o modo Ilimitado. Essas classes fornecem mais capacidade de computação do que as classes de instância anteriores, db.t2. Elas são desenvolvidas pelo Nitro System da AWS, uma combinação de hardware dedicado e hipervisor leve.
- db.t2 – classes de instância que fornecem um nível básico de performance, com capacidade de intermitência para o uso total da CPU. As instâncias db.t2 são configuradas para o modo Ilimitado. Recomendamos só usar essas classes de instância para servidores de desenvolvimento e de teste ou outros servidores que não sejam de produção.

  Para os mecanismos de bancos de dados do RDS para MariaDB, do RDS para MySQL, do RDS para SQL Server e do RDS para PostgreSQL, o Amazon RDS iniciará o processo de fim de suporte para essa classe de instância de banco de dados usando a programação a seguir. Com

relação a todas as instâncias de bancos de dados do RDS que usam essa classe de instância, recomendamos fazer upgrade para uma classe de instância de banco de dados de geração mais recente o quanto antes.

Ação ou recomendação	Datas
A partir dessa data, o Amazon RDS começou a fazer upgrade automaticamente das instâncias usando db.t2 para a classe de instância db.t3 da geração mais recente. Não é mais possível criar instâncias de banco de dados usando a classe de instância db.t2.	1.º de junho de 2024
O Amazon RDS encerra o suporte para db.t2.	31 de dezembro de 2024

> **Note**
>
> As classes de instância de Bancos de Dados que usam o Nitro System AWS (db.m5, db.r5, db.t3) são limitadas na workload combinada de leitura e escrita.

Para especificações de hardware para classes de instância de banco de dados, consulte Especificações de hardware para classes de instância de banco de dados .

Tipos de classe de instância do Optimized Reads

Os seguintes tipos de classe de instância do Optimized Reads disponíveis são:

- db.r6gd: classes de instância com processadores Graviton2 da AWS. Essas classes de instância são ideais para executar workloads que fazem uso intenso de memória e oferecem armazenamento ao nível do bloco SSD baseado em NVME local para aplicações que precisam de armazenamento local de alta velocidade e baixa latência.
- db.r6id: classes de instância com processadores Intel Xeon escaláveis de terceira geração. Essas classes de instância são certificadas pela SAP e ideais para executar workloads com uso intenso de memória. Elas oferecem memória máxima de 1 TiB e até 7,6 TB de armazenamento SSD baseado em NVMe de conexão direta.

Mecanismos de banco de dados compatíveis para classes de instância de banco de dados

As considerações a seguir são específicas sobre o mecanismo de banco de dados para classes de instâncias de banco de dados:

Db2

O suporte a classes de instância de banco de dados varia de acordo com a versão e a edição do Db2. Para obter informações sobre o suporte a classes de instância por versão e edição, consulte Classes de instância do Amazon RDS para Db2.

Microsoft SQL Server

O suporte a classes de instância de banco de dados varia de acordo com a versão e a edição do SQL Server. Para obter informações sobre o suporte a classes de instância por versão e edição, consulte Suporte a classes de instâncias de banco de dados para o Microsoft SQL Server.

Oracle

O suporte a classes de instância de banco de dados varia de acordo com a versão e a edição do Oracle Database. O RDS for Oracle é compatível com as classesde instância otimizada para memória adicionais. Essas classes têm nomes no formato db.r5.*instance_size*.tpc*threads_per_core*mem*ratio*. Para a contagem de vCPU e alocação de memória para cada classe otimizada, consulte Classes de instância de banco de dados compatíveis do RDS para Oracle.

RDS Custom

Para receber informações sobre as classes de instância de banco de dados compatíveis com o RDS Custom, consulte Suporte a classes de instâncias de banco de dados do RDS Custom for Oracle e Compatibilidade com classes de instâncias de banco de dados do RDS Custom for SQL Server.

Na tabela a seguir, você pode encontrar detalhes sobre as classes de instâncias de banco de dados do Amazon RDS com suporte para cada mecanismo de banco de dados do Amazon RDS. A célula de cada mecanismo contém um dos seguintes valores:

Sim

A classe de instância é compatível com todas as versões do mecanismo de banco de dados.

Não

A classe de instância não é compatível com o mecanismo de banco de dados.

specific-versions

A classe de instância é compatível somente com as versões de banco de dados especificadas do mecanismo.

O Amazon RDS descontinua periodicamente as versões principais e secundárias do mecanismo do banco de dados. Talvez nem todas as Regiões da AWS sejam compatíveis com versões anteriores do mecanismo. Para obter informações sobre as versões atuais compatíveis, consulte os tópicos para os mecanismos de banco de dados individuais: versões do MariaDB, versões do Microsoft SQL Server, versões do MySQL, versões do Oracle e versões do PostgreSQL.

Tópicos

- Mecanismos de banco de dados compatíveis com classes de instância de uso geral
- Mecanismos de banco de dados compatíveis com classes de instância otimizada para memória
- Mecanismos de banco de dados compatíveis com classes de instância otimizada para computação
- Mecanismos de banco de dados compatíveis para classes de instância de desempenho expansível
- Mecanismos de banco de dados compatíveis com classes de instância de leituras otimizadas

Mecanismos de banco de dados compatíveis com classes de instância de uso geral

As tabelas a seguir mostram os bancos de dados e as versões de banco de dados compatíveis com as classes de instância de uso geral.

db.m7g: classes de instância de uso geral com processadores Graviton3 da AWS

Classe de instância	Db2	MariaDB	Micros SQL Server	MySQL	Orac	PostgreSQL
db.m7g.1(xlarge	Não	MariaDB versões 10.11, 10.6.10 e versões 10.6	Não	MySQL 8.0.28 e posterior	Não	Todas as versões do PostgreSQ

Classe de instância	Db2	MariaDB	Micros SQL Server	MySQL	Orac	PostgreSQL
		posteriores, 10.5.17 e versões 10.5 posteriores, 10.4.26 e versões 10.4 posteriores				L 16 e 15, 14.5 e versões 14 posteriores e 13.4 e versões 13 posteriores
db.m7g.1 xlarge	Não	MariaDB versões 10.11, 10.6.10 e versões 10.6 posteriores, 10.5.17 e versões 10.5 posteriores, 10.4.26 e versões 10.4 posteriores	Não	MySQL 8.0.28 e posterior	Não	Todas as versões do PostgreSQ L 16 e 15, 14.5 e versões 14 posteriores e 13.4 e versões 13 posteriores
db.m7g.8 large	Não	MariaDB versões 10.11, 10.6.10 e versões 10.6 posteriores, 10.5.17 e versões 10.5 posteriores, 10.4.26 e versões 10.4 posteriores	Não	MySQL 8.0.28 e posterior	Não	Todas as versões do PostgreSQ L 16 e 15, 14.5 e versões 14 posteriores e 13.4 e versões 13 posteriores
db.m7g.4 large	Não	MariaDB versões 10.11, 10.6.10 e versões 10.6 posteriores, 10.5.17 e versões 10.5 posteriores, 10.4.26 e versões 10.4 posteriores	Não	MySQL 8.0.28 e posterior	Não	Todas as versões do PostgreSQ L 16 e 15, 14.5 e versões 14 posteriores e 13.4 e versões 13 posteriores

Classe de instância	Db2	MariaDB	Micros SQL Server	MySQL	Orac	PostgreSQL
db.m7g.2x large	Não	MariaDB versões 10.11, 10.6.10 e versões 10.6 posteriores, 10.5.17 e versões 10.5 posteriores, 10.4.26 e versões 10.4 posteriores	Não	MySQL 8.0.28 e posterior	Não	Todas as versões do PostgreSQ L 16 e 15, 14.5 e versões 14 posteriores e 13.4 e versões 13 posteriores
db.m7g.xl arge	Não	MariaDB versões 10.11, 10.6.10 e versões 10.6 posteriores, 10.5.17 e versões 10.5 posteriores, 10.4.26 e versões 10.4 posteriores	Não	MySQL 8.0.28 e posterior	Não	Todas as versões do PostgreSQ L 16 e 15, 14.5 e versões 14 posteriores e 13.4 e versões 13 posteriores
db.m7g.la rge	Não	MariaDB versões 10.11, 10.6.10 e versões 10.6 posteriores, 10.5.17 e versões 10.5 posteriores, 10.4.26 e versões 10.4 posteriores	Não	MySQL 8.0.28 e posterior	Não	Todas as versões do PostgreSQ L 16 e 15, 14.5 e versões 14 posteriores e 13.4 e versões 13 posteriores

db.m6g: classes de instância de uso geral com processadores Graviton2 da AWS

Classe de instância	Db2	MariaDB	Micros SQL Server	MySQL	Ora	PostgreSQL
db.m6g.1(xlarge	Não	Todas as versões do MariaDB 10.11, 10.6, 10.5 e 10.4	Não	MySQL 8.0.23 e posteriores	Nãc	Todas as versões do PostgreSQL 16, 15, 14 e 13; e 12.7 e

Classe de instância	Db2	MariaDB	Micros SQL Server	MySQL	Ora	PostgreSQL
						versões 12 posterior es
db.m6g.1 xlarge	Não	Todas as versões do MariaDB 10.11, 10.6, 10.5 e 10.4	Não	MySQL 8.0.23 e posteriores	Não	Todas as versões do PostgreSQL 16, 15, 14 e 13; e 12.7 e versões 12 posterior es
db.m6g.8 large	Não	Todas as versões do MariaDB 10.11, 10.6, 10.5 e 10.4	Não	MySQL 8.0.23 e posteriores	Não	Todas as versões do PostgreSQL 16, 15, 14 e 13; e 12.7 e versões 12 posterior es
db.m6g.4 large	Não	Todas as versões do MariaDB 10.11, 10.6, 10.5 e 10.4	Não	MySQL 8.0.23 e posteriores	Não	Todas as versões do PostgreSQL 16, 15, 14 e 13; e 12.7 e versões 12 posterior es
db.m6g.2 large	Não	Todas as versões do MariaDB 10.11, 10.6, 10.5 e 10.4	Não	MySQL 8.0.23 e posteriores	Não	Todas as versões do PostgreSQL 16, 15, 14 e 13; e 12.7 e versões 12 posterior es
db.m6g.xl arge	Não	Todas as versões do MariaDB 10.11, 10.6, 10.5 e 10.4	Não	MySQL 8.0.23 e posteriores	Não	Todas as versões do PostgreSQL 16, 15, 14 e 13; e 12.7 e versões 12 posterior es

Classe de instância	Db2	MariaDB	Micros SQL Server	MySQL	Ora	PostgreSQL
db.m6g.large	Não	Todas as versões do MariaDB 10.11, 10.6, 10.5 e 10.4	Não	MySQL 8.0.23 e posteriores	Não	Todas as versões do PostgreSQL 16, 15, 14 e 13; e 12.7 e versões 12 posteriores

db.m6gd: classes de instância de uso geral com processadores Graviton2 da AWS e armazenamento SSD

Classe de instância	Db2	MariaDB	Micros SQL Server	MySQL	Ora	PostgreSQL
db.m6gd.16xlarge	Não	MariaDB versões 10.11, 10.6.7 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Não	MySQL 8.0.28 e posterior	Não	Todas as versões do PostgreSQL 16, 15, e 14; 13.7 e versões 13 posteriores; e 13.4
db.m6gd.12xlarge	Não	MariaDB versões 10.11, 10.6.7 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Não	MySQL 8.0.28 e posterior	Não	Todas as versões do PostgreSQL 16, 15, e 14; 13.7 e versões 13 posteriores; e 13.4
db.m6gd.8xlarge	Não	MariaDB versões 10.11, 10.6.7 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posteriores,	Não	MySQL 8.0.28 e posterior	Não	Todas as versões do PostgreSQL 16, 15, e 14; 13.7 e versões 13 posteriores; e 13.4

Classe de instância	Db2	MariaDB	Micros SQL Server	MySQL	Ora	PostgreSQL
		10.4.25 e versões 10.4 posteriores				
db.m6gd.4xlarge	Não	MariaDB versões 10.11, 10.6.7 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Não	MySQL 8.0.28 e posterior	Não	Todas as versões do PostgreSQL 16, 15, e 14; 13.7 e versões 13 posteriores; e 13.4
db.m6gd.2xlarge	Não	MariaDB versões 10.11, 10.6.7 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Não	MySQL 8.0.28 e posterior	Não	Todas as versões do PostgreSQL 16, 15, e 14; 13.7 e versões 13 posteriores; e 13.4
db.m6gd.xlarge	Não	MariaDB versões 10.11, 10.6.7 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Não	MySQL 8.0.28 e posterior	Não	Todas as versões do PostgreSQL 16, 15, e 14; 13.7 e versões 13 posteriores; e 13.4
db.m6gd.large	Não	MariaDB versões 10.11, 10.6.7 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Não	MySQL 8.0.28 e posterior	Não	Todas as versões do PostgreSQL 16, 15, e 14; 13.7 e versões 13 posteriores; e 13.4

db.m6id: classes de instância de uso geral com processadores Intel Xeon escaláveis de terceira geração e armazenamento SSD

Classe de instância	Db2	MariaDB	Micros SQL Server	MySQL	Ora	PostgreSQL
db.m6id.32xlarge	Não	MariaDB 10.6.10 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Não	MySQL versão 8.0.28 e posteriores	Não	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores e 13.7 e versões 13 posteriores
db.m6id.24xlarge	Não	MariaDB 10.6.10 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Não	MySQL versão 8.0.28 e posteriores	Não	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores e 13.7 e versões 13 posteriores
db.m6id.16xlarge	Não	MariaDB 10.6.10 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Não	MySQL versão 8.0.28 e posteriores	Não	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores e 13.7 e versões 13 posteriores
db.m6id.12xlarge	Não	MariaDB 10.6.10 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Não	MySQL versão 8.0.28 e posteriores	Não	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores e 13.7 e versões 13 posteriores
db.m6id.8xlarge	Não	MariaDB 10.6.10 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Não	MySQL versão 8.0.28 e posteriores	Não	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores e 13.7 e versões 13 posteriores
db.m6id.4xlarge	Não	MariaDB 10.6.10 e versões 10.6 posteriores, 10.5.16 e versões 10.5	Não	MySQL versão 8.0.28 e posteriores	Não	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14

Classe de instância	Db2	MariaDB	Micros SQL Server	MySQL	Ora	PostgreSQL
		posteriores, 10.4.25 e versões 10.4 posteriores				posteriores e 13.7 e versões 13 posteriores
db.m6id.2xlarge	Não	MariaDB 10.6.10 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Não	MySQL versão 8.0.28 e posteriores	Não	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores e 13.7 e versões 13 posteriores
db.m6id.xlarge	Não	MariaDB 10.6.10 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Não	MySQL versão 8.0.28 e posteriores	Não	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores e 13.7 e versões 13 posteriores
db.m6id.large	Não	MariaDB 10.6.10 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Não	MySQL versão 8.0.28 e posteriores	Não	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores e 13.7 e versões 13 posteriores

db.m6idn: classes de instância de uso geral com processadores escaláveis Intel Xeon de terceira geração, armazenamento SSD e otimização de rede

Classe de instância	Db	MariaDB	Micros SQL Server	MySQL	Orac	PostgreSQL
db.m6idn.32xlarge	Não	MariaDB versão 10.6.8 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Não	MySQL versão 8.0.28 e posteriores	Não	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores e 13.7 e versões 13 posteriores

Classe de instância	Db	MariaDB	Micros SQL Server	MySQL	Orac	PostgreSQL
db.m6idn.24xlarge	Nã	MariaDB versão 10.6.8 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Não	MySQL versão 8.0.28 e posteriores	Não	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores e 13.7 e versões 13 posteriores
db.m6idn.16xlarge	Nã	MariaDB versão 10.6.8 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Não	MySQL versão 8.0.28 e posteriores	Não	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores e 13.7 e versões 13 posteriores
db.m6idn.12xlarge	Nã	MariaDB versão 10.6.8 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Não	MySQL versão 8.0.28 e posteriores	Não	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores e 13.7 e versões 13 posteriores
db.m6idn.8xlarge	Sir	MariaDB versão 10.6.8 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Não	MySQL versão 8.0.28 e posteriores	Não	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores e 13.7 e versões 13 posteriores
db.m6idn.4xlarge	Sir	MariaDB versão 10.6.8 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Não	MySQL versão 8.0.28 e posteriores	Não	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores e 13.7 e versões 13 posteriores

Classe de instância	Db	MariaDB	Micros SQL Server	MySQL	Orac	PostgreSQL
db.m6idn.2xlarge	Sir	MariaDB versão 10.6.8 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Não	MySQL versão 8.0.28 e posteriores	Não	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores e 13.7 e versões 13 posteriores
db.m6idn.xlarge	Sir	MariaDB versão 10.6.8 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Não	MySQL versão 8.0.28 e posteriores	Não	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores e 13.7 e versões 13 posteriores
db.m6idn.large	Sir	MariaDB versão 10.6.8 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Não	MySQL versão 8.0.28 e posteriores	Não	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores e 13.7 e versões 13 posteriores

db.m6in: classes de instância de uso geral com processadores escaláveis Intel Xeon de terceira geração e otimização de rede

Classe de instância	Db	MariaDB	Micros SQL Server	MySQL	Orac	PostgreSQL
db.m6in.32xlarge	Nã	MariaDB versão 10.6.8 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Não	MySQL versão 8.0.28 e posteriores	Não	Todas as versões 16 e 15 do PostgreSQL, 14.3 e versões 14 posteriores, 13.7 e versões 13 posteriores, 12.11 e versões 12 posteriores

Classe de instância	Db	MariaDB	Micros SQL Serve	MySQL	Ora	PostgreSQL
						e 11.16 e versões 11 posteriores
db.m6in.24xlarge	Nã	MariaDB versão 10.6.8 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Não	MySQL versão 8.0.28 e posteriores	Não	Todas as versões 16 e 15 do PostgreSQL, 14.3 e versões 14 posteriores, 13.7 e versões 13 posteriores, 12.11 e versões 12 posteriores e 11.16 e versões 11 posteriores
db.m6in.16xlarge	Nã	MariaDB versão 10.6.8 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Não	MySQL versão 8.0.28 e posteriores	Não	Todas as versões 16 e 15 do PostgreSQL, 14.3 e versões 14 posteriores, 13.7 e versões 13 posteriores, 12.11 e versões 12 posteriores e 11.16 e versões 11 posteriores
db.m6in.16xlarge	Nã	MariaDB versão 10.6.8 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Não	MySQL versão 8.0.28 e posteriores	Não	Todas as versões 16 e 15 do PostgreSQL, 14.3 e versões 14 posteriores, 13.7 e versões 13 posteriores, 12.11 e versões 12 posteriores e 11.16 e versões 11 posteriores

Classe de instância	Db	MariaDB	Micros SQL Serve	MySQL	Ora	PostgreSQL
db.m6in.8xlarge	Sir	MariaDB versão 10.6.8 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Não	MySQL versão 8.0.28 e posteriores	Não	Todas as versões 16 e 15 do PostgreSQL, 14.3 e versões 14 posteriores, 13.7 e versões 13 posteriores, 12.11 e versões 12 posteriores e 11.16 e versões 11 posteriores
db.m6in.4xlarge	Sir	MariaDB versão 10.6.8 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Não	MySQL versão 8.0.28 e posteriores	Não	Todas as versões 16 e 15 do PostgreSQL, 14.3 e versões 14 posteriores, 13.7 e versões 13 posteriores, 12.11 e versões 12 posteriores e 11.16 e versões 11 posteriores
db.m6i.2xlarge	Sir	MariaDB versão 10.6.8 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Não	MySQL versão 8.0.28 e posteriores	Não	Todas as versões 16 e 15 do PostgreSQL, 14.3 e versões 14 posteriores, 13.7 e versões 13 posteriores, 12.11 e versões 12 posteriores e 11.16 e versões 11 posteriores

Classe de instância	Db	MariaDB	Micros SQL Serve	MySQL	Ora	PostgreSQL
db.m6 em xlarge	Sir	MariaDB versão 10.6.8 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posterior es, 10.4.25 e versões 10.4 posteriores	Não	MySQL versão 8.0.28 e posteriores	Não	Todas as versões 16 e 15 do PostgreSQL, 14.3 e versões 14 posterior es, 13.7 e versões 13 posteriores, 12.11 e versões 12 posteriores e 11.16 e versões 11 posteriores
db.m6in.l arge	Sir	MariaDB versão 10.6.8 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posterior es, 10.4.25 e versões 10.4 posteriores	Não	MySQL versão 8.0.28 e posteriores	Não	Todas as versões 16 e 15 do PostgreSQL, 14.3 e versões 14 posterior es, 13.7 e versões 13 posteriores, 12.11 e versões 12 posteriores e 11.16 e versões 11 posteriores

db.m6i: classes de instância de uso geral com processadores Intel Xeon escaláveis de terceira geração

Classe de instância	Db	MariaDB	Micros SQL Serve	MySQL	Oracle	PostgreSQL
db.m6i.32 xlarge	Sir	MariaDB versões 10.11, 10.6.7 e versões 10.6 posteriores, 10.5.15 e versões 10.5 posteriores, 10.4.24 e versões 10.4 posterior es	Sim	MySQL versão 8.0.28 e posteriores	Oracle Databa 19c	Todas as versões do PostgreSQL 16, 15, e 14; 13.4, 12.8 e 11.13 e versões 11 posteriores

Classe de instância	Db	MariaDB	Micros SQL Server	MySQL	Oracle	PostgreSQL
db.m6i.24 xlarge	Sir	MariaDB versões 10.11, 10.6.7 e versões 10.6 posteriores, 10.5.15 e versões 10.5 posteriores, 10.4.24 e versões 10.4 posterior es	Sim	MySQL versão 8.0.28 e posteriores	Oracle Databa 19c	Todas as versões do PostgreSQL 16, 15, e 14; 13.4, 12.8 e 11.13 e versões 11 posteriores
db.m6i.16 xlarge	Sir	MariaDB versões 10.11, 10.6.7 e versões 10.6 posteriores, 10.5.15 e versões 10.5 posteriores, 10.4.24 e versões 10.4 posterior es	Sim	MySQL versão 8.0.28 e posteriores	Oracle Databa 19c	Todas as versões do PostgreSQL 16, 15, e 14; 13.4, 12.8 e 11.13 e versões 11 posteriores
db.m6i.12 xlarge	Sir	MariaDB versões 10.11, 10.6.7 e versões 10.6 posteriores, 10.5.15 e versões 10.5 posteriores, 10.4.24 e versões 10.4 posterior es	Sim	MySQL versão 8.0.28 e posteriores	Oracle Databa 19c	Todas as versões do PostgreSQL 16, 15, e 14; 13.4, 12.8 e 11.13 e versões 11 posteriores
db.m6i.8x large	Sir	MariaDB versões 10.11, 10.6.7 e versões 10.6 posteriores, 10.5.15 e versões 10.5 posteriores, 10.4.24 e versões 10.4 posterior es	Sim	MySQL versão 8.0.28 e posteriores	Oracle Databa 19c	Todas as versões do PostgreSQL 16, 15, e 14; 13.4, 12.8 e 11.13 e versões 11 posteriores

Classe de instância	Db	MariaDB	Micros SQL Server	MySQL	Oracle	PostgreSQL
db.m6i.4x large	Sir	MariaDB versões 10.11, 10.6.7 e versões 10.6 posteriores, 10.5.15 e versões 10.5 posteriores, 10.4.24 e versões 10.4 posterior es	Sim	MySQL versão 8.0.28 e posteriores	Oracle Databa 19c	Todas as versões do PostgreSQL 16, 15, e 14; 13.4, 12.8 e 11.13 e versões 11 posteriores
db.m6i.2x large	Sir	MariaDB versões 10.11, 10.6.7 e versões 10.6 posteriores, 10.5.15 e versões 10.5 posteriores, 10.4.24 e versões 10.4 posterior es	Sim	MySQL versão 8.0.28 e posteriores	Oracle Databa 19c	Todas as versões do PostgreSQL 16, 15, e 14; 13.4, 12.8 e 11.13 e versões 11 posteriores
db.m6i.xl arge	Sir	MariaDB versões 10.11, 10.6.7 e versões 10.6 posteriores, 10.5.15 e versões 10.5 posteriores, 10.4.24 e versões 10.4 posterior es	Sim	MySQL versão 8.0.28 e posteriores	Oracle Databa 19c	Todas as versões do PostgreSQL 16, 15, e 14; 13.4, 12.8 e 11.13 e versões 11 posteriores
db.m6i.la rge	Sir	MariaDB versões 10.11, 10.6.7 e versões 10.6 posteriores, 10.5.15 e versões 10.5 posteriores, 10.4.24 e versões 10.4 posterior es	Sim	MySQL versão 8.0.28 e posteriores	Oracle Databa 19c	Todas as versões do PostgreSQL 16, 15, e 14; 13.4, 12.8 e 11.13 e versões 11 posteriores

db.m5d: classes de instância de uso geral com processadores Intel Xeon Platinum e armazenamento SSD

Classe de instância	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.m5d.24xlarge	Não	MariaDB versões 10.11, 10.6.7 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Sim	MySQL 8.0.28 e posterior	Sim	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores, 13.7 e versões 13 posteriores e 13.4
db.m5d.16xlarge	Não	MariaDB versões 10.11, 10.6.7 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Sim	MySQL 8.0.28 e posterior	Sim	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores, 13.7 e versões 13 posteriores e 13.4
db.m5d.12xlarge	Não	MariaDB versões 10.11, 10.6.7 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Sim	MySQL 8.0.28 e posterior	Sim	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores, 13.7 e versões 13 posteriores e 13.4
db.m5d.8xlarge	Não	MariaDB versões 10.11, 10.6.7 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Sim	MySQL 8.0.28 e posterior	Sim	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores, 13.7 e versões 13 posteriores e 13.4
db.m5d.4xlarge	Não	MariaDB versões 10.11, 10.6.7 e versões 10.6 posteriores, 10.5.16 e	Sim	MySQL 8.0.28 e posterior	Sim	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores, 13.7

Classe de instância	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
		versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores				e versões 13 posteriores e 13.4
db.m5d.2xlarge	Não	MariaDB versões 10.11, 10.6.7 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Sim	MySQL 8.0.28 e posterior	Sim	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores, 13.7 e versões 13 posteriores e 13.4
db.m5d.xlarge	Não	MariaDB versões 10.11, 10.6.7 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Sim	MySQL 8.0.28 e posterior	Sim	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores, 13.7 e versões 13 posteriores e 13.4
db.m5d.large	Não	MariaDB versões 10.11, 10.6.7 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Sim	MySQL 8.0.28 e posterior	Sim	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores, 13.7 e versões 13 posteriores e 13.4

db.m5: classes de instância de uso geral com processadores Intel Xeon Platinum de 2,5 GHz

Classe de instância	Db2	Maria	Microsoft SQL Server	MyS	Oracl	PostgreSQL
db.m5.24x large	Não	Sim	Sim	Sim	Sim	Todas as versões do PostgreSQL 16, 15, 14, 13, 12 e 11; 10.17 e versões 10 posteriores; 9.6.22 e versões 9 posteriores
db.m5.16x large	Não	Sim	Sim	Sim	Sim	Todas as versões do PostgreSQL 16, 15, 14, 13, 12 e 11; 10.17 e versões 10 posteriores; 9.6.22 e versões 9 posteriores
db.m5.12x large	Não	Sim	Sim	Sim	Sim	Todas as versões do PostgreSQL 16, 15, 14, 13, 12 e 11; 10.17 e versões 10 posteriores; 9.6.22 e versões 9 posteriores
db.m5.8xlarge	Não	Sim	Sim	Sim	Sim	Todas as versões do PostgreSQL 16, 15, 14, 13, 12 e 11; 10.17 e versões 10 posteriores; 9.6.22 e versões 9 posteriores
db.m5.4xlarge	Não	Sim	Sim	Sim	Sim	Todas as versões do PostgreSQL 16, 15, 14, 13, 12 e 11; 10.17 e versões 10 posteriores; 9.6.22 e versões 9 posteriores
db.m5.2xlarge	Não	Sim	Sim	Sim	Sim	Todas as versões do PostgreSQL 16, 15, 14, 13, 12 e 11; 10.17 e versões 10 posteriores; 9.6.22 e versões 9 posteriores
db.m5.xlarge	Não	Sim	Sim	Sim	Sim	Todas as versões do PostgreSQL 16, 15, 14, 13, 12 e 11; 10.17 e versões 10 posteriores; 9.6.22 e versões 9 posteriores
db.m5.large	Não	Sim	Sim	Sim	Sim	Todas as versões do PostgreSQL 16, 15, 14, 13, 12 e 11; 10.17 e versões 10 posteriores; 9.6.22 e versões 9 posteriores

db.m4: classes de instância de uso geral com processadores Intel Xeon escaláveis

Classe de instância	Db	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.m4.16 large	Nã	Preterido	Preterido	Preterido	Preter	Preterido
db.m4.10 large	Nã	Preterido	Preterido	Preterido	Preter	Preterido
db.m4.4x arge	Nã	Preterido	Preterido	Preterido	Preter	Preterido
db.m4.2x arge	Nã	Preterido	Preterido	Preterido	Preter	Preterido
db.m4.xla rge	Nã	Preterido	Preterido	Preterido	Preter	Preterido
db.m4.lar ge	Nã	Preterido	Preterido	Preterido	Preter	Preterido

db.m3: classes de instância de uso geral

Classe de instância	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.m3.2xl arge	Não	Não	Preterido	Sim	Preterido	Preterido
db.m3.xla rge	Não	Não	Preterido	Sim	Preterido	Preterido
db.m3.lar ge	Não	Não	Preterido	Sim	Preterido	Preterido

Classe de instância	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.m3.medium	Não	Não	Preterido	Sim	Preterido	Preterido

Mecanismos de banco de dados compatíveis com classes de instância otimizada para memória

As tabelas a seguir mostram os bancos de dados e as versões de banco de dados compatíveis com as classes de instância otimizada para memória.

db.z1d: classes de instâncias otimizadas para memória

Classe de instância	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.z1d.1 xlarge	Não	Não	Sim	Não	Sim	Não
db.z1d.6 large	Não	Não	Sim	Não	Sim	Não
db.z1d.3 large	Não	Não	Sim	Não	Sim	Não
db.z1d.2 large	Não	Não	Sim	Não	Sim	Não
db.z1d.xl arge	Não	Não	Sim	Não	Sim	Não
db.z1d.la rge	Não	Não	Sim	Não	Sim	Não

db.x2g: classes de instância otimizada para memória desenvolvidas por processadores Graviton2 da AWS

Classe de instância	Db	MariaDB	Micros SQL Server	MySQL	Ora	PostgreSQL
db.x2g.1 xlarge	Nã	Todas as versões do MariaDB 10.11, 10.6, 10.5 e 10.4	Não	MySQL 8.0.25 e posteriores	Nã	Todas as versões do PostgreSQL 16, 15, 14 e 13; e 12.7 e versões 12 posteriores
db.x2g.1 xlarge	Nã	Todas as versões do MariaDB 10.11, 10.6, 10.5 e 10.4	Não	MySQL 8.0.25 e posteriores	Nã	Todas as versões do PostgreSQL 16, 15, 14 e 13; e 12.7 e versões 12 posteriores
db.x2g.8 large	Nã	Todas as versões do MariaDB 10.11, 10.6, 10.5 e 10.4	Não	MySQL 8.0.25 e posteriores	Nã	Todas as versões do PostgreSQL 16, 15, 14 e 13; e 12.7 e versões 12 posteriores
db.x2g.4 large	Nã	Todas as versões do MariaDB 10.11, 10.6, 10.5 e 10.4	Não	MySQL 8.0.25 e posteriores	Nã	Todas as versões do PostgreSQL 16, 15, 14 e 13; e 12.7 e versões 12 posteriores
db.x2g.2 large	Nã	Todas as versões do MariaDB 10.11, 10.6, 10.5 e 10.4	Não	MySQL 8.0.25 e posteriores	Nã	Todas as versões do PostgreSQL 16, 15, 14 e 13; e 12.7 e versões 12 posteriores
db.x2g.x arge	Nã	Todas as versões do MariaDB 10.11, 10.6, 10.5 e 10.4	Não	MySQL 8.0.25 e posteriores	Nã	Todas as versões do PostgreSQL 16, 15, 14 e 13; e 12.7 e versões 12 posteriores

Classe de instância	Db	MariaDB	Micros SQL Serve	MySQL	Ora	PostgreSQL
db.x2g.la rge	Nã	Todas as versões do MariaDB 10.11, 10.6, 10.5 e 10.4	Não	MySQL 8.0.25 e posteriores	Não	Todas as versões do PostgreSQL 16, 15, 14 e 13; e 12.7 e versões 12 posteriores

db.x2idn: classes de instância otimizada para memória com processadores Intel Xeon escaláveis de terceira geração

Classe de instância	Db	MariaDB	Micros SQL Serve	MySQL	Oracle	PostgreSQL
db.x2idn. 32xlarge	Nã	Todas as versões do MariaDB 10.11, 10.6.7 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Não	MySQL 8.0.28 e posterio	Some Enterp e Editior	PostgreSQL versões 15, 14.6 e 13.9
db.x2idn. 24xlarge	Nã	Todas as versões do MariaDB 10.11, 10.6.7 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Não	MySQL 8.0.28 e posterio	Some Enterp e Editior	PostgreSQL versões 15, 14.6 e 13.9
db.x2idn. 16xlarge	Nã	Todas as versões do MariaDB 10.11, 10.6.7 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Não	MySQL 8.0.28 e posterio	Some Enterp e Editior	PostgreSQL versões 15, 14.6 e 13.9

db.x2iedn: classes de instância otimizada para memória com SSDs baseados em NVMe COM processadores Intel Xeon escaláveis de terceira geração

Classe de instância	DI	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.x2iedn .32xlarge	Si	Todas as versões do MariaDB 10.11, 10.6.7 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Somente edições Enterprise e Standard, SQL Server 2014 12.00 e posterior	MySQL 8.0.28 e posteric	Somer Enterp e Editior	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores, 13.7 e versões 13 posterior es e 13.4
db.x2iedn .24xlarge	Si	Todas as versões do MariaDB 10.11, 10.6.7 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Somente edições Enterprise e Standard, SQL Server 2014 12.00 e posterior	MySQL 8.0.28 e posteric	Somer Enterp e Editior	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores, 13.7 e versões 13 posterior es e 13.4
db.x2iedn .16xlarge	Si	Todas as versões do MariaDB 10.11, 10.6.7 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Somente edições Enterprise e Standard, SQL Server 2014 12.00 e posterior	MySQL 8.0.28 e posteric	Somer Enterp e Editior	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores, 13.7 e versões 13 posterior es e 13.4
db.x2iedn .8xlarge	Si	Todas as versões do MariaDB 10.11, 10.6.7 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Somente edições Enterprise e Standard, SQL Server 2014 12.00 e posterior	MySQL 8.0.28 e posteric	Somer Enterp e Editior	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores, 13.7 e versões 13 posterior es e 13.4

Classe de instância	DI	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.x2iedn .4xlarge	Si	Todas as versões do MariaDB 10.11, 10.6.7 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Somente edições Enterprise e Standard, SQL Server 2014 12.00 e posterior	MySQL 8.0.28 e posteric	Enterp e Edition e Standa Edition 2 (SE2)	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores, 13.7 e versões 13 posterior es e 13.4
db.x2iedn .2xlarge	Si	Todas as versões do MariaDB 10.11, 10.6.7 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Somente edições Enterprise e Standard, SQL Server 2014 12.00 e posterior	MySQL 8.0.28 e posteric	Enterp e Edition e Standa Edition 2 (SE2)	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores, 13.7 e versões 13 posterior es e 13.4
db.x2iedn .xlarge	Si	Todas as versões do MariaDB 10.11, 10.6.7 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Somente edições Enterprise e Standard, SQL Server 2014 12.00 e posterior	MySQL 8.0.28 e posteric	Enterp e Edition e Standa Edition 2 (SE2)	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores, 13.7 e versões 13 posterior es e 13.4

db.x2iezn: classes de instância otimizada para memória com processadores Intel Xeon escaláveis de terceira geração

Classe de instância	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.x2iezn.8xlarge	Não	Não	Não	Não	Somente Enterprise Edition	Não
db.x2iezn.6xlarge	Não	Não	Não	Não	Somente Enterprise Edition	Não
db.x2iezn.4xlarge	Não	Não	Não	Não	Enterprise Edition e Standard Edition 2 (SE2)	Não
db.x2iezn.2xlarge	Não	Não	Não	Não	Enterprise Edition e Standard Edition 2 (SE2)	Não

db.x1e – classes de instância otimizada para memória

Classe de instância	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.x1e.32xlarge	Não	Não	Sim	Não	Sim	Não
db.x1e.16xlarge	Não	Não	Sim	Não	Sim	Não
db.x1e.8xlarge	Não	Não	Sim	Não	Sim	Não
db.x1e.4xlarge	Não	Não	Sim	Não	Sim	Não
db.x1e.2xlarge	Não	Não	Sim	Não	Sim	Não

Classe de instância	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.x1e.xlarge	Não	Não	Sim	Não	Sim	Não

db.x1 – classes de instância otimizada para memória

Classe de instância	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.x1.32xlarge	Não	Não	Sim	Não	Sim	Não
db.x1.16xlarge	Não	Não	Sim	Não	Sim	Não

db.r7g: classes de instância otimizada para memória com processadores Graviton3 da AWS

Classe de instância	Db	MariaDB	Micros SQL Server	MySQL	Ora	PostgreSQL
db.r7g.1 xlarge	Na	MariaDB versões 10.11, 10.6.10 e versões 10.6 posteriores, 10.5.17 e versões 10.5 posteriores, 10.4.26 e versões 10.4 posteriores	Não	MySQL 8.0.28 e posterior	Não	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores e 13.4 e versões 13 posteriores
db.r7g.1 xlarge	Na	MariaDB versões 10.11, 10.6.10 e versões 10.6 posteriores, 10.5.17 e versões 10.5 posteriores,	Não	MySQL 8.0.28 e posterior	Não	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores e 13.4 e versões 13 posteriores

Classe de instância	DB	MariaDB	Micros SQL Server	MySQL	Ora	PostgreSQL
		10.4.26 e versões 10.4 posteriores				
db.r7g.8 large	Na	MariaDB versões 10.11, 10.6.10 e versões 10.6 posteriores, 10.5.17 e versões 10.5 posteriores, 10.4.26 e versões 10.4 posteriores	Não	MySQL 8.0.28 e posterior	Não	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores e 13.4 e versões 13 posteriores
db.r7g.4 large	Na	MariaDB versões 10.11, 10.6.10 e versões 10.6 posteriores, 10.5.17 e versões 10.5 posteriores, 10.4.26 e versões 10.4 posteriores	Não	MySQL 8.0.28 e posterior	Não	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores e 13.4 e versões 13 posteriores
db.r7g.2 large	Na	MariaDB versões 10.11, 10.6.10 e versões 10.6 posteriores, 10.5.17 e versões 10.5 posteriores, 10.4.26 e versões 10.4 posteriores	Não	MySQL 8.0.28 e posterior	Não	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores e 13.4 e versões 13 posteriores
db.r7g.xl arge	Na	MariaDB versões 10.11, 10.6.10 e versões 10.6 posteriores, 10.5.17 e versões 10.5 posteriores, 10.4.26 e versões 10.4 posteriores	Não	MySQL 8.0.28 e posterior	Não	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores e 13.4 e versões 13 posteriores

Classe de instância	Db	MariaDB	Micros SQL Server	MySQL	Ora	PostgreSQL
db.r7g.la rge	Nã	MariaDB versões 10.11, 10.6.10 e versões 10.6 posteriores, 10.5.17 e versões 10.5 posteriores, 10.4.26 e versões 10.4 posteriores	Não	MySQL 8.0.28 e posterior	Nã	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores e 13.4 e versões 13 posteriores

db.r6g: classes de instância otimizada para memória com processadores Graviton2 da AWS

Classe de instância	Db2	MariaDB	Micros SQL Server	MySQL	Ora	PostgreSQL
db.r6g.16 xlarge	Não	Todas as versões do MariaDB 10.11, 10.6, 10.5 e 10.4	Não	MySQL 8.0.23 e posteriores	Não	Todas as versões do PostgreSQL 16, 15, 14 e 13; e 12.7 e versões 12 posteriores
db.r6g.12 xlarge	Não	Todas as versões do MariaDB 10.11, 10.6, 10.5 e 10.4	Não	MySQL 8.0.23 e posteriores	Não	Todas as versões do PostgreSQL 16, 15, 14 e 13; e 12.7 e versões 12 posteriores
db.r6g.8x large	Não	Todas as versões do MariaDB 10.11, 10.6, 10.5 e 10.4	Não	MySQL 8.0.23 e posteriores	Não	Todas as versões do PostgreSQL 16, 15, 14 e 13; e 12.7 e versões 12 posteriores
db.r6g.4x large	Não	Todas as versões do MariaDB 10.11, 10.6, 10.5 e 10.4	Não	MySQL 8.0.23 e posteriores	Não	Todas as versões do PostgreSQL 16, 15, 14 e 13; e 12.7 e versões 12 posteriores

Classe de instância	Db2	MariaDB	Micros SQL Server	MySQL	Ora	PostgreSQL
db.r6g.2x large	Não	Todas as versões do MariaDB 10.11, 10.6, 10.5 e 10.4	Não	MySQL 8.0.23 e posteriores	Não	Todas as versões do PostgreSQL 16, 15, 14 e 13; e 12.7 e versões 12 posteriores
db.r6g.xl arge	Não	Todas as versões do MariaDB 10.11, 10.6, 10.5 e 10.4	Não	MySQL 8.0.23 e posteriores	Não	Todas as versões do PostgreSQL 16, 15, 14 e 13; e 12.7 e versões 12 posteriores
db.r6g.la rge	Não	Todas as versões do MariaDB 10.11, 10.6, 10.5 e 10.4	Não	MySQL 8.0.23 e posteriores	Não	Todas as versões do PostgreSQL 16, 15, 14 e 13; e 12.7 e versões 12 posteriores

db.r6gd: classes de instância otimizada para memória com processadores Graviton2 da AWS

Classe de instância	DI	MariaDB	Micros SQL Server	MySQL	Ora	PostgreSQL
db.r6gd.1 6xlarge	Na	MariaDB versões 10.11, 10.6.7 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posterior es, 10.4.25 e versões 10.4 posteriores	Não	MySQL 8.0.28 e posterior	Não	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores, 13.7 e versões 13 posteriores e 13.4
db.r6gd.1 2xlarge	Na	MariaDB versões 10.11, 10.6.7 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posterior es, 10.4.25 e versões 10.4 posteriores	Não	MySQL 8.0.28 e posterior	Não	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores, 13.7 e

Classe de instância	DI	MariaDB	Micros SQL Server	MySQL	Ora	PostgreSQL
						versões 13 posteriores e 13.4
db.r6gd.8 xlarge	Na	MariaDB versões 10.11, 10.6.7 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posterior es, 10.4.25 e versões 10.4 posteriores	Não	MySQL 8.0.28 e posterior	Não	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores, 13.7 e versões 13 posteriores e 13.4
db.r6gd.4 xlarge	Na	MariaDB versões 10.11, 10.6.7 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posterior es, 10.4.25 e versões 10.4 posteriores	Não	MySQL 8.0.28 e posterior	Não	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores, 13.7 e versões 13 posteriores e 13.4
db.r6gd.2 xlarge	Na	MariaDB versões 10.11, 10.6.7 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posterior es, 10.4.25 e versões 10.4 posteriores	Não	MySQL 8.0.28 e posterior	Não	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores, 13.7 e versões 13 posteriores e 13.4
db.r6gd.x large	Na	MariaDB versões 10.11, 10.6.7 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posterior es, 10.4.25 e versões 10.4 posteriores	Não	MySQL 8.0.28 e posterior	Não	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores, 13.7 e versões 13 posteriores e 13.4

Classe de instância	Db	MariaDB	Micros SQL Server	MySQL	Ora	PostgreSQL
db.r6gd.large	Na	MariaDB versões 10.11, 10.6.7 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Não	MySQL 8.0.28 e posterior	Não	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores, 13.7 e versões 13 posteriores e 13.4

db.r6id: classes de instância otimizada para memória com processadores Intel Xeon escaláveis de terceira geração

Classe de instância	Db	MariaDB	Micros SQL Server	MySQL	Ora	PostgreSQL
db.r6id.32xlarge	Não	MariaDB 10.6.10 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Não	MySQL versão 8.0.28 e posteriores	Não	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores e 13.7 e versões 13 posteriores
db.r6id.32xlarge	Não	MariaDB 10.6.10 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Não	MySQL versão 8.0.28 e posteriores	Não	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores e 13.7 e versões 13 posteriores
db.r6id.16xlarge	Não	MariaDB 10.6.10 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Não	MySQL versão 8.0.28 e posteriores	Não	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores e 13.7 e versões 13 posteriores

Classe de instância	Db	MariaDB	Micros SQL Server	MySQL	Ora	PostgreSQL
db.r6id.12xlarge	Nã	MariaDB 10.6.10 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Não	MySQL versão 8.0.28 e posteriores	Nãc	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores e 13.7 e versões 13 posteriores
db.r6id.8xlarge	Nã	MariaDB 10.6.10 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Não	MySQL versão 8.0.28 e posteriores	Nãc	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores e 13.7 e versões 13 posteriores
db.r6id.4xlarge	Nã	MariaDB 10.6.10 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Não	MySQL versão 8.0.28 e posteriores	Nãc	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores e 13.7 e versões 13 posteriores
db.r6id.2xlarge	Nã	MariaDB 10.6.10 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Não	MySQL versão 8.0.28 e posteriores	Nãc	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores e 13.7 e versões 13 posteriores
db.r6id.xlarge	Nã	MariaDB 10.6.10 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Não	MySQL versão 8.0.28 e posteriores	Nãc	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores e 13.7 e versões 13 posteriores

Classe de instância	Db	MariaDB	Micros SQL Server	MySQL	Ora	PostgreSQL
db.r6id.l arge	Nã	MariaDB 10.6.10 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Não	MySQL versão 8.0.28 e posteriores	Nãc	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores e 13.7 e versões 13 posteriores

db.r6idn: classes de instância otimizada para memória com processadores Intel Xeon escaláveis de terceira geração

Classe de instância	Db	MariaDB	Micros SQL Server	MySQL	Ora	PostgreSQL
db.r6idn. 32xlarge	Sim	MariaDB versão 10.6.8 e versões 10.6 posterior es, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Não	MySQL versão 8.0.28 e posterior es	Nãc	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores e 13.7 e versões 13 posteriores
db.r6idn. 24xlarge	Sim	MariaDB versão 10.6.8 e versões 10.6 posterior es, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Não	MySQL versão 8.0.28 e posterior es	Nãc	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores e 13.7 e versões 13 posteriores
db.r6idn. 16xlarge	Sim	MariaDB versão 10.6.8 e versões 10.6 posterior es, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Não	MySQL versão 8.0.28 e posterior es	Nãc	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores e 13.7 e versões 13 posteriores
db.r6idn. 12xlarge	Sim	MariaDB versão 10.6.8 e versões 10.6 posterior	Não	MySQL versão	Nãc	Todas as versões do PostgreSQL 16 e 15,

Classe de instância	Db	MariaDB	Micros SQL Server	MySQL	Ora	PostgreSQL
		es, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores		8.0.28 e posterior es		14.5 e versões 14 posteriores e 13.7 e versões 13 posteriores
db.r6idn. 12xlarge	Sim	MariaDB versão 10.6.8 e versões 10.6 posterior es, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Não	MySQL versão 8.0.28 e posterior es	Não	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores e 13.7 e versões 13 posteriores
db.r6idn. 4xlarge	Sim	MariaDB versão 10.6.8 e versões 10.6 posterior es, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Não	MySQL versão 8.0.28 e posterior es	Não	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores e 13.7 e versões 13 posteriores
db.r6i.2x large	Sim	MariaDB versão 10.6.8 e versões 10.6 posterior es, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Não	MySQL versão 8.0.28 e posterior es	Não	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores e 13.7 e versões 13 posteriores
db.r6idn. xlarge	Sim	MariaDB versão 10.6.8 e versões 10.6 posterior es, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Não	MySQL versão 8.0.28 e posterior es	Não	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores e 13.7 e versões 13 posteriores

db.r6in: classes de instância otimizada para memória com processadores Intel Xeon escaláveis de terceira geração

Classe de instância	Db2	MariaDB	Micros SQL Server	MySQL	Ora	PostgreSQL
db.r6in.32xlarge	Sim	MariaDB versão 10.6.8 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Não	MySQL versão 8.0.28 e posteriores	Não	Todas as versões 16 e 15 do PostgreSQL, 14.3 e versões 14 posteriores, 13.7 e versões 13 posteriores, 12.11 e versões 12 posteriores e 11.16 e versões 11 posteriores
db.r6in.24xlarge	Sim	MariaDB versão 10.6.8 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Não	MySQL versão 8.0.28 e posteriores	Não	Todas as versões 16 e 15 do PostgreSQL, 14.3 e versões 14 posteriores, 13.7 e versões 13 posteriores, 12.11 e versões 12 posteriores e 11.16 e versões 11 posteriores
db.r6in.24xlarge	Sim	MariaDB versão 10.6.8 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Não	MySQL versão 8.0.28 e posteriores	Não	Todas as versões 16 e 15 do PostgreSQL, 14.3 e versões 14 posteriores, 13.7 e versões 13 posteriores, 12.11 e versões 12 posteriores e 11.16 e versões 11 posteriores
db.r6in.24xlarge	Sim	MariaDB versão 10.6.8 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Não	MySQL versão 8.0.28 e posteriores	Não	Todas as versões 16 e 15 do PostgreSQL, 14.3 e versões 14 posteriores, 13.7 e versões 13 posteriores, 12.11 e versões 12 posteriores e 11.16 e versões 11 posteriores

Classe de instância	Db	MariaDB	Micros SQL Server	MySQL	Ora	PostgreSQL
db.r6in.8xlarge	Sim	MariaDB versão 10.6.8 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Não	MySQL versão 8.0.28 e posteriores	Não	Todas as versões 16 e 15 do PostgreSQL, 14.3 e versões 14 posteriores, 13.7 e versões 13 posteriores, 12.11 e versões 12 posteriores e 11.16 e versões 11 posteriores
db.r6in.4xlarge	Sim	MariaDB versão 10.6.8 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Não	MySQL versão 8.0.28 e posteriores	Não	Todas as versões 16 e 15 do PostgreSQL, 14.3 e versões 14 posteriores, 13.7 e versões 13 posteriores, 12.11 e versões 12 posteriores e 11.16 e versões 11 posteriores
db.r6in.2xlarge	Sim	MariaDB versão 10.6.8 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Não	MySQL versão 8.0.28 e posteriores	Não	Todas as versões 16 e 15 do PostgreSQL, 14.3 e versões 14 posteriores, 13.7 e versões 13 posteriores, 12.11 e versões 12 posteriores e 11.16 e versões 11 posteriores
db.r6in.xlarge	Sim	MariaDB versão 10.6.8 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Não	MySQL versão 8.0.28 e posteriores	Não	Todas as versões 16 e 15 do PostgreSQL, 14.3 e versões 14 posteriores, 13.7 e versões 13 posteriores, 12.11 e versões 12 posteriores e 11.16 e versões 11 posteriores

Classe de instância	Db2	MariaDB	Micros SQL Server	MySQL	Ora	PostgreSQL
db.r6in.large	Sim	MariaDB versão 10.6.8 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Não	MySQL versão 8.0.28 e posteriores	Não	Todas as versões 16 e 15 do PostgreSQL, 14.3 e versões 14 posteriores, 13.7 e versões 13 posteriores, 12.11 e versões 12 posteriores e 11.16 e versões 11 posteriores

db.r6i: classes de instância otimizada para memória pré-configuradas para alto nível de memória, armazenamento e E/S

Classe de instância	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r6i.8xlarge.tpc2.mem4x	Não	Não	Não	Não	Somente EE	Não
db.r6i.8xlarge.tpc2.mem3x	Não	Não	Não	Não	Somente EE	Não
db.r6i.6xlarge.tpc2.mem4x	Não	Não	Não	Não	Somente EE	Não
db.r6i.4xlarge.tpc2.mem4x	Não	Não	Não	Não	EE e SE2 BYOL	Não

Classe de instância	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r6i.4xlarge.tpc2.mem3x	Não	Não	Não	Não	EE e SE2 BYOL	Não
db.r6i.4xlarge.tpc2.mem2x	Não	Não	Não	Não	EE e SE2 BYOL	Não
db.r6i.2xlarge.tpc2.mem8x	Não	Não	Não	Não	EE e SE2 BYOL	Não
db.r6i.2xlarge.tpc2.mem4x	Não	Não	Não	Não	EE e SE2 BYOL	Não
db.r6i.2xlarge.tpc1.mem2x	Não	Não	Não	Não	EE e SE2 BYOL	Não
db.r6i.xlarge.tpc2.mem4x	Não	Não	Não	Não	EE e SE2 BYOL	Não
db.r6i.xlarge.tpc2.mem2x	Não	Não	Não	Não	EE e SE2 BYOL	Não
db.r6i.large.tpc1.mem2x	Não	Não	Não	Não	EE e SE2 BYOL	Não

db.r6i: classes de instância otimizada para memória otimizadas para memória

Classe de instânci	DI	MariaDB	Micros SQL Serve	MySQL	Ora	PostgreSQL
db.r6i.3 xlarge	Si	MariaDB versões 10.11, 10.6.7 e versões 10.6 posteriores, 10.5.15 e versões 10.5 posteriores, 10.4.24 e versões 10.4 posteriores	Sim	MySQL versão 8.0.28 e posterior es	Sim	Todas as versões do PostgreSQL 16, 15 e 14, 13.4 e versões posteriores, 12.8 e versões 12, 11.13 e versões 11 posterior es e 10.21 e versões 10 posteriores
db.r6i.2 xlarge	Si	MariaDB versões 10.11, 10.6.7 e versões 10.6 posteriores, 10.5.15 e versões 10.5 posteriores, 10.4.24 e versões 10.4 posteriores	Sim	MySQL versão 8.0.28 e posterior es	Sim	Todas as versões do PostgreSQL 16, 15 e 14, 13.4 e versões posteriores, 12.8 e versões 12, 11.13 e versões 11 posterior es e 10.21 e versões 10 posteriores
db.r6i.1 xlarge	Si	MariaDB versões 10.11, 10.6.7 e versões 10.6 posteriores, 10.5.15 e versões 10.5 posteriores, 10.4.24 e versões 10.4 posteriores	Sim	MySQL versão 8.0.28 e posterior es	Sim	Todas as versões do PostgreSQL 16, 15 e 14, 13.4 e versões posteriores, 12.8 e versões 12, 11.13 e versões 11 posterior es e 10.21 e versões 10 posteriores
db.r6i.1 xlarge	Si	MariaDB versões 10.11, 10.6.7 e versões 10.6 posteriores, 10.5.15 e versões 10.5 posteriores, 10.4.24 e versões 10.4 posteriores	Sim	MySQL versão 8.0.28 e posterior es	Sim	Todas as versões do PostgreSQL 16, 15 e 14, 13.4 e versões posteriores, 12.8 e versões 12, 11.13 e versões 11 posterior es e 10.21 e versões 10 posteriores

Classe de instânci	DI	MariaDB	Micros SQL Serve	MySQL	Ora	PostgreSQL
db.r6i.8 large	Si	MariaDB versões 10.11, 10.6.7 e versões 10.6 posteriores, 10.5.15 e versões 10.5 posteriores, 10.4.24 e versões 10.4 posteriores	Sim	MySQL versão 8.0.28 e posterior es	Sim	Todas as versões do PostgreSQL 16, 15 e 14, 13.4 e versões posteriores, 12.8 e versões 12, 11.13 e versões 11 posterior es e 10.21 e versões 10 posteriores
db.r6i.4 large	Si	MariaDB versões 10.11, 10.6.7 e versões 10.6 posteriores, 10.5.15 e versões 10.5 posteriores, 10.4.24 e versões 10.4 posteriores	Sim	MySQL versão 8.0.28 e posterior es	Sim	Todas as versões do PostgreSQL 16, 15 e 14, 13.4 e versões posteriores, 12.8 e versões 12, 11.13 e versões 11 posterior es e 10.21 e versões 10 posteriores
db.r6i.2 large	Si	MariaDB versões 10.11, 10.6.7 e versões 10.6 posteriores, 10.5.15 e versões 10.5 posteriores, 10.4.24 e versões 10.4 posteriores	Sim	MySQL versão 8.0.28 e posterior es	Sim	Todas as versões do PostgreSQL 16, 15 e 14, 13.4 e versões posteriores, 12.8 e versões 12, 11.13 e versões 11 posterior es e 10.21 e versões 10 posteriores
db.r6i.xl arge	Si	MariaDB versões 10.11, 10.6.7 e versões 10.6 posteriores, 10.5.15 e versões 10.5 posteriores, 10.4.24 e versões 10.4 posteriores	Sim	MySQL versão 8.0.28 e posterior es	Sim	Todas as versões do PostgreSQL 16, 15 e 14, 13.4 e versões posteriores, 12.8 e versões 12, 11.13 e versões 11 posterior es e 10.21 e versões 10 posteriores

Classe de instânci	DI	MariaDB	Micros SQL Serve	MySQL	Ora	PostgreSQL
db.r6i.la rge	Si	MariaDB versões 10.11, 10.6.7 e versões 10.6 posteriores, 10.5.15 e versões 10.5 posteriores, 10.4.24 e versões 10.4 posteriores	Sim	MySQL versão 8.0.28 e posterior es	Sim	Todas as versões do PostgreSQL 16, 15 e 14, 13.4 e versões posteriores, 12.8 e versões 12, 11.13 e versões 11 posterior es e 10.21 e versões 10 posteriores

db.r5d: classes de instância otimizada para memória

Classe de instânci	DI	MariaDB	Micros SQL Serve	MySQL	Ora	PostgreSQL
db.r5d.2 xlarge	Na	MariaDB versões 10.11, 10.6.7 e versões 10.6 posterior es, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Sim	MySQL 8.0.28 e posterio	Sim	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores, 13.7 e versões 13 posteriores e 13.4
db.r5d.1 xlarge	Na	MariaDB versões 10.11, 10.6.7 e versões 10.6 posterior es, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Sim	MySQL 8.0.28 e posterio	Sim	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores, 13.7 e versões 13 posteriores e 13.4
db.r5d.1 xlarge	Na	MariaDB versões 10.11, 10.6.7 e versões 10.6 posterior es, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Sim	MySQL 8.0.28 e posterio	Sim	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores, 13.7 e versões 13 posteriores e 13.4

Classe de instância	DI	MariaDB	Micros SQL Server	MySQL	Ora	PostgreSQL
db.r5d.8 large	Na	MariaDB versões 10.11, 10.6.7 e versões 10.6 posterior es, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Sim	MySQL 8.0.28 e posterio	Sim	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores, 13.7 e versões 13 posteriores e 13.4
db.r5d.4 large	Na	MariaDB versões 10.11, 10.6.7 e versões 10.6 posterior es, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Sim	MySQL 8.0.28 e posterio	Sim	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores, 13.7 e versões 13 posteriores e 13.4
db.r5d.2 large	Na	MariaDB versões 10.11, 10.6.7 e versões 10.6 posterior es, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Sim	MySQL 8.0.28 e posterio	Sim	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores, 13.7 e versões 13 posteriores e 13.4
db.r5d.x arge	Na	MariaDB versões 10.11, 10.6.7 e versões 10.6 posterior es, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Sim	MySQL 8.0.28 e posterio	Sim	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores, 13.7 e versões 13 posteriores e 13.4
db.r5d.la rge	Na	MariaDB versões 10.11, 10.6.7 e versões 10.6 posterior es, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Sim	MySQL 8.0.28 e posterio	Sim	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores, 13.7 e versões 13 posteriores e 13.4

db.r5: classes de instância otimizada para memória pré-configuradas para alta memória, armazenamento e E/S

Classe de instância	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r5b.8xlarge.tpc2.mem3x	Não	Não	Não	Não	Sim	Não
db.r5b.6xlarge.tpc2.mem4x	Não	Não	Não	Não	Sim	Não
db.r5b.4xlarge.tpc2.mem4x	Não	Não	Não	Não	Sim	Não
db.r5b.4xlarge.tpc2.mem3x	Não	Não	Não	Não	Sim	Não
db.r5b.4xlarge.tpc2.mem2x	Não	Não	Não	Não	Sim	Não
db.r5b.2xlarge.tpc2.mem8x	Não	Não	Não	Não	Sim	Não
db.r5b.2xlarge.tpc2.mem4x	Não	Não	Não	Não	Sim	Não
db.r5b.2xlarge.tpc1.mem2x	Não	Não	Não	Não	Sim	Não

Classe de instância	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r5b.xlarge.tpc2.mem4x	Não	Não	Não	Não	Sim	Não
db.r5b.xlarge.tpc2.mem2x	Não	Não	Não	Não	Sim	Não
db.r5b.large.tpc1.mem2x	Não	Não	Não	Não	Sim	Não

db.r5b – classes de instância otimizada para memória

Classe de instância	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r5b.24xlarge	Não	MariaDB versões 10.11, 10,6.5 e versões 10.6 posteriores, 10.5.12 e versões 10.5 posteriores, 10.4.24 e versões 10.4 posteriores, 10.3.34 e versões 10.3 posteriores	Sim	MySQL 8.0.25 e posteriores	Sim	Todas as versões do PostgreSQL 16, 15, 14 e 13; e 12.7 e versões 12 posteriores
db.r5b.16xlarge	Não	MariaDB versões 10.11, 10,6.5 e versões 10.6 posteriores, 10.5.12 e versões 10.5 posteriores, 10.4.24 e versões 10.4 posterior	Sim	MySQL 8.0.25 e posteriores	Sim	Todas as versões do PostgreSQL 16, 15, 14 e 13; e 12.7 e versões 12 posteriores

Classe de instância	Db2	MariaDB	Micros SQL Server	MySQL	Orac	PostgreSQL
		es, 10.3.34 e versões 10.3 posteriores				
db.r5b.12xlarge	Não	MariaDB versões 10.11, 10,6.5 e versões 10.6 posteriores, 10.5.12 e versões 10.5 posteriores, 10.4.24 e versões 10.4 posteriores, 10.3.34 e versões 10.3 posteriores	Sim	MySQL 8.0.25 e posteriores	Sim	Todas as versões do PostgreSQL 16, 15, 14 e 13; e 12.7 e versões 12 posteriores
db.r5b.8xlarge	Não	MariaDB versões 10.11, 10,6.5 e versões 10.6 posteriores, 10.5.12 e versões 10.5 posteriores, 10.4.24 e versões 10.4 posteriores, 10.3.34 e versões 10.3 posteriores	Sim	MySQL 8.0.25 e posteriores	>Sin	Todas as versões do PostgreSQL 16, 15, 14 e 13; e 12.7 e versões 12 posteriores
db.r5b.4xlarge	Não	MariaDB versões 10.11, 10,6.5 e versões 10.6 posteriores, 10.5.12 e versões 10.5 posteriores, 10.4.24 e versões 10.4 posteriores, 10.3.34 e versões 10.3 posteriores	Sim	MySQL 8.0.25 e posteriores	Sim	Todas as versões do PostgreSQL 16, 15, 14 e 13; e 12.7 e versões 12 posteriores
db.r5b.2xlarge	Não	MariaDB versões 10.11, 10,6.5 e versões 10.6 posteriores, 10.5.12 e versões 10.5 posteriores, 10.4.24 e versões 10.4 posteriores, 10.3.34 e versões 10.3 posteriores	Sim	MySQL 8.0.25 e posteriores	Sim	Todas as versões do PostgreSQL 16, 15, 14 e 13; e 12.7 e versões 12 posteriores

Classe de instância	Db2	MariaDB	Micros SQL Server	MySQL	Orac	PostgreSQL
db.r5b.xlarge	Não	MariaDB versões 10.11, 10,6.5 e versões 10.6 posteriores, 10.5.12 e versões 10.5 posteriores, 10.4.24 e versões 10.4 posteriores, 10.3.34 e versões 10.3 posteriores	Sim	MySQL 8.0.25 e posteriores	Sim	Todas as versões do PostgreSQL 16, 15, 14 e 13; e 12.7 e versões 12 posteriores
db.r5b.large	Não	MariaDB versões 10.11, 10,6.5 e versões 10.6 posteriores, 10.5.12 e versões 10.5 posteriores, 10.4.24 e versões 10.4 posteriores, 10.3.34 e versões 10.3 posteriores	Sim	MySQL 8.0.25 e posteriores	Sim	Todas as versões do PostgreSQL 16, 15, 14 e 13; e 12.7 e versões 12 posteriores

db.r5: classes de instância otimizada para memória pré-configuradas para alta memória, armazenamento e E/S

Classe de instância	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r5.12xlarge.tpc2.mem2x	Não	Não	Não	Não	Sim	Não
db.r5.8xlarge.tpc2.mem3x	Não	Não	Não	Não	Sim	Não
db.r5.6xlarge.tpc2.mem4x	Não	Não	Não	Não	Sim	Não

Classe de instância	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r5.4xlarge.tpc2.mem4x	Não	Não	Não	Não	Sim	Não
db.r5.4xlarge.tpc2.mem3x	Não	Não	Não	Não	Sim	Não
db.r5.4xlarge.tpc2.mem2x	Não	Não	Não	Não	Sim	Não
db.r5.2xlarge.tpc2.mem8x	Não	Não	Não	Não	Sim	Não
db.r5.2xlarge.tpc2.mem4x	Não	Não	Não	Não	Sim	Não
db.r5.2xlarge.tpc1.mem2x	Não	Não	Não	Não	Sim	Não
db.r5.xlarge.tpc2.mem4x	Não	Não	Não	Não	Sim	Não
db.r5.xlarge.tpc2.mem2x	Não	Não	Não	Não	Sim	Não
db.r5.large.tpc1.mem2x	Não	Não	Não	Não	Sim	Não

db.r5: classes de instância otimizada para memória

Classe de instância	Db2	MariaD	Microsoft SQL Server	MySC	Oracle	PostgreSQL
db.r5.24xlarge	Não	Sim	Sim	Sim	Sim	Todas as versões do PostgreSQL 16, 15, 14, 13, 12 e 11; 10.17 e versões 10 posteriores; 9.6.22 e versões 9 posteriores
db.r5.16xlarge	Não	Sim	Sim	Sim	Sim	Todas as versões do PostgreSQL 16, 15, 14, 13, 12 e 11; 10.17 e versões 10 posteriores; 9.6.22 e versões 9 posteriores
db.r5.12xlarge	Não	Sim	Sim	Sim	Sim	Todas as versões do PostgreSQL 16, 15, 14, 13, 12 e 11; 10.17 e versões 10 posteriores; 9.6.22 e versões 9 posteriores
db.r5.8xlarge	Não	Sim	Sim	Sim	Sim	Todas as versões do PostgreSQL 16, 15, 14, 13, 12 e 11; 10.17 e versões 10 posteriores; 9.6.22 e versões 9 posteriores
db.r5.4xlarge	Não	Sim	Sim	Sim	Sim	Todas as versões do PostgreSQL 16, 15, 14, 13, 12 e 11; 10.17 e versões 10 posteriores; 9.6.22 e versões 9 posteriores
db.r5.2xlarge	Não	Sim	Sim	Sim	Sim	Todas as versões do PostgreSQL 16, 15, 14, 13, 12 e 11; 10.17 e versões 10 posteriores; 9.6.22 e versões 9 posteriores
db.r5.xlarge	Não	Sim	Sim	Sim	Sim	Todas as versões do PostgreSQL 16, 15, 14, 13, 12 e 11; 10.17 e versões 10 posteriores; 9.6.22 e versões 9 posteriores
db.r5.large	Não	Sim	Sim	Sim	Sim	Todas as versões do PostgreSQL 16, 15, 14, 13, 12 e 11; 10.17 e versões 10 posteriores; 9.6.22 e versões 9 posteriores

db.r4 – classes de instância otimizada para memória

Classe de instância	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r4.16xlarge	Não	Preterido	Preterido	Preterido	Preteric	Preterido
db.r4.8xlarge	Não	Preterido	Preterido	Preterido	Preteric	Preterido
db.r4.4xlarge	Não	Preterido	Preterido	Preterido	Preteric	Preterido
db.r4.2xlarge	Não	Preterido	Preterido	Preterido	Preteric	Preterido
db.r4.xlarge	Não	Preterido	Preterido	Preterido	Preteric	Preterido
db.r4.large	Não	Preterido	Preterido	Preterido	Preteric	Preterido

db.r3 – classes de instância otimizada para memória

Classe de instância	Db2	MariaDB	Microsoft SQL Server	MySC	Oracle	PostgreSQL
db.r3.8xlarge**	Não	Todas as versões do MariaDB 10.6, 10.5, 10.4 e 10.3	Preterido	Sim	Preterido	Preterido
db.r3.4xlarge	Não	Todas as versões do MariaDB 10.6, 10.5, 10.4 e 10.3	Preterido	Sim	Preterido	Preterido

Classe de instância	Db2	MariaDB	Microsoft SQL Server	MySQ	Oracle	PostgreSQL
db.r3.2xlarge	Não	Todas as versões do MariaDB 10.6, 10.5, 10.4 e 10.3	Preterido	Sim	Preterido	Preterido
db.r3.xlarge	Não	Todas as versões do MariaDB 10.6, 10.5, 10.4 e 10.3	Preterido	Sim	Preterido	Preterido
db.r3.large	Não	Todas as versões do MariaDB 10.6, 10.5, 10.4 e 10.3	Preterido	Sim	Preterido	Preterido

Mecanismos de banco de dados compatíveis com classes de instância otimizada para computação

As tabelas a seguir mostram os bancos de dados e as versões de banco de dados compatíveis com as classes de instância otimizada para computação.

db.c6gd: classes de instâncias otimizadas para computação (somente para implantações de clusters de banco de dados multi-AZ)

Classe de instância	Db2	Maria	Microsoft SQL Server	MySQL	Orac	PostgreSQL
db.c6gd.16xlarge	Não	Não	Não	MySQL 8.0.28 e posterior	Não	Todas as versões 16 e 15 do PostgreSQL; 14.5 e versões 14 posteriores; 13.4 e 13.7 e versões 13 posteriores
db.c6gd.12xlarge	Não	Não	Não	MySQL 8.0.28 e posterior	Não	Todas as versões 16 e 15 do PostgreSQL; 14.5 e versões 14

Classe de instância	Db2	Maria	Microsoft SQL Server	MySQL	Orac	PostgreSQL
						posteriores; 13.4 e 13.7 e versões 13 posteriores
db.c6gd.8xlarge	Não	Não	Não	MySQL 8.0.28 e posterior	Não	Todas as versões 16 e 15 do PostgreSQL; 14.5 e versões 14 posteriores; 13.4 e 13.7 e versões 13 posteriores
db.c6gd.4xlarge	Não	Não	Não	MySQL 8.0.28 e posterior	Não	Todas as versões 16 e 15 do PostgreSQL; 14.5 e versões 14 posteriores; 13.4 e 13.7 e versões 13 posteriores
db.c6gd.2xlarge	Não	Não	Não	MySQL 8.0.28 e posterior	Não	Todas as versões 16 e 15 do PostgreSQL; 14.5 e versões 14 posteriores; 13.4 e 13.7 e versões 13 posteriores
db.c6gd.xlarge	Não	Não	Não	MySQL 8.0.28 e posterior	Não	Todas as versões 16 e 15 do PostgreSQL; 14.5 e versões 14 posteriores; 13.4 e 13.7 e versões 13 posteriores
db.c6gd.large	Não	Não	Não	MySQL 8.0.28 e posterior	Não	Todas as versões 16 e 15 do PostgreSQL; 14.5 e versões 14 posteriores; 13.4 e 13.7 e versões 13 posteriores
db.c6gd.medium	Não	Não	Não	MySQL 8.0.28 e posterior	Não	Todas as versões 16 e 15 do PostgreSQL; 14.5 e versões 14 posteriores; 13.4 e 13.7 e versões 13 posteriores

Mecanismos de banco de dados compatíveis para classes de instância de desempenho expansível

As tabelas a seguir mostram os bancos de dados e as versões de banco de dados compatíveis com as classes de instância de desempenho expansível.

db.t4g: classes de instância de desempenho expansível com processadores Graviton2 da AWS

Classe de instância	Db2	MariaDB	Microsoft SQL Server	MySQL	Orac	PostgreSQL
db.t4g: classes de instância de desempenho expansível com processadores Graviton2 da AWS						
db.t4g.2xlarge	Não	Todas as versões do MariaDB 10.11, 10.6, 10.5 e 10.4	Não	MySQL 8.0.25 e posteriores	Não	Todas as versões do PostgreSQL 16, 15, 14 e 13; e 12.7 e versões 12 posteriores
db.t4g.xlarge	Não	Todas as versões do MariaDB 10.11, 10.6, 10.5 e 10.4	Não	MySQL 8.0.25 e posteriores	Não	Todas as versões do PostgreSQL 16, 15, 14 e 13; e 12.7 e versões 12 posteriores
db.t4g.large	Não	Todas as versões do MariaDB 10.11, 10.6, 10.5 e 10.4	Não	MySQL 8.0.25 e posteriores	Não	Todas as versões do PostgreSQL 16, 15, 14 e 13; e 12.7 e versões 12 posteriores
db.t4g.medium	Não	Todas as versões do MariaDB 10.11, 10.6, 10.5 e 10.4	Não	MySQL 8.0.25 e posteriores	Não	Todas as versões do PostgreSQL 16, 15, 14 e 13; e 12.7 e versões 12 posteriores

Classe de instância	Db2	MariaDB	Microsoft SQL Server	MySQL	Orac	PostgreSQL
db.t4g.small	Não	Todas as versões do MariaDB 10.11, 10.6, 10.5 e 10.4	Não	MySQL 8.0.25 e posteriores	Não	Todas as versões do PostgreSQL 16, 15, 14 e 13; e 12.7 e versões 12 posteriores
db.t4g.micro	Não	Todas as versões do MariaDB 10.11, 10.6, 10.5 e 10.4	Não	MySQL 8.0.25 e posteriores	Não	Todas as versões do PostgreSQL 16, 15, 14 e 13; e 12.7 e versões 12 posteriores

db.t3: classes de instância expansível

Classe de instância	Db2	Maria	Microsoft SQL Server	MyS(	Oracle	PostgreSQL
db.t3.2xlarge	Sim	Sim	Sim	Sim	Sim	Todas as versões do PostgreSQL 16, 15, 14, 13, 12, 11 e 10; 9.6.22 e versões 9 posteriores
db.t3.xlarge	Sim	Sim	Sim	Sim	Sim	Todas as versões do PostgreSQL 16, 15, 14, 13, 12, 11 e 10; 9.6.22 e versões 9 posteriores
db.t3.large	Sim	Sim	Sim	Sim	Sim	Todas as versões do PostgreSQL 16, 15, 14, 13, 12, 11 e 10; 9.6.22 e versões 9 posteriores
db.t3.medium	Sim	Sim	Sim	Sim	Sim	Todas as versões do PostgreSQL 16, 15, 14, 13, 12, 11 e 10; 9.6.22 e versões 9 posteriores

Classe de instância	Db2	Maria	Microsoft SQL Server	MyS(	Oracle	PostgreSQL
db.t3.sma ll	Sim	Sim	Sim	Sim	Sim	Todas as versões do PostgreSQL 16, 15, 14, 13, 12, 11 e 10; 9.6.22 e versões 9 posteriores
db.t3.mic ro	Não	Sim	Sim	Sim	Somente no Oracle Database 12c versão 1 (12.1.0.2), que foi descontin uado	Todas as versões do PostgreSQL 16, 15, 14, 13, 12, 11 e 10; 9.6.22 e versões 9 posteriores

db.t2: classes de instância expansível

Classe de instância	Db2	MariaDB	Microsof SQL Server	MySQL	Oracle	PostgreSQL
db.t2.2xl arge	Não	Preterido	Não	Preterido	Preterid	Preterido
db.t2.xla rge	Não	Preterido	Não	Preterido	Preterid	Preterido
db.t2.lar ge	Não	Preterido	Preteridc	Preterido	Preterid	Preterido
db.t2.med ium	Não	Preterido	Preteridc	Preterido	Preterid	Preterido
db.t2.sma ll	Não	Preterido	Preteridc	Preterido	Preterid	Preterido

Classe de instância	Db2	MariaDB	Microsof SQL Server	MySQL	Oracle	PostgreSQL
db.t2.micro	Não	Preterido	Preterido	Preterido	Preterid	Preterido

Mecanismos de banco de dados compatíveis com classes de instância de leituras otimizadas

As tabelas a seguir mostram os bancos de dados e as versões de banco de dados compatíveis com as classes de instância de leituras otimizadas.

db.r6gd: classes de instância otimizada para memória que comportam leituras otimizadas e com processadores Graviton2 da AWS

Classe de instância	DI	MariaDB	Micros SQL Server	MySQL	Ora	PostgreSQL
db.r6gd.16xlarge	Na	MariaDB versões 10.11, 10.6.7 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Não	MySQL 8.0.28 e posterior	Não	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores, 13.7 e versões 13 posteriores e 13.4
db.r6gd.12xlarge	Na	MariaDB versões 10.11, 10.6.7 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Não	MySQL 8.0.28 e posterior	Não	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores, 13.7 e versões 13 posteriores e 13.4
db.r6gd.8xlarge	Na	MariaDB versões 10.11, 10.6.7 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posterior	Não	MySQL 8.0.28 e posterior	Não	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14

Classe de instância	DI	MariaDB	Micros SQL Server	MySQL	Ora	PostgreSQL
		es, 10.4.25 e versões 10.4 posteriores				posteriores, 13.7 e versões 13 posteriores e 13.4
db.r6gd.4 xlarge	Na	MariaDB versões 10.11, 10.6.7 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posterior es, 10.4.25 e versões 10.4 posteriores	Não	MySQL 8.0.28 e posterior	Não	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores, 13.7 e versões 13 posteriores e 13.4
db.r6gd.2 xlarge	Na	MariaDB versões 10.11, 10.6.7 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posterior es, 10.4.25 e versões 10.4 posteriores	Não	MySQL 8.0.28 e posterior	Não	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores, 13.7 e versões 13 posteriores e 13.4
db.r6gd.x large	Na	MariaDB versões 10.11, 10.6.7 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posterior es, 10.4.25 e versões 10.4 posteriores	Não	MySQL 8.0.28 e posterior	Não	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores, 13.7 e versões 13 posteriores e 13.4
db.r6gd.l arge	Na	MariaDB versões 10.11, 10.6.7 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posterior es, 10.4.25 e versões 10.4 posteriores	Não	MySQL 8.0.28 e posterior	Não	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores, 13.7 e versões 13 posteriores e 13.4

db.r6id: classes de instância otimizada para memória que comportam leituras otimizadas e com processadores Intel Xeon escaláveis de terceira geração

Classe de instância	Db	MariaDB	Micros SQL Server	MySQL	Ora	PostgreSQL
db.r6id.32xlarge	Nã	MariaDB 10.6.10 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Não	MySQL versão 8.0.28 e posteriores	Não	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores e 13.7 e versões 13 posteriores
db.r6id.32xlarge	Nã	MariaDB 10.6.10 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Não	MySQL versão 8.0.28 e posteriores	Não	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores e 13.7 e versões 13 posteriores
db.r6id.16xlarge	Nã	MariaDB 10.6.10 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Não	MySQL versão 8.0.28 e posteriores	Não	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores e 13.7 e versões 13 posteriores
db.r6id.12xlarge	Nã	MariaDB 10.6.10 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Não	MySQL versão 8.0.28 e posteriores	Não	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores e 13.7 e versões 13 posteriores
db.r6id.8xlarge	Nã	MariaDB 10.6.10 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Não	MySQL versão 8.0.28 e posteriores	Não	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores e 13.7 e versões 13 posteriores

Classe de instância	Db	MariaDB	Micros SQL Server	MySQL	Ora	PostgreSQL
db.r6id.4xlarge	Nã	MariaDB 10.6.10 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Não	MySQL versão 8.0.28 e posteriores	Nãc	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores e 13.7 e versões 13 posteriores
db.r6id.2xlarge	Nã	MariaDB 10.6.10 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Não	MySQL versão 8.0.28 e posteriores	Nãc	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores e 13.7 e versões 13 posteriores
db.r6id.xlarge	Nã	MariaDB 10.6.10 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Não	MySQL versão 8.0.28 e posteriores	Nãc	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores e 13.7 e versões 13 posteriores
db.r6id.large	Nã	MariaDB 10.6.10 e versões 10.6 posteriores, 10.5.16 e versões 10.5 posteriores, 10.4.25 e versões 10.4 posteriores	Não	MySQL versão 8.0.28 e posteriores	Nãc	Todas as versões do PostgreSQL 16 e 15, 14.5 e versões 14 posteriores e 13.7 e versões 13 posteriores

Determinar o suporte para classes de instância de bancos de dados nas Regiões da AWS

Para determinar as classes de instância de bancos de dados compatíveis com cada mecanismo de banco de dados em uma Região da AWSespecífica, você pode usar uma das várias abordagens. Você pode usar o AWS Management Console, a página Definição de preço do Amazon RDS ou o comando describe-orderable-db-instance-options para a AWS Command Line Interface (AWS CLI).

> **Note**
>
> Quando você executa operações com a AWS Management Console, ela mostra automaticamente as classes de instância de bancos de dados compatíveis com um mecanismo de banco de dados específico, uma versão do mecanismo de banco de dados e a Região da AWS. São exemplos de operação que você pode realizar: criação e modificação de uma instância de banco de dados.

Sumário

- Usar a página de definição de preço do Amazon RDS para determinar o suporte para classe de instância de banco de dados em Regiões da AWS
- Usar a AWS CLI para determinar o suporte para classe de instância de banco de dados em Regiões da AWS
  - Listando as classes de instância de bancos de dados compatíveis com uma versão específica do mecanismo de banco de dados em uma Região da AWS
  - Listagem das versões do mecanismo de banco de dados que oferecem suporte a uma classe de instância de bancos de dados específica em uma Região da AWS
  - Listar as Regiões da AWS que comportam um mecanismo de banco de dados e classe de instância específicos

Usar a página de definição de preço do Amazon RDS para determinar o suporte para classe de instância de banco de dados em Regiões da AWS

Você pode usar a página Definição de preço do Amazon RDS para determinar as classes de instância de bancos de dados compatíveis com cada mecanismo de bancos de dados em uma Região da AWS específica.

Para usar a página de definição de preço para determinar as classes de instância de Bancos de Dados compatíveis com cada mecanismo em uma região

1. Acesse a definição de preço do Amazon RDS.
2. Na seção Calculadora de Preços da AWS para Amazon RDS, escolha Crie sua estimativa personalizada agora.
3. Em Escolher uma região, escolha uma Região da AWS.

4. Em Localizar um serviço, insira **`Amazon RDS`**.
5. Escolha Configurar para sua opção de configuração e mecanismo de banco de dados.
6. Use a seção de instâncias compatíveis para ver as classes de instância de banco de dados aceitas.
7. (Opcional) Escolha outras opções na calculadora e selecione Salvar e visualizar resumo ou Salvar e adicionar serviço.

Usar a AWS CLI para determinar o suporte para classe de instância de banco de dados em Regiões da AWS

Você pode usar a AWS CLI para determinar quais classes de instância de bancos de dados são compatíveis com versões específicas de mecanismos de banco de dados em uma Região da AWS. A tabela a seguir mostra os valores válidos do mecanismo de banco de dados.

Nomes de mecanismos	Valores do mecanismo nos comandos CLI	Mais informações sobre as versões
Db2	`db2-ae` `db2-se`	Db2 em versões do Amazon RDS
MariaDB	`mariadb`	MariaDB em versões do Amazon RDS

Microsoft SQL Server	`sqlserver-ee`
	`sqlserver-se`
	`sqlserver-ex`
	`sqlserver-web`
		Versões do Microsoft SQL Server no Amazon RDS

MySQL	`mysql`	MySQL em versões do Amazon RDS
Oracle	`oracle-ee` `oracle-se2`	Notas de lançamento do Amazon RDS para Oracle

Nomes de mecanismos	Valores do mecanismo nos comandos CLI	Mais informações sobre as versões
PostgreSQL	`postgres`	Versões disponíveis do banco de dados PostgreSQL

Para obter informações sobre nomes de Região da AWS, consulte AWSRegiões de .

Os exemplos a seguir demonstram como determinar o suporte de classe de instância de bancos de dados em uma Região da AWS usando o comando da AWS CLI describe-orderable-db-instance-options.

> **Note**
>
> Para limitar a saída, os exemplos a seguir mostram resultados apenas para o tipo de armazenamento SSD (`gp2`) de uso geral. Se necessário, você pode alterar o tipo de armazenamento para SSD de uso geral (`gp3`), IOPS provisionadas (`io1`), Block Express com IOPS provisionadas (`io2`) ou magnético (`standard`) nos comandos.

Tópicos

- Listando as classes de instância de bancos de dados compatíveis com uma versão específica do mecanismo de banco de dados em uma Região da AWS
- Listagem das versões do mecanismo de banco de dados que oferecem suporte a uma classe de instância de bancos de dados específica em uma Região da AWS
- Listar as Regiões da AWS que comportam um mecanismo de banco de dados e classe de instância específicos

Listando as classes de instância de bancos de dados compatíveis com uma versão específica do mecanismo de banco de dados em uma Região da AWS

Para listar as classes de instância de bancos de dados compatíveis com uma versão específica do mecanismo de banco de dados em uma Região da AWS, execute o comando a seguir.

Para Linux, macOS ou Unix:

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

Para Windows:

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

Por exemplo, o comando a seguir lista as classes de instância de banco de dados compatíveis com a versão 13.6 do mecanismo de banco de dados do RDS para PostgreSQL no Leste dos EUA (N. da Virgínia).

Para Linux, macOS ou Unix:

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

Para Windows:

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

Listagem das versões do mecanismo de banco de dados que oferecem suporte a uma classe de instância de bancos de dados específica em uma Região da AWS

Para listar as versões do mecanismo de banco de dados compatíveis com uma classe de instância de bancos de dados específica em uma Região da AWS, execute o comando a seguir.

Para Linux, macOS ou Unix:

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

Para Windows:

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

Por exemplo, o comando a seguir lista as versões do mecanismo de banco de dados do RDS para o mecanismo de banco de dados PostgreSQL que oferecem suporte para a classe de instância de bancos de dados db.r5.large no Leste dos EUA (N. da Virgínia).

Para Linux, macOS ou Unix:

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

Para Windows:

```
aws rds describe-orderable-db-instance-options --engine postgres --db-instance-class
 db.m7g.large ^
```

```
    --query "*[].{EngineVersion:EngineVersion,StorageType:StorageType}|[?
StorageType=='gp2']|[].{EngineVersion:EngineVersion}" ^
    --output text ^
    --region us-east-1
```

Listar as Regiões da AWS que comportam um mecanismo de banco de dados e classe de instância específicos

O script do Bash a seguir lista todas as Regiões da AWS que comportam a combinação específica de mecanismo de banco de dados e classe de instância.

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

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

O exemplo de saída a seguir verifica o suporte regional para o RDS para MySQL usando a classe de instância db.r8g.large.

```
./check_region_support.sh mysql db.r8g.large
ap-south-1 doesn't support db.r8g.large for mysql.
eu-north-1 doesn't support db.r8g.large for mysql.
eu-west-3 doesn't support db.r8g.large for mysql.
```

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

Alterar a classe de instância de banco de dados

Você pode alterar a CPU e a memória disponíveis para uma instância de banco de dados alterando a classe de instância de banco de dados. Para alterar a classe da instância de banco de dados, modifique a instância de banco de dados seguindo as instruções em Modificar uma instância de banco de dados do Amazon RDS.

Configurar o processador para uma classe de instância de banco de dados no RDS para Oracle

As classes de instância de banco de dados do Amazon RDS oferecem suporte à Tecnologia Hyper-Threading Intel, que permite a execução de vários threads simultaneamente em um único núcleo de CPU Intel Xeon. Cada thread é representado como uma CPU virtual (vCPU) na instância de banco de dados. Uma instância de banco de dados tem um número padrão de núcleos de CPU, que varia de acordo com a classe de instância de banco de dados. Por exemplo, uma classe de instância de banco de dados db.m4.xlarge tem dois núcleos de CPU e dois threads por núcleo por padrão, totalizando quatro vCPUs.

> **Note**
>
> Cada vCPU é um hyperthread de um núcleo de CPU Intel Xeon.

Tópicos

- Visão geral da configuração do processador para RDS para Oracle
- Classes de instância de banco de dados compatíveis com a configuração do processador
- Configuração dos núcleos e threads de CPU por núcleo de CPU para uma classe de instância de banco de dados

Visão geral da configuração do processador para RDS para Oracle

Ao usar o RDS para Oracle, geralmente você pode encontrar uma classe de instância de banco de dados que tem uma combinação de memória e um número de vCPUs para atender às suas workloads. No entanto, você também pode especificar os seguintes recursos de processador a fim de otimizar a instância de banco de dados do RDS para Oracle para workloads ou necessidades empresariais específicas:

- Número de núcleos de CPU – você pode personalizar o número de núcleos de CPU para a instância. Você pode fazer isso para otimizar potencialmente os custos de licenciamento do software com uma instância de banco de dados que tem quantidades de RAM suficientes para workloads com uso intensivo de memória, mas menos núcleos de CPU.
- Threads por núcleo – você pode desativar a Tecnologia Hyper-Threading Intel especificando um único thread por núcleo de CPU. Você pode fazer isso para determinadas workloads, como workloads de high-performance computing (HPC - computação de alta performance).

Você pode controlar o número de núcleos e de threads de CPU para cada núcleo separadamente. Você pode definir um ou ambos em uma solicitação. Após uma configuração ser associada a uma instância de banco de dados, a configuração persiste até que você a altere.

As configurações de processador para uma instância de banco de dados são associadas a snapshots de instância de banco de dados. Quando um snapshot é restaurado, sua instância de banco de dados restaurada usa as configurações de recursos do processador usadas quando o snapshot foi obtido.

Se você alterar a classe da instância de banco de dados para uma instância de banco de dados com configurações de processador não padrão, especifique as configurações de processador padrão ou especifique explicitamente as configurações de processador no momento de modificar. Esse requisito garante que você esteja ciente dos custos de licenciamento de terceiros que possam ser incorridos ao modificar a instância de banco de dados.

Não há uma cobrança adicional ou reduzida para especificar atributos de processador em uma instância de banco de dados do RDS para Oracle. Você será cobrado da mesma forma que para as instâncias de banco de dados executadas com configurações de CPU padrão.

Classes de instância de banco de dados compatíveis com a configuração do processador

Só será possível configurar o número de núcleos e de threads de CPU por núcleo quando as seguintes condições forem atendidas:

- Você está configurando uma instância de banco de dados do RDS para Oracle. Para obter informações sobre as classes de instância de banco de dados compatíveis com diferentes edições do Oracle Database, consulte Classes de instância de banco de dados do RDS para Oracle.
- A instância de banco de dados está usando a opção de licenciamento traga a sua própria licença (BYOL) do RDS para Oracle. Para ter mais informações sobre as opções de licenciamento da Oracle, consulte Opções de licenciamento do RDS para Oracle.
- Sua instância de banco de dados não pertence a uma das classes de instância db.r5 ou db.r5b que têm configurações de processador predefinidas. Essas classes de instância têm nomes no formato db.r5.*instance_size*.tpc*threads_per_core*.mem*ratio* ou db.r5b.*instance_size*.tpc*threads_per_core*.mem*ratio*. Por exemplo, db.r5.xlarge.tpc2.mem4x é pré-configurada com dois threads por núcleo (tpc2) e 4x a quantidade de memória da classe de instância db.r5.xlarge padrão. Não é possível configurar os recursos do processador dessas classes de instância otimizada. Para ter mais informações, consulte Classes de instância de banco de dados compatíveis do RDS para Oracle.

É possível usar o comando da AWS CLI a seguir para mostrar as vCPUs padrão, núcleos, threads por núcleo e números válidos de núcleos para uma classe de instância. Substitua *r7i.48xlarge* no comando de exemplo pelo nome da sua classe de instância.

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

Na tabela a seguir, você pode encontrar as classes de instância de banco de dados que oferecem suporte à configuração de um número de núcleos e threads de CPU por núcleo. Você também pode encontrar o valor padrão e os valores válidos para o número de núcleos de CPU e de threads de CPU por núcleo para cada classe de instância de banco de dados.

DB instance class	vCPUs padrão	Núcleos de CPU padrão	Threads padrão por núcleo	Número válido de núcleos de CPU	Número válido de threads por núcleo
db.m6i: classes de instância otimizada para memória					
db.m6i.large	2	1	2	1	1, 2
db.m6i.xlarge	4	2	2	2	1, 2
db.m6i.2xlarge	8	4	2	2, 4	1, 2
db.m6i.4xlarge	16	8	2	2, 4, 6, 8	1, 2
db.m6i.4xlarge	16	8	2	2, 4, 6, 8	1, 2
db.m6i.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
db.m6i.12 xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
db.m6i.16 xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
db.m6i.24 xlarge	96	48	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2
db.m6i.32 xlarge	128	64	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40,	1, 2

DB instance class	vCPUs padrão	Núcleos de CPU padrão	Threads padrão por núcleo	Número válido de núcleos de CPU	Número válido de threads por núcleo
				42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64	
db.m5: classes de instância de uso geral					
db.m5.large	2	1	2	1	1, 2
db.m5.xlarge	4	2	2	2	1, 2
db.m5.2xlarge	8	4	2	2, 4	1, 2
db.m5.4xlarge	16	8	2	2, 4, 6, 8	1, 2
db.m5.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
db.m5.12x large	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
db.m5.16x large	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
db.m5.24x large	96	48	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2
db.m5d: classes de instância de uso geral					
db.m5d.large	2	1	2	1	1, 2
db.m5d.xlarge	4	2	2	2	1, 2
db.m5d.2x large	8	4	2	2, 4	1, 2

DB instance class	vCPUs padrão	Núcleos de CPU padrão	Threads padrão por núcleo	Número válido de núcleos de CPU	Número válido de threads por núcleo
db.m5d.4x large	16	8	2	2, 4, 6, 8	1, 2
db.m5d.8x large	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
db.m5d.12 xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
db.m5d.16 xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
db.m5d.24 xlarge	96	48	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2
db.m4: classes de instância de uso geral					
db.m4.10x large	40	20	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20	1, 2
db.m4.16x large	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
db.r7i: classes de instância otimizada para memória					
db.r7i.large	2	1	2	1	1, 2
db.r7i.xlarge	4	2	2	1, 2	1, 2
db.r7i.2xlarge	8	4	2	1, 2, 3, 4	1, 2

DB instance class	vCPUs padrão	Núcleos de CPU padrão	Threads padrão por núcleo	Número válido de núcleos de CPU	Número válido de threads por núcleo
db.r7i.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
db.r7i.8xlarge	32	16	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1, 2
db.r7i.12xlarge	48	24	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24	1, 2
db.r7i.16xlarge	64	32	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1, 2
db.r7i.24xlarge	96	48	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48	1, 2
db.r7i.48xlarge	192	96	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64, 66, 68, 70, 72, 74, 76, 78, 80, 82, 84, 86, 88, 90, 92, 94, 96	1, 2

DB instance class	vCPUs padrão	Núcleos de CPU padrão	Threads padrão por núcleo	Número válido de núcleos de CPU	Número válido de threads por núcleo
db.r6i: classes de instância otimizada para memória					
db.r6i.large	2	1	2	1	1, 2
db.r6i.xlarge	4	2	2	1, 2	1, 2
db.r6i.2xlarge	8	4	2	2, 4	1, 2
db.r6i.4xlarge	16	8	2	2, 4, 6, 8	1, 2
db.r6i.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
db.r6i.12xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
db.r6i.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
db.r6i.24xlarge	96	48	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2
db.r6i.32xlarge	128	64	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64	1, 2
db.r5: classes de instância otimizada para memória					
db.r5.large	2	1	2	1	1, 2
db.r5.xlarge	4	2	2	2	1, 2

DB instance class	vCPUs padrão	Núcleos de CPU padrão	Threads padrão por núcleo	Número válido de núcleos de CPU	Número válido de threads por núcleo
db.r5.2xlarge	8	4	2	2, 4	1, 2
db.r5.4xlarge	16	8	2	2, 4, 6, 8	1, 2
db.r5.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
db.r5.12xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
db.r5.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
db.r5.24xlarge	96	48	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2
db.r5: classes de instância otimizada para memória					
db.r5b.large	2	1	2	1	1, 2
db.r5b.xlarge	4	2	2	2	1, 2
db.r5b.2xlarge	8	4	2	2, 4	1, 2
db.r5b.4xlarge	16	8	2	2, 4, 6, 8	1, 2
db.r5b.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
db.r5b.12 xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2

DB instance class	vCPUs padrão	Núcleos de CPU padrão	Threads padrão por núcleo	Número válido de núcleos de CPU	Número válido de threads por núcleo
db.r5b.16 xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
db.r5b.24 xlarge	96	48	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2
db.r5d: classes de instância otimizada para memória					
db.r5d.large	2	1	2	1	1, 2
db.r5d.xlarge	4	2	2	2	1, 2
db.r5d.2xlarge	8	4	2	2, 4	1, 2
db.r5d.4xlarge	16	8	2	2, 4, 6, 8	1, 2
db.r5d.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
db.r5d.12 xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
db.r5d.16 xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
db.r5d.24 xlarge	96	48	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2
db.r4 – classes de instância otimizada para memória					

DB instance class	vCPUs padrão	Núcleos de CPU padrão	Threads padrão por núcleo	Número válido de núcleos de CPU	Número válido de threads por núcleo
db.r4.large	2	1	2	1	1, 2
db.r4.xlarge	4	2	2	1, 2	1, 2
db.r4.2xlarge	8	4	2	1, 2, 3, 4	1, 2
db.r4.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
db.r4.8xlarge	32	16	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1, 2
db.r4.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
db.r3 – classes de instância otimizada para memória					
db.r3.large	2	1	2	1	1, 2
db.r3.xlarge	4	2	2	1, 2	1, 2
db.r3.2xlarge	8	4	2	1, 2, 3, 4	1, 2
db.r3.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
db.r3.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
db.x2idn: classes de instância otimizada para memória					
db.x2idn. 16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2

DB instance class	vCPUs padrão	Núcleos de CPU padrão	Threads padrão por núcleo	Número válido de núcleos de CPU	Número válido de threads por núcleo
db.x2idn.24xlarge	96	48	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2
db.x2idn.32xlarge	128	64	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64	1, 2
db.x2iedn: classes de instância otimizada para memória					
db.x2iedn.xlarge	4	2	2	1, 2	1, 2
db.x2iedn.2xlarge	8	4	2	2, 4	1, 2
db.x2iedn.4xlarge	16	8	2	2, 4, 6, 8	1, 2
db.x2iedn.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
db.x2iedn.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
db.x2iedn.24xlarge	96	48	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2

DB instance class	vCPUs padrão	Núcleos de CPU padrão	Threads padrão por núcleo	Número válido de núcleos de CPU	Número válido de threads por núcleo
db.x2iedn .32xlarge	128	64	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64	1, 2
db.x2iezn: classes de instância otimizada para memória					
db.x2iezn .2xlarge	8	4	2	2, 4	1, 2
db.x2iezn .4xlarge	16	8	2	2, 4, 6, 8	1, 2
db.x2iezn .6xlarge	24	12	2	2, 4, 6, 8, 10, 12	1, 2
db.x2iezn .8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
db.x2iezn .12xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
db.x1 – classes de instância otimizada para memória					
db.x1.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
db.x1.32xlarge	128	64	2	4, 8, 12, 16, 20, 24, 28, 32, 36, 40, 44, 48, 52, 56, 60, 64	1, 2
db.x1e – classes de instância otimizada para memória					

DB instance class	vCPUs padrão	Núcleos de CPU padrão	Threads padrão por núcleo	Número válido de núcleos de CPU	Número válido de threads por núcleo
db.x1e.xlarge	4	2	2	1, 2	1, 2
db.x1e.2xlarge	8	4	2	1, 2, 3, 4	1, 2
db.x1e.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
db.x1e.8xlarge	32	16	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1, 2
db.x1e.16 xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
db.x1e.32 xlarge	128	64	2	4, 8, 12, 16, 20, 24, 28, 32, 36, 40, 44, 48, 52, 56, 60, 64	1, 2
db.z1d: classes de instâncias otimizadas para memória					
db.z1d.large	2	1	2	1	1, 2
db.z1d.xlarge	4	2	2	2	1, 2
db.z1d.2xlarge	8	4	2	2, 4	1, 2
db.z1d.3xlarge	12	6	2	2, 4, 6	1, 2
db.z1d.6xlarge	24	12	2	2, 4, 6, 8, 10, 12	1, 2
db.z1d.12 xlarge	48	24	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2

> **Note**
>
> Você pode usar o AWS CloudTrail para monitorar e auditar alterações na configuração do processo de instâncias de banco de dados do Amazon RDS para Oracle. Para ter mais informações sobre o uso de CloudTrail, consulte Monitorar chamadas de API do Amazon RDSno AWS CloudTrail.

Configuração dos núcleos e threads de CPU por núcleo de CPU para uma classe de instância de banco de dados

Você pode configurar o número de núcleos e threads de CPU por núcleo para a classe de instância de banco de dados ao executar as seguintes operações:

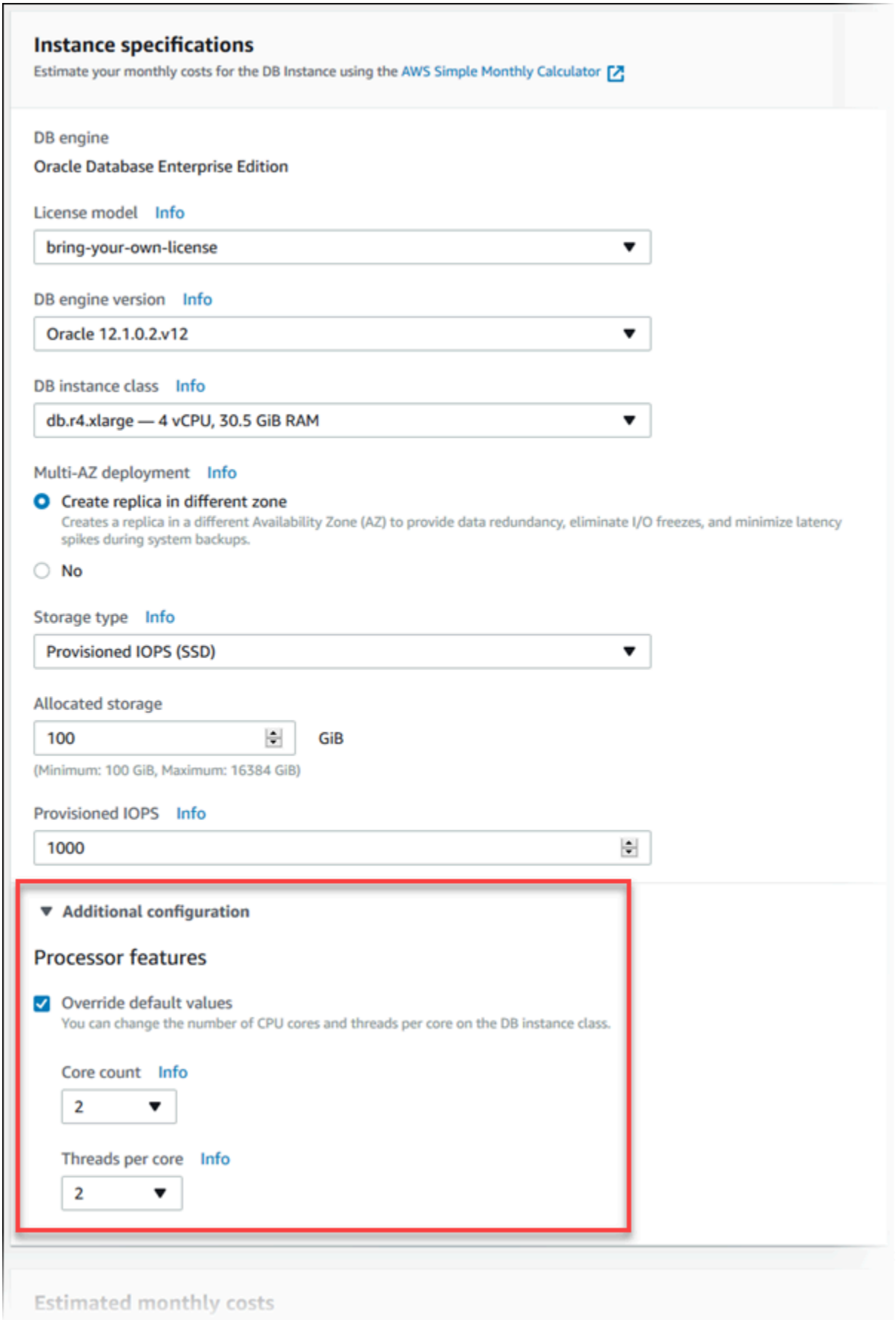
- Criar uma instância de banco de dados do Amazon RDS
- Modificar uma instância de banco de dados do Amazon RDS
- Restaurar uma instância de banco de dados
- Restaurar uma instância de banco de dados para um momento especificado no Amazon RDS

> **Note**
>
> Ao modificar uma instância de banco de dados para configurar o número de núcleos ou de threads de CPU por núcleo, há uma breve interrupção da instância de banco de dados.

Você pode definir os núcleos e os threads de CPU por núcleo de CPU para uma classe de instância de banco de dados usando o AWS Management Console, a AWS CLI ou a API do RDS.

Console

Ao criar, modificar ou restaurar uma instância de banco de dados, você define a classe de banco de dados no AWS Management Console. A seção Instance specifications (Especificações da instância) mostra opções para o processador. A imagem a seguir mostra as opções de recursos de processador.

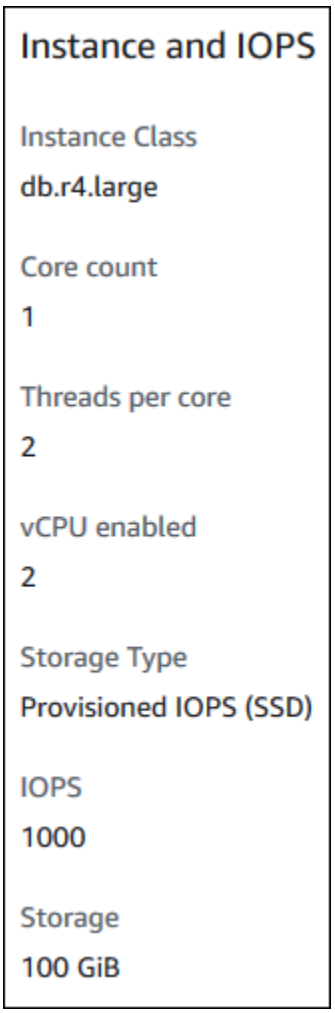

Instance specifications
Estimate your monthly costs for the DB Instance using the AWS Simple Monthly Calculator
DB engine
Oracle Database Enterprise Edition
License model Info
bring-your-own-license
DB engine version Info
Oracle 12.1.0.2.v12
DB instance class Info
db.r4.xlarge — 4 vCPU, 30.5 GiB RAM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

Defina as opções a seguir para os valores apropriados de sua classe de instância de banco de dados em Processor features (Recursos do processador):

- Core count (Contagem de núcleos) – defina o número de núcleos de CPU usando esta opção. O valor deve ser igual ou menor que o número máximo de núcleos de CPU para a classe de instância de banco de dados.
- Threads per core (Threads por núcleo) – especifique 2 para habilitar vários threads por núcleo ou especifique 1 para desabilitar vários threads por núcleo.

Ao modificar ou restaurar uma instância de banco de dados, você também pode definir núcleos de CPU e threads por núcleo de CPU para as configurações padrão da classe da instância.

Ao visualizar os detalhes de uma instância de banco de dados no console, veja as informações de processador da classe de instância de banco de dados na guia Configuration (Configuração). A imagem a seguir mostra uma classe de instância de banco de dados com um núcleo de CPU e vários threads por núcleo habilitados.



Para instâncias de banco de dados Oracle, as informações de processador aparecem apenas para instâncias de banco de dados Traga sua própria licença (BYOL).

AWS CLI

Você pode definir os recursos de processador de uma instância de banco de dados ao executar um dos seguintes comandos da AWS CLI:

- create-db-instance
- modify-db-instance
- restore-db-instance-from-db-snapshot
- restore-db-instance-from-s3
- restore-db-instance-to-point-in-time

Para configurar o processador de uma classe de instância de banco de dados para uma instância de banco de dados usando a AWS CLI, inclua a opção `--processor-features` no comando. Especifique o número de núcleos de CPU com o nome do recurso `coreCount` e especifique se os vários threads por núcleo estão habilitados com o nome do recurso `threadsPerCore`.

A opção tem a seguinte sintaxe.

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

Veja a seguir exemplos que configuram o processador:

Exemplos

- Configuração do número de núcleos de CPU para uma instância de banco de dados
- Configuração do número de núcleos de CPU e desabilitação de vários threads para uma instância de banco de dados
- Visualização dos valores válidos do processador para uma classe de instância de banco de dados
- Retorno às configurações de processador padrão para uma instância de banco de dados
- Retorno para o número padrão de núcleos de CPU para uma instância de banco de dados
- Retorno para o número padrão de threads por núcleo para uma instância de banco de dados

Configuração do número de núcleos de CPU para uma instância de banco de dados

Example

O exemplo a seguir modifica `mydbinstance` configurando o número de núcleos de CPU para 4. As alterações são aplicadas imediatamente usando `--apply-immediately`. Caso queira aplicar as alterações durante a janela de manutenção programada, omita a opção `--apply-immediately`.

Para Linux, macOS ou Unix:

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

Para Windows:

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

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

Configuração do número de núcleos de CPU e desabilitação de vários threads para uma instância de banco de dados

Example

O exemplo a seguir modifica `mydbinstance` definindo o número de núcleos de CPU para `4` e desabilitando vários threads por núcleo. As alterações são aplicadas imediatamente usando `--apply-immediately`. Caso queira aplicar as alterações durante a janela de manutenção programada, omita a opção `--apply-immediately`.

Para Linux, macOS ou Unix:

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

Para Windows:

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

Visualização dos valores válidos do processador para uma classe de instância de banco de dados

Example

Você pode visualizar os valores válidos de processador para uma classe de instância de banco de dados específica executando o comando describe-orderable-db-instance-options e especificando a classe da instância para a opção `--db-instance-class`. Por exemplo, o resultado do comando a seguir mostra as opções de processador para a classe de instância db.r3.large.

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

O seguinte é o resultado do exemplo do comando em formato JSON.


```
    {
                "SupportsIops": true,
                "MaxIopsPerGib": 50.0,
                "LicenseModel": "bring-your-own-license",
                "DBInstanceClass": "db.r3.large",
                "SupportsIAMDatabaseAuthentication": false,
                "MinStorageSize": 100,
                "AvailabilityZones": [
                    {
                        "Name": "us-west-2a"
                    },
                    {
                        "Name": "us-west-2b"
                    },
                    {
                        "Name": "us-west-2c"
                    }
                ],
                "EngineVersion": "12.1.0.2.v2",
                "MaxStorageSize": 32768,
                "MinIopsPerGib": 1.0,
                "MaxIopsPerDbInstance": 40000,
                "ReadReplicaCapable": false,
                "AvailableProcessorFeatures": [
                    {
                        "Name": "coreCount",
                        "DefaultValue": "1",
                        "AllowedValues": "1"
                    },
                    {
                        "Name": "threadsPerCore",
                        "DefaultValue": "2",
                        "AllowedValues": "1,2"
                    }
                ],
                "SupportsEnhancedMonitoring": true,
                "SupportsPerformanceInsights": false,
                "MinIopsPerDbInstance": 1000,
                "StorageType": "io1",
                "Vpc": false,
                "SupportsStorageEncryption": true,
                "Engine": "oracle-ee",
```

```
                "MultiAZCapable": true
    }
```

Além disso, você pode executar os seguintes comandos para informações de processador de classe de instância de banco de dados:

- describe-db-instances – mostra as informações do processador da instância de banco de dados especificada.
- describe-db-snapshots – mostra as informações do processador do snapshot de banco de dados especificado.
- describe-valid-db-instance-modifications – mostra as modificações válidas no processador da instância de banco de dados especificada.

Na saída dos comandos anteriores, os valores dos recursos do processador não serão nulos somente se as seguintes condições forem atendidas:

- Você está usando uma instância de banco de dados do RDS para Oracle.
- A instância de banco de dados do RDS para Oracle é compatível com a alteração de valores do processador.
- As configurações atuais de núcleo e de thread de CPU são definidas como valores não padrão.

Se as condições anteriores não forem atendidas, você poderá obter o tipo de instância usando describe-db-instances. É possível obter as informações do processador para esse tipo de instância executando a operação describe-instance-types do EC2.

Retorno às configurações de processador padrão para uma instância de banco de dados

Example

O exemplo a seguir modifica a `mydbinstance` retornando sua classe de instância de banco de dados para os valores de processador padrão dela. As alterações são aplicadas imediatamente usando `--apply-immediately`. Caso queira aplicar as alterações durante a janela de manutenção programada, omita a opção `--apply-immediately`.

Para Linux, macOS ou Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
```

```
    --use-default-processor-features \
    --apply-immediately
```

Para Windows:

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

Retorno para o número padrão de núcleos de CPU para uma instância de banco de dados

Example

O exemplo a seguir modifica `mydbinstance` retornando sua classe de instância de banco de dados para o número padrão de núcleos de CPU dela. A configuração de threads por núcleo não foi alterada. As alterações são aplicadas imediatamente usando `--apply-immediately`. Caso queira aplicar as alterações durante a janela de manutenção programada, omita a opção `--apply-immediately`.

Para Linux, macOS ou Unix:

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

Para Windows:

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

Retorno para o número padrão de threads por núcleo para uma instância de banco de dados

Example

O exemplo a seguir modifica `mydbinstance` retornando sua classe de instância de banco de dados para o número padrão de threads por núcleo dela. O número da configuração de núcleos de CPU não é alterado. As alterações são aplicadas imediatamente usando `--apply-immediately`. Caso

queira aplicar as alterações durante a janela de manutenção programada, omita a opção `--apply-immediately`.

Para Linux, macOS ou Unix:

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

Para Windows:

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

API do RDS

É possível definir os recursos do processador de uma instância de banco de dados ao chamar uma das seguintes operações da API do Amazon RDS:

- CreateDBInstance
- ModifyDBInstance
- RestoreDBInstanceFromDBSnapshot
- RestoreDBInstanceFromS3
- RestoreDBInstanceToPointInTime

Para configurar os recursos do processador de uma classe de instância de banco de dados para uma instância de banco de dados usando a API do Amazon RDS, inclua o parâmetro `ProcessFeatures` na chamada.

O parâmetro tem a seguinte sintaxe.

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

Especifique o número de núcleos de CPU com o nome do atributo `coreCount` e especifique se os vários threads por núcleo estão habilitados com o nome do atributo `threadsPerCore`.

É possível visualizar os valores válidos do processador de uma classe de instância de banco de dados específica executando a operação DescribeOrderableDBInstanceOptions e especificando a classe da instância para o parâmetro `DBInstanceClass`. Você também pode usar as seguintes operações:

- DescribeDBInstances – mostra as informações do processador da instância de banco de dados especificada.
- DescribeDBSnapshots – mostra as informações do processador do snapshot de banco de dados especificado.
- DescribeValidDBInstanceModifications – mostra as modificações válidas no processador da instância de banco de dados especificada.

Na saída das operações anteriores, os valores dos recursos do processador não serão nulos somente se as seguintes condições forem atendidas:

- Você está usando uma instância de banco de dados do RDS para Oracle.
- A instância de banco de dados do RDS para Oracle é compatível com a alteração de valores do processador.
- As configurações atuais de núcleo e de thread de CPU são definidas como valores não padrão.

Se as condições anteriores não forem atendidas, você poderá obter o tipo de instância usando DescribeDBInstances. É possível obter as informações do processador para esse tipo de instância executando a operação DescribeInstanceTypes do EC2.

Especificações de hardware para classes de instância de banco de dados

Nas tabelas desta seção, é possível encontrar detalhes de hardware sobre as classes de instância de banco de dados do Amazon RDS.

Para obter informações sobre o suporte a mecanismos de banco de dados do Amazon RDS para cada classe de instância de banco de dados, consulte Mecanismos de banco de dados compatíveis para classes de instância de banco de dados.

Tópicos

- Especificações de hardware para classes de instância de banco de dados
- Especificações de hardware para as classes de instância de uso geral

- Especificações de hardware para as classes de instância otimizada para memória
- Especificações de hardware para as classes de instância otimizada para computação
- Especificações de hardware para as classes de instância de desempenho expansível

Especificações de hardware para classes de instância de banco de dados

A terminologia a seguir é usada para descrever especificações de hardware para classes de instâncias de banco de dados:

vCPU

O número de unidades de processamento central (CPUs) virtuais. Uma CPU virtual é uma unidade de capacidade que pode ser usada para comparar classes de instância de banco de dados. Em vez de comprar ou alugar um determinado processador para usar durante vários meses ou anos, você está alugando a capacidade de acordo com a hora. Nosso objetivo é tornar uma quantidade consistente e específica da capacidade da CPU disponível, dentro dos limites de hardware subjacente real.

ECU

A medida relativa da potência de todo o processamento de uma instância do Amazon EC2. Para que os desenvolvedores tenham facilidade ao comparar a capacidade da CPU entre diferentes classes de instância, definimos uma unidade computacional do Amazon EC2. A quantidade de CPU que é alocada para determinada instância é expressa em termos dessas Unidades computacionais do EC2. No momento, uma ECU fornece a capacidade de CPU equivalente a de um processador 2007 Xeon ou 2007 Opteron de 1,0–1,2 GHz.

Memória (GiB)

A memória RAM, em gibibytes, alocada à instância de banco de dados. Geralmente, há uma proporção consistente entre a memória e a vCPU. Por exemplo, considere a classe de instância db.r4, que possui uma proporção entre memória e vCPU semelhante à da classe de instância db.r5. No entanto, para a maioria dos casos de uso, a classe de instância db.r5 fornece uma performance melhor e mais consistente do que a classe de instância db.r4.

Otimizado para EBS

A instância de banco de dados usa uma pilha de configuração otimizada e fornece capacidade adicional dedicada para E/S do Amazon EBS. Essa otimização proporciona a melhor performance para seus volumes do EBS ao minimizar a contenção entre a E/S e outro tráfego de sua

instância. Consulte mais informações sobre instâncias otimizadas para o Amazon EBS em Instâncias otimizadas para Amazon EBS no Guia do usuário do Amazon EC2.

As instâncias otimizadas para EBS têm uma linha de base e uma taxa máxima de IOPS. A taxa máxima de IOPS é imposta no nível da instância de banco de dados. Um conjunto de volumes do EBS que se combinam para ter uma taxa de IOPS maior do que o máximo não pode exceder o limite no nível da instância. Por exemplo, se o máximo de IOPS para uma classe específica de instância de banco de dados for 40.000 e você anexar quatro volumes do EBS de 64.000 IOPS, o máximo de IOPS será 40.000 em vez de 256.000. Para saber o máximo de IOPS para cada tipo de instância do EC2, consulte Tipos de instância compatíveis no Guia do usuário do Amazon EC2 para instâncias do Linux.

Máx. Largura de banda EBS (Mbps)

A largura de banda EBS máxima em megabits por segundo. Divida em oito para obter a taxa de transferência esperada em megabytes por segundo.

> **Important**
>
> Os volumes SSD (gp2) de uso geral para instâncias de banco de dados do Amazon RDS têm um limite de taxa de transferência de 250 MiB/s na maioria dos casos. Porém, o limite da taxa de transferência pode variar de acordo com o tipo de volume. Para obter mais informações, consulte Tipos de volume do Amazon EBS no Guia do usuário do Amazon EC2.

Largura de banda de rede

A velocidade da rede em relação a outras classes de instância de banco de dados.

Especificações de hardware para as classes de instância de uso geral

As tabelas a seguir mostram as especificações de computação, memória, armazenamento e largura de banda para as classes de instância de uso geral.

db.m8g: classes de instância de uso geral com processadores Graviton4 da AWS

Classe de instância	vCPU	ECU	Memória (GiB)	Armazenamento da instância (GiB)	Máx. Largura de banda EBS (Mbps)	Largura de banda da rede (Gbps)
db.m8g.48xlarge	192	—	768	Otimizada somente para EBS	40.000	50
db.m8g.24xlarge	96	—	512	Otimizada somente para EBS	30.000	40
db.m8g.16xlarge	64	—	384	Otimizada somente para EBS	20.000	30
db.m8g.12xlarge	48	—	256	Otimizada somente para EBS	15.000	22,5
db.m8g.8xlarge	32	—	128	Otimizada somente para EBS	10.000	15
db.m8g.4xlarge*	16	—	64	Otimizada somente para EBS	Até 10 mil	Até 15
db.m8g.2xlarge*	8	—	32	Otimizada somente para EBS	Até 10 mil	Até 15
db.m8g.xlarge*	4	—	16	Otimizada somente para EBS	Até 10 mil	Até 12,5

Classe de instância	vCPU	ECU	Memória (GiB)	Armazenamento da instância (GiB)	Máx. Largura de banda EBS (Mbps)	Largura de banda da rede (Gbps)
db.m8g.large*	2	—	8	Otimizada somente para EBS	Até 10 mil	Até 12,5

db.m7i: classes de instância de uso geral com processadores Intel Xeon escaláveis de quarta geração

Classe de instância	vCPU	ECU	Memória (GiB)	Armazenamento da instância (GiB)	Máx. Largura de banda EBS (Mbps)	Largura de banda da rede (Gbps)
db.m7i.48xlarge	192	—	768	Otimizada somente para EBS	40.000	50
db.m7i.24xlarge	96	—	384	Otimizada somente para EBS	30.000	37.5
db.m7i.16xlarge	64	—	256	Otimizada somente para EBS	20.000	25
db.m7i.12xlarge	48	—	192	Otimizada somente para EBS	15.000	18,75
db.m7i.8xlarge	32	—	128	Otimizada somente para EBS	10.000	12,5

Classe de instância	vCPU	ECU	Memória (GiB)	Armazenamento da instância (GiB)	Máx. Largura de banda EBS (Mbps)	Largura de banda da rede (Gbps)
db.m7i.4xlarge	16	—	64	Otimizada somente para EBS	Até 10 mil	Até 12,5
db.m7i.2xlarge	8	—	32	Otimizada somente para EBS	Até 10 mil	Até 12,5
db.m7i.xlarge	4	—	16	Otimizada somente para EBS	Até 10 mil	Até 12,5
db.m7i.large	2	—	8	Otimizada somente para EBS	Até 10 mil	Até 12,5

db.m7g: classes de instância de uso geral com processadores Graviton3 da AWS

Classe de instância	vCPU	ECU	Memória (GiB)	Armazenamento da instância (GiB)	Máx. Largura de banda EBS (Mbps)	Largura de banda da rede (Gbps)
db.m7g.16xlarge	64	—	256	Otimizada somente para EBS	20.000	30
db.m7g.12xlarge	48	—	192	Otimizada somente para EBS	15.000	22,5

Classe de instância	vCPU	ECU	Memória (GiB)	Armazenamento da instância (GiB)	Máx. Largura de banda EBS (Mbps)	Largura de banda da rede (Gbps)
db.m7g.8xlarge	32	—	128	Otimizada somente para EBS	10.000	15
db.m7g.4xlarge	16	—	64	Otimizada somente para EBS	Até 10 mil	Até 15
db.m7g.2xlarge*	8	—	32	Otimizada somente para EBS	Até 10 mil	Até 15
db.m7g.xlarge*	4	—	16	Otimizada somente para EBS	Até 10 mil	Até 12,5
db.m7g.large*	2	—	8	Otimizada somente para EBS	Até 10 mil	Até 12,5

db.m6g: classes de instância de uso geral com processadores Graviton2 da AWS

Classe de instância	vCPU	ECU	Memória (GiB)	Armazenamento da instância (GiB)	Máx. Largura de banda EBS (Mbps)	Largura de banda da rede (Gbps)
db.m6g.16xlarge	64	—	256	Otimizada somente para EBS	19.000	25

Classe de instância	vCPU	ECU	Memória (GiB)	Armazenamento da instância (GiB)	Máx. Largura de banda EBS (Mbps)	Largura de banda da rede (Gbps)
db.m6g.12xlarge	48	—	192	Otimizada somente para EBS	13.500	20
db.m6g.8xlarge	32	—	128	Otimizada somente para EBS	9.000	12
db.m6g.4xlarge	16	—	64	Otimizada somente para EBS	4.750	Até 10
db.m6g.2xlarge*	8	—	32	Otimizada somente para EBS	Até 4.750	Até 10
db.m6g.xlarge*	4	—	16	Otimizada somente para EBS	Até 4.750	Até 10
db.m6g.large*	2	—	8	Otimizada somente para EBS	Até 4.750	Até 10

db.m6gd: classes de instância de uso geral com processadores Graviton2 da AWS e armazenamento SSD

Classe de instância	vCPU	ECU	Memória (GiB)	Armazenamento da instância (GiB)	Máx. Largura de banda EBS (Mbps)	Largura de banda da rede (Gbps)
db.m6gd.16xlarge	64	—	256	SSD de NVMe de 2 x 1.900	19.000	25
db.m6gd.12xlarge	48	—	192	SSD de NVMe de 2 x 1.425	13.500	20
db.m6gd.8xlarge	32	—	128	SSD de NVMe de 1 x 1.900	9.000	12
db.m6gd.4xlarge*	16	—	64	SSD de NVMe de 1 x 950	4.750	Até 10
db.m6gd.2xlarge*	8	—	32	SSD de NVMe de 1 x 474	Até 4.750	Até 10
db.m6gd.xlarge*	4	—	16	SSD de NVMe de 1 x 237	Até 4.750	Até 10
db.m6gd.large*	2	—	8	SSD de NVMe de 1 x 118	Até 4.750	Até 10

db.m6id: classes de instância de uso geral com processadores Intel Xeon escaláveis de terceira geração e armazenamento SSD

Classe de instância	vCPU	ECU	Memória (GiB)	Armazenamento da instância (GiB)	Máx. Largura de banda EBS (Mbps)	Largura de banda da rede (Gbps)
db.m6id.32xlarge	128	—	512	SSD de NVMe de 4 x 1.900	40.000	50

Classe de instância	vCPU	ECU	Memória (GiB)	Armazenamento da instância (GiB)	Máx. Largura de banda EBS (Mbps)	Largura de banda da rede (Gbps)
db.m6id.24xlarge	96	—	384	SSD de NVMe de 4 x 1.425	30.000	37.5
db.m6id.16xlarge	64	—	256	SSD de NVMe de 2 x 1.900	20.000	25
db.m6id.12xlarge	48	—	192	SSD de NVMe de 2 x 1.425	15.000	18,75
db.m6id.8xlarge	32	—	128	SSD de NVMe de 1 x 1.900	10.000	12,5
db.m6id.4xlarge*	16	—	64	SSD de NVMe de 1 x 950	Até 10 mil	Até 12,5
db.m6id.4xlarge*	8	—	32	SSD de NVMe de 1 x 474	Até 10 mil	Até 12,5
db.m6id.xlarge*	4	—	16	SSD de NVMe de 1 x 237	Até 10 mil	Até 12,5
db.m6id.large*	2	—	8	SSD de NVMe de 1 x 118	Até 10 mil	Até 12,5

db.m6idn: classes de instância de uso geral com processadores escaláveis Intel Xeon de terceira geração, armazenamento SSD e otimização de rede

Classe de instância	vCPU	ECU	Memória (GiB)	Armazenamento da instância (GiB)	Máx. Largura de banda EBS (Mbps)	Largura de banda da rede (Gbps)
db.m6idn.32xlarge	128	—	512	SSD de NVMe de 4 x 1.900	80.000	200
db.m6idn.24xlarge	96	—	384	SSD de NVMe de 4 x 1.425	60.000	150
db.m6idn.16xlarge	64	—	256	SSD de NVMe de 2 x 1.900	40.000	100
db.m6idn.12xlarge	48	—	192	SSD de NVMe de 2 x 1.425	30.000	75
db.m6idn.8xlarge	32	—	128	1 x SSD NVMe de 1.900	20.000	50
db.m6idn.4xlarge*	16	—	64	SSD de NVMe de 1 x 950	Até 20 mil	Até 50
db.m6idn.2xlarge*	8	—	32	SSD de NVMe de 1 x 474	Até 20 mil	Até 40
db.m6idn.xlarge*	4	—	16	SSD de NVMe de 1 x 237	Até 20 mil	Até 30
db.m6idn.large*	2	—	8	SSD de NVMe de 1 x 118	Até 20 mil	Até 25

db.m6in: classes de instância de uso geral com processadores escaláveis Intel Xeon de terceira geração e otimização de rede

Classe de instância	vCPU	ECU	Memória (GiB)	Armazenamento da instância (GiB)	Máx. Largura de banda EBS (Mbps)	Largura de banda da rede (Gbps)
db.m6in.32xlarge	128	—	512	Otimizada somente para EBS	80.000	200
db.m6in.24xlarge	96	—	384	Otimizada somente para EBS	60.000	150
db.m6in.16xlarge	64	—	256	Otimizada somente para EBS	40.000	100
db.m6in.16xlarge	48	—	192	Otimizada somente para EBS	30.000	75
db.m6in.8xlarge	32	—	128	Otimizada somente para EBS	20.000	50
db.m6in.4xlarge*	16	—	64	Otimizada somente para EBS	Até 20 mil	Até 50
db.m6in.2xlarge*	8	—	32	Otimizada somente para EBS	Até 20 mil	Até 40
db.m6in.xlarge*	4	—	16	Otimizada somente para EBS	Até 20 mil	Até 30

Classe de instância	vCPU	ECU	Memória (GiB)	Armazenamento da instância (GiB)	Máx. Largura de banda EBS (Mbps)	Largura de banda da rede (Gbps)
db.m6in.large*	2	—	8	Otimizada somente para EBS	Até 20 mil	Até 25

db.m6i: classes de instância de uso geral com processadores Intel Xeon escaláveis de terceira geração

Classe de instância	vCPU	ECU	Memória (GiB)	Armazenamento da instância (GiB)	Máx. Largura de banda EBS (Mbps)	Largura de banda da rede (Gbps)
db.m6i.32xlarge	128	—	512	Otimizada somente para EBS	40.000	50
db.m6i.24xlarge	96	—	384	Otimizada somente para EBS	30.000	37.5
db.m6i.16xlarge	64	—	256	Otimizada somente para EBS	20.000	25
db.m6i.12xlarge	48	—	192	Otimizada somente para EBS	15.000	18,75
db.m6i.8xlarge	32	—	128	Otimizada somente para EBS	10.000	12,5

Classe de instância	vCPU	ECU	Memória (GiB)	Armazenamento da instância (GiB)	Máx. Largura de banda EBS (Mbps)	Largura de banda da rede (Gbps)
db.m6i.4xlarge*	16	—	64	Otimizada somente para EBS	Até 10 mil	Até 12,5
db.m6i.2xlarge*	8	—	32	Otimizada somente para EBS	Até 10 mil	Até 12,5
db.m6i.xlarge*	4	—	16	Otimizada somente para EBS	Até 10 mil	Até 12,5
db.m6i.large*	2	—	8	Otimizada somente para EBS	Até 10 mil	Até 12,5

db.m5d: classes de instância de uso geral com processadores Intel Xeon Platinum e armazenamento SSD

Classe de instância	vCPU	ECU	Memória (GiB)	Armazenamento da instância (GiB)	Máx. Largura de banda EBS (Mbps)	Largura de banda da rede (Gbps)
db.m5d.24xlarge	96	345	384	SSD de NVMe de 4 x 900	19.000	25
db.m5d.16xlarge	64	262	256	SSD de NVMe de 4 x 600	13.600	20
db.m5d.12xlarge	48	173	192	SSD de NVMe de 2 x 900	9.500	10

Classe de instância	vCPU	ECU	Memória (GiB)	Armazenamento da instância (GiB)	Máx. Largura de banda EBS (Mbps)	Largura de banda da rede (Gbps)
db.m5d.8xlarge	32	131	128	SSD de NVMe de 2 x 600	6.800	10
db.m5d.4xlarge	16	61	64	SSD de NVMe de 2 x 300	4.750	Até 10
db.m5d.2xlarge	8	31	32	SSD de NVMe de 1 x 300	Até 4.750	Até 10
db.m5d.xlarge*	4	15	16	SSD de NVMe de 1 x 150	Até 4.750	Até 10
db.m5d.large*	2	10	8	SSD de NVMe de 1 x 75	Até 4.750	Até 10

db.m5: classes de instância de uso geral com processadores Intel Xeon Platinum

Classe de instância	vCPU	ECU	Memória (GiB)	Armazenamento da instância (GiB)	Máx. Largura de banda EBS (Mbps)	Largura de banda da rede (Gbps)
db.m5.24xlarge	96	345	384	Otimizada somente para EBS	19.000	25
db.m5.16xlarge	64	262	256	Otimizada somente para EBS	13.600	20

Classe de instância	vCPU	ECU	Memória (GiB)	Armazenamento da instância (GiB)	Máx. Largura de banda EBS (Mbps)	Largura de banda da rede (Gbps)
db.m5.12xlarge	48	173	192	Otimizada somente para EBS	9.500	10
db.m5.8xlarge	32	131	128	Otimizada somente para EBS	6.800	10
db.m5.4xlarge	16	61	64	Otimizada somente para EBS	4.750	Até 10
db.m5.2xlarge*	8	31	32	Otimizada somente para EBS	Até 4.750	Até 10
db.m5.xlarge*	4	15	16	Otimizada somente para EBS	Até 4.750	Até 10
db.m5.large*	2	10	8	Otimizada somente para EBS	Até 4.750	Até 10

db.m4: classes de instância de uso geral com processadores escaláveis Intel Xeon

Classe de instância	vCPU	ECU	Memória (GiB)	Armazenamento da instância (GiB)	Máx. Largura de banda EBS (Mbps)	Largura de banda da rede (Gbps)
db.m4.16xlarge	64	188	256	Otimizada somente para EBS	10.000	25
db.m4.10xlarge	40	124.5	160	Otimizada somente para EBS	4.000	10
db.m4.4xlarge	16	53.5	64	Otimizada somente para EBS	2.000	Alto
db.m4.2xlarge	8	25.5	32	Otimizada somente para EBS	1.000	Alto
db.m4.xlarge	4	13	16	Otimizada somente para EBS	750	Alto
db.m4.large	2	6.5	8	Otimizada somente para EBS	450	Moderada

db.m3: classes de instância de uso geral

Classe de instância	vCPU	ECU	Memória (GiB)	Armazenamento da instância (GiB)	Máx. Largura de banda EBS (Mbps)	Largura de banda da rede (Gbps)
db.m3.2xlarge	8	26	30	Otimizada somente para EBS	1.000	Alto
db.m3.xlarge	4	13	15	Otimizada somente para EBS	500	Alto
db.m3.large	2	6.5	7,5	Somente EBS	—	Moderada
db.m3.medium	1	3	3,75	Somente EBS	—	Moderada

* Esses tipos de instância de banco de dados podem oferecer suporte a um desempenho máximo por 30 minutos pelo menos uma vez a cada 24 horas. Consulte mais informações sobre o desempenho de referência dos tipos de instância subjacentes do EC2 em Instâncias otimizadas para Amazon EBS no Guia do usuário do Amazon EC2.

Especificações de hardware para as classes de instância otimizada para memória

As tabelas a seguir mostram as especificações de computação, memória, armazenamento e largura de banda para as classes de instância otimizada para memória.

db.z1d: classes de instâncias otimizadas para memória

Classe de instância	vCPU	ECU	Memória (GiB)	Armazenamento da instância (GiB)	Máx. Largura de banda EBS (Mbps)	Largura de banda da rede (Gbps)
db.z1d.12xlarge	48	271	384	SSD de NVMe de 2 x 900	14.000	25

Classe de instância	vCPU	ECU	Memória (GiB)	Armazenamento da instância (GiB)	Máx. Largura de banda EBS (Mbps)	Largura de banda da rede (Gbps)
db.z1d.6xlarge	24	134	192	SSD de NVMe de 1 x 900	7.000	10
db.z1d.3xlarge	12	75	96	SSD de NVMe de 1 x 450	3.500	Até 10
db.z1d.2xlarge	8	53	64	SSD de NVMe de 1 x 300	2,333	Até 10
db.z1d.xlarge*	4	28	32	SSD de NVMe de 1 x 150	Até 2.333	Até 10
db.z1d.large*	2	15	16	SSD de NVMe de 1 x 75	Até 2.333	Até 10

db.x2g: classes de instância otimizada para memória com processadores Graviton2 da AWS.

Classe de instância	vCPU	ECU	Memória (GiB)	Armazenamento da instância (GiB)	Máx. Largura de banda EBS (Mbps)	Largura de banda da rede (Gbps)
db.x2g.16xlarge	64	—	1024	Otimizada somente para EBS	19.000	25
db.x2g.12xlarge	48	—	768	Otimizada somente para EBS	14.250	20

Classe de instância	vCPU	ECU	Memória (GiB)	Armazenamento da instância (GiB)	Máx. Largura de banda EBS (Mbps)	Largura de banda da rede (Gbps)
db.x2g.8xlarge	32	—	512	Otimizada somente para EBS	9.500	12
db.x2g.4xlarge	16	—	256	Otimizada somente para EBS	4.750	Até 10
db.x2g.2xlarge	8	—	128	Otimizada somente para EBS	Até 4.750	Até 10
db.x2g.xlarge	4	—	64	Otimizada somente para EBS	Até 4.750	Até 10
db.x2g.large	2	—	32	Otimizada somente para EBS	Até 4.750	Até 10

db.x2idn: classes de instância otimizada para memória com processadores escaláveis Intel Xeon de 3ª geração.

Classe de instância	vCPU	ECU	Memória (GiB)	Armazenamento da instância (GiB)	Máx. Largura de banda EBS (Mbps)	Largura de banda da rede (Gbps)
db.x2idn.32xlarge	128	—	2.048	SSD de NVMe de 2 x 1.900	80.000	100

Classe de instância	vCPU	ECU	Memória (GiB)	Armazenamento da instância (GiB)	Máx. Largura de banda EBS (Mbps)	Largura de banda da rede (Gbps)
db.x2idn.24xlarge	96	—	1.536	SSD de NVMe de 2 x 1.425	60.000	75
db.x2idn.16xlarge	64	—	1,024	SSD de NVMe de 1 x 1.900	40.000	50

db.x2iedn: classes de instância otimizada para memória com SSDs baseados em NVMe com processadores escaláveis Intel Xeon de 3ª geração.

Classe de instância	vCPU	ECU	Memória (GiB)	Armazenamento da instância (GiB)	Máx. Largura de banda EBS (Mbps)	Largura de banda da rede (Gbps)
db.x2iedn.32xlarge	128	—	4.096	SSD de NVMe de 2 x 1.900	80.000	100
db.x2iedn.24xlarge	96	—	3.072	SSD de NVMe de 2 x 1.425	60.000	75
db.x2iedn.16xlarge	64	—	2.048	SSD de NVMe de 1 x 1.900	40.000	50
db.x2iedn.8xlarge	32	—	1,024	SSD de NVMe de 1 x 950	20.000	25
db.x2iedn.4xlarge	16	—	512	SSD de NVMe de 1 x 475	Até 20 mil	Até 25
db.x2iedn.2xlarge	8	—	256	SSD de NVMe de 1 x 237	Até 20 mil	Até 25

Classe de instância	vCPU	ECU	Memória (GiB)	Armazenamento da instância (GiB)	Máx. Largura de banda EBS (Mbps)	Largura de banda da rede (Gbps)
db.x2iedn.xlarge	4	—	128	SSD de NVMe de 1 x 118	Até 20 mil	Até 25

db.x2iezn: classes de instância otimizada para memória com processadores escaláveis Intel Xeon de 2ª geração.

Classe de instância	vCPU	ECU	Memória (GiB)	Armazenamento da instância (GiB)	Máx. Largura de banda EBS (Mbps)	Largura de banda da rede (Gbps)
db.x2iezn.12xlarge	>48	—	1.536	Otimizada somente para EBS	19.000	100
db.x2iezn.8xlarge	32	—	1,024	Otimizada somente para EBS	12.000	75
db.x2iezn.6xlarge	24	—	768	Otimizada somente para EBS	Até 9,500	50
db.x2iezn.4xlarge	16	—	512	Otimizada somente para EBS	Até 4.750	Até 25
db.x2iezn.2xlarge	8	—	256	Otimizada somente para EBS	Até 3.170	Até 25

db.x1e – classes de instância otimizada para memória

Classe de instância	vCPU	ECU	Memória (GiB)	Armazenamento da instância (GiB)	Máx. Largura de banda EBS (Mbps)	Largura de banda da rede (Gbps)
db.x1e.32xlarge	128	340	3,904	Otimizada somente para EBS	14.000	25
db.x1e.16xlarge	64	179	1,952	Otimizada somente para EBS	7.000	10
db.x1e.8xlarge	32	91	976	Otimizada somente para EBS	3.500	Até 10
db.x1e.4xlarge	16	47	488	Otimizada somente para EBS	1,750	Até 10
db.x1e.2xlarge	8	23	244	Otimizada somente para EBS	1.000	Até 10
db.x1e.xlarge	4	12	122	Otimizada somente para EBS	500	Até 10

db.x1 – classes de instância otimizada para memória

Classe de instância	vCPU	ECU	Memória (GiB)	Armazenamento da instância (GiB)	Máx. Largura de banda EBS (Mbps)	Largura de banda da rede (Gbps)
db.x1.32xlarge	128	349	1,952	Otimizada somente para EBS	14.000	25
db.x1.16xlarge	64	174.5	976	Otimizada somente para EBS	7.000	10

db.r8g: classes de instância otimizada para memória com processadores Graviton4 da AWS

Classe de instância	vCPU	ECU	Memória (GiB)	Armazenamento da instância (GiB)	Máx. Largura de banda EBS (Mbps)	Largura de banda da rede (Gbps)
db.r8g.48xlarge	192	—	1536	Otimizada somente para EBS	40.000	50
db.r8g.24xlarge	96	—	768	Otimizada somente para EBS	30.000	40
db.r8g.16xlarge	64	—	512	Otimizada somente para EBS	20.000	30
db.r8g.12xlarge	48	—	384	Otimizada somente para EBS	15.000	22,5

Classe de instância	vCPU	ECU	Memória (GiB)	Armazenamento da instância (GiB)	Máx. Largura de banda EBS (Mbps)	Largura de banda da rede (Gbps)
db.r8g.8xlarge	32	—	256	Otimizada somente para EBS	10.000	15
db.r8g.4xlarge*	16	—	128	Otimizada somente para EBS	Até 10 mil	Até 15
db.r8g.2xlarge*	8	—	64	Otimizada somente para EBS	Até 10 mil	Até 15
db.r8g.xlarge*	4	—	32	Otimizada somente para EBS	Até 10 mil	Até 12,5
db.r8g.large*	2	—	16	Otimizada somente para EBS	Até 10 mil	Até 12,5

db.r7i: classes de instância otimizada para memória com processadores Intel Xeon escaláveis de quarta geração

Classe de instância	vCPU	ECU	Memória (GiB)	Armazenamento da instância (GiB)	Máx. Largura de banda EBS (Mbps)	Largura de banda da rede (Gbps)
db.r7i.48xlarge	192	—	1536	Otimizada somente para EBS	40.000	50

Classe de instância	vCPU	ECU	Memória (GiB)	Armazenamento da instância (GiB)	Máx. Largura de banda EBS (Mbps)	Largura de banda da rede (Gbps)
db.r7i.24xlarge	96	—	768	Otimizada somente para EBS	30.000	37.5
db.r7i.16xlarge	64	—	512	Otimizada somente para EBS	20.000	25
db.r7i.12xlarge	48	—	384	Otimizada somente para EBS	15.000	18,75
db.r7i.8xlarge	32	—	256	Otimizada somente para EBS	10.000	12,5
db.r7i.4xlarge	16	—	128	Otimizada somente para EBS	Até 10 mil	Até 12,5
db.r7i.2xlarge	8	—	64	Otimizada somente para EBS	Até 10 mil	Até 12,5
db.r7i.xlarge	4	—	32	Otimizada somente para EBS	Até 10 mil	Até 12,5
db.r7i.large	2	—	16	Otimizada somente para EBS	Até 10 mil	Até 12,5

db.r7g: classes de instância otimizada para memória com processadores Graviton3 da AWS

Classe de instância	vCPU	ECU	Memória (GiB)	Armazenamento da instância (GiB)	Máx. Largura de banda EBS (Mbps)	Largura de banda da rede (Gbps)
db.r7g.16xlarge	64	—	512	Otimizada somente para EBS	20.000	30
db.r7g.12xlarge	48	—	384	Otimizada somente para EBS	15.000	22,5
db.r7g.8xlarge	32	—	256	Otimizada somente para EBS	10.000	15
db.r7g.4xlarge	16	—	128	Otimizada somente para EBS	Até 10 mil	Até 15
db.r7g.2xlarge*	8	—	64	Otimizada somente para EBS	Até 10 mil	Até 15
db.r7g.xlarge*	4	—	32	Otimizada somente para EBS	Até 10 mil	Até 12,5
db.r7g.large*	2	—	16	Otimizada somente para EBS	Até 10 mil	Até 12,5

db.r6g: classes de instância otimizada para memória com processadores Graviton2 da AWS

Classe de instância	vCPU	ECU	Memória (GiB)	Armazenamento da instância (GiB)	Máx. Largura de banda EBS (Mbps)	Largura de banda da rede (Gbps)
db.r6g.16xlarge	64	—	512	Otimizada somente para EBS	19.000	25
db.r6g.12xlarge	48	—	384	Otimizada somente para EBS	13.500	20
db.r6g.8xlarge	32	—	256	Otimizada somente para EBS	9.000	12
db.r6g.4xlarge	16	—	128	Otimizada somente para EBS	4.750	Até 10
db.r6g.2xlarge*	8	—	64	Otimizada somente para EBS	Até 4.750	Até 10
db.r6g.xlarge*	4	—	32	Otimizada somente para EBS	Até 4.750	Até 10
db.r6g.large*	2	—	16	Otimizada somente para EBS	Até 4.750	Até 10

db.r6gd: classes de instância otimizada para memória com processadores Graviton2 da AWS e armazenamento de memória SSD

Classe de instância	vCPU	ECU	Memória (GiB)	Armazenamento da instância (GiB)	Máx. Largura de banda EBS (Mbps)	Largura de banda da rede (Gbps)
db.r6gd.16xlarge	64	—	512	SSD de NVMe de 2 x 1.900	19.000	25
db.r6gd.12xlarge	48	—	384	SSD de NVMe de 2 x 1.425	13.500	20
db.r6gd.8xlarge	32	—	256	SSD de NVMe de 1 x 1.900	9.000	12
db.r6gd.4xlarge	16	—	128	SSD de NVMe de 1 x 950	4.750	Até 10
db.r6gd.2xlarge	8	—	64	SSD de NVMe de 1 x 474	Até 4.750	Até 10
db.r6gd.xlarge	4	—	32	SSD de NVMe de 1 x 237	Até 4.750	Até 10
db.r6gd.large	2	—	16	SSD de NVMe de 1 x 118	Até 4.750	Até 10

db.r6id: classes de instância otimizada para memória com processadores escaláveis Intel Xeon de 3ª geração

Classe de instância	vCPU	ECU	Memória (GiB)	Armazenamento da instância (GiB)	Máx. Largura de banda EBS (Mbps)	Largura de banda da rede (Gbps)
db.r6id.32xlarge	128	—	1,024	SSD de NVMe de 4 x 1.900	40.000	50

Classe de instância	vCPU	ECU	Memória (GiB)	Armazenamento da instância (GiB)	Máx. Largura de banda EBS (Mbps)	Largura de banda da rede (Gbps)
db.r6id.32xlarge	96	—	768	SSD de NVMe de 4 x 1.425	30.000	37.5
db.r6id.16xlarge	64	—	512	SSD de NVMe de 2 x 1.900	20.000	25
db.r6id.12xlarge	48	—	384	SSD de NVMe de 2 x 1.425	15.000	18,75
db.r6id.8xlarge	32	—	256	SSD de NVMe de 1 x 1.900	10.000	12,5
db.r6id.4xlarge*	16	—	128	SSD de NVMe de 1 x 950	Até 10 mil	Até 12,5
db.r6id.2xlarge*	8	—	64	SSD de NVMe de 1 x 474	Até 10 mil	Até 12,5
db.r6id.xlarge*	4	—	32	SSD de NVMe de 1 x 237	Até 10 mil	Até 12,5
db.r6id.large*	2	—	16	SSD de NVMe de 1 x 118	Até 10 mil	Até 12,5

db.r6idn: classes de instância otimizada para memória com processadores escaláveis Intel Xeon de 3ª geração, armazenamento SSD e otimização de rede

Classe de instância	vCPU	ECU	Memória (GiB)	Armazenamento da instância (GiB)	Máx. Largura de banda EBS (Mbps)	Largura de banda da rede (Gbps)
db.r6idn.32xlarge	128	—	1,024	SSD de NVMe de 4 x 1.900	80.000	200
db.r6idn.24xlarge	96	—	768	SSD de NVMe de 4 x 1.425	60.000	150
db.r6idn.16xlarge	64	—	512	SSD de NVMe de 2 x 1.900	40.000	100
db.r6idn.12xlarge	48	—	384	SSD de NVMe de 2 x 1.425	30.000	75
db.r6idn.12xlarge	32	—	256	SSD de NVMe de 1 x 1.900	20.000	50
db.r6idn.4xlarge*	16	—	128	SSD de NVMe de 1 x 950	Até 20 mil	Até 50
db.r6idn.12xlarge	8	—	64	SSD de NVMe de 1 x 474	Até 20 mil	Até 40
db.r6idn.xlarge*	4	—	32	SSD de NVMe de 1 x 237	Até 20 mil	Até 30
db.r6idn.large*	2	—	16	SSD de NVMe de 1 x 118	Até 20 mil	Até 25

db.r6in: classes de instância otimizada para memória com processadores escaláveis Intel Xeon de 3ª geração e otimização de rede

Classe de instância	vCPU	ECU	Memória (GiB)	Armazenamento da instância (GiB)	Máx. Largura de banda EBS (Mbps)	Largura de banda da rede (Gbps)
db.r6in.32xlarge	128	—	1,024	Otimizada somente para EBS	80.000	200
db.r6in.24xlarge	96	—	768	Otimizada somente para EBS	60.000	150
db.r6in.24xlarge	64	—	512	Otimizada somente para EBS	40.000	100
db.r6in.24xlarge	48	—	384	Otimizada somente para EBS	30.000	75
db.r6in.8xlarge	32	—	256	Otimizada somente para EBS	20.000	50
db.r6in.4xlarge*	16	—	128	Otimizada somente para EBS	Até 20 mil	Até 50
db.r6in.2xlarge*	8	—	64	Otimizada somente para EBS	Até 20 mil	Até 40
db.r6in.xlarge*	4	—	32	Otimizada somente para EBS	Até 20 mil	Até 30

Classe de instância	vCPU	ECU	Memória (GiB)	Armazenamento da instância (GiB)	Máx. Largura de banda EBS (Mbps)	Largura de banda da rede (Gbps)
db.r6in.large*	2	—	16	Otimizada somente para EBS	Até 20 mil	Até 25

db.r6i: classes de instância otimizada para memória pré-configuradas para alto nível de memória, armazenamento e E/S

Classe de instância	vCPU	ECU	Memória (GiB)	Armazenamento da instância (GiB)	Máx. Largura de banda EBS (Mbps)	Largura de banda da rede (Gbps)
db.r6i.8xlarge.tpc2.mem4x	32	—	1024	Otimizada somente para EBS	40.000	50
db.r6i.8xlarge.tpc2.mem3x	32	—	768	Otimizada somente para EBS	30.000	37.5
db.r6i.6xlarge.tpc2.mem4x	24	—	768	Otimizada somente para EBS	30.000	37.5
db.r6i.4xlarge.tpc2.mem4x	16	—	512	Otimizada somente para EBS	20.000	25
db.r6i.4xlarge.tpc2.mem3x	16	—	384	Otimizada somente para EBS	15.000	18,75

Classe de instância	vCPU	ECU	Memória (GiB)	Armazenamento da instância (GiB)	Máx. Largura de banda EBS (Mbps)	Largura de banda da rede (Gbps)
db.r6i.4xlarge.tpc2.mem2x	16	—	256	Otimizada somente para EBS	10.000	12,5
db.r6i.2xlarge.tpc2.mem8x	8	—	512	Otimizada somente para EBS	20.000	12,5
db.r6i.2xlarge.tpc2.mem4x	8	—	256	Otimizada somente para EBS	10.000	12,5
db.r6i.2xlarge.tpc1.mem2x	8	—	128	Otimizada somente para EBS	Até 10 mil	12,5
db.r6i.xlarge.tpc2.mem4x	4	—	128	Otimizada somente para EBS	Até 10 mil	12,5
db.r6i.xlarge.tpc2.mem2x	4	—	64	Otimizada somente para EBS	Até 10 mil	12,5
db.r6i.large.tpc1.mem2x	2	—	32	Otimizada somente para EBS	Até 10 mil	12,5

db.r6i: classes de instância otimizada para memória com processadores Intel Xeon escaláveis de terceira geração

Classe de instância	vCPU	ECU	Memória (GiB)	Armazenamento da instância (GiB)	Máx. Largura de banda EBS (Mbps)	Largura de banda da rede (Gbps)
db.r6i.32xlarge	128	—	1,024	Otimizada somente para EBS	40.000	50
db.r6i.24xlarge	96	—	768	Otimizada somente para EBS	30.000	37.5
db.r6i.16xlarge	64	—	512	Otimizada somente para EBS	20.000	25
db.r6i.12xlarge	48	—	384	Otimizada somente para EBS	15.000	18,75
db.r6i.8xlarge	32	—	256	Otimizada somente para EBS	10.000	12,5
db.r6i.4xlarge*	16	—	128	Otimizada somente para EBS	Até 10 mil	Até 12,5
db.r6i.2xlarge*	8	—	64	Otimizada somente para EBS	Até 10 mil	Até 12,5
db.r6i.xlarge*	4	—	32	Otimizada somente para EBS	Até 10 mil	Até 12,5

Classe de instância	vCPU	ECU	Memória (GiB)	Armazenamento da instância (GiB)	Máx. Largura de banda EBS (Mbps)	Largura de banda da rede (Gbps)
db.r6i.large*	2	—	16	Otimizada somente para EBS	Até 10 mil	Até 12,5

db.r5d: classes de instância otimizada para memória com processadores Intel Xeon Platinum e armazenamento SSD

Classe de instância	vCPU	ECU	Memória (GiB)	Armazenamento da instância (GiB)	Máx. Largura de banda EBS (Mbps)	Largura de banda da rede (Gbps)
db.r5d.24xlarge	96	347	768	SSD de NVMe de 4 x 900	19.000	25
db.r5d.16xlarge	64	264	512	SSD de NVMe de 4 x 600	13.600	20
db.r5d.12xlarge	48	173	384	SSD de NVMe de 2 x 900	9.500	10
db.r5d.8xlarge	32	132	256	SSD de NVMe de 2 x 600	6.800	10
db.r5d.4xlarge	16	71	128	SSD de NVMe de 2 x 300	4.750	Até 10
db.r5d.2xlarge*	8	38	64	SSD de NVMe de 1 x 300	Até 4.750	Até 10
db.r5d.xlarge*	4	19	32	SSD de NVMe de 1 x 150	Até 4.750	Até 10

Classe de instância	vCPU	ECU	Memória (GiB)	Armazenamento da instância (GiB)	Máx. Largura de banda EBS (Mbps)	Largura de banda da rede (Gbps)
db.r5d.large*	2	10	16	SSD de NVMe de 1 x 75	Até 4.750	Até 10

db.r5: classes de instância otimizada para memória Oracle pré-configuradas para alta memória, armazenamento e E/S

Classe de instância	vCPU	ECU	Memória (GiB)	Armazenamento da instância (GiB)	Máx. Largura de banda EBS (Mbps)	Largura de banda da rede (Gbps)
db.r5b.8xlarge.tpc2.mem3x	32	—	768	Otimizada somente para EBS	60.000	25
db.r5b.6xlarge.tpc2.mem4x	24	—	768	Otimizada somente para EBS	60.000	25
db.r5b.4xlarge.tpc2.mem4x	16	—	512	Otimizada somente para EBS	40.000	20
db.r5b.4xlarge.tpc2.mem3x	16	—	384	Otimizada somente para EBS	30.000	10
db.r5b.4xlarge.tpc2.mem2x	16	—	256	Otimizada somente para EBS	20.000	10

Classe de instância	vCPU	ECU	Memória (GiB)	Armazenamento da instância (GiB)	Máx. Largura de banda EBS (Mbps)	Largura de banda da rede (Gbps)
db.r5b.2xlarge.tpc2.mem8x	8	—	512	Otimizada somente para EBS	40.000	20
db.r5b.2xlarge.tpc2.mem4x	8	—	256	Otimizada somente para EBS	20.000	10
db.r5b.2xlarge.tpc1.mem2x	8	—	128	Otimizada somente para EBS	10.000	Até 10
db.r5b.xlarge.tpc2.mem4x	4	—	128	Otimizada somente para EBS	10.000	Até 10
db.r5b.xlarge.tpc2.mem2x	4	—	64	Otimizada somente para EBS	Até 10 mil	Até 10
db.r5b.large.tpc1.mem2x	2	—	32	Otimizada somente para EBS	Até 10 mil	Até 10

db.r5b: classes de instância otimizada para memória com processadores Intel Xeon Platinum e armazenamento SSD

Classe de instância	vCPU	ECU	Memória (GiB)	Armazenamento da instância (GiB)	Máx. Largura de banda EBS (Mbps)	Largura de banda da rede (Gbps)
db.r5b.24xlarge	96	347	768	Otimizada somente para EBS	60.000	25
db.r5b.16xlarge	64	264	512	Otimizada somente para EBS	40.000	20
db.r5b.12xlarge	48	173	384	Otimizada somente para EBS	30.000	10
db.r5b.8xlarge	32	132	256	Otimizada somente para EBS	20.000	10
db.r5b.4xlarge	16	71	128	Otimizada somente para EBS	10.000	Até 10
db.r5b.2xlarge*	8	38	64	Otimizada somente para EBS	Até 10 mil	Até 10
db.r5b.xlarge*	4	19	32	Otimizada somente para EBS	Até 10 mil	Até 10
db.r5b.large*	2	10	16	Otimizada somente para EBS	Até 10 mil	Até 10

db.r5: classes de instância otimizada para memória pré-configuradas para alta memória, armazenamento e E/S

Classe de instância	vCPU	ECU	Memória (GiB)	Armazenamento da instância (GiB)	Máx. Largura de banda EBS (Mbps)	Largura de banda da rede (Gbps)
db.r5.12xlarge.tpc2.mem2x	48	—	768	Otimizada somente para EBS	19.000	25
db.r5.8xlarge.tpc2.mem3x	32	—	768	Otimizada somente para EBS	19.000	25
db.r5.6xlarge.tpc2.mem4x	24	—	768	Otimizada somente para EBS	19.000	25
db.r5.4xlarge.tpc2.mem4x	16	—	512	Otimizada somente para EBS	13.600	20
db.r5.4xlarge.tpc2.mem3x	16	—	384	Otimizada somente para EBS	9.500	10
db.r5.4xlarge.tpc2.mem2x	16	—	256	Otimizada somente para EBS	6.800	10
db.r5.2xlarge.tpc2.mem8x	8	—	512	Otimizada somente para EBS	13.600	20

Classe de instância	vCPU	ECU	Memória (GiB)	Armazenamento da instância (GiB)	Máx. Largura de banda EBS (Mbps)	Largura de banda da rede (Gbps)
db.r5.2xlarge.tpc2.mem4x	8	—	256	Otimizada somente para EBS	6.800	10
db.r5.2xlarge.tpc1.mem2x	8	—	128	Otimizada somente para EBS	4.750	Até 10
db.r5.xlarge.tpc2.mem4x	4	—	128	Otimizada somente para EBS	4.750	Até 10
db.r5.xlarge.tpc2.mem2x	4	—	64	Otimizada somente para EBS	Até 4.750	Até 10
db.r5.large.tpc1.mem2x	2	—	32	Otimizada somente para EBS	Até 4.750	Até 10

db.r5: classes de instância otimizada para memória

Classe de instância	vCPU	ECU	Memória (GiB)	Armazenamento da instância (GiB)	Máx. Largura de banda EBS (Mbps)	Largura de banda da rede (Gbps)
db.r5.24xlarge	96	347	768	Otimizada somente para EBS	19.000	25

Classe de instância	vCPU	ECU	Memória (GiB)	Armazenamento da instância (GiB)	Máx. Largura de banda EBS (Mbps)	Largura de banda da rede (Gbps)
db.r5.16xlarge	64	264	512	Otimizada somente para EBS	13.600	20
db.r5.12xlarge	48	173	384	Otimizada somente para EBS	9.500	12
db.r5.8xlarge	32	132	256	Otimizada somente para EBS	6.800	10
db.r5.4xlarge	16	71	128	Otimizada somente para EBS	4.750	Até 10
db.r5.2xlarge*	8	38	64	Otimizada somente para EBS	Até 4.750	Até 10
db.r5.xlarge*	4	19	32	Otimizada somente para EBS	Até 4.750	Até 10
db.r5.large*	2	10	16	Otimizada somente para EBS	Até 4.750	Até 10

db.r4: classes de instância otimizada para memória com processadores Intel Xeon Platinum e armazenamento SSD

Classe de instância	vCPU	ECU	Memória (GiB)	Armazenamento da instância (GiB)	Máx. Largura de banda EBS (Mbps)	Largura de banda da rede (Gbps)
db.r4.16xlarge	64	195	488	Otimizada somente para EBS	14.000	25
db.r4.8xlarge	32	99	244	Otimizada somente para EBS	7.000	10
db.r4.4xlarge	16	53	122	Otimizada somente para EBS	3.500	Até 10
db.r4.2xlarge	8	27	61	Otimizada somente para EBS	1.700	Até 10
db.r4.xlarge	4	13,5	30,5	Otimizada somente para EBS	850	Até 10
db.r4.large	2	7	15,25	Otimizada somente para EBS	425	Até 10

db.r3 – classes de instância otimizada para memória

Classe de instância	vCPU	ECU	Memória (GiB)	Armazenamento da instância (GiB)	Máx. Largura de banda EBS (Mbps)	Largura de banda da rede (Gbps)
db.r3.8xlarge**	32	104	244	Somente EBS	—	10

Classe de instância	vCPU	ECU	Memória (GiB)	Armazenamento da instância (GiB)	Máx. Largura de banda EBS (Mbps)	Largura de banda da rede (Gbps)
db.r3.4xlarge	16	52	122	Otimizada somente para EBS	2.000	Alto
db.r3.2xlarge	8	26	61	Otimizada somente para EBS	1.000	Alto
db.r3.xlarge	4	13	30.5	Otimizada somente para EBS	500	Moderada
db.r3.large	2	6.5	15.25	Otimizada somente para EBS	—	Moderada

* Esses tipos de instância de banco de dados podem oferecer suporte a um desempenho máximo por 30 minutos pelo menos uma vez a cada 24 horas. Consulte mais informações sobre o desempenho de referência dos tipos de instância subjacentes do EC2 em Instâncias otimizadas para Amazon EBS no Guia do usuário do Amazon EC2.

** A classe de instância de banco de dados r3.8xlarge não possui largura de banda dedicada do EBS e, portanto, não oferece otimização do EBS. Para essa classe de instância, o tráfego de rede e o trafego do Amazon EBS compartilham a mesma interface de rede de 10 gigabits.

Especificações de hardware para as classes de instância otimizada para computação

As tabelas a seguir mostram as especificações de computação, memória, armazenamento e largura de banda para as classes de instância otimizada para computação.

db.c6gd: classes de instâncias otimizadas para computação (somente para implantações de clusters de banco de dados multi-AZ)

Classe de instância	vCPU	ECU	Memória (GiB)	Armazenamento da instância (GiB)	Máx. Largura de banda EBS (Mbps)	Largura de banda da rede (Gbps)
db.c6gd.16xlarge	64	—	128	SSD de NVMe de 2 x 1.900	19.000	25
db.c6gd.12xlarge	48	—	96	SSD de NVMe de 2 x 1.425	13.500	20
db.c6gd.8xlarge	32	—	64	SSD de NVMe de 1 x 1.900	9.000	12
db.c6gd.4xlarge	16	—	32	SSD de NVMe de 1 x 950	4.750	Até 10
db.c6gd.2xlarge	8	—	16	SSD de NVMe de 1 x 474	Até 4.750	Até 10
db.c6gd.xlarge	4	—	8	SSD de NVMe de 1 x 237	Até 4.750	Até 10
db.c6gd.large	2	—	4	SSD de NVMe de 1 x 118	Até 4.750	Até 10
db.c6gd.medium	1	—	2	NVMe SSD de 1 x 59	Até 4.750	Até 10

Especificações de hardware para as classes de instância de desempenho expansível

As tabelas a seguir mostram as especificações de computação, memória, armazenamento e largura de banda para as classes de instância de desempenho expansível.

db.t4g: classes de instância de desempenho expansível com processadores Graviton2 da AWS

Classe de instância	vCPU	ECU	Memória (GiB)	Armazenamento da instância (GiB)	Máx. Largura de banda EBS (Mbps)	Largura de banda da rede (Gbps)
db.t4g.2xlarge*	8	—	32	Otimizada somente para EBS	Até 2.780	Até 5
db.t4g.xlarge*	4	—	16	Otimizada somente para EBS	Até 2.780	Até 5
db.t4g.large*	2	—	8	Otimizada somente para EBS	Até 2.780	Até 5
db.t4g.medium*	2	—	4	Otimizada somente para EBS	Até 2.085	Até 5
db.t4g.small*	2	—	2	Otimizada somente para EBS	Até 2.085	Até 5
db.t4g.micro*	2	—	1	Otimizada somente para EBS	Até 2.085	Até 5

db.t3: classes de instância expansível

Classe de instância	vCPU	ECU	Memória (GiB)	Armazenamento da instância (GiB)	Máx. Largura de banda EBS (Mbps)	Largura de banda da rede (Gbps)
db.t3.2xlarge*	8	Variáv	32	Otimizada somente para EBS	Até 2.048	Até 5
db.t3.xlarge*	4	Variáv	16	Otimizada somente para EBS	Até 2.048	Até 5
db.t3.large*	2	Variáv	8	Otimizada somente para EBS	Até 2.048	Até 5
db.t3.medium*	2	Variáv	4	Otimizada somente para EBS	Até 1.536	Até 5
db.t3.small*	2	Variáv	2	Otimizada somente para EBS	Até 1.536	Até 5
db.t3.micro*	2	Variáv	1	Otimizada somente para EBS	Até 1.536	Até 5

db.t2: classes de instância expansível

Classe de instância	vCPU	ECU	Memória (GiB)	Armazenamento da instância (GiB)	Máx. Largura de banda EBS (Mbps)	Largura de banda da rede (Gbps)
db.t2.2xlarge	8	Variáv	32	Somente EBS	—	Moderada

Classe de instância	vCPU	ECU	Memória (GiB)	Armazenamento da instância (GiB)	Máx. Largura de banda EBS (Mbps)	Largura de banda da rede (Gbps)
db.t2.xlarge	4	Variáv	16	Somente EBS	—	Moderada
db.t2.large	2	Variáv	8	Somente EBS	—	Moderada
db.t2.medium	2	Variáv	4	Somente EBS	—	Moderada
db.t2.small	1	Variáv	2	Somente EBS	—	Baixo
db.t2.micro	1	Variáv	1	Somente EBS	—	Baixo

* Esses tipos de instância de banco de dados podem oferecer suporte a um desempenho máximo por 30 minutos pelo menos uma vez a cada 24 horas. Consulte mais informações sobre o desempenho de referência dos tipos de instância subjacentes do EC2 em Instâncias otimizadas para Amazon EBS no Guia do usuário do Amazon EC2.

Armazenamento de instâncias de banco de dados do Amazon RDS

As instâncias de banco de dados do Amazon RDS para Db2, MySQL, MariaDB, PostgreSQL, Oracle e Microsoft SQL Server utilizam volumes do Amazon Elastic Block Store (Amazon EBS) para armazenamento de logs e banco de dados.

Em alguns casos, a workload do banco de dados pode não conseguir alcançar 100% das IOPS provisionadas. Para obter mais informações, consulte Fatores que afetam a performance de armazenamento.

Para obter mais informações sobre a definição de preço de armazenamento de instâncias, consulte Definição de preço do Amazon RDS.

Tipos de armazenamento do Amazon RDS

O Amazon RDS oferece três tipos de armazenamento: SSD de IOPS provisionadas (também conhecido como io1 e io2 Block Express), SSD de uso geral (também conhecido como gp2 e gp3) e magnético (também conhecido como padrão). Eles diferem em características de performance e preço, o que significa que você pode adaptar o custo e a performance de armazenamento às necessidades da workload do banco de dados. É possível criar instâncias de banco de dados do Db2, do MySQL, do MariaDB, do Oracle, do SQL Server e do PostgreSQL RDS com até 64 tebibytes (TiB) de armazenamento. O RDS para Db2 não é compatível com os tipos de armazenamento magnético e gp2.

A lista a seguir descreve resumidamente os três tipos de armazenamento:

- IOPS provisionadas: o armazenamento de IOPS provisionadas foi projetado para atender às necessidades de workloads de uso intenso de E/S, em especial workloads de banco de dados que exigem baixa latência de E/S e throughput de E/S consistente. O armazenamento de IOPS provisionadas é mais adequado para ambientes de produção.

  Para obter mais informações sobre o armazenamento de IOPS provisionadas, incluindo os intervalos de tamanho de armazenamento, consulte Armazenamento SSD de IOPS provisionadas.
- SSD de uso geral: os volumes de SSD de uso geral oferecem armazenamento econômico ideal para uma ampla variedade de workloads em execução em instâncias de banco de dados de médio porte. O armazenamento de uso geral é mais adequado para ambientes de desenvolvimento e teste.

Para obter mais informações sobre o armazenamento SSD de uso geral, incluindo os intervalos de tamanho de armazenamento, consulte Armazenamento SSD de uso geral.

- Magnético – o Amazon RDS também é compatível com o armazenamento magnético para retrocompatibilidade. Recomendamos que você use SSD de uso geral ou SSD de IOPS provisionadas para toda e qualquer nova necessidade de armazenamento. A quantidade máxima de armazenamento permitida para instâncias de banco de dados no armazenamento magnético é de 3 TiB. Para obter mais informações, consulte Armazenamento magnético (herdado, não recomendado).

Quando você seleciona SSD de uso geral ou SSD de IOPS provisionadas, dependendo do mecanismo selecionado e da quantidade de armazenamento solicitada, o Amazon RDS distribui automaticamente vários volumes para melhorar a performance, conforme mostrado na tabela a seguir.

Mecanismo do banco de dados	Tamanho de armazenamento do Amazon RDS	Número de volumes provisionados
Db2	Menor que 400 GiB	1
Db2	De 400 GiB a 65.536 GiB	4
MariaDB, MySQL e PostgreSQL	Menor que 400 GiB	1
MariaDB, MySQL e PostgreSQL	De 400 GiB a 65.536 GiB	4
Oracle	Menor que 200 GiB	1
Oracle	De 200 GiB a 65.536 GiB	4
SQL Server	Any	1

Quando você modifica um volume de SSD de uso geral ou de SSD de IOPS provisionadas, ele passa por uma sequência de estados. Quando o volume está no estado `optimizing`, a performance de volume fica entre as especificações de configuração de origem e de destino. A performance de volume transitório não será menor do que a menor das duas especificações.

> **Important**
>
> Quando você modifica o armazenamento de uma instância para que ele passe de um para quatro volumes, ou quando você modifica uma instância usando armazenamento magnético, o Amazon RDS não usa o recurso Volumes Elásticos. Em vez disso, o Amazon RDS provisiona novos volumes e move de forma transparente os dados do volume antigo para os novos volumes. Essa operação consome uma quantidade significativa de IOPS e throughput dos volumes antigos e novos. Dependendo do tamanho do volume e da quantidade de workloads do banco de dados presente durante a modificação, essa operação pode consumir grande quantidade de IOPS, aumentar significativamente a latência de E/S e levar várias horas para ser concluída, enquanto a instância do RDS permanece no estado `Modifying`.

Armazenamento SSD de IOPS provisionadas

Para uma aplicação de produção que exija uma performance de E/S rápido e consistente, recomendamos o armazenamento de IOPS provisionadas. O armazenamento de IOPS provisionadas é um tipo de armazenamento que oferece performance previsível e latência consistentemente baixa. O armazenamento de IOPS provisionadas é otimizado para workloads de processamento de transações on-line (OLTP) que exigem performance consistente. As IOPS provisionadas ajudam a ajustar a performance dessas workloads.

Ao criar uma instância de banco de dados, você especifica a taxa de IOPS e o tamanho do volume. O Amazon RDS fornecerá essa taxa de IOPS para a instância de banco de dados até você alterá-la.

O Amazon RDS oferece dois tipos de armazenamento SSD com IOPS provisionadas: Armazenamento io2 Block Express (recomendado) e armazenamento io1 (geração anterior).

Armazenamento io2 Block Express (recomendado)

Em relação a workloads com uso intenso de E/S e sensíveis à latência, é possível usar o armazenamento SSD de IOPS provisionadas io2 Block Express e alcançar até 256 mil operações

de E/S por segundo (IOPS). O throughput de volumes io2 Block Express varia com base na quantidade de IOPS provisionadas por volume e no tamanho das operações de E/S que estão sendo executadas.

Todos os volumes io2 do RDS baseados no AWS Nitro System são volumes io2 Block Express e oferecem latência média inferior a um milissegundo. As instâncias de banco de dados não baseadas no AWS Nitro System são volumes io2.

A tabela a seguir mostra o intervalo de IOPS provisionadas e o throughput máximo de cada mecanismo de banco de dados e o intervalo de tamanho de armazenamento.

Mecanismo do banco de dados	Intervalo do tamanho do armazenamento	Intervalo de IOPS provisionadas	Throughput máximo
Db2, MariaDB, MySQL e PostgreSQL	De 100 GiB a 65.536 GiB	De 1.000 a 256.000 IOPS	16.000 MiB/s
Oracle	De 100 GiB a 199 GiB	De 1.000 a 199.000 IOPS	16.000 MiB/s
Oracle	De 200 GiB a 65.536 GiB	De 1.000 a 256.000 IOPS	16.000 MiB/s
SQL Server	De 20 GiB a 65.536 GiB	De 1.000 a 256.000 IOPS	16.000 MiB/s

As faixas de IOPS e de tamanho de armazenamento têm as seguintes restrições:

- A proporção de IOPS com relação ao armazenamento alocado (em GiB) não deve ser maior que 1000:1. Para instâncias de banco de dados não baseadas no AWS Nitro System, a proporção é de 500:1.
- As IOPS máximas podem ser provisionadas com volumes de 256 GiB e acima (1.000 IOPS × 256 GiB = 256.000 IOPS). Para instâncias de banco de dados não baseadas no AWS Nitro System, o máximo de IOPS é alcançado em 512 GiB (500 IOPS x 512 GiB = 256.000 IOPS).
- A throughput é dimensionada proporcionalmente até 0,256 MiB/s por IOPS provisionadas. O throughput máximo de 4.000 MiB/s pode ser alcançada em 256.000 IOPS com um tamanho de E/S de 16 KiB e 16.000 IOPS ou superior com um tamanho de E/S de 256 KiB. Para instâncias de

banco de dados não baseadas no AWS Nitro System, o throughput máximo de 2.000 MiB/s pode ser alcançado em 128.000 IOPS com um tamanho de E/S de 16 KiB.

- Se você estiver usando autoescalabilidade de armazenamento, as mesmas taxas entre IOPS e limite máximo de armazenamento (em GiB) também se aplicarão. Para obter mais informações sobre autoescalabilidade automática de armazenamento, consulte Gerenciar a capacidade automaticamente com a escalabilidade automática de armazenamento do Amazon RDS.

Os volumes do Amazon RDS io2 Block Express estão disponíveis nas seguintes Regiões da AWS:

- Ásia-Pacífico (Hong Kong)
- Asia Pacific (Mumbai)
- Ásia-Pacífico (Seul)
- Ásia-Pacífico (Singapura)
- Ásia-Pacífico (Sydney)
- Ásia-Pacífico (Tóquio)
- Canadá (Central)
- Europa (Frankfurt)
- Europa (Irlanda)
- Europe (London)
- Europe (Stockholm)
- Oriente Médio (Barém)
- Leste dos EUA (Ohio)
- Leste dos EUA (N. da Virgínia)
- Oeste dos EUA (N. da Califórnia)
- Oeste dos EUA (Oregon)

armazenamento io1 (geração anterior)

Para workloads com uso intenso de E/S, você pode usar o armazenamento SSD io1 de IOPS provisionadas e alcançar até 256.000 operações de E/S por segundo (IOPS). O throughput de volumes io1 varia com base na quantidade de IOPS provisionadas por volume e no tamanho das operações de E/S que estão sendo executadas. Recomendamos usar o armazenamento io2 Block Express quando ele estiver disponível.

A tabela a seguir mostra o intervalo de IOPS provisionadas e o throughput máximo de cada mecanismo de banco de dados e o intervalo de tamanho de armazenamento.

Mecanismo do banco de dados	Intervalo do tamanho do armazenamento	Intervalo de IOPS provisionadas	Throughput máximo
Db2, MariaDB, MySQL e PostgreSQL	De 100 GiB a 399 GiB	De 1.000 a 19.950 IOPS	500 MiB/s
Db2, MariaDB, MySQL e PostgreSQL	De 400 GiB a 65.536 GiB	De 1.000 a 256.000 IOPS	4.000 MiB/s
Oracle	De 100 GiB a 199 GiB	De 1.000 a 9.950 IOPS	500 MiB/s
Oracle	De 200 GiB a 65.536 GiB	De 1.000 a 256.000 IOPS[1]	4.000 MiB/s
SQL Server	De 20 GiB a 16.384 GiB	De 1.000 a 64.000 IOPS[2]	1.000 MiB/s

> **Note**
>
> [1] Para o Oracle, é possível provisionar o máximo de 256.000 IOPS somente no tipo de instância r5b.
> [2] Para o SQL Server, o máximo de 64.000 IOPS é garantido apenas em instâncias baseadas no Nitro dos tipos de instância m5*, m6i, r5*, r6i e z1d. Outros tipos de instâncias garantem performance de até 32.000 IOPS.

As faixas de IOPS e de tamanho de armazenamento têm as seguintes restrições:

- A proporção de IOPS com relação ao armazenamento alocado (em GiB) deve ser de 1 a 50 no RDS for SQL Server e de 0,5 a 50 em outros mecanismos de banco de dados RDS.
- Se você estiver usando autoescalabilidade de armazenamento, as mesmas taxas entre IOPS e limite máximo de armazenamento (em GiB) também se aplicarão.

Para obter mais informações sobre autoescalabilidade automática de armazenamento, consulte Gerenciar a capacidade automaticamente com a escalabilidade automática de armazenamento do Amazon RDS.

Combinação de armazenamento de IOPS provisionadas com implantações multi-AZ ou réplicas de leitura

Para casos de uso de OLTP de produção, recomendamos o uso de implantações multi-AZ para tolerância a falhas aprimorada com armazenamento de IOPS provisionadas para performance rápida e previsível.

Também é possível usar o armazenamento de IOPS provisionadas com réplicas de leitura para MySQL, MariaDB ou PostgreSQL. O tipo de armazenamento para uma réplica de leitura é independente daquele da instância de banco de dados primária. Por exemplo, você pode usar o SSD de uso geral para réplicas de leitura com uma instância do banco de dados primária que usa armazenamento SSD de IOPS provisionadas para reduzir custos. No entanto, a performance da réplica de leitura nesse caso pode ser diferente daquela de uma configuração em que a instância de banco de dados primária e as réplicas de leitura usam o armazenamento de IOPS provisionadas.

Custos do armazenamento de IOPS provisionadas

Com o armazenamento de IOPS provisionadas, você é cobrado pelos recursos provisionados se usá-los ou não durante um certo mês.

Para obter mais informações sobre a definição de preço, consulte Definição de preço do Amazon RDS.

Com obter a melhor performance do armazenamento de IOPS provisionadas do Amazon RDS

Se a sua workload for limitada a E/S, usar o armazenamento de IOPS provisionadas pode aumentar o número de solicitações de E/S que o sistema é capaz de processar simultaneamente. O aumento da simultaneidade permite a diminuição da latência pois as solicitações de E/S passam menos tempo em uma fila. A latência menor permite compromissos de banco de dados mais rápidos, o que melhora o tempo de resposta e permite uma maior taxa de transferência de banco de dados.

O armazenamento de IOPS provisionadas permite reservar a capacidade de E/S com a especificação de IOPS. No entanto, como qualquer outro atributo de capacidade do sistema, a taxa

de transferência máxima sob carga será restringida pelo recurso que for consumido primeiro. Esse recurso pode ser largura de banda da rede, CPU, memória ou recursos internos do banco de dados.

Armazenamento SSD de uso geral

O armazenamento de uso geral oferece armazenamento econômico e aceitável para a maioria das workloads de banco de dados não sensíveis à latência nem à performance.

> **Note**
>
> As instâncias de banco de dados que usam o armazenamento de uso geral podem experimentar uma latência muito mais longa que as instâncias que usam o armazenamento de IOPS provisionadas. Se você precisar de uma instância de banco de dados com latência mínima após essas operações, recomendamos o uso de Armazenamento SSD de IOPS provisionadas.

O Amazon RDS oferece dois tipos de armazenamento de uso geral: Armazenamento gp3 (recomendado) e Armazenamento gp2 (geração anterior).

Armazenamento gp3 (recomendado)

Ao usar volumes de armazenamento gp3 de uso geral, é possível personalizar a performance do armazenamento independentemente da capacidade de armazenamento. A performance de armazenamento é a combinação de operações de E/S por segundo (IOPS) e a rapidez com que o volume pode realizar leituras e gravações (throughput de armazenamento). Em volumes de armazenamento gp3, o Amazon RDS fornece uma performance de armazenamento de referência de 3.000 IOPS e 125 MiB/s.

Para cada mecanismo de banco de dados do RDS, exceto o RDS para SQL Server, quando o tamanho do armazenamento de volumes gp3 atinge determinado limite, a performance do armazenamento de referência aumenta. Isso ocorre por conta da distribuição de volumes, em que o armazenamento usa quatro volumes em vez de um. O RDS para SQL Server não é compatível com a distribuição de volume e, portanto, não tem um valor limite. Em volumes distribuídos, o Amazon RDS oferece uma performance de armazenamento de referência de 12.000 IOPS e 500 MiB/s.

A performance do armazenamento para volumes gp3 nos mecanismos de banco de dados do Amazon RDS, incluindo o limite, é mostrada na tabela a seguir.

Mecanismo de banco de dados	Tamanho de armazenamento	Performance de armazenamento de referência	Intervalo de IOPS provisionadas	Variedade do throughput de armazenamento provisionado
Db2, MariaDB, MySQL e PostgreSQL	De 20 GiB a 399 GiB	3.000 IOPS/125 MiB/s	N/D	N/D
Db2, MariaDB, MySQL e PostgreSQL	De 400 GiB a 65.536 GiB	12.000 IOPS/500 MiB/s	De 12.000 a 64.000 IOPS	De 500 a 4.000 MiB/s
Oracle	De 20 GiB a 199 GiB	3.000 IOPS/125 MiB/s	N/D	N/D
Oracle	De 200 GiB a 65.536 GiB	12.000 IOPS/500 MiB/s	De 12.000 a 64.000 IOPS	De 500 a 4.000 MiB/s
SQL Server	De 20 GiB a 16.384 GiB	3.000 IOPS/125 MiB/s	De 3.000 a 16.000 IOPS	De 125 a 1.000 MiB/s

Para cada mecanismo de banco de dados, exceto o RDS para SQL Server, você pode provisionar IOPS e throughput de armazenamento adicionais quando o tamanho do armazenamento estiver igual ou superior ao valor limite. Para o RDS para SQL Server, você pode provisionar IOPS e throughput de armazenamento adicionais para qualquer tamanho de armazenamento disponível. Para todos os mecanismos de banco de dados, você paga apenas pela performance adicional do armazenamento provisionado. Para obter mais informações, consulte Definição de preço do Amazon RDS.

Embora as IOPS provisionadas e o throughput de armazenamento adicionados não dependam do tamanho do armazenamento, eles estão relacionados entre si. Quando você aumenta as IOPS acima de 32.000 para MariaDB e MySQL, o valor do throughput de armazenamento aumenta automaticamente de 500 MiBps. Por exemplo, quando você define as IOPS como 40.000 no RDS para MySQL, o throughput de armazenamento deve ser de pelo menos 625 MiBps. O aumento automático não ocorre para instâncias de banco de dados do Db2, do Oracle, do PostgreSQL e do SQL Server.

Para clusters de banco de dados multi-AZ, o Amazon RDS define automaticamente o valor de throughput com base nas IOPS que você provisiona. Não é possível modificar o valor de throughput.

Os valores de performance de armazenamento para volumes gp3 no RDS têm as seguintes restrições:

- A proporção máxima do throughput de armazenamento em relação ao IOPS é de 0,25 para todos os mecanismos de banco de dados compatíveis.
- A proporção mínima de IOPS para o armazenamento alocado (em GiB) é de 0,5 no RDS para SQL Server. Não há proporção mínima para os outros mecanismos de banco de dados compatíveis.
- A proporção máxima de IOPS em relação ao armazenamento alocado é de 500 para todos os mecanismos de banco de dados compatíveis.
- Se você estiver usando autoescalabilidade de armazenamento, as mesmas taxas entre IOPS e limite máximo de armazenamento (em GiB) também se aplicarão.

  Para obter mais informações sobre autoescalabilidade automática de armazenamento, consulte Gerenciar a capacidade automaticamente com a escalabilidade automática de armazenamento do Amazon RDS.

Armazenamento gp2 (geração anterior)

Quando suas aplicações não precisam de alto desempenho de armazenamento, você pode usar o armazenamento SSD gp2 de uso geral. A performance de E/S de referência para o armazenamento gp2 é de 3 IOPS para cada GiB, com no mínimo 100 IOPS. Esse relacionamento significa que volumes maiores têm uma performance melhor. Por exemplo, a performance basal para um volume de 100 GiB é 300 IOPS. A performance basal para um volume de 1.000 GiB é de 3.000 IOPS.

Os volumes gp2 individuais abaixo de 1.000 GiB de tamanho também têm a capacidade de intermitência de até 3.000 IOPS durante períodos prolongados. O saldo de créditos E/S de volume determina a performance de intermitência. Para obter uma descrição mais detalhada de como a performance de referência e o saldo de créditos de E/S afetam a performance, consulte a publicação Understanding burst vs. baseline performance with Amazon RDS and gp2 (Conceitos básicos da intermitência com relação à performance de referência com o Amazon RDS e gp2) no Blog do banco de dados da AWS.

Muitas workloads nunca esgotam o saldo de intermitência. No entanto, algumas workloads podem esgotar o saldo de créditos de armazenamento de intermitência de 3.000 IOPS, portanto você deve planejar sua capacidade de armazenamento para atender às necessidades das workloads.

Para volumes gp2 com mais de 4.000 GiB, a performance de referência é maior que a performance de expansão. Para esses volumes, a intermitência é irrelevante porque a performance da linha de base é melhor que a performance de intermitência de 3.000 IOPS. No entanto, para instâncias de banco de dados de determinados mecanismos e tamanhos, o armazenamento é distribuído em quatro volumes, fornecendo quatro vezes o throughput básico e quatro vezes as IOPS de intermitência de um único volume.

A performance do armazenamento para volumes gp2 de vários tamanhos de armazenamento nos mecanismos de banco de dados do Amazon RDS é mostrada na tabela a seguir.

Mecanismo de banco de dados	Tamanho de armazenamento do RDS	Intervalo de IOPS de referência	Intervalo de throughput de referência	IOPS de intermitência
MariaDB, MySQL e PostgreSQL	De 5 GiB a 399 GiB[1]	De 100 a 1.197 IOPS	De 128 a 250 MiB/s	3.000
MariaDB, MySQL e PostgreSQL	De 400 GiB a 1.335 GiB	De 1.200 a 4.005 IOPS	De 512 a 1 mil MiB/s	12.000
MariaDB, MySQL e PostgreSQL	De 1.336 GiB a 3.999 GiB	De 4.008 a 11.997 IOPS	1.000 MiB/s	12.000
MariaDB, MySQL e PostgreSQL	De 4.000 GiB a 65.536 GiB	De 12.000 a 64.000 IOPS	1.000 MiB/s	N/D[2]
Oracle	De 20 GiB a 199 GiB	De 100 a 597 IOPS	De 128 a 250 MiB/s	3.000
Oracle	De 200 GiB a 1.335 GiB	De 600 a 4.005 IOPS	De 500 a 1.000 MiB/s	12.000
Oracle	De 1.336 GiB a 3.999 GiB	De 4.008 a 11.997 IOPS	1.000 MiB/s	12.000

Mecanismo de banco de dados	Tamanho de armazenamento do RDS	Intervalo de IOPS de referência	Intervalo de throughput de referência	IOPS de intermitência
Oracle	De 4.000 GiB a 65.536 GiB	De 12.000 a 64.000 IOPS	1.000 MiB/s	N/D[2]
SQL Server	De 20 GiB a 333 GiB	De 100 a 999 IOPS	De 128 a 250 MiB/s	3.000
SQL Server	De 334 GiB a 999 GiB	De 1.002 a 2.997 IOPS	250 MiB/s	3.000
SQL Server	De 1.000 GiB a 16.384 GiB	De 3.000 a 16.000 IOPS	250 MiB/s	N/D[2]

> **Note**
>
> [1] Usando o AWS Management Console, é possível criar instâncias de banco de dados com um tamanho mínimo de armazenamento de 5 GiB no nível gratuito para as classes de instância de banco de dados db.t3.micro e db.t4g.micro. Caso contrário, o tamanho mínimo de armazenamento é de 20 GiB. Essa limitação não se aplica à AWS CLI e à API do RDS.
> [2] O desempenho basal do volume excede o desempenho de expansão máxima.

Comparar os tipos de armazenamento em unidades de estado sólido (SSD)

A tabela a seguir mostra casos de uso e características de performance para os volumes de armazenamento SSD usados pelo Amazon RDS.

Característica	IOPS provision adas (io2 Block Express)	IOPS provision adas (io1)	Uso geral (gp3)	Uso geral (gp2)
Descrição	O maior desempenho no portfólio de	Performance consistente de armazenam	Flexibilidade no provisionamento de armazenam	Oferece IOPS de intermitência

Característica	IOPS provisionadas (io2 Block Express)	IOPS provisionadas (io1)	Uso geral (gp3)	Uso geral (gp2)
	armazenamento do RDS (IOPS, throughput, latência) Projetado para workloads transacionais sensíveis à latência	ento (IOPS, throughput, latência) Projetado para workloads transacionais sensíveis à latência	ento, IOPS e throughput de forma independente Equilibra a performance de preço para uma ampla variedade de workloads transacionais.	Equilibra a performance de preço para uma ampla variedade de workloads transacionais.
Casos de uso	Workloads transacionais essenciais para os negócios que exigem latência de menos de um milissegundo e desempenho de IOPS sustentado de até 256.000 IOPS	Workloads transacionais que exigem desempenho de IOPS sustentado de até 256.000 IOPS	Ampla variedade de workloads executadas em bancos de dados relacionais de médio porte em ambientes de desenvolvimento/teste	Ampla variedade de workloads executadas em bancos de dados relacionais de médio porte em ambientes de desenvolvimento/teste
Latência	Submilissegundo, fornecido consistentemente 99,9% do tempo	Milissegundo de um dígito, fornecido consistentemente 99,9% do tempo	Milissegundo de um dígito, fornecido consistentemente 99% do tempo	Milissegundo de um dígito, fornecido consistentemente 99% do tempo

Característica	IOPS provisionadas (io2 Block Express)	IOPS provisionadas (io1)	Uso geral (gp3)	Uso geral (gp2)
Tamanho do volume	De 100 GiB a 65.536 GiB	De 100 GiB a 65.536 GiB (De 20 GiB a 16.384 GiB no RDS para SQL Server)	De 20 GiB a 65.536 GiB (16.384 GiB no RDS para SQL Server)	De 20 GiB a 65.536 GiB (16.384 GiB no RDS para SQL Server)
Máximo de IOPS	256.000	256.000 (64.000 no RDS para SQL Server)	64.000 (16.000 no RDS para SQL Server)	64.000 (16.000 no RDS para SQL Server) **Note** Você não pode provisionar IOPS diretamente no armazenamento gp2. O IOPS varia com o tamanho de armazenamento alocado.

Característica	IOPS provisionadas (io2 Block Express)	IOPS provisionadas (io1)	Uso geral (gp3)	Uso geral (gp2)
Throughput máximo	Escalas com base em IOPS provisionadas de até 4.000 MB/s A throughput é dimensionada proporcionalmente até 0,256 MiB/s por IOPS provisionadas. O throughput máximo de 4.000 MiB/s pode ser alcançada em 256.000 IOPS com um tamanho de E/S de 16 KiB e 16.000 IOPS ou superior com um tamanho de E/S de 256 KiB. Para instâncias não baseadas no AWS Nitro System, o throughput máximo de 2.000 MiB/s pode ser alcançado	Escalas com base em IOPS provisionadas de até 4.000 MB/s	Fornecer throughput adicional de até 4 mil MB/s (1 mil MB/s no RDS para SQL Server)	1.000 MB/s (250 MB/s no RDS para SQL Server)

Característica	IOPS provisionadas (io2 Block Express)	IOPS provisionadas (io1)	Uso geral (gp3)	Uso geral (gp2)
	em 128.000 IOPS com um tamanho de E/S de 16 KiB.			
AWS CLI e nome da API do RDS	io2	io1	gp3	gp2

Armazenamento magnético (herdado, não recomendado)

O Amazon RDS também suporta armazenamento magnético para retrocompatibilidade. Recomendamos que você use SSD de uso geral ou SSD de IOPS provisionadas para toda e qualquer nova necessidade de armazenamento. A seguir estão algumas limitações para o armazenamento magnético:

- Não permite escalar o armazenamento ao usar o mecanismo de banco de dados do SQL Server.
- Não permite converter em um tipo de armazenamento diferente ao usar o mecanismo de banco de dados do SQL Server.
- Não é compatível com a escalabilidade automática de armazenamento.
- Não é compatível com os volumes elásticos.
- Limitado a um tamanho máximo de 3 TiB.
- Limitado a um máximo de 1.000 IOPS.

Volume de log dedicado (DLV)

Use um volume de log dedicado (DLV) para uma instância de banco de dados que usa o armazenamento de IOPS provisionadas (PIOPS) utilizando o console do Amazon RDS, AWS CLI a ou a API do Amazon RDS. Um DLV move os logs de transações do banco de dados do PostgreSQL e logs binários e redo logs do MySQL/MariaDB para um volume de armazenamento separado do volume que contém as tabelas do banco de dados. Um DLV torna o registro de gravação de transações mais eficiente e consistente. Os DLVs são ideais para bancos de dados com grande

armazenamento alocado, altos requisitos de I/O por segundo (IOPS) ou cargas de trabalho sensíveis à latência.

Os DLVs são compatíveis com armazenamento PIOPS (io1 e io2 Block Express) e são criados com um tamanho fixo de 1.000 GiB e 3.000 IOPS provisionadas.

> **Note**
>
> Os DLVs não comportam o Armazenamento de uso geral (gp2 e gp3).

O Amazon RDS é compatível com DLVs em todas as Regiões da AWS para as seguintes versões:

- 10.6.7 e versões 10.6 posteriores
- MySQL 8.0.28 e versões 8.0 posteriores
- PostgreSQL 13.10 e versões 13 posteriores, 14.7 e versões 14 posteriores, 15.2 e versões 15 posteriores e 16.1 e versões 16 posteriores

O Amazon RDS on é compatível com implantações multi-AZ. Quando você modifica ou cria uma instância Multi-AZ, um DLV é criado tanto para a primária quanto para a secundária.

O RDS oferece suporte a DLVs com réplicas de leitura. Se a instância de banco de dados primária tiver um DLV ativado, todas as réplicas de leitura criadas após a ativação do DLV também terão um DLV. Qualquer réplica de leitura criada antes da mudança para o DLV não a terá habilitada, a menos que seja explicitamente modificada para isso. Recomendamos que todas as réplicas de leitura anexadas a uma instância primária antes da ativação do DLV também sejam modificadas manualmente para terem um DLV.

Depois de modificar a configuração de DLV para uma instância de banco de dados, a instância de banco de dados deve ser reinicializada.

Para ter informações sobre como habilitar o DLV, consulte Usando um volume de registro dedicado (DLV).

Monitoramento da performance de armazenamento

O Amazon RDS fornece várias métricas que você pode usar para avaliar a performance da sua instância de banco de dados. Você pode visualizar as métricas na página de resumo da sua

instância no Console de gerenciamento do Amazon RDS. Você também pode usar o Amazon CloudWatch para monitorar essas métricas. Para obter mais informações, consulte Visualizar métricas no console do Amazon RDS. O Monitoramento aprimorado fornece métricas de E/S mais detalhadas; para obter mais informações, consulte Monitorar métricas do SO com o monitoramento avançado.

As métricas a seguir são úteis para monitorar o armazenamento da sua instância de banco de dados:

- IOPS – o número de operações de E/S concluídas por segundo. Essa métrica é relatada como as IOPS médias para um determinado intervalo de tempo. O Amazon RDS relata IOPS de leitura e de gravação separadamente em intervalos de um minuto. A IOPS total é a soma da IOPS de leitura e gravação. Os valores típicos da IOPS variam de zero a dezenas de milhares por segundo.
- Latência – o tempo decorrido entre o envio de uma solicitação de E/S e a sua conclusão. Essa métrica é relatada como a latência média para um determinado intervalo de tempo. O Amazon RDS relata a latência de leitura e gravação separadamente em intervalos de um minuto. Os valores típicos de latência estão em milissegundos (ms).
- Taxa de transferência – o número de bytes por segundo transferidos para ou a partir do disco. Essa métrica é relatada como a taxa de transferência média para um determinado intervalo de tempo. O Amazon RDS relata o throughput de leitura e de gravação separadamente em intervalos de um minuto usando unidades de bytes por segundo (B/s). Os valores típicos da taxa de transferência variam de zero à largura de banda máxima do canal de E/S.
- Comprimento da fila – o número de solicitações de E/S na fila aguardando por ser atendidas. Essas são solicitações de E/S que foram enviadas pelo aplicativo, mas não foram enviadas ao dispositivo porque o mesmo está ocupado atendendo a outras solicitações de E/S. O tempo gasto esperando na fila é um componente de latência e tempo de serviço (não disponível como uma métrica). Essa métrica é relatada como o comprimento da fila médio para um determinado intervalo de tempo. O Amazon RDS relata o comprimento da fila em intervalos de um minuto. Os valores típicos de comprimento da fila variam de zero a várias centenas.

Os valores de IOPS medidos são independentes do tamanho da operação de E/S individual. Isso significa que ao medir a performance de E/S, observe o throughput da instância, não apenas o número de operações de E/S.

Fatores que afetam a performance de armazenamento

As atividades do sistema, a workload do banco de dados e a classe da instância de banco de dados podem afetar a performance de armazenamento.

Atividades do sistema

As seguintes atividades relacionadas ao sistema consomem capacidade de E/S e podem reduzir a performance da instância de banco de dados em andamento:

- Criação da espera multi-AZ
- Criação de réplica de leitura
- Alteração dos tipos de armazenamento

workload do banco de dados

Em alguns casos o design do banco de dados ou da aplicação resulta em problemas de simultaneidade, travamento ou outras formas de contenção do banco de dados. Nesses casos, talvez não seja possível usar toda a largura de banda provisionada diretamente. Além disso, você pode encontrar as seguintes situações relacionadas à workload:

- O limite da taxa de transferência do tipo de instância subjacente é acessado.
- A profundidade da fila é consistentemente menor que 1, pois a aplicação não está conduzindo operações de E/S suficientes.
- Você vivencia contenção de consulta no banco de dados mesmo que haja capacidade de E/S não usada.

Em alguns casos, não há um recurso de sistema que esteja dentro ou próximo de um limite, e a adição de threads não aumenta a taxa de transação do banco de dados. Nesses casos, o gargalo é mais provável a contenção no banco de dados. As formas mais comuns são bloqueio de linha e contenção de bloqueio de página de índice, mas há muitas outras possibilidades. Se essa for a sua situação, procure a orientação de um especialista em ajuste de performance de banco de dados.

Classe de instância de banco de dados

Para obter a máxima performance da instância de banco de dados Amazon RDS, selecione um tipo de instância da geração atual com largura de banda suficiente para oferecer suporte ao seu tipo de armazenamento. Por exemplo, você pode escolher instâncias otimizadas para Amazon EBS e instâncias com conectividade de rede de 10 gigabit.

> **Important**
>
> Dependendo da classe de instância que você está usando, poderá observar uma performance de IOPS menor do que o máximo que o RDS permite provisionar. Para obter informações específicas sobre a performance de IOPS para classes de instância de banco de dados, consulte Instâncias otimizadas para Amazon EBS no Guia do usuário do Amazon EC2. Recomendamos que você determine as IOPS máximas para a classe de instância antes de definir um valor de IOPS provisionadas para sua instância de banco de dados.

Recomendamos usar a geração mais recente das instâncias para obter a melhor performance. As instâncias de banco de dados de geração anteriores também podem ter um armazenamento máximo menor.

Alguns sistemas de arquivos mais antigos de 32 bits podem ter capacidades de armazenamento mais baixas. Para determinar a capacidade de armazenamento da sua instância de banco de dados, use o comando describe-valid-db-instance-modifications da AWS CLI.

A tabela a seguir mostra o armazenamento máximo para o qual cada classe de instância de banco de dados pode ser expandida em cada mecanismo de banco de dados.

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

A tabela a seguir mostra algumas exceções para o armazenamento máximo (em TiB). Todas as instâncias de banco de dados do RDS para Microsoft SQL Server, exceto o armazenamento io2 Block Express, têm um armazenamento máximo de 16 TiB. Portanto, não há entradas para o SQL Server.

Classe de instância	Db2	MariaDB	MySQL	Oracle	PostgreSQL
db.m3: classes de instância padrão					
db.t4g: classes de instância expansível					
db.t4g.medium	N/D	16	16	N/D	32
db.t4g.small	N/D	16	16	N/D	16
db.t4g.micro	N/D	6	6	N/D	6
db.t3: classes de instância expansível					
db.t3.medium	32	16	16	32	32
db.t3.small	32	16	16	32	16
db.t3.micro	N/D	6	6	32	6
db.t2: classes de instância expansível					

Para obter mais detalhes sobre todas as classes de instâncias compatíveis, consulte Instâncias de banco de dados de gerações anteriores.

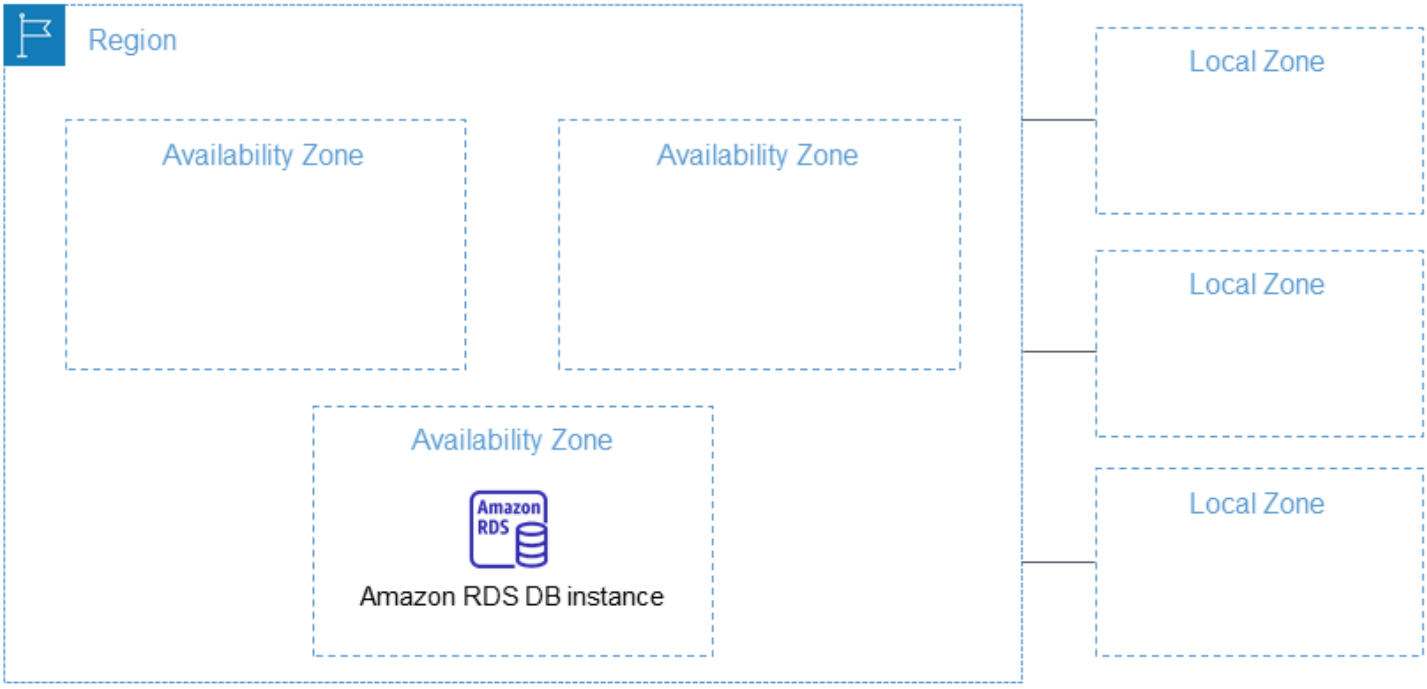
Regiões, zonas de disponibilidade e Local Zones

Os recursos de computação em nuvem da Amazon são hospedados em vários locais no mundo todo. Esses locais são compostos por regiões da AWS, zonas de disponibilidade e zonas locais. Cada região da AWS é uma área geográfica separada. Cada região da AWS contém vários locais isolados conhecidos como zonas de disponibilidade.

> **Note**
>
> Para obter informações sobre como localizar as zonas de disponibilidade para uma região da AWS, consulte Como descrever zonas de disponibilidade na documentação do Amazon EC2.

Usando o Local Zones, é possível colocar recursos, como computação e armazenamento, em vários locais mais próximos dos usuários. O Amazon RDS permite que você coloque recursos, como instâncias de banco de dados, e dados em vários locais. Os recursos não são replicados entre regiões da AWS, a menos que você especifique isso.

A Amazon opera datacenters de última geração e altamente disponíveis. Embora sejam raras, podem ocorrer falhas que afetam a disponibilidade das instâncias de banco de dados que estiverem no mesmo local. Se você hospeda todas as instâncias de banco de dados em um único local afetado por essa falha, nenhuma delas estará disponível.



É importante lembrar que cada região AWS é totalmente independente. Qualquer atividade do Amazon RDS que você iniciar (por exemplo, criar instâncias de banco de dados ou listar instâncias de banco de dados disponíveis) ocorre apenas em sua região da AWS padrão atual. A região da AWS padrão pode ser alterada no console ou definindo a variável de ambiente `AWS_DEFAULT_REGION`. Ou ele pode ser substituído usando o parâmetro `--region` pela AWS Command Line Interface (AWS CLI). Para obter mais informações, consulte Configurar a AWS Command Line Interface, especificamente as seções sobre variáveis de ambiente e opções de linha de comando.

O Amazon RDS é compatível com regiões especiais da AWSchamadas AWS GovCloud (US). Elas são projetadas para permitir que clientes e agências governamentais dos EUA transfiram workloads mais confidenciais para a nuvem. A região AWS GovCloud (US) atende aos seus requisitos específicos de regulamentação e conformidade do governo dos EUA. Para obter mais informações, consulte O que é o AWS GovCloud (US)?

Para criar ou trabalhar com uma instância de bancos de dados do Amazon RDS em uma região da AWS específica, use o endpoint de serviço regional correspondente.

AWSRegiões de

Cada região da AWS é projetada para ser isolada das outras regiões da AWS. Esse design proporciona o máximo de tolerância a falhas e estabilidade possível.

Ao visualizar os recursos, você vê apenas os recursos que estão vinculados à região da AWS especificada. Isso ocorre porque as regiões da AWS são isoladas entre si e nós não replicamos os recursos entre regiões da AWS automaticamente.

Disponibilidade de regiões

A tabela a seguir mostra as regiões da AWS em que o Amazon RDS está disponível no momento e o endpoint para cada região.

Nome da região	Região	Endpoint	Protocolo	
Leste dos EUA (Ohio)	us-east-2	rds.us-east-2.amazonaws.com	HTTPS	
		rds-fips.us-east-2.api.aws	HTTPS	
		rds.us-east-2.api.aws	HTTPS	

Nome da região	Região	Endpoint	Protocolo
		rds-fips.us-east-2.amazonaws.com	HTTPS
Leste dos EUA (Norte da Virgínia)	us-east-1	rds.us-east-1.amazonaws.com	HTTPS
		rds-fips.us-east-1.api.aws	HTTPS
		rds-fips.us-east-1.amazonaws.com	HTTPS
		rds.us-east-1.api.aws	HTTPS
Oeste dos EUA (N. da Califórnia)	us-west-1	rds.us-west-1.amazonaws.com	HTTPS
		rds.us-west-1.api.aws	HTTPS
		rds-fips.us-west-1.amazonaws.com	HTTPS
		rds-fips.us-west-1.api.aws	HTTPS
Oeste dos EUA (Oregon)	us-west-2	rds.us-west-2.amazonaws.com	HTTPS
		rds-fips.us-west-2.amazonaws.com	HTTPS
		rds.us-west-2.api.aws	HTTPS
		rds-fips.us-west-2.api.aws	HTTPS
África (Cidade do Cabo)	af-south-1	rds.af-south-1.amazonaws.com	HTTPS
		rds.af-south-1.api.aws	HTTPS
Ásia-Pacífico (Hong Kong)	ap-east-1	rds.ap-east-1.amazonaws.com	HTTPS
		rds.ap-east-1.api.aws	HTTPS

Nome da região	Região	Endpoint	Protocolo
Ásia-Pacífico (Hyderabad)	ap-south-2	rds.ap-south-2.amazonaws.com rds.ap-south-2.api.aws	HTTPS HTTPS
Ásia-Pacífico (Jacarta)	ap-southeast-3	rds.ap-southeast-3.amazonaws.com rds.ap-southeast-3.api.aws	HTTPS HTTPS
Ásia-Pacífico (Melbourne)	ap-southeast-4	rds.ap-southeast-4.amazonaws.com rds.ap-southeast-4.api.aws	HTTPS HTTPS
Ásia-Pacífico (Mumbai)	ap-south-1	rds.ap-south-1.amazonaws.com rds.ap-south-1.api.aws	HTTPS HTTPS
Ásia-Pacífico (Osaka)	ap-northeast-3	rds.ap-northeast-3.amazonaws.com rds.ap-northeast-3.api.aws	HTTPS HTTPS
Ásia-Pacífico (Seul)	ap-northeast-2	rds.ap-northeast-2.amazonaws.com rds.ap-northeast-2.api.aws	HTTPS HTTPS
Ásia-Pacífico (Singapura)	ap-southeast-1	rds.ap-southeast-1.amazonaws.com rds.ap-southeast-1.api.aws	HTTPS HTTPS
Ásia-Pacífico (Sydney)	ap-southeast-2	rds.ap-southeast-2.amazonaws.com rds.ap-southeast-2.api.aws	HTTPS HTTPS

Nome da região	Região	Endpoint	Protocolo
Ásia-Pacífico (Tóquio)	ap-northeast-1	rds.ap-northeast-1.amazonaws.com	HTTPS
		rds.ap-northeast-1.api.aws	HTTPS
Canadá (Central)	ca-central-1	rds.ca-central-1.amazonaws.com	HTTPS
		rds.ca-central-1.api.aws	HTTPS
		rds-fips.ca-central-1.api.aws	HTTPS
		rds-fips.ca-central-1.amazonaws.com	HTTPS
Oeste do Canadá (Calgary)	ca-west-1	rds.ca-west-1.amazonaws.com	HTTPS
		rds-fips.ca-west-1.amazonaws.com	HTTPS
Europa (Frankfurt)	eu-central-1	rds.eu-central-1.amazonaws.com	HTTPS
		rds.eu-central-1.api.aws	HTTPS
Europa (Irlanda)	eu-west-1	rds.eu-west-1.amazonaws.com	HTTPS
		rds.eu-west-1.api.aws	HTTPS
Europa (Londres)	eu-west-2	rds.eu-west-2.amazonaws.com	HTTPS
		rds.eu-west-2.api.aws	HTTPS
Europa (Milão)	eu-south-1	rds.eu-south-1.amazonaws.com	HTTPS
		rds.eu-south-1.api.aws	HTTPS
Europa (Paris)	eu-west-3	rds.eu-west-3.amazonaws.com	HTTPS
		rds.eu-west-3.api.aws	HTTPS

Nome da região	Região	Endpoint	Protocolo
Europa (Espanha)	eu-south-2	rds.eu-south-2.amazonaws.com	HTTPS
		rds.eu-south-2.api.aws	HTTPS
Europa (Estocolmo)	eu-north-1	rds.eu-north-1.amazonaws.com	HTTPS
		rds.eu-north-1.api.aws	HTTPS
Europa (Zurique)	eu-central-2	rds.eu-central-2.amazonaws.com	HTTPS
		rds.eu-central-2.api.aws	HTTPS
Israel (Tel Aviv)	il-central-1	rds.il-central-1.amazonaws.com	HTTPS
		rds.il-central-1.api.aws	HTTPS
Oriente Médio (Barém)	me-south-1	rds.me-south-1.amazonaws.com	HTTPS
		rds.me-south-1.api.aws	HTTPS
Oriente Médio (Emirados Árabes Unidos)	me-central-1	rds.me-central-1.amazonaws.com	HTTPS
		rds.me-central-1.api.aws	HTTPS
América do Sul (São Paulo)	sa-east-1	rds.sa-east-1.amazonaws.com	HTTPS
		rds.sa-east-1.api.aws	HTTPS
AWS GovCloud (Leste dos EUA)	us-gov-east-1	rds.us-gov-east-1.amazonaws.com	HTTPS
		rds.us-gov-east-1.api.aws	HTTPS

Nome da região	Região	Endpoint	Protocolo	
AWS GovCloud (Oeste dos EUA)	us-gov-west-1	rds.us-gov-west-1.amazonaws.com rds.us-gov-west-1.api.aws	HTTPS HTTPS	

Se você não especificar explicitamente um endpoint, o endpoint Oeste dos EUA (Oregon) será considerado o padrão.

Quando você trabalha com uma instância de banco de dados usando a AWS CLI ou operações de API, certifique-se de especificar o endpoint regional.

Zonas de disponibilidade

Ao criar uma instância de banco de dados, é possível escolher uma zona de disponibilidade ou fazer com que o Amazon RDS escolha uma para você. Uma zona de disponibilidade é representada por um código de região da AWS seguido por um identificador de letra (por exemplo, `us-east-1a`).

Use o comando describe-availability-zones do Amazon EC2, como se segue, para descrever as zonas de disponibilidade da região especificada que estão disponíveis para sua conta.

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

Por exemplo, para descrever as zonas de disponibilidade da região Leste dos EUA (N. da Virgínia) (us-east-1) ativado para sua conta, execute o seguinte comando:

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

Você não pode escolher as zonas de disponibilidade para as instâncias de banco de dados primária e secundária em uma implantação de banco de dados multi-AZ. O Amazon RDS as escolhe para você aleatoriamente. Para obter mais informações sobre implantações Multi-AZ, consulte Configurar e gerenciar uma implantação multi-AZ para o Amazon RDS.

> **Note**
>
> A seleção aleatória de zonas de disponibilidade pelo RDS não garante uma distribuição uniforme de instâncias de banco de dados entre zonas de disponibilidade em uma única conta ou grupo de sub-rede de banco de dados. Você pode solicitar um AZ específico ao criar ou modificar uma instância Single-AZ e pode usar grupos de sub-redes de banco de dados mais específicos para instâncias Multi-AZ. Para ter mais informações, consulte Criar uma instância de banco de dados do Amazon RDS e Modificar uma instância de banco de dados do Amazon RDS.

Zonas Locais

Uma zona local é uma extensão de uma região AWS que está geograficamente próxima de seus usuários. É possível estender qualquer VPC da região principal da AWS para as zonas locais. Para isso, crie uma sub-rede e atribua-a à zona local da AWS. Quando você criar uma sub-rede em uma zona local, sua VPC também será estendida para essa zona local. A sub-rede na zona local funciona da mesma forma que outras sub-redes na VPC.

Quando criar uma instância de banco de dados, você poderá escolher uma sub-rede em uma zona local. As zonas locais têm suas próprias conexões com a Internet e suporte no AWS Direct Connect. Assim, os recursos criados em uma zona local podem atender usuários locais com comunicações de latência muito baixa. Para obter mais informações, consulte Zonas locais da AWS.

Uma zona local é representada por um código de região da AWS seguido por um identificador que indica o local, por exemplo, `us-west-2-lax-1a`.

> **Note**
>
> Uma zona local não pode ser incluída em uma implantação Multi-AZ.

Como usar uma zona local

1. Habilite a zona local no console do Amazon EC2.

   Para obter mais informações, consulte Habilitação de zonas locais no Guia do usuário do Amazon EC2.
2. Crie uma sub-rede na zona local.

Para obter mais informações, consulte Criar uma sub-rede na VPC no Guia do usuário da Amazon VPC.

3. Crie um grupo de sub-redes de banco de dados na zona local.

   Ao criar um grupo de sub-redes de banco de dados, escolha o grupo de zonas de disponibilidade para a zona local.

   Para obter mais informações, consulte Criar uma instância de banco de dados em uma VPC.

4. Crie uma instância de banco de dados que use o grupo de sub-redes de banco de dados na zona local.

   Para obter mais informações, consulte Criar uma instância de banco de dados do Amazon RDS.

> **Important**
>
> Atualmente, a única zona local da AWS onde o Amazon RDS está disponível é Los Angeles na região Oeste dos EUA (Oregon).

Recursos compatíveis com o Amazon RDS por Região da AWS e com o mecanismo de banco de dados

O suporte para recursos e opções do Amazon RDS varia entre Regiões da AWS e versões específicas de cada mecanismo de banco de dados. Para identificar o suporte e a disponibilidade da versão do mecanismo de banco de dados do RDS em determinada Região da AWS, você pode usar as seções a seguir.

Os recursos do Amazon RDS são diferentes dos recursos e das opções nativos do mecanismo. Para ter mais informações sobre os recursos e opções nativos do mecanismo, consulte Engine-native features (Recursos nativos do mecanismo).

Regiões e mecanismos de banco de dados compatíveis

- Convenções de tabela
- Referência rápida do recurso
- Regiões e mecanismos de banco de dados compatíveis com implantações azul/verde do Amazon RDS
- Regiões e mecanismos de banco de dados compatíveis com backups automatizados entre regiões no Amazon RDS
- Regiões e mecanismos de banco de dados compatíveis com réplicas de leitura entre regiões no Amazon RDS
- Regiões e mecanismos de banco de dados compatíveis com fluxos de atividades de banco de dados no Amazon RDS
- Regiões e mecanismos de banco de dados compatíveis com o modo de pilha dupla no Amazon RDS
- Regiões e mecanismos de banco de dados compatíveis com a exportação de snapshots para o S3 no Amazon RDS
- Regiões e mecanismos de banco de dados compatíveis com a autenticação de banco de dados do IAM no Amazon RDS
- Regiões e mecanismos de banco de dados compatíveis com a autenticação Kerberos no Amazon RDS
- Regiões e mecanismos de banco de dados compatíveis com clusters de banco de dados multi-AZ no Amazon RDS

- Regiões e mecanismos de banco de dados compatíveis com o Insights de Performance no Amazon RDS
- Regiões e mecanismos de banco de dados compatíveis com o RDS Custom
- Regiões e mecanismos de banco de dados compatíveis com o Amazon RDS Proxy
- Regiões e mecanismos de banco de dados compatíveis com a integração do Secrets Manager com o Amazon RDS
- Regiões e mecanismos de banco de dados que permitem Integrações ETL zero com o Amazon RDS
- Recursos nativos do mecanismo no Amazon RDS

Convenções de tabela

As tabelas nas seções de recursos usam estes padrões para especificar números de versão e nível de disponibilidade:

- Versão x.y: a versão específica por si só está disponível.
- Versão x.y e posteriores: a versão especificada e todas as versões secundárias posteriores da mesma versão principal são compatíveis. Por exemplo, “versão 10.11 e posterior” significa que as versões 10.11, 10.11.1 e 10.12 estão disponíveis.

Referência rápida do recurso

A tabela de referência rápida a seguir indica cada recurso e mecanismo de banco de dados do RDS disponível. A disponibilidade da região e da versão específica é exibida nas seções do recurso posteriores.

Atribut	RDS para Db2	RDS para MariaDB	RDS para MySQL	RDS para Oracle	RDS para PostgreSQL	RDS para SQL Server
Implar ões azul/ verd e	Indisponível	Disponível	Disponível	Indisponível	Disponível	Indisponível

Atribut	RDS para Db2	RDS para MariaDB	RDS para MySQL	RDS para Oracle	RDS para PostgreSQL	RDS para SQL Server
Backu autom ados entre regiõe	Disponível	Disponível	Disponível	Disponível	Disponível	Disponível
Réplic de leitura entre regiõe	Indisponível	Disponível	Disponível	Disponível	Disponível	Disponível
Fluxos de ativida s do banco de dados	Não disponível	Não disponível	Não disponível	Disponível	Indisponível	Disponível
Modo de pilha dupla	Indisponível	Disponível	Disponível	Disponível	Disponível	Disponível
Export o snapsl para o Amazo S3	Indisponível	Disponível	Disponível	Indisponível	Disponível	Indisponível

Atribut	RDS para Db2	RDS para MariaDB	RDS para MySQL	RDS para Oracle	RDS para PostgreSQL	RDS para SQL Server
Autent ção do banco de dados do AWS Identit and Acces Manag t (IAM)	Indisponível	Disponível	Disponível	Indisponível	Disponível	Indisponível
Autent ção de Kerbe	Disponível	Indisponível	Disponível	Disponível	Disponível	Disponível
Cluste de banco de dados multi-AZ	Não disponível	Não disponível	Disponível	Indisponível	Disponível	Indisponível
Insight de Perfor ce	Indisponível	Disponível	Disponível	Disponível	Disponível	Disponível

Atribut	RDS para Db2	RDS para MariaDB	RDS para MySQL	RDS para Oracle	RDS para PostgreSQL	RDS para SQL Server
RDS Custor	Não disponível	Não disponível	Não disponível	Disponível	Indisponível	Disponível
RDS Proxy	Indisponível	Disponível	Disponível	Indisponível	Disponível	Disponível
Integra o do Secret Manag	Disponível	Disponível	Disponível	Disponível	Disponível	Disponível

Regiões e mecanismos de banco de dados compatíveis com implantações azul/verde do Amazon RDS

Uma implantação azul/verde copia um ambiente de banco de dados de produção em um ambiente de teste separado e sincronizado. Usando as implantações azul/verde do Amazon, você pode fazer alterações no banco de dados no ambiente de teste sem afetar o ambiente de produção. Por exemplo, você pode atualizar a versão principal ou secundária do mecanismo de banco de dados, alterar os parâmetros do banco de dados ou fazer alterações no esquema no ambiente de teste. Quando estiver pronto, você poderá promover o ambiente de teste como o novo ambiente de banco de dados de produção. Para ter mais informações, consulte Usar implantações azul/verde do Amazon RDS para atualizações de banco de dados.

As implantações azuis/verdes são compatíveis com todas as Regiões da AWS.

As implantações azuis/verdes não são compatíveis com os seguintes mecanismos:

- RDS para Db2
- RDS para SQL Server
- RDS para Oracle

Tópicos

- Implantações azuis/verdes com RDS para MariaDB

- Implantações azuis/verdes com RDS para MySQL
- Implantações azuis/verdes com RDS para PostgreSQL

Implantações azuis/verdes com RDS para MariaDB

Para o RDS para MariaDB, as implantações azuis/verdes são compatíveis com as seguintes versões:

- RDS para MariaDB 11.4 (todas as versões disponíveis)
- RDS para MariaDB versão 10.2 e 10 posteriores

Implantações azuis/verdes com RDS para MySQL

Para o RDS para MySQL, as implantações azuis/verdes são compatíveis com as seguintes versões:

- RDS para MySQL 8.4 (todas as versões disponíveis)
- RDS para MySQL 8.0 (todas as versões disponíveis)
- RDS para MySQL 5.7 (todas as versões disponíveis)

Implantações azuis/verdes com RDS para PostgreSQL

Para o RDS para PostgreSQL, as implantações azuis/verdes são compatíveis com a versão 11.1 e todas as versões principais e secundárias posteriores.

> **Note**
>
> Em determinadas condições, o RDS para PostgreSQL usa replicação lógica em vez de replicação física para manter o ambiente verde sincronizado com o ambiente azul. Para ter mais informações, consulte ???.

Regiões e mecanismos de banco de dados compatíveis com backups automatizados entre regiões no Amazon RDS

Ao usar a replicação de backup no Amazon RDS, você pode configurar sua instância de banco de dados do RDS para replicar snapshots e logs de transações para uma região de destino. Quando a

replicação de backup é configurada para uma instância de banco de dados, o RDS inicia uma cópia entre regiões de todos os snapshots e logs de transação assim que estiverem prontos. Para ter mais informações, consulte Replicar backups automatizados para outra Região da AWS.

Para ter informações sobre as Regiões da AWS onde a replicação de backup está disponível, consulte Replicar backups automatizados para outra Região da AWS.

Tópicos

- Replicação de backup com RDS para Db2
- Replicação de backup com RDS para MariaDB
- Replicação de backup com RDS para MySQL
- Replicação de backup com RDS para Oracle
- Replicação de backup com RDS para PostgreSQL
- Replicação de backup com RDS para SQL Server

Replicação de backup com RDS para Db2

O Amazon RDS é compatível com a replicação de backup para todas as versões disponíveis atualmente do RDS para Db2.

Replicação de backup com RDS para MariaDB

O Amazon RDS é compatível com a replicação de backup para todas as versões disponíveis atualmente do RDS para MariaDB.

Replicação de backup com RDS para MySQL

O Amazon RDS é compatível com a replicação de backup para todas as versões disponíveis atualmente do RDS para MySQL.

Replicação de backup com RDS para Oracle

O Amazon RDS é compatível com a replicação de backup para todas as versões disponíveis atualmente do RDS para Oracle.

Replicação de backup com RDS para PostgreSQL

O Amazon RDS é compatível com a replicação de backup para todas as versões disponíveis atualmente do RDS para PostgreSQL.

Replicação de backup com RDS para SQL Server

O Amazon RDS é compatível com a replicação de backup para todas as versões disponíveis atualmente do RDS para SQL Server.

Regiões e mecanismos de banco de dados compatíveis com réplicas de leitura entre regiões no Amazon RDS

Usando as réplicas de leitura entre regiões no Amazon RDS, você pode criar uma réplica de leitura do MariaDB, do MySQL, do Oracle, do PostgreSQL ou do SQL Server em uma região da diferente da instância de banco de dados de origem. Para ter mais informações sobre réplicas de leitura entre regiões, inclusive considerações sobre regiões de origem e destino, consulte Criar uma réplica de leitura em uma Região da AWS diferente.

As réplicas de leitura entre regiões não estão disponíveis com os seguintes mecanismos:

- RDS para Db2

Tópicos

- Réplicas de leitura entre regiões com o RDS para MariaDB
- Réplicas de leitura entre regiões com o RDS para MySQL
- Réplicas de leitura entre regiões com o RDS para Oracle
- Réplicas de leitura entre regiões com o RDS para PostgreSQL
- Réplicas de leitura entre regiões com o RDS para SQL Server

Réplicas de leitura entre regiões com o RDS para MariaDB

As réplicas de leitura entre regiões com RDS para MariaDB estão disponíveis em todas as regiões para as seguintes versões:

- RDS para MariaDB 11.4 (todas as versões disponíveis)
- RDS para MariaDB 10.11 (todas as versões disponíveis)
- RDS para MariaDB 10.6 (todas as versões disponíveis)
- RDS para MariaDB 10.5 (todas as versões disponíveis)
- RDS para MariaDB 10.4 (todas as versões disponíveis)

Réplicas de leitura entre regiões com o RDS para MySQL

As réplicas de leitura entre regiões com RDS para MySQL estão disponíveis em todas as regiões para as seguintes versões:

- RDS para MySQL 8.4 (todas as versões disponíveis)
- RDS para MySQL 8.0 (todas as versões disponíveis)
- RDS para MySQL 5.7 (todas as versões disponíveis)

Réplicas de leitura entre regiões com o RDS para Oracle

As réplicas de leitura entre regiões com o RDS para Oracle estão disponíveis em todas as Regiões da AWS para todas as versões de banco de dados compatíveis que usam a Enterprise Edition. As réplicas são aceitas somente em não CDBs e na configuração de locatário único da arquitetura do CDB. As réplicas de leitura entre regiões não são compatíveis com a configuração de multilocatário da arquitetura de CDB.

Para ter mais informações sobre requisitos adicionais para réplicas de leitura entre regiões com o RDS para Oracle, consulte Requisitos e considerações sobre réplicas do RDS para Oracle.

Réplicas de leitura entre regiões com o RDS para PostgreSQL

As réplicas de leitura entre regiões com RDS para PostgreSQL estão disponíveis em todas as regiões para as seguintes versões:

- RDS para PostgreSQL 17 (todas as versões disponíveis)
- RDS para PostgreSQL 16 (todas as versões disponíveis)
- RDS para PostgreSQL 15 (todas as versões disponíveis)
- RDS para PostgreSQL 14 (todas as versões disponíveis)
- RDS para PostgreSQL 13 (todas as versões disponíveis)
- RDS para PostgreSQL 12 (todas as versões disponíveis)
- RDS para PostgreSQL 11 (todas as versões disponíveis)
- RDS para PostgreSQL 10 (todas as versões disponíveis)

Réplicas de leitura entre regiões com o RDS para SQL Server

As réplicas de leitura entre regiões com RDS para SQL Server estão disponíveis em todas as regiões, exceto as seguintes:

- Africa (Cape Town)
- Ásia-Pacífico (Hong Kong)
- Ásia-Pacífico (Hyderabad)
- Ásia-Pacífico (Jacarta)
- Ásia-Pacífico (Melbourne)
- Oeste do Canadá (Calgary)
- Europa (Milão)
- Europa (Espanha)
- Europa (Zurique)
- Israel (Tel Aviv)
- Oriente Médio (Bahrein)
- Oriente Médio (Emirados Árabes Unidos)

As réplicas de leitura entre regiões com o RDS para SQL Server estão disponíveis para as seguintes versões usando o Microsoft SQL Server Enterprise Edition:

- RDS para SQL Server 2022
- RDS para SQL Server 2019 (versão 15.00.4073.23 e posteriores)
- RDS para SQL Server 2017 (versão 14.00.3281.6 e posteriores)
- RDS para SQL Server 2016 (versão 13.00.6300.2 e posteriores)

Regiões e mecanismos de banco de dados compatíveis com fluxos de atividades de banco de dados no Amazon RDS

Ao usar fluxos de atividade de banco de dados no Amazon RDS, é possível monitorar e definir alarmes para atividades de auditoria nos bancos de dados Oracle e SQL Server. Para ter mais informações, consulte Visão geral dos Database Activity Streams.

Fluxos de atividades de banco de dados não estão disponíveis com os seguintes mecanismos:

- RDS para Db2
- RDS para MariaDB
- RDS para MySQL
- RDS para PostgreSQL.

Tópicos

- Fluxos de atividade de banco de dados com o RDS para Oracle
- Fluxos de atividade de banco de dados com o RDS para SQL Server

Fluxos de atividade de banco de dados com o RDS para Oracle

As regiões e as versões do mecanismo a seguir estão disponíveis para fluxos de atividade de banco de dados com o RDS para Oracle.

Para ter mais informações sobre os requisitos adicionais para fluxos de atividade de banco de dados com o RDS para Oracle, consulte Visão geral dos Database Activity Streams.

Região	RDS para Oracle 21c	RDS para Oracle 19c
Leste dos EUA (Norte da Virgínia)	Indisponível	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 e posterior, usando Enterprise Edition (EE) ou Standard Edition 2 (SE2)
Leste dos EUA (Ohio)	Indisponível	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 e posterior, usando Enterprise Edition (EE) ou Standard Edition 2 (SE2)
Oeste dos EUA (N. da Califórnia)	Indisponível	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 e posterior, usando Enterprise Edition (EE) ou Standard Edition 2 (SE2)
Oeste dos EUA (Oregon)	Indisponível	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 e posterior, usando

Região	RDS para Oracle 21c	RDS para Oracle 19c
		Enterprise Edition (EE) ou Standard Edition 2 (SE2)
África (Cidade do Cabo)	Indisponível	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 e posterior, usando Enterprise Edition (EE) ou Standard Edition 2 (SE2)
Ásia-Pacífico (Hong Kong)	Indisponível	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 e posterior, usando Enterprise Edition (EE) ou Standard Edition 2 (SE2)
Ásia-Pacífico (Hyderabad)	Indisponível	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 e posterior, usando Enterprise Edition (EE) ou Standard Edition 2 (SE2)
Ásia-Pacífico (Jacarta)	Indisponível	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 e posterior, usando Enterprise Edition (EE) ou Standard Edition 2 (SE2)
Ásia-Pacífico (Malásia)	Não disponível	Não disponível
Ásia-Pacífico (Melbourne)	Não disponível	Não disponível
Ásia-Pacífico (Mumbai)	Indisponível	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 e posterior, usando Enterprise Edition (EE) ou Standard Edition 2 (SE2)
Ásia-Pacífico (Osaka)	Indisponível	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 e posterior, usando Enterprise Edition (EE) ou Standard Edition 2 (SE2)

Região	RDS para Oracle 21c	RDS para Oracle 19c
Ásia-Pacífico (Seul)	Indisponível	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 e posterior, usando Enterprise Edition (EE) ou Standard Edition 2 (SE2)
Ásia-Pacífico (Singapura)	Indisponível	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 e posterior, usando Enterprise Edition (EE) ou Standard Edition 2 (SE2)
Ásia-Pacífico (Sydney)	Indisponível	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 e posterior, usando Enterprise Edition (EE) ou Standard Edition 2 (SE2)
Ásia-Pacífico (Tóquio)	Indisponível	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 e posterior, usando Enterprise Edition (EE) ou Standard Edition 2 (SE2)
Canadá (Central)	Indisponível	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 e posterior, usando Enterprise Edition (EE) ou Standard Edition 2 (SE2)
Oeste do Canadá (Calgary)	Indisponível	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 e posterior, usando Enterprise Edition (EE) ou Standard Edition 2 (SE2)
China (Pequim)	Não disponível	Não disponível
China (Ningxia)	Não disponível	Não disponível

Região	RDS para Oracle 21c	RDS para Oracle 19c
Europa (Frankfurt)	Indisponível	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 e posterior, usando Enterprise Edition (EE) ou Standard Edition 2 (SE2)
Europa (Irlanda)	Indisponível	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 e posterior, usando Enterprise Edition (EE) ou Standard Edition 2 (SE2)
Europa (Londres)	Indisponível	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 e posterior, usando Enterprise Edition (EE) ou Standard Edition 2 (SE2)
Europa (Milão)	Indisponível	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 e posterior, usando Enterprise Edition (EE) ou Standard Edition 2 (SE2)
Europa (Paris)	Indisponível	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 e posterior, usando Enterprise Edition (EE) ou Standard Edition 2 (SE2)
Europa (Espanha)	Indisponível	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 e posterior, usando Enterprise Edition (EE) ou Standard Edition 2 (SE2)
Europa (Estocolmo)	Indisponível	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 e posterior, usando Enterprise Edition (EE) ou Standard Edition 2 (SE2)
Europa (Zurique)	Não disponível	Não disponível

Região	RDS para Oracle 21c	RDS para Oracle 19c
Ásia-Pacífico (Melbourne)	Não disponível	Não disponível
Oriente Médio (Barém)	Indisponível	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 e posterior, usando Enterprise Edition (EE) ou Standard Edition 2 (SE2)
Oriente Médio (Emirados Árabes Unidos)	Indisponível	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 e posterior, usando Enterprise Edition (EE) ou Standard Edition 2 (SE2)
América do Sul (São Paulo)	Indisponível	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 e posterior, usando Enterprise Edition (EE) ou Standard Edition 2 (SE2)
AWS GovCloud (Leste dos EUA)	Não disponível	Não disponível
AWS GovCloud (Oeste dos EUA)	Não disponível	Não disponível

Fluxos de atividade de banco de dados com o RDS para SQL Server

As regiões e as versões do mecanismo a seguir estão disponíveis para fluxos de atividade de banco de dados com o RDS para SQL Server.

Para ter mais informações sobre os requisitos adicionais para fluxos de atividade de banco de dados com o RDS para SQL Server, consulte Visão geral dos Database Activity Streams.

Região	RDS para SQL Server 2019	RDS para SQL Server 2017	RDS para SQL Server 2016
Leste dos EUA (Norte da Virgínia)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis

Região	RDS para SQL Server 2019	RDS para SQL Server 2017	RDS para SQL Server 2016
Leste dos EUA (Ohio)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Oeste dos EUA (N. da Califórnia)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Oeste dos EUA (Oregon)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
África (Cidade do Cabo)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Hong Kong)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Hyderabad)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Jacarta)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Malásia)	Não disponível	Não disponível	Não disponível
Ásia-Pacífico (Melbourne)	Não disponível	Não disponível	Não disponível
Ásia-Pacífico (Mumbai)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Osaka)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Seul)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis

Região	RDS para SQL Server 2019	RDS para SQL Server 2017	RDS para SQL Server 2016
Ásia-Pacífico (Singapura)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Sydney)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Tóquio)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Canadá (Central)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Oeste do Canadá (Calgary)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
China (Pequim)	Não disponível	Não disponível	Não disponível
China (Ningxia)	Não disponível	Não disponível	Não disponível
Europa (Frankfurt)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Irlanda)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Londres)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Milão)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Paris)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Espanha)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis

Região	RDS para SQL Server 2019	RDS para SQL Server 2017	RDS para SQL Server 2016
Europa (Estocolmo)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Zurique)	Não disponível	Não disponível	Não disponível
Israel (Tel Aviv)	Não disponível	Não disponível	Não disponível
Oriente Médio (Barém)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Oriente Médio (Emirados Árabes Unidos)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
América do Sul (São Paulo)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
AWS GovCloud (Leste dos EUA)	Não disponível	Não disponível	Não disponível
AWS GovCloud (Oeste dos EUA)	Não disponível	Não disponível	Não disponível

Regiões e mecanismos de banco de dados compatíveis com o modo de pilha dupla no Amazon RDS

Ao usar o modo de duas pilhas no RDS, os recursos podem se comunicar com a instância de banco de dados por Internet Protocol versão 4 (IPv4), Internet Protocol versão 6 (IPv6) ou ambos. Para obter mais informações, consulte Modo de pilha dupla.

Tópicos

- Modo de duas pilhas com o RDS para Db2
- Modo de duas pilhas com RDS para MariaDB
- Modo de duas pilhas com RDS para MySQL

- Modo de duas pilhas com RDS para Oracle
- Modo de duas pilhas com RDS para PostgreSQL
- Modo de duas pilhas com RDS para SQL Server

O IPv6 não é compatível com nenhum mecanismo de banco de dados do Amazon RDS nas seguintes Regiões da AWS:

- Ásia-Pacífico (Hyderabad)
- Oriente Médio (Emirados Árabes Unidos)
- Ásia-Pacífico (Malásia)
- Ásia-Pacífico (Melbourne)
- Oeste do Canadá (Calgary)
- Europa (Zurique)
- Israel (Tel Aviv)

Modo de duas pilhas com o RDS para Db2

As regiões e as versões do mecanismo a seguir estão disponíveis para o modo de duas pilhas com o RDS para Db2.

Região	RDS para Db2 11.5				
Leste dos EUA (Norte da Virgínia)	Todas as versões disponíveis				
Leste dos EUA (Ohio)	Todas as versões disponíveis				
Oeste dos EUA (Norte da Califórnia)	Todas as versões disponíveis				

Região	RDS para Db2 11.5				
Oeste dos EUA (Oregon)	Todas as versões disponíveis				
África (Cidade do Cabo)	Todas as versões disponíveis				
Ásia-Pacífico (Hong Kong)	Todas as versões disponíveis				
Ásia-Pacífico (Hyderabad)	Indisponível				
Ásia-Pacífico (Jacarta)	Todas as versões disponíveis				
Ásia-Pacífico (Malásia)	Indisponível				
Ásia-Pacífico (Melbourne)	Indisponível				
Ásia-Pacífico (Mumbai)	Todas as versões disponíveis				
Ásia-Pacífico (Osaka)	Todas as versões disponíveis				
Ásia-Pacífico (Seul)	Todas as versões disponíveis				

Região	RDS para Db2 11.5				
Ásia-Pacífico (Singapura)	Todas as versões disponíveis				
Ásia-Pacífico (Sydney)	Todas as versões disponíveis				
Ásia-Pacífico (Tóquio)	Todas as versões disponíveis				
Canadá (Central)	Todas as versões disponíveis				
Oeste do Canadá (Calgary)	Indisponível				
China (Pequim)	Indisponível				
China (Ningxia)	Indisponível				
Europa (Frankfurt)	Todas as versões disponíveis				
Europa (Irlanda)	Todas as versões disponíveis				

Região	RDS para Db2 11.5				
Europa (Londres)	Todas as versões disponíveis				
Europa (Milão)	Todas as versões disponíveis				
Europa (Paris)	Todas as versões disponíveis				
Europa (Espanha)	Indisponível				
Europa (Estocolmo)	Todas as versões disponíveis				
Europa (Zurique)	Indisponível				
Israel (Tel Aviv)	Indisponível				
Oriente Médio (Bahrein)	Todas as versões disponíveis				
Oriente Médio (Emirados Árabes Unidos)	Indisponível				

Região	RDS para Db2 11.5				
América do Sul (São Paulo)	Todas as versões disponíveis				
AWS GovCloud (Leste dos EUA)	Indisponível				
AWS GovCloud (Oeste dos EUA)	Indisponível				

Modo de duas pilhas com RDS para MariaDB

As regiões e as versões do mecanismo a seguir estão disponíveis para o modo de duas pilhas com o RDS para MariaDB.

Região	RDS para MariaDB 11.4	RDS para MariaDB 10.11	RDS para MariaDB 10.6	RDS para MariaDB 10.5	RDS para MariaDB 10.4
Leste dos EUA (Norte da Virgínia)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Leste dos EUA (Ohio)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Oeste dos EUA (Norte da Califórnia)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis

Região	RDS para MariaDB 11.4	RDS para MariaDB 10.11	RDS para MariaDB 10.6	RDS para MariaDB 10.5	RDS para MariaDB 10.4
Oeste dos EUA (Oregon)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
África (Cidade do Cabo)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Hong Kong)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Hyderabad)	Não disponível	Não disponível	Não disponível	Não disponível	Não disponível
Ásia-Pacífico (Jacarta)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Malásia)	Não disponível	Não disponível	Não disponível	Não disponível	Não disponível
Ásia-Pacífico (Melbourne)	Não disponível	Não disponível	Não disponível	Não disponível	Não disponível
Ásia-Pacífico (Mumbai)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Osaka)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis

Região	RDS para MariaDB 11.4	RDS para MariaDB 10.11	RDS para MariaDB 10.6	RDS para MariaDB 10.5	RDS para MariaDB 10.4
Ásia-Pacífico (Seul)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Singapura)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Sydney)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Tóquio)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Canadá (Central)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Oeste do Canadá (Calgary)	Não disponível	Não disponível	Não disponível	Não disponível	Não disponível
China (Pequim)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
China (Ningxia)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Frankfurt)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis

Região	RDS para MariaDB 11.4	RDS para MariaDB 10.11	RDS para MariaDB 10.6	RDS para MariaDB 10.5	RDS para MariaDB 10.4
Europa (Irlanda)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Londres)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Milão)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Paris)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Espanha)	Não disponível	Não disponível	Não disponível	Não disponível	Não disponível
Europa (Estocolmo)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Zurique)	Não disponível	Não disponível	Não disponível	Não disponível	Não disponível
Israel (Tel Aviv)	Não disponível	Não disponível	Não disponível	Não disponível	Não disponível
Oriente Médio (Bahrein)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis

Região	RDS para MariaDB 11.4	RDS para MariaDB 10.11	RDS para MariaDB 10.6	RDS para MariaDB 10.5	RDS para MariaDB 10.4
Oriente Médio (Emirados Árabes Unidos)	Não disponível	Não disponível	Não disponível	Não disponível	Não disponível
América do Sul (São Paulo)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
AWS GovCloud (Leste dos EUA)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
AWS GovCloud (Oeste dos EUA)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis

Modo de duas pilhas com RDS para MySQL

As regiões e as versões do mecanismo a seguir estão disponíveis para o modo de duas pilhas com o RDS para MySQL.

Região	RDS para MySQL 8.4	RDS para MySQL 8.0	RDS para MySQL 5.7 (com suporte estendido do RDS)
Leste dos EUA (Norte da Virgínia)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis

Região	RDS para MySQL 8.4	RDS para MySQL 8.0	RDS para MySQL 5.7 (com suporte estendido do RDS)
Leste dos EUA (Ohio)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Oeste dos EUA (Norte da Califórnia)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Oeste dos EUA (Oregon)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
África (Cidade do Cabo)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Hong Kong)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Hyderabad)	Não disponível	Não disponível	Não disponível
Ásia-Pacífico (Jacarta)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Malásia)	Não disponível	Não disponível	Não disponível
Ásia-Pacífico (Melbourne)	Não disponível	Não disponível	Não disponível
Ásia-Pacífico (Mumbai)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Osaka)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Seul)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis

Região	RDS para MySQL 8.4	RDS para MySQL 8.0	RDS para MySQL 5.7 (com suporte estendido do RDS)
Ásia-Pacífico (Singapura)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Sydney)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Tóquio)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Canadá (Central)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Oeste do Canadá (Calgary)	Não disponível	Não disponível	Não disponível
China (Pequim)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
China (Ningxia)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Frankfurt)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Irlanda)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Londres)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Milão)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Paris)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis

Região	RDS para MySQL 8.4	RDS para MySQL 8.0	RDS para MySQL 5.7 (com suporte estendido do RDS)
Europa (Espanha)	Não disponível	Não disponível	Não disponível
Europa (Estocolmo)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Zurique)	Não disponível	Não disponível	Não disponível
Israel (Tel Aviv)	Não disponível	Não disponível	Não disponível
Oriente Médio (Bahrein)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Oriente Médio (Emirados Árabes Unidos)	Não disponível	Não disponível	Não disponível
América do Sul (São Paulo)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
AWS GovCloud (Leste dos EUA)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
AWS GovCloud (Oeste dos EUA)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis

Modo de duas pilhas com RDS para Oracle

As regiões e as versões do mecanismo a seguir estão disponíveis para o modo de duas pilhas com o RDS para Oracle.

Região	RDS para Oracle 21c	RDS para Oracle 19c
Leste dos EUA (Norte da Virgínia)	Todas as versões disponíveis	Todas as versões disponíveis

Região	RDS para Oracle 21c	RDS para Oracle 19c
Leste dos EUA (Ohio)	Todas as versões disponíveis	Todas as versões disponíveis
Oeste dos EUA (Norte da Califórnia)	Todas as versões disponíveis	Todas as versões disponíveis
Oeste dos EUA (Oregon)	Todas as versões disponíveis	Todas as versões disponíveis
África (Cidade do Cabo)	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Hong Kong)	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Hyderabad)	Não disponível	Não disponível
Ásia-Pacífico (Jacarta)	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Malásia)	Não disponível	Não disponível
Ásia-Pacífico (Melbourne)	Não disponível	Não disponível
Ásia-Pacífico (Mumbai)	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Osaka)	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Seul)	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Singapura)	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Sydney)	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Tóquio)	Todas as versões disponíveis	Todas as versões disponíveis
Canadá (Central)	Todas as versões disponíveis	Todas as versões disponíveis
Oeste do Canadá (Calgary)	Não disponível	Não disponível
China (Pequim)	Todas as versões disponíveis	Todas as versões disponíveis
China (Ningxia)	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Frankfurt)	Todas as versões disponíveis	Todas as versões disponíveis

Região	RDS para Oracle 21c	RDS para Oracle 19c
Europa (Irlanda)	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Londres)	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Milão)	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Paris)	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Espanha)	Não disponível	Não disponível
Europa (Estocolmo)	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Zurique)	Não disponível	Não disponível
Israel (Tel Aviv)	Não disponível	Não disponível
Oriente Médio (Bahrein)	Todas as versões disponíveis	Todas as versões disponíveis
Oriente Médio (Emirados Árabes Unidos)	Não disponível	Não disponível
América do Sul (São Paulo)	Todas as versões disponíveis	Todas as versões disponíveis
AWS GovCloud (Leste dos EUA)	Todas as versões disponíveis	Todas as versões disponíveis
AWS GovCloud (Oeste dos EUA)	Todas as versões disponíveis	Todas as versões disponíveis

Modo de duas pilhas com RDS para PostgreSQL

As regiões e as versões do mecanismo a seguir estão disponíveis para o modo de duas pilhas com o RDS para PostgreSQL.

Região	RDS para PostgreSQL 17	RDS para PostgreSQL 16	RDS para PostgreSQL 15	RDS para PostgreSQL 14	RDS para PostgreSQL 13	RDS para PostgreSQL 12	RDS para PostgreSQL 11	RDS para PostgreSQL 10
Leste dos EUA (Norte da Virgínia)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Leste dos EUA (Ohio)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Oeste dos EUA (Norte da Califórnia)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Oeste dos EUA (Oregon)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
África (Cidade do Cabo)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis

Região	RDS para PostgreSQL 17	RDS para PostgreSQL 16	RDS para PostgreSQL 15	RDS para PostgreSQL 14	RDS para PostgreSQL 13	RDS para PostgreSQL 12	RDS para PostgreSQL 11	RDS para PostgreSQL 10
Ásia-Pacífico (Hong Kong)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Hyderabad)	Não disponível	Não disponível	Não disponível	Não disponível	Não disponível	Não disponível	Não disponível	Não disponível
Ásia-Pacífico (Jacarta)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Malásia)	Não disponível	Não disponível	Não disponível	Não disponível	Não disponível	Não disponível	Não disponível	Não disponível
Ásia-Pacífico (Melbourne)	Não disponível	Não disponível	Não disponível	Não disponível	Não disponível	Não disponível	Não disponível	Não disponível
Ásia-Pacífico (Mumbai)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis

Região	RDS para PostgreSQL 17	RDS para PostgreSQL 16	RDS para PostgreSQL 15	RDS para PostgreSQL 14	RDS para PostgreSQL 13	RDS para PostgreSQL 12	RDS para PostgreSQL 11	RDS para PostgreSQL 10
Ásia-Pacífico (Osaka)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Seul)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Singapura)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Sydney)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Tóquio)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis

Região	RDS para PostgreSQL 17	RDS para PostgreSQL 16	RDS para PostgreSQL 15	RDS para PostgreSQL 14	RDS para PostgreSQL 13	RDS para PostgreSQL 12	RDS para PostgreSQL 11	RDS para PostgreSQL 10
Canadá (Central)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Oeste do Canadá (Calgary)	Não disponível	Não disponível	Não disponível	Não disponível	Não disponível	Não disponível	Não disponível	Não disponível
China (Pequim)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
China (Ningxia)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Frankfurt)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis

Região	RDS para PostgreSQL 17	RDS para PostgreSQL 16	RDS para PostgreSQL 15	RDS para PostgreSQL 14	RDS para PostgreSQL 13	RDS para PostgreSQL 12	RDS para PostgreSQL 11	RDS para PostgreSQL 10
Europa (Irlanda)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Londres)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Milão)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Paris)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Espanha)	Não disponível	Não disponível	Não disponível	Não disponível	Não disponível	Não disponível	Não disponível	Não disponível
Europa (Estocolmo)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis

Região	RDS para PostgreSQL 17	RDS para PostgreSQL 16	RDS para PostgreSQL 15	RDS para PostgreSQL 14	RDS para PostgreSQL 13	RDS para PostgreSQL 12	RDS para PostgreSQL 11	RDS para PostgreSQL 10
Europa (Zurique)	Não disponível	Não disponível	Não disponível	Não disponível	Não disponível	Não disponível	Não disponível	Não disponível
Israel (Tel Aviv)	Não disponível	Não disponível	Não disponível	Não disponível	Não disponível	Não disponível	Não disponível	Não disponível
Oriente Médio (Bahrein)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Oriente Médio (Emirados Árabes Unidos)	Não disponível	Não disponível	Não disponível	Não disponível	Não disponível	Não disponível	Não disponível	Não disponível
América do Sul (São Paulo)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
AWS GovCloud (Leste dos EUA)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis

Região	RDS para PostgreSQL 17	RDS para PostgreSQL 16	RDS para PostgreSQL 15	RDS para PostgreSQL 14	RDS para PostgreSQL 13	RDS para PostgreSQL 12	RDS para PostgreSQL 11	RDS para PostgreSQL 10
AWS GovCloud (Oeste dos EUA)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis

Modo de duas pilhas com RDS para SQL Server

As regiões e as versões do mecanismo a seguir estão disponíveis para o modo de duas pilhas com o RDS para SQL Server.

Região	RDS para SQL Server 2019	RDS para SQL Server 2017	RDS para SQL Server 2016
Leste dos EUA (Norte da Virgínia)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Leste dos EUA (Ohio)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Oeste dos EUA (Norte da Califórnia)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Oeste dos EUA (Oregon)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
África (Cidade do Cabo)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Hong Kong)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis

Região	RDS para SQL Server 2019	RDS para SQL Server 2017	RDS para SQL Server 2016
Ásia-Pacífico (Hyderabad)	Não disponível	Não disponível	Não disponível
Ásia-Pacífico (Jacarta)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Malásia)	Não disponível	Não disponível	Não disponível
Ásia-Pacífico (Melbourne)	Não disponível	Não disponível	Não disponível
Ásia-Pacífico (Mumbai)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Osaka)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Seul)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Singapura)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Sydney)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Tóquio)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Canadá (Central)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Oeste do Canadá (Calgary)	Não disponível	Não disponível	Não disponível

Região	RDS para SQL Server 2019	RDS para SQL Server 2017	RDS para SQL Server 2016
China (Pequim)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
China (Ningxia)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Frankfurt)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Irlanda)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Londres)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Milão)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Paris)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Espanha)	Não disponível	Não disponível	Não disponível
Europa (Estocolmo)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Zurique)	Não disponível	Não disponível	Não disponível
Israel (Tel Aviv)	Não disponível	Não disponível	Não disponível
Oriente Médio (Bahrein)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Oriente Médio (Emirados Árabes Unidos)	Não disponível	Não disponível	Não disponível

Região	RDS para SQL Server 2019	RDS para SQL Server 2017	RDS para SQL Server 2016
América do Sul (São Paulo)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
AWS GovCloud (Leste dos EUA)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
AWS GovCloud (Oeste dos EUA)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis

Regiões e mecanismos de banco de dados compatíveis com a exportação de snapshots para o S3 no Amazon RDS

Você pode exportar dados de snapshots de banco de dados do RDS para um bucket do Amazon S3. Você pode exportar todos os tipos de snapshots de banco de dados, inclusive snapshots manuais, snapshots automatizados do sistema e snapshots criados pelo AWS Backup. Depois que os dados são exportados, você pode analisar os dados exportados diretamente por meio de ferramentas, como Amazon Athena ou Amazon Redshift Spectrum. Para ter mais informações, consulte Exportação de dados de snapshot de banco de dados para o Amazon S3 para Amazon RDS.

A exportação de snapshots para o S3 não está disponível com os seguintes mecanismos:

- RDS para Db2
- RDS para Oracle
- RDS para SQL Server

Tópicos

- Exportar snapshots para o S3 com o RDS para MariaDB
- Exportar snapshots para o S3 com o RDS para MySQL
- Exportar snapshots para o S3 com o RDS para PostgreSQL

Exportar snapshots para o S3 com o RDS para MariaDB

As regiões e as versões do mecanismo a seguir estão disponíveis para exportar snapshots para o S3 com RDS para MariaDB.

Região	RDS para MariaDB 11.4	RDS para MariaDB 10.11	RDS para MariaDB 10.6	RDS para MariaDB 10.5	RDS para MariaDB 10.4
Leste dos EUA (Norte da Virgínia)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Leste dos EUA (Ohio)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Oeste dos EUA (Norte da Califórnia)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Oeste dos EUA (Oregon)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
África (Cidade do Cabo)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Hong Kong)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Hyderabad)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis

Região	RDS para MariaDB 11.4	RDS para MariaDB 10.11	RDS para MariaDB 10.6	RDS para MariaDB 10.5	RDS para MariaDB 10.4
Ásia-Pacífico (Jacarta)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Malásia)	Não disponível	Não disponível	Não disponível	Não disponível	Não disponível
Ásia-Pacífico (Melbourne)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Mumbai)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia Pacifico (Osaka)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Seul)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Singapura)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Sydney)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Tóquio)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis

Região	RDS para MariaDB 11.4	RDS para MariaDB 10.11	RDS para MariaDB 10.6	RDS para MariaDB 10.5	RDS para MariaDB 10.4
Canadá (Central)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Oeste do Canadá (Calgary)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
China (Pequim)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
China (Ningxia)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Frankfurt)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Irlanda)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Londres)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Milão)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Paris)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis

Região	RDS para MariaDB 11.4	RDS para MariaDB 10.11	RDS para MariaDB 10.6	RDS para MariaDB 10.5	RDS para MariaDB 10.4
Europa (Espanha)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Estocolmo)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Zurique)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Israel (Tel Aviv)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Oriente Médio (Bahrein)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Oriente Médio (Emirados Árabes Unidos)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
América do Sul (São Paulo)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
AWS GovCloud (Leste dos EUA)	Não disponível	Não disponível	Não disponível	Não disponível	Não disponível

Região	RDS para MariaDB 11.4	RDS para MariaDB 10.11	RDS para MariaDB 10.6	RDS para MariaDB 10.5	RDS para MariaDB 10.4
AWS GovCloud (Oeste dos EUA)	Não disponível	Não disponível	Não disponível	Não disponível	Não disponível

Exportar snapshots para o S3 com o RDS para MySQL

As regiões e as versões do mecanismo a seguir estão disponíveis para exportar snapshots para o S3 com RDS para MySQL.

Região	RDS para MySQL 8.4	RDS para MySQL 8.0	RDS para MySQL 5.7 (com suporte estendido do RDS)
Leste dos EUA (Ohio)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Leste dos EUA (Norte da Virgínia)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Oeste dos EUA (Norte da Califórnia)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Oeste dos EUA (Oregon)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
África (Cidade do Cabo)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Hong Kong)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Hyderabad)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis

Região	RDS para MySQL 8.4	RDS para MySQL 8.0	RDS para MySQL 5.7 (com suporte estendido do RDS)
Ásia-Pacífico (Jacarta)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Malásia)	Não disponível	Não disponível	Não disponível
Ásia-Pacífico (Melbourne)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Mumbai)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia Pacifico (Osaka)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Seul)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Singapura)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Sydney)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Tóquio)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Canadá (Central)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Oeste do Canadá (Calgary)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
China (Pequim)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis

Região	RDS para MySQL 8.4	RDS para MySQL 8.0	RDS para MySQL 5.7 (com suporte estendido do RDS)
China (Ningxia)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Frankfurt)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Irlanda)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Londres)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Milão)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Paris)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Espanha)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Estocolmo)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Zurique)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Israel (Tel Aviv)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Oriente Médio (Bahrein)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis

Região	RDS para MySQL 8.4	RDS para MySQL 8.0	RDS para MySQL 5.7 (com suporte estendido do RDS)
Oriente Médio (Emirados Árabes Unidos)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
América do Sul (São Paulo)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
AWS GovCloud (Leste dos EUA)	Não disponível	Não disponível	Não disponível
AWS GovCloud (Oeste dos EUA)	Não disponível	Não disponível	Não disponível

Exportar snapshots para o S3 com o RDS para PostgreSQL

As regiões e as versões do mecanismo a seguir estão disponíveis para exportar snapshots para o S3 com RDS para PostgreSQL.

Região	RDS para PostgreSQL 17	RDS para PostgreSQL 16	RDS para PostgreSQL 15	RDS para PostgreSQL 14	RDS para PostgreSQL 13	RDS para PostgreSQL 12	RDS para PostgreSQL 11	RDS para PostgreSQL 10
Leste dos EUA (Ohio)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Leste dos EUA (Norte	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões

Região	RDS para PostgreSQL 17	RDS para PostgreSQL 16	RDS para PostgreSQL 15	RDS para PostgreSQL 14	RDS para PostgreSQL 13	RDS para PostgreSQL 12	RDS para PostgreSQL 11	RDS para PostgreSQL 10
da Virgínia)	disponíveis	disponíveis	disponíveis	disponíveis	disponíveis	disponíveis	disponíveis	disponíveis
Oeste dos EUA (Norte da Califórnia)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Oeste dos EUA (Oregon)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
África (Cidade do Cabo)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Hong Kong)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis

Região	RDS para PostgreSQL 17	RDS para PostgreSQL 16	RDS para PostgreSQL 15	RDS para PostgreSQL 14	RDS para PostgreSQL 13	RDS para PostgreSQL 12	RDS para PostgreSQL 11	RDS para PostgreSQL 10
Ásia-Pacífico (Hyderabad)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Jacarta)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Malásia)	Não disponível	Não disponível	Não disponível	Não disponível	Não disponível	Não disponível	Não disponível	Não disponível
Ásia-Pacífico (Melbourne)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Mumbai)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia Pacifico (Osaka)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis

Região	RDS para PostgreSQL 17	RDS para PostgreSQL 16	RDS para PostgreSQL 15	RDS para PostgreSQL 14	RDS para PostgreSQL 13	RDS para PostgreSQL 12	RDS para PostgreSQL 11	RDS para PostgreSQL 10
Ásia-Pacífico (Seul)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Singapura)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Sydney)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Tóquio)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Canadá (Central)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis

Região	RDS para PostgreSQL 17	RDS para PostgreSQL 16	RDS para PostgreSQL 15	RDS para PostgreSQL 14	RDS para PostgreSQL 13	RDS para PostgreSQL 12	RDS para PostgreSQL 11	RDS para PostgreSQL 10
Oeste do Canadá (Calgary)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
China (Pequim)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
China (Ningxia)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Frankfurt)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Irlanda)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis

Região	RDS para PostgreSQL 17	RDS para PostgreSQL 16	RDS para PostgreSQL 15	RDS para PostgreSQL 14	RDS para PostgreSQL 13	RDS para PostgreSQL 12	RDS para PostgreSQL 11	RDS para PostgreSQL 10
Europa (Londres)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Milão)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Paris)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Espanha)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Estocolmo)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis

Região	RDS para PostgreSQL 17	RDS para PostgreSQL 16	RDS para PostgreSQL 15	RDS para PostgreSQL 14	RDS para PostgreSQL 13	RDS para PostgreSQL 12	RDS para PostgreSQL 11	RDS para PostgreSQL 10
Europa (Zurique)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Israel (Tel Aviv)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Oriente Médio (Bahrein)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Oriente Médio (Emirados Árabes Unidos)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
América do Sul (São Paulo)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis

Região	RDS para PostgreSQL 17	RDS para PostgreSQL 16	RDS para PostgreSQL 15	RDS para PostgreSQL 14	RDS para PostgreSQL 13	RDS para PostgreSQL 12	RDS para PostgreSQL 11	RDS para PostgreSQL 10
AWS GovCloud (Leste dos EUA)	Não disponível	Não disponível	Não disponível	Não disponível	Não disponível	Não disponível	Não disponível	Não disponível
AWS GovCloud (Oeste dos EUA)	Não disponível	Não disponível	Não disponível	Não disponível	Não disponível	Não disponível	Não disponível	Não disponível

Regiões e mecanismos de banco de dados compatíveis com a autenticação de banco de dados do IAM no Amazon RDS

Usando a autenticação de banco de dados do IAM Database do Amazon RDS, você pode autenticá-la sem uma senha quando se conectar a uma instância de banco de dados. Em vez disso, você usa um token de autenticação. Para ter mais informações, consulte Autenticação do banco de dados do IAM para MariaDB, MySQL e PostgreSQL.

A autenticação de banco de dados do IAM não está disponível com os seguintes mecanismos:

- RDS para Db2
- RDS para Oracle
- RDS para SQL Server

Tópicos

- Autenticação de banco de dados do IAM com RDS para MariaDB
- Autenticação de banco de dados do IAM com RDS para MySQL
- Autenticação de banco de dados do IAM para RDS para PostgreSQL

Autenticação de banco de dados do IAM com RDS para MariaDB

As regiões e as versões do mecanismo a seguir estão disponíveis para a autenticação de banco de dados do IAM com o RDS para MariaDB.

Região	RDS para MariaDB 11.4	RDS para MariaDB 10.11	RDS para MariaDB 10.6	RDS para MariaDB 10.5	RDS para MariaDB 10.4
Leste dos EUA (Norte da Virgínia)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Não disponível	Não disponível
Leste dos EUA (Ohio)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Não disponível	Não disponível
Oeste dos EUA (Norte da Califórnia)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Não disponível	Não disponível
Oeste dos EUA (Oregon)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Não disponível	Não disponível
África (Cidade do Cabo)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Não disponível	Não disponível
Ásia-Pacífico (Hong Kong)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Não disponível	Não disponível
Ásia-Pacífico (Hyderabad)	Não disponível	Não disponível	Não disponível	Não disponível	Não disponível
Ásia-Pacífico (Jacarta)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Não disponível	Não disponível

Região	RDS para MariaDB 11.4	RDS para MariaDB 10.11	RDS para MariaDB 10.6	RDS para MariaDB 10.5	RDS para MariaDB 10.4
Ásia-Pacífico (Malásia)	Todas as versões disponíveis	Todas as versões disponíveis	Não disponível	Não disponível	Não disponível
Ásia-Pacífico (Melbourne)	Não disponível	Não disponível	Não disponível	Não disponível	Não disponível
Ásia-Pacífico (Mumbai)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Não disponível	Não disponível
Ásia Pacifico (Osaka)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Não disponível	Não disponível
Ásia-Pacífico (Seul)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Não disponível	Não disponível
Ásia-Pacífico (Singapura)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Não disponível	Não disponível
Ásia-Pacífico (Sydney)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Não disponível	Não disponível
Ásia-Pacífico (Tóquio)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Não disponível	Não disponível
Canadá (Central)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Não disponível	Não disponível

Região	RDS para MariaDB 11.4	RDS para MariaDB 10.11	RDS para MariaDB 10.6	RDS para MariaDB 10.5	RDS para MariaDB 10.4
Oeste do Canadá (Calgary)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Não disponível	Não disponível
China (Pequim)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Não disponível	Não disponível
China (Ningxia)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Não disponível	Não disponível
Europa (Frankfurt)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Não disponível	Não disponível
Europa (Irlanda)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Não disponível	Não disponível
Europa (Londres)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Não disponível	Não disponível
Europa (Milão)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Não disponível	Não disponível
Europa (Paris)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Não disponível	Não disponível
Europa (Espanha)	Não disponível	Não disponível	Não disponível	Não disponível	Não disponível

Região	RDS para MariaDB 11.4	RDS para MariaDB 10.11	RDS para MariaDB 10.6	RDS para MariaDB 10.5	RDS para MariaDB 10.4
Europa (Estocolmo)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Não disponível	Não disponível
Europa (Zurique)	Não disponível	Não disponível	Não disponível	Não disponível	Não disponível
Israel (Tel Aviv)	Não disponível	Não disponível	Não disponível	Não disponível	Não disponível
Oriente Médio (Bahrein)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Não disponível	Não disponível
Oriente Médio (Emirados Árabes Unidos)	Não disponível	Não disponível	Não disponível	Não disponível	Não disponível
América do Sul (São Paulo)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Não disponível	Não disponível
AWS GovCloud (Leste dos EUA)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Não disponível	Não disponível
AWS GovCloud (Oeste dos EUA)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Não disponível	Não disponível

Autenticação de banco de dados do IAM com RDS para MySQL

A autenticação de banco de dados do IAM com o RDS para MySQL está disponível em todas as regiões para as seguintes versões:

- RDS para MySQL 8.4: todas as versões disponíveis
- RDS para MySQL 8.0: todas as versões disponíveis
- RDS para MySQL 5.7: todas as versões disponíveis

Autenticação de banco de dados do IAM para RDS para PostgreSQL

A autenticação de banco de dados do IAM com o RDS para PostgreSQL está disponível em todas as regiões para as seguintes versões:

- RDS para PostgreSQL 17: todas as versões disponíveis
- RDS para PostgreSQL 16: todas as versões disponíveis
- RDS para PostgreSQL 15: todas as versões disponíveis
- RDS para PostgreSQL 14: todas as versões disponíveis
- RDS para PostgreSQL 13: todas as versões disponíveis
- RDS para PostgreSQL 12: todas as versões disponíveis
- RDS para PostgreSQL 11: todas as versões disponíveis
- RDS para PostgreSQL 10: todas as versões disponíveis

Regiões e mecanismos de banco de dados compatíveis com a autenticação Kerberos no Amazon RDS

Usando a autenticação do Kerberos no Amazon RDS, você pode dar suporte à autenticação externa de usuários de banco de dados usando Kerberos e Microsoft Active Directory. Usar o Kerberos e o Active Directory oferece os benefícios do logon único e da autenticação centralizada dos usuários do banco de dados.

A autenticação de Kerberos não está disponível com os seguintes mecanismos:

- RDS para MariaDB

Embora a maioria das regiões da AWS esteja ativa por padrão para a conta da AWS, determinadas regiões são ativadas somente quando você as seleciona manualmente. Essas regiões são chamadas de regiões de adesão. Entretanto, as regiões que estão ativas por padrão, assim que sua conta da AWS é criada, são chamadas de regiões comerciais ou simplesmente de regiões. Em relação a regiões de adesão, é necessário usar uma entidade principal de serviço regionalizado do formulário `directoryservice.rds.region_name.amazonaws.com`. Por exemplo, para a África (Cidade do Cabo), é necessário adicionar a entidade principal de serviço `directoryservice.rds.region-af-south-1.amazonaws.com` à política de confiança. Para ter mais informações, consulte Autenticação de Kerberos.

Tópicos

- Autenticação Kerberos com RDS para Db2
- Autenticação de Kerberos com RDS para MySQL
- Autenticação de Kerberos com RDS para Oracle
- Autenticação de Kerberos com RDS para PostgreSQL
- Autenticação de Kerberos com RDS para SQL Server

Autenticação Kerberos com RDS para Db2

As regiões e as versões do mecanismo a seguir estão disponíveis para a autenticação Kerberos com o RDS para Db2.

Região	RDS para Db2 11.5
Leste dos EUA (Ohio)	Todas as versões
Leste dos EUA (Norte da Virgínia)	Todas as versões
Oeste dos EUA (N. da Califórnia)	Todas as versões
Oeste dos EUA (Oregon)	Todas as versões
África (Cidade do Cabo)	Indisponível
Ásia-Pacífico (Hong Kong)	Indisponível
Ásia-Pacífico (Hyderabad)	Indisponível

Região	RDS para Db2 11.5
Ásia-Pacífico (Jacarta)	Indisponível
Ásia-Pacífico (Melbourne)	Indisponível
Ásia-Pacífico (Mumbai)	Todas as versões
Ásia-Pacífico (Osaka)	Indisponível
Ásia-Pacífico (Seul)	Todas as versões
Ásia-Pacífico (Singapura)	Todas as versões
Ásia-Pacífico (Sydney)	Todas as versões
Ásia-Pacífico (Tóquio)	Todas as versões
Canadá (Central)	Todas as versões
Oeste do Canadá (Calgary)	Indisponível
China (Pequim)	Todas as versões
China (Ningxia)	Todas as versões
Europa (Frankfurt)	Todas as versões
Europa (Irlanda)	Todas as versões
Europa (Londres)	Todas as versões
Europa (Milão)	Indisponível
Europa (Paris)	Indisponível
Europa (Espanha)	Indisponível
Europa (Estocolmo)	Todas as versões
Europa (Zurique)	Indisponível

Região	RDS para Db2 11.5
Israel (Tel Aviv)	Indisponível
Oriente Médio (Barém)	Indisponível
Oriente Médio (Emirados Árabes Unidos)	Indisponível
América do Sul (São Paulo)	Todas as versões
AWS GovCloud (Leste dos EUA)	Indisponível
AWS GovCloud (Oeste dos EUA)	Indisponível

Autenticação de Kerberos com RDS para MySQL

As regiões e as versões do mecanismo a seguir estão disponíveis para a autenticação Kerberos com o RDS para MySQL.

Região	RDS para MySQL 8.0	RDS para MySQL 5.7	RDS para MySQL 5.6
Leste dos EUA (Ohio)	Todas as versões	Todas as versões	Todas as versões
Leste dos EUA (Norte da Virgínia)	Todas as versões	Todas as versões	Todas as versões
Oeste dos EUA (N. da Califórnia)	Todas as versões	Todas as versões	Todas as versões
Oeste dos EUA (Oregon)	Todas as versões	Todas as versões	Todas as versões
África (Cidade do Cabo)	Todas as versões	Todas as versões	Todas as versões
Ásia-Pacífico (Hong Kong)	Todas as versões	Todas as versões	Todas as versões

Região	RDS para MySQL 8.0	RDS para MySQL 5.7	RDS para MySQL 5.6
Ásia-Pacífico (Hyderabad)	Todas as versões	Todas as versões	Todas as versões
Ásia-Pacífico (Jacarta)	Todas as versões	Todas as versões	Todas as versões
Ásia-Pacífico (Melbourne)	Todas as versões	Todas as versões	Todas as versões
Ásia-Pacífico (Mumbai)	Todas as versões	Todas as versões	Todas as versões
Ásia-Pacífico (Osaka)	Não disponível	Não disponível	Não disponível
Ásia-Pacífico (Seul)	Todas as versões	Todas as versões	Todas as versões
Ásia-Pacífico (Singapura)	Todas as versões	Todas as versões	Todas as versões
Ásia-Pacífico (Sydney)	Todas as versões	Todas as versões	Todas as versões
Ásia-Pacífico (Tóquio)	Todas as versões	Todas as versões	Todas as versões
Canadá (Central)	Todas as versões	Todas as versões	Todas as versões
Oeste do Canadá (Calgary)	Não disponível	Não disponível	Não disponível
China (Pequim)	Todas as versões	Todas as versões	Todas as versões
China (Ningxia)	Todas as versões	Todas as versões	Todas as versões
Europa (Frankfurt)	Todas as versões	Todas as versões	Todas as versões
Europa (Irlanda)	Todas as versões	Todas as versões	Todas as versões
Europa (Londres)	Todas as versões	Todas as versões	Todas as versões

Região	RDS para MySQL 8.0	RDS para MySQL 5.7	RDS para MySQL 5.6
Europa (Milão)	Todas as versões	Todas as versões	Todas as versões
Europa (Paris)	Todas as versões	Todas as versões	Todas as versões
Europa (Espanha)	Todas as versões	Todas as versões	Todas as versões
Europa (Estocolmo)	Todas as versões	Todas as versões	Todas as versões
Europa (Zurique)	Todas as versões	Todas as versões	Todas as versões
Israel (Tel Aviv)	Todas as versões	Todas as versões	Todas as versões
Oriente Médio (Barém)	Todas as versões	Todas as versões	Todas as versões
Oriente Médio (Emirados Árabes Unidos)	Todas as versões	Todas as versões	Todas as versões
América do Sul (São Paulo)	Todas as versões	Todas as versões	Todas as versões
AWS GovCloud (Leste dos EUA)	Não disponível	Não disponível	Não disponível
AWS GovCloud (Oeste dos EUA)	Não disponível	Não disponível	Não disponível

Autenticação de Kerberos com RDS para Oracle

As regiões e as versões do mecanismo a seguir estão disponíveis para a autenticação Kerberos com o RDS para Oracle.

Região	RDS para Oracle 21c	RDS para Oracle 19c
Leste dos EUA (Ohio)	Todas as versões	Todas as versões

Região	RDS para Oracle 21c	RDS para Oracle 19c
Leste dos EUA (Norte da Virgínia)	Todas as versões	Todas as versões
Oeste dos EUA (N. da Califórnia)	Todas as versões	Todas as versões
Oeste dos EUA (Oregon)	Todas as versões	Todas as versões
África (Cidade do Cabo) (região de adesão)	Todas as versões	Todas as versões
Ásia-Pacífico (Hong Kong) (região de adesão)	Todas as versões	Todas as versões
Ásia-Pacífico (Hyderabad) (região de adesão)	Todas as versões	Todas as versões
Ásia-Pacífico (Jacarta) (região de adesão)	Todas as versões	Todas as versões
Ásia-Pacífico (Melbourne) (região de adesão)	Todas as versões	Todas as versões
Ásia-Pacífico (Mumbai)	Todas as versões	Todas as versões
Ásia-Pacífico (Osaka)	Não disponível	Não disponível
Ásia-Pacífico (Seul)	Todas as versões	Todas as versões
Ásia-Pacífico (Singapura)	Todas as versões	Todas as versões
Ásia-Pacífico (Sydney)	Todas as versões	Todas as versões
Ásia-Pacífico (Tóquio)	Todas as versões	Todas as versões
Canadá (Central)	Todas as versões	Todas as versões
Oeste do Canadá (Calgary)	Não disponível	Não disponível

Região	RDS para Oracle 21c	RDS para Oracle 19c
China (Pequim)	Não disponível	Não disponível
China (Ningxia)	Não disponível	Não disponível
Europa (Frankfurt)	Todas as versões	Todas as versões
Europa (Irlanda)	Todas as versões	Todas as versões
Europa (Londres)	Todas as versões	Todas as versões
Europa (Milão) (região de adesão)	Todas as versões	Todas as versões
Europa (Paris)	Não disponível	Não disponível
Europa (Espanha) (região de adesão)	Todas as versões	Todas as versões
Europa (Estocolmo)	Todas as versões	Todas as versões
Europa (Zurique) (região de adesão)	Todas as versões	Todas as versões
Israel (Tel Aviv) (região de adesão)	Todas as versões	Todas as versões
Oriente Médio (Bahrein) (região de adesão)	Todas as versões	Todas as versões
Oriente Médio (Emirados Árabes Unidos) (região de adesão)	Todas as versões	Todas as versões
América do Sul (São Paulo)	Todas as versões	Todas as versões
AWS GovCloud (Leste dos EUA)	Todas as versões	Todas as versões

Região	RDS para Oracle 21c	RDS para Oracle 19c
AWS GovCloud (Oeste dos EUA)	Todas as versões	Todas as versões

Autenticação de Kerberos com RDS para PostgreSQL

As regiões e as versões do mecanismo a seguir estão disponíveis para a autenticação Kerberos com o RDS para PostgreSQL.

Região	RDS para PostgreSQL 16	RDS para PostgreSQL 15	RDS para PostgreSQL 14	RDS para PostgreSQL 13	RDS para PostgreSQL 12	RDS para PostgreSQL 11	RDS para PostgreSQL 10
Leste dos EUA (Ohio)	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões
Leste dos EUA (Norte da Virgínia)	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões
Oeste dos EUA (N. da Califórnia)	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões
Oeste dos EUA (Oregon)	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões
África (Cidade do Cabo)	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões

Região	RDS para PostgreSQL 16	RDS para PostgreSQL 15	RDS para PostgreSQL 14	RDS para PostgreSQL 13	RDS para PostgreSQL 12	RDS para PostgreSQL 11	RDS para PostgreSQL 10
Ásia-Pacífico (Hong Kong)	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões
Ásia-Pacífico (Hyderabad)	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões
Ásia-Pacífico (Jacarta)	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões
Ásia-Pacífico (Melbourne)	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões
Ásia-Pacífico (Mumbai)	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões
Ásia-Pacífico (Osaka)	Não disponível	Não disponível	Não disponível	Não disponível	Não disponível	Não disponível	Não disponível
Ásia-Pacífico (Seul)	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões

Região	RDS para PostgreSQL 16	RDS para PostgreSQL 15	RDS para PostgreSQL 14	RDS para PostgreSQL 13	RDS para PostgreSQL 12	RDS para PostgreSQL 11	RDS para PostgreSQL 10
Ásia-Pacífico (Singapura)	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões
Ásia-Pacífico (Sydney)	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões
Ásia-Pacífico (Tóquio)	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões
Canadá (Central)	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões
Oeste do Canadá (Calgary)	Não disponível	Não disponível	Não disponível	Não disponível	Não disponível	Não disponível	Não disponível
China (Pequim)	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões
China (Ningxia)	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões
Europa (Frankfurt)	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões
Europa (Irlanda)	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões

Região	RDS para PostgreSQL 16	RDS para PostgreSQL 15	RDS para PostgreSQL 14	RDS para PostgreSQL 13	RDS para PostgreSQL 12	RDS para PostgreSQL 11	RDS para PostgreSQL 10
Europa (Londres)	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões
Europa (Milão)	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões
Europa (Paris)	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões
Europa (Espanha)	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões
Europa (Estocolmo)	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões
Europa (Zurique)	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões
Israel (Tel Aviv)	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões
Oriente Médio (Barém)	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões
Oriente Médio (Emirados Árabes Unidos)	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões

Região	RDS para PostgreSQL 16	RDS para PostgreSQL 15	RDS para PostgreSQL 14	RDS para PostgreSQL 13	RDS para PostgreSQL 12	RDS para PostgreSQL 11	RDS para PostgreSQL 10
América do Sul (São Paulo)	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões	Todas as versões
AWS GovCloud (Leste dos EUA)	Não disponível	Não disponível	Não disponível	Não disponível	Não disponível	Não disponível	Não disponível
AWS GovCloud (Oeste dos EUA)	Não disponível	Não disponível	Não disponível	Não disponível	Não disponível	Não disponível	Não disponível

Autenticação de Kerberos com RDS para SQL Server

As regiões e as versões do mecanismo a seguir estão disponíveis para a autenticação Kerberos com o RDS para SQL Server.

Região	RDS para SQL Server 2022	RDS para SQL Server 2019	RDS para SQL Server 2017	RDS para SQL Server 2016
Leste dos EUA (Ohio)	Todas as versões	Todas as versões	Todas as versões	Todas as versões
Leste dos EUA (Norte da Virgínia)	Todas as versões	Todas as versões	Todas as versões	Todas as versões
Oeste dos EUA (N. da Califórnia)	Todas as versões	Todas as versões	Todas as versões	Todas as versões

Região	RDS para SQL Server 2022	RDS para SQL Server 2019	RDS para SQL Server 2017	RDS para SQL Server 2016
Oeste dos EUA (Oregon)	Todas as versões	Todas as versões	Todas as versões	Todas as versões
África (Cidade do Cabo)	Todas as versões	Todas as versões	Todas as versões	Todas as versões
Ásia-Pacífico (Hong Kong)	Todas as versões	Todas as versões	Todas as versões	Todas as versões
Ásia-Pacífico (Hyderabad)	Todas as versões	Todas as versões	Todas as versões	Todas as versões
Ásia-Pacífico (Jacarta)	Todas as versões	Todas as versões	Todas as versões	Todas as versões
Ásia-Pacífico (Melbourne)	Todas as versões	Todas as versões	Todas as versões	Todas as versões
Ásia-Pacífico (Mumbai)	Todas as versões	Todas as versões	Todas as versões	Todas as versões
Ásia-Pacífico (Osaka)	Todas as versões	Todas as versões	Todas as versões	Todas as versões
Ásia-Pacífico (Seul)	Todas as versões	Todas as versões	Todas as versões	Todas as versões
Ásia-Pacífico (Singapura)	Todas as versões	Todas as versões	Todas as versões	Todas as versões
Ásia-Pacífico (Sydney)	Todas as versões	Todas as versões	Todas as versões	Todas as versões
Ásia-Pacífico (Tóquio)	Todas as versões	Todas as versões	Todas as versões	Todas as versões

Região	RDS para SQL Server 2022	RDS para SQL Server 2019	RDS para SQL Server 2017	RDS para SQL Server 2016
Canadá (Central)	Todas as versões	Todas as versões	Todas as versões	Todas as versões
Oeste do Canadá (Calgary)	Não disponível	Não disponível	Não disponível	Não disponível
China (Pequim)	Todas as versões	Todas as versões	Todas as versões	Todas as versões
China (Ningxia)	Todas as versões	Todas as versões	Todas as versões	Todas as versões
Europa (Frankfurt)	Todas as versões	Todas as versões	Todas as versões	Todas as versões
Europa (Irlanda)	Todas as versões	Todas as versões	Todas as versões	Todas as versões
Europa (Londres)	Todas as versões	Todas as versões	Todas as versões	Todas as versões
Europa (Milão)	Todas as versões	Todas as versões	Todas as versões	Todas as versões
Europa (Paris)	Todas as versões	Todas as versões	Todas as versões	Todas as versões
Europa (Espanha)	Todas as versões	Todas as versões	Todas as versões	Todas as versões
Europa (Estocolmo)	Todas as versões	Todas as versões	Todas as versões	Todas as versões
Europa (Zurique)	Todas as versões	Todas as versões	Todas as versões	Todas as versões

Região	RDS para SQL Server 2022	RDS para SQL Server 2019	RDS para SQL Server 2017	RDS para SQL Server 2016
Israel (Tel Aviv)	Não disponível	Não disponível	Não disponível	Não disponível
Oriente Médio (Barém)	Todas as versões	Todas as versões	Todas as versões	Todas as versões
Oriente Médio (Emirados Árabes Unidos)	Todas as versões	Todas as versões	Todas as versões	Todas as versões
América do Sul (São Paulo)	Todas as versões	Todas as versões	Todas as versões	Todas as versões
AWS GovCloud (Leste dos EUA)	Todas as versões	Todas as versões	Todas as versões	Todas as versões
AWS GovCloud (Oeste dos EUA)	Todas as versões	Todas as versões	Todas as versões	Todas as versões

Regiões e mecanismos de banco de dados compatíveis com clusters de banco de dados multi-AZ no Amazon RDS

Uma implantação de clusters de banco de dados multi-AZ é um modo de implantação de alta disponibilidade do Amazon RDS com duas instâncias de banco de dados em espera legíveis. Um cluster de banco de dados multi-AZ tem uma instância de banco de dados do gravador e duas instâncias de banco de dados do leitor em três zonas de disponibilidade diferentes na mesma região. Clusters de banco de dados multi-AZ oferecem alta disponibilidade, maior capacidade para workloads de leitura e menor latência do gravação quando comparados com implantação de instância de banco de dados multi-AZ. Para ter mais informações, consulte Implantações de cluster de banco de dados multi-AZ para o Amazon RDS.

Clusters de banco de dados multi-AZ não estão disponíveis com os seguintes mecanismos:

- RDS para Db2
- RDS para MariaDB

- RDS para Oracle
- RDS para SQL Server

Tópicos

- Clusters de banco de dados multi-AZ com RDS para MySQL
- Clusters de banco de dados multi-AZ com RDS para PostgreSQL

Clusters de banco de dados multi-AZ com RDS para MySQL

As regiões e as versões do mecanismo a seguir estão disponíveis para clusters de banco de dados multi-AZ com o RDS para MySQL.

Região	RDS para MySQL 8.4	RDS para MySQL 8.0
Leste dos EUA (Norte da Virgínia)	Todas as versões disponíveis	Todas as versões disponíveis
Leste dos EUA (Ohio)	Todas as versões disponíveis	Todas as versões disponíveis
Oeste dos EUA (Norte da Califórnia)	Não disponível	Não disponível
Oeste dos EUA (Oregon)	Todas as versões disponíveis	Todas as versões disponíveis
África (Cidade do Cabo)	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Hong Kong)	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Hyderabad)	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Jacarta)	Todas as versões disponíveis	Todas as versões disponíveis

Região	RDS para MySQL 8.4	RDS para MySQL 8.0
Ásia-Pacífico (Malásia)	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Melbourne)	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Mumbai)	Todas as versões disponíveis	Todas as versões disponíveis
Ásia Pacifico (Osaka)	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Seul)	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Singapura)	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Sydney)	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Tóquio)	Todas as versões disponíveis	Todas as versões disponíveis
Canadá (Central)	Todas as versões disponíveis	Todas as versões disponíveis
Canadá (Central)	Todas as versões disponíveis	Todas as versões disponíveis
Oeste do Canadá (Calgary)	Todas as versões disponíveis	Todas as versões disponíveis
China (Pequim)	Todas as versões disponíveis	Todas as versões disponíveis
China (Ningxia)	Todas as versões disponíveis	Todas as versões disponíveis

Região	RDS para MySQL 8.4	RDS para MySQL 8.0
Europa (Frankfurt)	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Irlanda)	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Londres)	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Milão)	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Paris)	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Espanha)	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Estocolmo)	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Zurique)	Todas as versões disponíveis	Todas as versões disponíveis
Israel (Tel Aviv)	Todas as versões disponíveis	Todas as versões disponíveis
Oriente Médio (Bahrein)	Todas as versões disponíveis	Todas as versões disponíveis
Oriente Médio (Emirados Árabes Unidos)	Todas as versões disponíveis	Todas as versões disponíveis
América do Sul (São Paulo)	Todas as versões disponíveis	Todas as versões disponíveis

Região	RDS para MySQL 8.4	RDS para MySQL 8.0
AWS GovCloud (Leste dos EUA)	Não disponível	Não disponível
AWS GovCloud (Oeste dos EUA)	Não disponível	Não disponível

É possível listar as versões disponíveis em uma região para determinada classe de instância de banco de dados usando a AWS CLI. Altere a classe da instância de banco de dados para mostrar as versões de mecanismo disponíveis para ela.

Para Linux, macOS ou Unix:

```
aws rds describe-orderable-db-instance-options \
--engine mysql \
--db-instance-class db.r5d.large \
--query '*[]|[?SupportsClusters == `true`].[EngineVersion]'  \
--output text
```

Para Windows:

```
aws rds describe-orderable-db-instance-options ^
--engine mysql ^
--db-instance-class db.r5d.large ^
--query "*[]|[?SupportsClusters == `true`].[EngineVersion]"  ^
--output text
```

Clusters de banco de dados multi-AZ com RDS para PostgreSQL

As regiões e as versões do mecanismo a seguir estão disponíveis para clusters de banco de dados multi-AZ para o RDS para PostgreSQL.

Região	RDS para PostgreSQL 17	RDS para PostgreSQL 16	RDS para PostgreSQL 15	RDS para PostgreSQL 14	RDS para PostgreSQL 13
Leste dos EUA (Norte da Virgínia)	Todas as versões 17 do PostgreSQL	Todas as versões 16 do PostgreSQL	Todas as versões do PostgreSQL 15	Versão 14.5 e posterior	Versões 13.4 e 13.7 e posteriores
Leste dos EUA (Ohio)	Todas as versões 17 do PostgreSQL	Todas as versões 16 do PostgreSQL	Todas as versões do PostgreSQL 15	Versão 14.5 e posterior	Versões 13.4 e 13.7 e posteriores
Oeste dos EUA (Norte da Califórnia)	Não disponível	Não disponível	Não disponível	Não disponível	Não disponível
Oeste dos EUA (Oregon)	Todas as versões 17 do PostgreSQL	Todas as versões 16 do PostgreSQL	Todas as versões do PostgreSQL 15	Versão 14.5 e posterior	Versões 13.4 e 13.7 e posteriores
África (Cidade do Cabo)	Todas as versões 17 do PostgreSQL	Todas as versões 16 do PostgreSQL	Todas as versões do PostgreSQL 15	Versão 14.5 e posterior	Versões 13.4 e 13.7 e posteriores
Ásia-Pacífico (Hong Kong)	Todas as versões 17 do PostgreSQL	Todas as versões 16 do PostgreSQL	Todas as versões do PostgreSQL 15	Versão 14.5 e posterior	Versões 13.4 e 13.7 e posteriores
Ásia-Pacífico (Hyderabad)	Todas as versões 17 do PostgreSQL	Todas as versões 16 do PostgreSQL	Todas as versões do PostgreSQL 15	Versão 14.5 e posterior	Versões 13.4 e 13.7 e posteriores

Região	RDS para PostgreSQL 17	RDS para PostgreSQL 16	RDS para PostgreSQL 15	RDS para PostgreSQL 14	RDS para PostgreSQL 13
Ásia-Pacífico (Jacarta)	Todas as versões 17 do PostgreSQL	Todas as versões 16 do PostgreSQL	Todas as versões do PostgreSQL 15	Versão 14.5 e posterior	Versões 13.4 e 13.7 e posteriores
Ásia-Pacífico (Malásia)	Todas as versões 17 do PostgreSQL	Todas as versões 16 do PostgreSQL	Todas as versões do PostgreSQL 15	Versão 14.5 e posterior	Versões 13.4 e 13.7 e posteriores
Ásia-Pacífico (Melbourne)	Todas as versões 17 do PostgreSQL	Todas as versões 16 do PostgreSQL	Todas as versões do PostgreSQL 15	Versão 14.5 e posterior	Versões 13.4 e 13.7 e posteriores
Ásia-Pacífico (Mumbai)	Todas as versões 17 do PostgreSQL	Todas as versões 16 do PostgreSQL	Todas as versões do PostgreSQL 15	Versão 14.5 e posterior	Versões 13.4 e 13.7 e posteriores
Ásia Pacifico (Osaka)	Todas as versões 17 do PostgreSQL	Todas as versões 16 do PostgreSQL	Todas as versões do PostgreSQL 15	Versão 14.5 e posterior	Versões 13.4 e 13.7 e posteriores
Ásia-Pacífico (Seul)	Todas as versões 17 do PostgreSQL	Todas as versões 16 do PostgreSQL	Todas as versões do PostgreSQL 15	Versão 14.5 e posterior	Versões 13.4 e 13.7 e posteriores
Ásia-Pacífico (Singapura)	Todas as versões 17 do PostgreSQL	Todas as versões 16 do PostgreSQL	Todas as versões do PostgreSQL 15	Versão 14.5 e posterior	Versões 13.4 e 13.7 e posteriores

Região	RDS para PostgreSQL 17	RDS para PostgreSQL 16	RDS para PostgreSQL 15	RDS para PostgreSQL 14	RDS para PostgreSQL 13
Ásia-Pacífico (Sydney)	Todas as versões 17 do PostgreSQL	Todas as versões 16 do PostgreSQL	Todas as versões do PostgreSQL 15	Versão 14.5 e posterior	Versões 13.4 e 13.7 e posteriores
Ásia-Pacífico (Tóquio)	Todas as versões 17 do PostgreSQL	Todas as versões 16 do PostgreSQL	Todas as versões do PostgreSQL 15	Versão 14.5 e posterior	Versões 13.4 e 13.7 e posteriores
Canadá (Central)	Todas as versões 17 do PostgreSQL	Todas as versões 16 do PostgreSQL	Todas as versões do PostgreSQL 15	Versão 14.5 e posterior	Versões 13.4 e 13.7 e posteriores
Oeste do Canadá (Calgary)	Todas as versões 17 do PostgreSQL	Todas as versões 16 do PostgreSQL	Todas as versões do PostgreSQL 15	Versão 14.5 e posterior	Versões 13.4 e 13.7 e posteriores
China (Pequim)	Todas as versões 17 do PostgreSQL	Todas as versões 16 do PostgreSQL	Todas as versões do PostgreSQL 15	Versão 14.5 e posterior	Versões 13.4 e 13.7 e posteriores
China (Ningxia)	Todas as versões 17 do PostgreSQL	Todas as versões 16 do PostgreSQL	Todas as versões do PostgreSQL 15	Versão 14.5 e posterior	Versões 13.4 e 13.7 e posteriores
Europa (Frankfurt)	Todas as versões 17 do PostgreSQL	Todas as versões 16 do PostgreSQL	Todas as versões do PostgreSQL 15	Versão 14.5 e posterior	Versões 13.4 e 13.7 e posteriores

Região	RDS para PostgreSQL 17	RDS para PostgreSQL 16	RDS para PostgreSQL 15	RDS para PostgreSQL 14	RDS para PostgreSQL 13
Europa (Irlanda)	Todas as versões 17 do PostgreSQL	Todas as versões 16 do PostgreSQL	Todas as versões do PostgreSQL 15	Versão 14.5 e posterior	Versões 13.4 e 13.7 e posteriores
Europa (Londres)	Todas as versões 17 do PostgreSQL	Todas as versões 16 do PostgreSQL	Todas as versões do PostgreSQL 15	Versão 14.5 e posterior	Versões 13.4 e 13.7 e posteriores
Europa (Milão)	Todas as versões 17 do PostgreSQL	Todas as versões 16 do PostgreSQL	Todas as versões do PostgreSQL 15	Versão 14.5 e posterior	Versões 13.4 e 13.7 e posteriores
Europa (Paris)	Todas as versões 17 do PostgreSQL	Todas as versões 16 do PostgreSQL	Todas as versões do PostgreSQL 15	Versão 14.5 e posterior	Versões 13.4 e 13.7 e posteriores
Europa (Espanha)	Todas as versões 17 do PostgreSQL	Todas as versões 16 do PostgreSQL	Todas as versões do PostgreSQL 15	Versão 14.5 e posterior	Versões 13.4 e 13.7 e posteriores
Europa (Estocolmo)	Todas as versões 17 do PostgreSQL	Todas as versões 16 do PostgreSQL	Todas as versões do PostgreSQL 15	Versão 14.5 e posterior	Versões 13.4 e 13.7 e posteriores
Europa (Zurique)	Todas as versões 17 do PostgreSQL	Todas as versões 16 do PostgreSQL	Todas as versões do PostgreSQL 15	Versão 14.5 e posterior	Versões 13.4 e 13.7 e posteriores

Região	RDS para PostgreSQL 17	RDS para PostgreSQL 16	RDS para PostgreSQL 15	RDS para PostgreSQL 14	RDS para PostgreSQL 13
Israel (Tel Aviv)	Todas as versões 17 do PostgreSQL	Todas as versões 16 do PostgreSQL	Todas as versões do PostgreSQL 15	Versão 14.5 e posterior	Versões 13.4 e 13.7 e posteriores
Oriente Médio (Bahrein)	Todas as versões 17 do PostgreSQL	Todas as versões 16 do PostgreSQL	Todas as versões do PostgreSQL 15	Versão 14.5 e posterior	Versões 13.4 e 13.7 e posteriores
Oriente Médio (Emirados Árabes Unidos)	Todas as versões 17 do PostgreSQL	Todas as versões 16 do PostgreSQL	Todas as versões do PostgreSQL 15	Versão 14.5 e posterior	Versões 13.4 e 13.7 e posteriores
América do Sul (São Paulo)	Todas as versões 17 do PostgreSQL	Todas as versões 16 do PostgreSQL	Todas as versões do PostgreSQL 15	Versão 14.5 e posterior	Versões 13.4 e 13.7 e posteriores
AWS GovCloud (Leste dos EUA)	Não disponível	Não disponível	Não disponível	Não disponível	Não disponível
AWS GovCloud (Oeste dos EUA)	Não disponível	Não disponível	Não disponível	Não disponível	Não disponível

É possível listar as versões disponíveis em uma região para determinada classe de instância de banco de dados usando a AWS CLI. Altere a classe da instância de banco de dados para mostrar as versões de mecanismo disponíveis para ela.

Para Linux, macOS ou Unix:

```
aws rds describe-orderable-db-instance-options \
--engine postgres \
--db-instance-class db.r5d.large \
--query '*[]|[?SupportsClusters == `true`].[EngineVersion]'  \
--output text
```

Para Windows:

```
aws rds describe-orderable-db-instance-options ^
--engine postgres ^
--db-instance-class db.r5d.large ^
--query "*[]|[?SupportsClusters == `true`].[EngineVersion]"  ^
--output text
```

Regiões e mecanismos de banco de dados compatíveis com o Insights de Performance no Amazon RDS

Important

A AWS anunciou a data de fim da vida útil do Insights de Performance: 30 de novembro de 2025. Após essa data, o Amazon RDS não oferecerá mais suporte à experiência do console, aos períodos de retenção flexíveis (de 1 a 24 meses) e aos preços associados do Insights de Performance. A API do Insights de Performance continuará existindo sem alterações de preço. Os custos da API do Insights de Performance aparecerão em sua fatura da com o custo do CloudWatch Database Insights.

Recomendamos que você atualize quaisquer instâncias de banco de dados usando o nível pago do Insights de Performance para o modo avançado do Database Insights antes de 30 de novembro de 2025. Para ter informações sobre como atualizar para o modo avançado do Database Insights, consulte Ativação do modo Avançado do Database Insights para Amazon RDS.

Se você não tomar nenhuma medida, as instâncias de banco de dados que estão usando o Insights de Performance usarão como padrão o modo padrão do Database Insights. Com o modo padrão do Database Insights, você pode perder o acesso ao histórico de dados

de desempenho após sete dias e talvez não consiga usar planos de execução e recursos de análise sob demanda no console do Amazon RDS. Após 30 de novembro de 2025, somente o modo avançado do Database Insights permitirá planos de execução e análise sob demanda.
Com o CloudWatch Database Insights, você pode monitorar a carga do banco de dados para sua frota de bancos de dados e analisar e solucionar problemas de desempenho em grande escala. Para ter mais informações sobre o Database Insights, consulte Monitoramento de bancos de dados do Amazon RDS com o CloudWatch Database Insights. Para obter informações de definição de preço, consulte Definição de preço do Amazon CloudWatch.

O Performance Insights expande os recursos de monitoramento do Amazon RDS existentes para ilustrar e ajudar você a analisar a performance do banco de dados. Com o painel do Performance Insights, você pode visualizar a carga do banco de dados em sua instância de banco de dados do Amazon RDS. Você também pode filtrá-la por esperas, instruções SQL, hosts ou usuários. Para obter mais informações, consulte Monitorar a carga de banco de dados com o Performance Insights no Amazon RDS.

O Performance Insights está disponível para todos os mecanismos de banco de dados do RDS, exceto o RDS para Db2.

Para os mecanismos de banco de dados disponíveis, o Performance Insights está disponível com todas as versões de mecanismos disponíveis e em todas as Regiões da AWS.

Para receber informações sobre a compatibilidade da região, do mecanismo de banco de dados e da classe de instância com os recursos do Insights de Performance, consulte O mecanismo de banco de dados do Amazon RDS, a região e a classe de instância são compatíveis com atributos do Insights de Performance..

Regiões e mecanismos de banco de dados compatíveis com o RDS Custom

O Amazon RDS Custom automatiza tarefas e operações de administração do banco de dados. Usando o RDS Custom, como administrador de banco de dados, você pode acessar e personalizar seu ambiente de banco de dados e sistema operacional. Com o RDS Custom, você pode personalizar para atender aos requisitos de aplicações herdadas, personalizadas e empacotadas. Para obter mais informações, consulte Amazon RDS Custom.

O RDS Custom é compatível somente com os seguintes mecanismos de banco de dados:

Tópicos

- Regiões e mecanismos de banco de dados compatíveis com o RDS Custom para Oracle
- Regiões e mecanismos de banco de dados compatíveis com o RDS Custom para SQL Server

Regiões e mecanismos de banco de dados compatíveis com o RDS Custom para Oracle

As regiões e as versões do mecanismo a seguir estão disponíveis para o RDS Custom para Oracle.

Região	Oracle Database 19c	Oracle Database 18c	Banco de Dados Oracle 12c
Leste dos EUA (Norte da Virgínia)	19c com a RU/RUR de janeiro de 2021 ou posterior	18c com a RU/RUR de janeiro de 2021 ou posterior	12.1 e 12.2 com a RU/RUR de janeiro de 2021 ou posterior
Leste dos EUA (Ohio)	19c com a RU/RUR de janeiro de 2021 ou posterior	18c com a RU/RUR de janeiro de 2021 ou posterior	12.1 e 12.2 com a RU/RUR de janeiro de 2021 ou posterior
Oeste dos EUA (Norte da Califórnia)	Não disponível	Não disponível	Não disponível
Oeste dos EUA (Oregon)	19c com a RU/RUR de janeiro de 2021 ou posterior	18c com a RU/RUR de janeiro de 2021 ou posterior	12.1 e 12.2 com a RU/RUR de janeiro de 2021 ou posterior
África (Cidade do Cabo)	Não disponível	Não disponível	Não disponível
Ásia-Pacífico (Hong Kong)	Não disponível	Não disponível	Não disponível
Ásia-Pacífico (Jacarta)	19c com a RU/RUR de janeiro de 2021 ou posterior	18c com a RU/RUR de janeiro de 2021 ou posterior	12.1 e 12.2 com a RU/RUR de janeiro de 2021 ou posterior

Região	Oracle Database 19c	Oracle Database 18c	Banco de Dados Oracle 12c
Ásia-Pacífico (Malásia)	Não disponível	Não disponível	Não disponível
Ásia-Pacífico (Melbourne)	Não disponível	Não disponível	Não disponível
Ásia-Pacífico (Mumbai)	19c com a RU/RUR de janeiro de 2021 ou posterior	18c com a RU/RUR de janeiro de 2021 ou posterior	12.1 e 12.2 com a RU/RUR de janeiro de 2021 ou posterior
Ásia-Pacífico (Osaka)	19c com a RU/RUR de janeiro de 2021 ou posterior	18c com a RU/RUR de janeiro de 2021 ou posterior	12.1 e 12.2 com a RU/RUR de janeiro de 2021 ou posterior
Ásia-Pacífico (Seul)	19c com a RU/RUR de janeiro de 2021 ou posterior	18c com a RU/RUR de janeiro de 2021 ou posterior	12.1 e 12.2 com a RU/RUR de janeiro de 2021 ou posterior
Ásia-Pacífico (Singapura)	19c com a RU/RUR de janeiro de 2021 ou posterior	18c com a RU/RUR de janeiro de 2021 ou posterior	12.1 e 12.2 com a RU/RUR de janeiro de 2021 ou posterior
Ásia-Pacífico (Sydney)	19c com a RU/RUR de janeiro de 2021 ou posterior	18c com a RU/RUR de janeiro de 2021 ou posterior	12.1 e 12.2 com a RU/RUR de janeiro de 2021 ou posterior
Ásia-Pacífico (Tóquio)	19c com a RU/RUR de janeiro de 2021 ou posterior	18c com a RU/RUR de janeiro de 2021 ou posterior	12.1 e 12.2 com a RU/RUR de janeiro de 2021 ou posterior
Canadá (Central)	19c com a RU/RUR de janeiro de 2021 ou posterior	18c com a RU/RUR de janeiro de 2021 ou posterior	12.1 e 12.2 com a RU/RUR de janeiro de 2021 ou posterior
Oeste do Canadá (Calgary)	Não disponível	Não disponível	Não disponível

Região	Oracle Database 19c	Oracle Database 18c	Banco de Dados Oracle 12c
China (Pequim)	Não disponível	Não disponível	Não disponível
China (Ningxia)	Não disponível	Não disponível	Não disponível
Europa (Frankfurt)	19c com a RU/RUR de janeiro de 2021 ou posterior	18c com a RU/RUR de janeiro de 2021 ou posterior	12.1 e 12.2 com a RU/RUR de janeiro de 2021 ou posterior
Europa (Irlanda)	19c com a RU/RUR de janeiro de 2021 ou posterior	18c com a RU/RUR de janeiro de 2021 ou posterior	12.1 e 12.2 com a RU/RUR de janeiro de 2021 ou posterior
Europa (Londres)	19c com a RU/RUR de janeiro de 2021 ou posterior	18c com a RU/RUR de janeiro de 2021 ou posterior	12.1 e 12.2 com a RU/RUR de janeiro de 2021 ou posterior
Europa (Milão)	19c com a RU/RUR de janeiro de 2021 ou posterior	18c com a RU/RUR de janeiro de 2021 ou posterior	12.1 e 12.2 com a RU/RUR de janeiro de 2021 ou posterior
Europa (Paris)	19c com a RU/RUR de janeiro de 2021 ou posterior	18c com a RU/RUR de janeiro de 2021 ou posterior	12.1 e 12.2 com a RU/RUR de janeiro de 2021 ou posterior
Europa (Estocolmo)	19c com a RU/RUR de janeiro de 2021 ou posterior	18c com a RU/RUR de janeiro de 2021 ou posterior	12.1 e 12.2 com a RU/RUR de janeiro de 2021 ou posterior
Israel (Tel Aviv)	Não disponível	Não disponível	Não disponível
Oriente Médio (Bahrein)	Não disponível	Não disponível	Não disponível
Oriente Médio (Emirados Árabes Unidos)	19c com a RU/RUR de janeiro de 2021 ou posterior	18c com a RU/RUR de janeiro de 2021 ou posterior	12.1 e 12.2 com a RU/RUR de janeiro de 2021 ou posterior

Região	Oracle Database 19c	Oracle Database 18c	Banco de Dados Oracle 12c
América do Sul (São Paulo)	19c com a RU/RUR de janeiro de 2021 ou posterior	18c com a RU/RUR de janeiro de 2021 ou posterior	12.1 e 12.2 com a RU/RUR de janeiro de 2021 ou posterior
AWS GovCloud (Leste dos EUA)	19c com a RU/RUR de janeiro de 2021 ou posterior	18c com a RU/RUR de janeiro de 2021 ou posterior	12.1 e 12.2 com a RU/RUR de janeiro de 2021 ou posterior
AWS GovCloud (Oeste dos EUA)	19c com a RU/RUR de janeiro de 2021 ou posterior	18c com a RU/RUR de janeiro de 2021 ou posterior	12.1 e 12.2 com a RU/RUR de janeiro de 2021 ou posterior

Regiões e mecanismos de banco de dados compatíveis com o RDS Custom para SQL Server

Você pode implantar o RDS Custom para SQL Server usando uma versão de mecanismo fornecida pelo RDS (RPEV) ou uma versão de mecanismo personalizada (CEV):

- Se você usar uma RPEV, ela incluirá a instalação da imagem de máquina da Amazon (AMI) padrão e do SQL Server. Se você personalizar ou modificar o sistema operacional (SO), suas alterações poderão não persistir durante a aplicação de patches, a restauração de snapshots ou a recuperação automática.
- Se você usar uma CEV, escolha sua própria AMI com o Microsoft SQL Server pré-instalado ou com um SQL Server que você instalar usando mídia própria. Ao usar uma CEV fornecida pela AWS, escolha a imagem (AMI) mais recente do Amazon EC2 disponibilizada pela AWS, que tenha a atualização cumulativa (CU) compatível com o RDS Custom para SQL Server. Com uma CEV, você pode personalizar a configuração do SO e do SQL Server para atender às necessidades da sua empresa.

As Regiões da AWS e as versões de mecanismo a seguir estão disponíveis para o RDS Custom para SQL Server. O suporte a versões de mecanismo depende do fato de você estar usando o RDS Custom para SQL Server com uma RPEV, CEV fornecida pela AWS ou CEV fornecida pelo cliente.

Região	RPEV	CEV fornecida pela AWS	CEV fornecida pelo cliente
Leste dos EUA (Norte da Virgínia)	SQL Server 2022 Enterprise, Standard ou Web, com CU9, CU13, CU14-GDR, CU15-GDR, CU16, CU17, CU18 e CU19. SQL Server 2019 Enterprise, Standard ou Web, com CU8, CU17, CU18, CU20, CU24, CU26, CU28-GDR, CU29-GDR, CU30 e CU32.	SQL Server 2022 Enterprise, Standard ou Web, com CU9, CU13, CU14-GDR, CU15-GDR, CU16, CU17, CU18 e CU19. SQL Server 2019 Enterprise, Standard ou Web, com CU8, CU17, CU18, CU20, CU24, CU26, CU28-GDR, CU29-GDR, CU30 e CU32.	SQL Server 2022 Enterprise, Standard ou Web, com CU9, CU13, CU14-GDR, CU15-GDR, CU16, CU17, CU18 e CU19. SQL Server 2019 Enterprise, Standard ou Web, com CU8, CU17, CU18, CU20, CU24, CU26, CU28-GDR, CU29-GDR, CU30 e CU32.
Leste dos EUA (Ohio)	SQL Server 2022 Enterprise, Standard ou Web, com CU9, CU13, CU14-GDR, CU15-GDR, CU16, CU17, CU18 e CU19. SQL Server 2019 Enterprise, Standard ou Web, com CU8, CU17, CU18, CU20, CU24, CU26, CU28-GDR, CU29-GDR, CU30 e CU32.	SQL Server 2022 Enterprise, Standard ou Web, com CU9, CU13, CU14-GDR, CU15-GDR, CU16, CU17, CU18 e CU19. SQL Server 2019 Enterprise, Standard ou Web, com CU8, CU17, CU18, CU20, CU24, CU26, CU28-GDR, CU29-GDR, CU30 e CU32.	SQL Server 2022 Enterprise, Standard ou Web, com CU9, CU13, CU14-GDR, CU15-GDR, CU16, CU17, CU18 e CU19. SQL Server 2019 Enterprise, Standard ou Web, com CU8, CU17, CU18, CU20, CU24, CU26, CU28-GDR, CU29-GDR, CU30 e CU32.
Oeste dos EUA (Norte da Califórnia)	SQL Server 2022 Enterprise, Standard ou Web, com CU9, CU13, CU14-GDR, CU15-GDR, CU16, CU17, CU18 e CU19.	SQL Server 2022 Enterprise, Standard ou Web, com CU9, CU13, CU14-GDR, CU15-GDR, CU16, CU17, CU18 e CU19.	SQL Server 2022 Enterprise, Standard ou Web, com CU9, CU13, CU14-GDR, CU15-GDR, CU16, CU17, CU18 e CU19.

Região	RPEV	CEV fornecida pela AWS	CEV fornecida pelo cliente
	SQL Server 2019 Enterprise, Standard ou Web, com CU8, CU17, CU18, CU20, CU24, CU26, CU28-GDR, CU29-GDR, CU30 e CU32.	SQL Server 2019 Enterprise, Standard ou Web, com CU8, CU17, CU18, CU20, CU24, CU26, CU28-GDR, CU29-GDR, CU30 e CU32.	SQL Server 2019 Enterprise, Standard ou Web, com CU8, CU17, CU18, CU20, CU24, CU26, CU28-GDR, CU29-GDR, CU30 e CU32.
Oeste dos EUA (Oregon)	SQL Server 2022 Enterprise, Standard ou Web, com CU9, CU13, CU14-GDR, CU15-GDR, CU16, CU17, CU18 e CU19. SQL Server 2019 Enterprise, Standard ou Web, com CU8, CU17, CU18, CU20, CU24, CU26, CU28-GDR, CU29-GDR, CU30 e CU32.	SQL Server 2022 Enterprise, Standard ou Web, com CU9, CU13, CU14-GDR, CU15-GDR, CU16, CU17, CU18 e CU19. SQL Server 2019 Enterprise, Standard ou Web, com CU8, CU17, CU18, CU20, CU24, CU26, CU28-GDR, CU29-GDR, CU30 e CU32.	SQL Server 2022 Enterprise, Standard ou Web, com CU9, CU13, CU14-GDR, CU15-GDR, CU16, CU17, CU18 e CU19. SQL Server 2019 Enterprise, Standard ou Web, com CU8, CU17, CU18, CU20, CU24, CU26, CU28-GDR, CU29-GDR, CU30 e CU32.

Região	RPEV	CEV fornecida pela AWS	CEV fornecida pelo cliente
África (Cidade do Cabo)	SQL Server 2022 Enterprise, Standard ou Web, com CU9, CU13, CU14-GDR, CU15-GDR, CU16, CU17, CU18 e CU19. SQL Server 2019 Enterprise, Standard ou Web, com CU8, CU17, CU18, CU20, CU24, CU26, CU28-GDR, CU29-GDR, CU30 e CU32.	SQL Server 2022 Enterprise, Standard ou Web, com CU9, CU13, CU14-GDR, CU15-GDR, CU16, CU17, CU18 e CU19. SQL Server 2019 Enterprise, Standard ou Web, com CU8, CU17, CU18, CU20, CU24, CU26, CU28-GDR, CU29-GDR, CU30 e CU32.	SQL Server 2022 Enterprise, Standard ou Web, com CU9, CU13, CU14-GDR, CU15-GDR, CU16, CU17, CU18 e CU19. SQL Server 2019 Enterprise, Standard ou Web, com CU8, CU17, CU18, CU20, CU24, CU26, CU28-GDR, CU29-GDR, CU30 e CU32.
Ásia-Pacífico (Hong Kong)	Não disponível	Não disponível	Não disponível
Ásia-Pacífico (Hyderabad)	Não disponível	Não disponível	Não disponível
Ásia-Pacífico (Jacarta)	Não disponível	Não disponível	Não disponível
Ásia-Pacífico (Malásia)	Não disponível	Não disponível	Não disponível
Ásia-Pacífico (Melbourne)	Não disponível	Não disponível	Não disponível

Região	RPEV	CEV fornecida pela AWS	CEV fornecida pelo cliente
Ásia-Pacífico (Mumbai)	SQL Server 2022 Enterprise, Standard ou Web, com CU9, CU13, CU14-GDR, CU15-GDR, CU16, CU17, CU18 e CU19. SQL Server 2019 Enterprise, Standard ou Web, com CU8, CU17, CU18, CU20, CU24, CU26, CU28-GDR, CU29-GDR, CU30 e CU32.	SQL Server 2022 Enterprise, Standard ou Web, com CU9, CU13, CU14-GDR, CU15-GDR, CU16, CU17, CU18 e CU19. SQL Server 2019 Enterprise, Standard ou Web, com CU8, CU17, CU18, CU20, CU24, CU26, CU28-GDR, CU29-GDR, CU30 e CU32.	SQL Server 2022 Enterprise, Standard ou Web, com CU9, CU13, CU14-GDR, CU15-GDR, CU16, CU17, CU18 e CU19. SQL Server 2019 Enterprise, Standard ou Web, com CU8, CU17, CU18, CU20, CU24, CU26, CU28-GDR, CU29-GDR, CU30 e CU32.
Ásia-Pacífico (Osaka)	SQL Server 2022 Enterprise, Standard ou Web, com CU9, CU13, CU14-GDR, CU15-GDR, CU16, CU17, CU18 e CU19. SQL Server 2019 Enterprise, Standard ou Web, com CU8, CU17, CU18, CU20, CU24, CU26, CU28-GDR, CU29-GDR, CU30 e CU32.	SQL Server 2022 Enterprise, Standard ou Web, com CU9, CU13, CU14-GDR, CU15-GDR, CU16, CU17, CU18 e CU19. SQL Server 2019 Enterprise, Standard ou Web, com CU8, CU17, CU18, CU20, CU24, CU26, CU28-GDR, CU29-GDR, CU30 e CU32.	SQL Server 2022 Enterprise, Standard ou Web, com CU9, CU13, CU14-GDR, CU15-GDR, CU16, CU17, CU18 e CU19. SQL Server 2019 Enterprise, Standard ou Web, com CU8, CU17, CU18, CU20, CU24, CU26, CU28-GDR, CU29-GDR, CU30 e CU32.

Região	RPEV	CEV fornecida pela AWS	CEV fornecida pelo cliente
Ásia-Pacífico (Seul)	SQL Server 2022 Enterprise, Standard ou Web, com CU9, CU13, CU14-GDR, CU15-GDR, CU16, CU17, CU18 e CU19. SQL Server 2019 Enterprise, Standard ou Web, com CU8, CU17, CU18, CU20, CU24, CU26, CU28-GDR, CU29-GDR, CU30 e CU32.	SQL Server 2022 Enterprise, Standard ou Web, com CU9, CU13, CU14-GDR, CU15-GDR, CU16, CU17, CU18 e CU19. SQL Server 2019 Enterprise, Standard ou Web, com CU8, CU17, CU18, CU20, CU24, CU26, CU28-GDR, CU29-GDR, CU30 e CU32.	SQL Server 2022 Enterprise, Standard ou Web, com CU9, CU13, CU14-GDR, CU15-GDR, CU16, CU17, CU18 e CU19. SQL Server 2019 Enterprise, Standard ou Web, com CU8, CU17, CU18, CU20, CU24, CU26, CU28-GDR, CU29-GDR, CU30 e CU32.
Ásia-Pacífico (Singapura)	SQL Server 2022 Enterprise, Standard ou Web, com CU9, CU13, CU14-GDR, CU15-GDR, CU16, CU17, CU18 e CU19. SQL Server 2019 Enterprise, Standard ou Web, com CU8, CU17, CU18, CU20, CU24, CU26, CU28-GDR, CU29-GDR, CU30 e CU32.	SQL Server 2022 Enterprise, Standard ou Web, com CU9, CU13, CU14-GDR, CU15-GDR, CU16, CU17, CU18 e CU19. SQL Server 2019 Enterprise, Standard ou Web, com CU8, CU17, CU18, CU20, CU24, CU26, CU28-GDR, CU29-GDR, CU30 e CU32.	SQL Server 2022 Enterprise, Standard ou Web, com CU9, CU13, CU14-GDR, CU15-GDR, CU16, CU17, CU18 e CU19. SQL Server 2019 Enterprise, Standard ou Web, com CU8, CU17, CU18, CU20, CU24, CU26, CU28-GDR, CU29-GDR, CU30 e CU32.

Região	RPEV	CEV fornecida pela AWS	CEV fornecida pelo cliente
Ásia-Pacífico (Sydney)	SQL Server 2022 Enterprise, Standard ou Web, com CU9, CU13, CU14-GDR, CU15-GDR, CU16, CU17, CU18 e CU19. SQL Server 2019 Enterprise, Standard ou Web, com CU8, CU17, CU18, CU20, CU24, CU26, CU28-GDR, CU29-GDR, CU30 e CU32.	SQL Server 2022 Enterprise, Standard ou Web, com CU9, CU13, CU14-GDR, CU15-GDR, CU16, CU17, CU18 e CU19. SQL Server 2019 Enterprise, Standard ou Web, com CU8, CU17, CU18, CU20, CU24, CU26, CU28-GDR, CU29-GDR, CU30 e CU32.	SQL Server 2022 Enterprise, Standard ou Web, com CU9, CU13, CU14-GDR, CU15-GDR, CU16, CU17, CU18 e CU19. SQL Server 2019 Enterprise, Standard ou Web, com CU8, CU17, CU18, CU20, CU24, CU26, CU28-GDR, CU29-GDR, CU30 e CU32.
Ásia-Pacífico (Tóquio)	SQL Server 2022 Enterprise, Standard ou Web, com CU9, CU13, CU14-GDR, CU15-GDR, CU16, CU17, CU18 e CU19. SQL Server 2019 Enterprise, Standard ou Web, com CU8, CU17, CU18, CU20, CU24, CU26, CU28-GDR, CU29-GDR, CU30 e CU32.	SQL Server 2022 Enterprise, Standard ou Web, com CU9, CU13, CU14-GDR, CU15-GDR, CU16, CU17, CU18 e CU19. SQL Server 2019 Enterprise, Standard ou Web, com CU8, CU17, CU18, CU20, CU24, CU26, CU28-GDR, CU29-GDR, CU30 e CU32.	SQL Server 2022 Enterprise, Standard ou Web, com CU9, CU13, CU14-GDR, CU15-GDR, CU16, CU17, CU18 e CU19. SQL Server 2019 Enterprise, Standard ou Web, com CU8, CU17, CU18, CU20, CU24, CU26, CU28-GDR, CU29-GDR, CU30 e CU32.

Região	RPEV	CEV fornecida pela AWS	CEV fornecida pelo cliente
Canadá (Central)	SQL Server 2022 Enterprise, Standard ou Web, com CU9, CU13, CU14-GDR, CU15-GDR, CU16, CU17, CU18 e CU19. SQL Server 2019 Enterprise, Standard ou Web, com CU8, CU17, CU18, CU20, CU24, CU26, CU28-GDR, CU29-GDR, CU30 e CU32.	SQL Server 2022 Enterprise, Standard ou Web, com CU9, CU13, CU14-GDR, CU15-GDR, CU16, CU17, CU18 e CU19. SQL Server 2019 Enterprise, Standard ou Web, com CU8, CU17, CU18, CU20, CU24, CU26, CU28-GDR, CU29-GDR, CU30 e CU32.	SQL Server 2022 Enterprise, Standard ou Web, com CU9, CU13, CU14-GDR, CU15-GDR, CU16, CU17, CU18 e CU19. SQL Server 2019 Enterprise, Standard ou Web, com CU8, CU17, CU18, CU20, CU24, CU26, CU28-GDR, CU29-GDR, CU30 e CU32.
Oeste do Canadá (Calgary)	Não disponível	Não disponível	Não disponível
China (Pequim)	Não disponível	Não disponível	Não disponível
China (Ningxia)	Não disponível	Não disponível	Não disponível
Europa (Frankfurt)	SQL Server 2022 Enterprise, Standard ou Web, com CU9, CU13, CU14-GDR, CU15-GDR, CU16, CU17, CU18 e CU19. SQL Server 2019 Enterprise, Standard ou Web, com CU8, CU17, CU18, CU20, CU24, CU26, CU28-GDR, CU29-GDR, CU30 e CU32.	SQL Server 2022 Enterprise, Standard ou Web, com CU9, CU13, CU14-GDR, CU15-GDR, CU16, CU17, CU18 e CU19. SQL Server 2019 Enterprise, Standard ou Web, com CU8, CU17, CU18, CU20, CU24, CU26, CU28-GDR, CU29-GDR, CU30 e CU32.	SQL Server 2022 Enterprise, Standard ou Web, com CU9, CU13, CU14-GDR, CU15-GDR, CU16, CU17, CU18 e CU19. SQL Server 2019 Enterprise, Standard ou Web, com CU8, CU17, CU18, CU20, CU24, CU26, CU28-GDR, CU29-GDR, CU30 e CU32.

Região	RPEV	CEV fornecida pela AWS	CEV fornecida pelo cliente
Europa (Irlanda)	SQL Server 2022 Enterprise, Standard ou Web, com CU9, CU13, CU14-GDR, CU15-GDR, CU16, CU17, CU18 e CU19. SQL Server 2019 Enterprise, Standard ou Web, com CU8, CU17, CU18, CU20, CU24, CU26, CU28-GDR, CU29-GDR, CU30 e CU32.	SQL Server 2022 Enterprise, Standard ou Web, com CU9, CU13, CU14-GDR, CU15-GDR, CU16, CU17, CU18 e CU19. SQL Server 2019 Enterprise, Standard ou Web, com CU8, CU17, CU18, CU20, CU24, CU26, CU28-GDR, CU29-GDR, CU30 e CU32.	SQL Server 2022 Enterprise, Standard ou Web, com CU9, CU13, CU14-GDR, CU15-GDR, CU16, CU17, CU18 e CU19. SQL Server 2019 Enterprise, Standard ou Web, com CU8, CU17, CU18, CU20, CU24, CU26, CU28-GDR, CU29-GDR, CU30 e CU32.
Europa (Londres)	SQL Server 2022 Enterprise, Standard ou Web, com CU9, CU13, CU14-GDR, CU15-GDR, CU16, CU17, CU18 e CU19. SQL Server 2019 Enterprise, Standard ou Web, com CU8, CU17, CU18, CU20, CU24, CU26, CU28-GDR, CU29-GDR, CU30 e CU32.	SQL Server 2022 Enterprise, Standard ou Web, com CU9, CU13, CU14-GDR, CU15-GDR, CU16, CU17, CU18 e CU19. SQL Server 2019 Enterprise, Standard ou Web, com CU8, CU17, CU18, CU20, CU24, CU26, CU28-GDR, CU29-GDR, CU30 e CU32.	SQL Server 2022 Enterprise, Standard ou Web, com CU9, CU13, CU14-GDR, CU15-GDR, CU16, CU17, CU18 e CU19. SQL Server 2019 Enterprise, Standard ou Web, com CU8, CU17, CU18, CU20, CU24, CU26, CU28-GDR, CU29-GDR, CU30 e CU32.
Europa (Milão)	Não disponível	Não disponível	Não disponível

Região	RPEV	CEV fornecida pela AWS	CEV fornecida pelo cliente
Europa (Paris)	SQL Server 2022 Enterprise, Standard ou Web, com CU9, CU13, CU14-GDR, CU15-GDR, CU16, CU17, CU18 e CU19. SQL Server 2019 Enterprise, Standard ou Web, com CU8, CU17, CU18, CU20, CU24, CU26, CU28-GDR, CU29-GDR, CU30 e CU32.	SQL Server 2022 Enterprise, Standard ou Web, com CU9, CU13, CU14-GDR, CU15-GDR, CU16, CU17, CU18 e CU19. SQL Server 2019 Enterprise, Standard ou Web, com CU8, CU17, CU18, CU20, CU24, CU26, CU28-GDR, CU29-GDR, CU30 e CU32.	SQL Server 2022 Enterprise, Standard ou Web, com CU9, CU13, CU14-GDR, CU15-GDR, CU16, CU17, CU18 e CU19. SQL Server 2019 Enterprise, Standard ou Web, com CU8, CU17, CU18, CU20, CU24, CU26, CU28-GDR, CU29-GDR, CU30 e CU32.
Europa (Espanha)	Não disponível	Não disponível	Não disponível
Europa (Estocolmo)	SQL Server 2022 Enterprise, Standard ou Web, com CU9, CU13, CU14-GDR, CU15-GDR, CU16, CU17, CU18 e CU19. SQL Server 2019 Enterprise, Standard ou Web, com CU8, CU17, CU18, CU20, CU24, CU26, CU28-GDR, CU29-GDR, CU30 e CU32.	SQL Server 2022 Enterprise, Standard ou Web, com CU9, CU13, CU14-GDR, CU15-GDR, CU16, CU17, CU18 e CU19. SQL Server 2019 Enterprise, Standard ou Web, com CU8, CU17, CU18, CU20, CU24, CU26, CU28-GDR, CU29-GDR, CU30 e CU32.	SQL Server 2022 Enterprise, Standard ou Web, com CU9, CU13, CU14-GDR, CU15-GDR, CU16, CU17, CU18 e CU19. SQL Server 2019 Enterprise, Standard ou Web, com CU8, CU17, CU18, CU20, CU24, CU26, CU28-GDR, CU29-GDR, CU30 e CU32.
Europa (Zurique)	Não disponível	Não disponível	Não disponível
Israel (Tel Aviv)	Não disponível	Não disponível	Não disponível

Região	RPEV	CEV fornecida pela AWS	CEV fornecida pelo cliente
Oriente Médio (Bahrein)	Não disponível	Não disponível	Não disponível
Oriente Médio (Emirados Árabes Unidos)	Não disponível	Não disponível	Não disponível
América do Sul (São Paulo)	SQL Server 2022 Enterprise, Standard ou Web, com CU9, CU13, CU14-GDR, CU15-GDR, CU16, CU17, CU18 e CU19. SQL Server 2019 Enterprise, Standard ou Web, com CU8, CU17, CU18, CU20, CU24, CU26, CU28-GDR, CU29-GDR, CU30 e CU32.	SQL Server 2022 Enterprise, Standard ou Web, com CU9, CU13, CU14-GDR, CU15-GDR, CU16, CU17, CU18 e CU19. SQL Server 2019 Enterprise, Standard ou Web, com CU8, CU17, CU18, CU20, CU24, CU26, CU28-GDR, CU29-GDR, CU30 e CU32.	SQL Server 2022 Enterprise, Standard ou Web, com CU9, CU13, CU14-GDR, CU15-GDR, CU16, CU17, CU18 e CU19. SQL Server 2019 Enterprise, Standard ou Web, com CU8, CU17, CU18, CU20, CU24, CU26, CU28-GDR, CU29-GDR, CU30 e CU32.
AWS GovCloud (Leste dos EUA)	Não disponível	Não disponível	Não disponível
AWS GovCloud (Oeste dos EUA)	Não disponível	Não disponível	Não disponível

Regiões e mecanismos de banco de dados compatíveis com o Amazon RDS Proxy

O proxy de Amazon RDS é um proxy de banco de dados totalmente gerenciado e altamente disponível que torna as aplicações mais escaláveis ao agrupar e compartilhar conexões de banco de dados estabelecidas. Para obter mais informações, consulte Amazon RDS Proxy.

O RDS Proxy não está disponível com os seguintes mecanismos:

- RDS para Db2
- RDS para Oracle

Tópicos

- RDS Proxy com RDS para MariaDB
- RDS Proxy com RDS para MySQL
- RDS Proxy com RDS para PostgreSQL
- RDS Proxy com RDS para SQL Server

RDS Proxy com RDS para MariaDB

As regiões e as versões do mecanismo a seguir estão disponíveis para o RDS Proxy com o RDS para MariaDB.

Região	RDS para MariaDB 11.4	RDS para MariaDB 10.11	RDS para MariaDB 10.6	RDS para MariaDB 10.5	RDS para MariaDB 10.4
Leste dos EUA (Norte da Virgínia)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Leste dos EUA (Ohio)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Oeste dos EUA (Norte da Califórnia)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Oeste dos EUA (Oregon)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis

Região	RDS para MariaDB 11.4	RDS para MariaDB 10.11	RDS para MariaDB 10.6	RDS para MariaDB 10.5	RDS para MariaDB 10.4
África (Cidade do Cabo)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Hong Kong)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Hyderabad)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Jacarta)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Malásia)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Melbourne)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Mumbai)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Osaka)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Seul)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis

Região	RDS para MariaDB 11.4	RDS para MariaDB 10.11	RDS para MariaDB 10.6	RDS para MariaDB 10.5	RDS para MariaDB 10.4
Ásia-Pacífico (Singapura)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Sydney)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Tailândia)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Tóquio)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Canadá (Central)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Oeste do Canadá (Calgary)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
China (Pequim)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
China (Ningxia)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Frankfurt)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis

Região	RDS para MariaDB 11.4	RDS para MariaDB 10.11	RDS para MariaDB 10.6	RDS para MariaDB 10.5	RDS para MariaDB 10.4
Europa (Irlanda)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Londres)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Milão)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Paris)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Espanha)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Estocolmo)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Zurique)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Israel (Tel Aviv)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Oriente Médio (Bahrein)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis

Região	RDS para MariaDB 11.4	RDS para MariaDB 10.11	RDS para MariaDB 10.6	RDS para MariaDB 10.5	RDS para MariaDB 10.4
Oriente Médio (Emirados Árabes Unidos)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
México (Central)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
América do Sul (São Paulo)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
AWS GovCloud (Leste dos EUA)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
AWS GovCloud (Oeste dos EUA)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis

RDS Proxy com RDS para MySQL

As regiões e as versões do mecanismo a seguir estão disponíveis para o RDS Proxy com o RDS para MySQL.

Região	RDS para MySQL 8.4	RDS para MySQL 8.0	RDS para MySQL 5.7
Leste dos EUA (Norte da Virgínia)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis

Região	RDS para MySQL 8.4	RDS para MySQL 8.0	RDS para MySQL 5.7
Leste dos EUA (Ohio)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Oeste dos EUA (Norte da Califórnia)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Oeste dos EUA (Oregon)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
África (Cidade do Cabo)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Hong Kong)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Hyderabad)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Jacarta)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Malásia)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Melbourne)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Mumbai)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Osaka)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Seul)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Singapura)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis

Região	RDS para MySQL 8.4	RDS para MySQL 8.0	RDS para MySQL 5.7
Ásia-Pacífico (Sydney)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Tailândia)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Tóquio)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Canadá (Central)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Oeste do Canadá (Calgary)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
China (Pequim)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
China (Ningxia)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Frankfurt)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Irlanda)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Londres)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Milão)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Paris)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Espanha)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis

Região	RDS para MySQL 8.4	RDS para MySQL 8.0	RDS para MySQL 5.7
Europa (Estocolmo)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Zurique)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Israel (Tel Aviv)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Oriente Médio (Bahrein)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Oriente Médio (Emirados Árabes Unidos)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
México (Central)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
América do Sul (São Paulo)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
AWS GovCloud (Leste dos EUA)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
AWS GovCloud (Oeste dos EUA)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis

RDS Proxy com RDS para PostgreSQL

As regiões e as versões do mecanismo a seguir estão disponíveis para o RDS Proxy com o RDS PostgreSQL.

Região	RDS para PostgreSQL 17	RDS para PostgreSQL 16	RDS para PostgreSQL 15	RDS para PostgreSQL 14	RDS para PostgreSQL 13	RDS para PostgreSQL 12	RDS para PostgreSQL 11	RDS para PostgreSQL 10
Leste dos EUA (Norte da Virgínia)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Leste dos EUA (Ohio)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Oeste dos EUA (Norte da Califórnia)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Oeste dos EUA (Oregon)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
África (Cidade do Cabo)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis

Região	RDS para PostgreSQL 17	RDS para PostgreSQL 16	RDS para PostgreSQL 15	RDS para PostgreSQL 14	RDS para PostgreSQL 13	RDS para PostgreSQL 12	RDS para PostgreSQL 11	RDS para PostgreSQL 10
Ásia-Pacífico (Hong Kong)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Hyderabad)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Jacarta)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Malásia)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Melbourne)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis

Região	RDS para PostgreSQL 17	RDS para PostgreSQL 16	RDS para PostgreSQL 15	RDS para PostgreSQL 14	RDS para PostgreSQL 13	RDS para PostgreSQL 12	RDS para PostgreSQL 11	RDS para PostgreSQL 10
Ásia-Pacífico (Mumbai)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Osaka)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Seul)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Singapura)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Sydney)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis

Região	RDS para PostgreSQL 17	RDS para PostgreSQL 16	RDS para PostgreSQL 15	RDS para PostgreSQL 14	RDS para PostgreSQL 13	RDS para PostgreSQL 12	RDS para PostgreSQL 11	RDS para PostgreSQL 10
Ásia-Pacífico (Tailândia)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Tóquio)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Canadá (Central)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Oeste do Canadá (Calgary)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
China (Pequim)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis

Região	RDS para PostgreSQL 17	RDS para PostgreSQL 16	RDS para PostgreSQL 15	RDS para PostgreSQL 14	RDS para PostgreSQL 13	RDS para PostgreSQL 12	RDS para PostgreSQL 11	RDS para PostgreSQL 10
China (Ningxia)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Frankfurt)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Irlanda)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Londres)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Milão)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis

Região	RDS para PostgreSQL 17	RDS para PostgreSQL 16	RDS para PostgreSQL 15	RDS para PostgreSQL 14	RDS para PostgreSQL 13	RDS para PostgreSQL 12	RDS para PostgreSQL 11	RDS para PostgreSQL 10
Europa (Paris)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Espanha)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Estocolmo)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Zurique)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Israel (Tel Aviv)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis

Região	RDS para PostgreSQL 17	RDS para PostgreSQL 16	RDS para PostgreSQL 15	RDS para PostgreSQL 14	RDS para PostgreSQL 13	RDS para PostgreSQL 12	RDS para PostgreSQL 11	RDS para PostgreSQL 10
Oriente Médio (Bahrein)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Oriente Médio (Emirados Árabes Unidos)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
México (Central)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
América do Sul (São Paulo)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
AWS GovCloud (Leste dos EUA)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis

Região	RDS para PostgreSQL 17	RDS para PostgreSQL 16	RDS para PostgreSQL 15	RDS para PostgreSQL 14	RDS para PostgreSQL 13	RDS para PostgreSQL 12	RDS para PostgreSQL 11	RDS para PostgreSQL 10
AWS GovCloud (Oeste dos EUA)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis

RDS Proxy com RDS para SQL Server

As regiões e as versões do mecanismo a seguir estão disponíveis para o RDS Proxy com o RDS para SQL Server.

Região	RDS para SQL Server 2019	RDS para SQL Server 2017	RDS para SQL Server 2016
Leste dos EUA (Norte da Virgínia)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Leste dos EUA (Ohio)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Oeste dos EUA (Norte da Califórnia)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Oeste dos EUA (Oregon)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
África (Cidade do Cabo)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Hong Kong)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis

Região	RDS para SQL Server 2019	RDS para SQL Server 2017	RDS para SQL Server 2016
Ásia-Pacífico (Hyderabad)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Jacarta)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Malásia)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Melbourne)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Mumbai)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Osaka)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Seul)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Singapura)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Sydney)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Tailândia)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Tóquio)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Canadá (Central)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis

Região	RDS para SQL Server 2019	RDS para SQL Server 2017	RDS para SQL Server 2016
Oeste do Canadá (Calgary)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
China (Pequim)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
China (Ningxia)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Frankfurt)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Irlanda)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Londres)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Milão)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Paris)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Espanha)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Estocolmo)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Zurique)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Israel (Tel Aviv)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis

Região	RDS para SQL Server 2019	RDS para SQL Server 2017	RDS para SQL Server 2016
Oriente Médio (Bahrein)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Oriente Médio (Emirados Árabes Unidos)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
México (Central)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
América do Sul (São Paulo)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
AWS GovCloud (Leste dos EUA)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
AWS GovCloud (Oeste dos EUA)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis

Regiões e mecanismos de banco de dados compatíveis com a integração do Secrets Manager com o Amazon RDS

Com o AWS Secrets Manager, é possível substituir credenciais codificadas em seu código, inclusive senhas de banco de dados, por uma chamada de API ao Secrets Manager para recuperar o segredo de forma programática. Para ter mais informações sobre o Secrets Manager, consulte o Guia do usuário do AWS Secrets Manager.

Você pode especificar que o Amazon RDS gerencie a senha de usuário principal no Secrets Manager para uma instância de banco de dados do Amazon RDS ou um cluster de banco de dados multi-AZ. O RDS gera a senha, a armazena no Secrets Manager e a alterna regularmente. Para ter mais informações, consulte Gerenciamento de senhas com Amazon RDS e AWS Secrets Manager.

A integração do Secrets Manager está disponível em todas as Regiões da AWS.

Regiões e mecanismos de banco de dados que permitem Integrações ETL zero com o Amazon RDS

As Integrações ETL zero do RDS são uma solução totalmente gerenciada para disponibilizar dados transacionais no Amazon Redshift ou no Amazon SageMaker depois que eles são gravados em uma instância de banco de dados do Amazon RDS. Para obter mais informações, consulte Integrações ETL zero.

Tópicos

- Integrações ETL zero com o RDS para MySQL

Integrações ETL zero com o RDS para MySQL

As regiões e as versões de mecanismo a seguir estão disponíveis para Integrações ETL zero do RDS para MySQL com o Amazon Redshift e o Amazon SageMaker.

Região	Integração ETL zero do Amazon Redshift com o RDS para MySQL 8.4	Integração ETL zero do Amazon Redshift com o RDS para MySQL 8.0	Integração ETL zero do Amazon SageMaker com o RDS para MySQL 8.4	Integração ETL zero do Amazon SageMaker com o RDS para MySQL 8.0
Leste dos EUA (Norte da Virgínia	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Leste dos EUA (Ohio)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis

Região	Integração ETL zero do Amazon Redshift com o RDS para MySQL 8.4	Integração ETL zero do Amazon Redshift com o RDS para MySQL 8.0	Integração ETL zero do Amazon SageMaker com o RDS para MySQL 8.4	Integração ETL zero do Amazon SageMaker com o RDS para MySQL 8.0
Oeste dos EUA (Norte da Califórn a)	Todas as versões disponíveis	Todas as versões disponíveis	Não disponível	Não disponível
Oeste dos EUA (Orego	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
África (Cidade do Cabo)	Todas as versões disponíveis	Todas as versões disponíveis	Não disponível	Não disponível
Ásia-Pacífico (Hong Kong)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Hyderad)	Não disponível	Não disponível	Não disponível	Não disponível

Região	Integração ETL zero do Amazon Redshift com o RDS para MySQL 8.4	Integração ETL zero do Amazon Redshift com o RDS para MySQL 8.0	Integração ETL zero do Amazon SageMaker com o RDS para MySQL 8.4	Integração ETL zero do Amazon SageMaker com o RDS para MySQL 8.0
Ásia-Pacífico (Jacart	Não disponível	Não disponível	Não disponível	Não disponível
Ásia-Pacífico (Malási	Não disponível	Não disponível	Não disponível	Não disponível
Ásia-Pacífico (Melbo e)	Não disponível	Não disponível	Não disponível	Não disponível
Ásia-Pacífico (Mumb	Todas as versões disponíveis	Todas as versões disponíveis	Não disponível	Não disponível
Ásia-Pacífico (Osaka	Todas as versões disponíveis	Todas as versões disponíveis	Não disponível	Não disponível
Ásia-Pacífico (Seul)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis

Região	Integração ETL zero do Amazon Redshift com o RDS para MySQL 8.4	Integração ETL zero do Amazon Redshift com o RDS para MySQL 8.0	Integração ETL zero do Amazon SageMaker com o RDS para MySQL 8.4	Integração ETL zero do Amazon SageMaker com o RDS para MySQL 8.0
Ásia-Pacífico (Singapura)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Sydne	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Ásia-Pacífico (Tóquio	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Canadá (Centra	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Oeste do Canadá (Calgar	Não disponível	Não disponível	Não disponível	Não disponível
China (Pequir	Não disponível	Não disponível	Não disponível	Não disponível
China (Ningxi	Não disponível	Não disponível	Não disponível	Não disponível

Região	Integração ETL zero do Amazon Redshift com o RDS para MySQL 8.4	Integração ETL zero do Amazon Redshift com o RDS para MySQL 8.0	Integração ETL zero do Amazon SageMaker com o RDS para MySQL 8.4	Integração ETL zero do Amazon SageMaker com o RDS para MySQL 8.0
Europa (Frankf t)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Irlanda	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Londre	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Milão)	Todas as versões disponíveis	Todas as versões disponíveis	Não disponível	Não disponível
Europa (Paris)	Todas as versões disponíveis	Todas as versões disponíveis	Não disponível	Não disponível
Europa (Espan	Não disponível	Não disponível	Não disponível	Não disponível
Europa (Estocc o)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
Europa (Zuriqu	Não disponível	Não disponível	Não disponível	Não disponível

Região	Integração ETL zero do Amazon Redshift com o RDS para MySQL 8.4	Integração ETL zero do Amazon Redshift com o RDS para MySQL 8.0	Integração ETL zero do Amazon SageMaker com o RDS para MySQL 8.4	Integração ETL zero do Amazon SageMaker com o RDS para MySQL 8.0
Israel (Tel Aviv)	Não disponível	Não disponível	Não disponível	Não disponível
Oriente Médio (Bahrei	Todas as versões disponíveis	Todas as versões disponíveis	Não disponível	Não disponível
Oriente Médio (Emira Árabes Unidos	Não disponível	Não disponível	Não disponível	Não disponível
Américo do Sul (São Paulo)	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis	Todas as versões disponíveis
AWS GovClo (Leste dos EUA)	Não disponível	Não disponível	Não disponível	Não disponível
AWS GovClo (Oeste dos EUA)	Não disponível	Não disponível	Não disponível	Não disponível

Recursos nativos do mecanismo no Amazon RDS

Os mecanismos de banco de dados do Amazon RDS também oferecem suporte a muitos dos recursos e funcionalidades mais comuns e nativos do mecanismo. Esses recursos são diferentes dos recursos nativos do Amazon RDS listados nesta página. Alguns recursos nativos do mecanismo podem ter suporte limitado ou privilégios restritos.

Para ter mais informações sobre os recursos nativos do mecanismo, consulte:

- Recursos do Amazon RDS para Db2
- Suporte ao recurso MariaDB no Amazon RDS
- Compatibilidade de recursos do MySQL no Amazon RDS
- Recursos do RDS for Oracle
- Trabalhar com recursos do PostgreSQL compatíveis com o Amazon RDS para PostgreSQL
- Recursos do Microsoft SQL Server no Amazon RDS

Faturamento da instância de banco de dados para Amazon RDS

As instâncias do Amazon RDS são cobradas com base em seguintes componentes:

- Horas de instância de banco de dados (por hora) – com base na classe da instância de banco de dados (por exemplo, db.t2.small ou db.m4.large). A definição de preço está listada em uma base por hora, mas é calculada em segundos e mostra o tempo no formato decimal. O uso do RDS é cobrado em incrementos de um segundo, com um mínimo de 10 minutos. Para ter mais informações, consulte Classes de instâncias de banco de dados .
- Armazenamento (por GiB por mês) – a capacidade de armazenamento provisionado para a sua instância de banco de dados. Se você dimensionar sua capacidade de armazenamento provisionada dentro do mês, sua fatura será rateada. Para obter mais informações, consulte Armazenamento de instâncias de banco de dados do Amazon RDS.
- Solicitações de entrada/saída (E/S) (por 1 milhão de solicitações): o número total de solicitações de E/S de armazenamento que você fez em um ciclo de faturamento, para somente o armazenamento magnético do Amazon RDS.
- IOPS provisionadas (por IOPS por mês): taxa de IOPS provisionadas, independentemente das IOPS consumidas, para armazenamento de IOPS provisionadas (SSD) do Amazon RDS e gp3 de uso geral. O armazenamento provisionado para volumes do EBS é cobrado em incrementos de um segundo, com um mínimo de 10 minutos.
- Armazenamento de backup (por GiB por mês) – o armazenamento de backup é o armazenamento associado a backups automatizados de banco de dados e a qualquer DB snapshot ativo que você tenha feito. Aumentar seu período de retenção de backup ou fazer snapshots de bancos de dados adicionais aumenta o armazenamento de backup utilizado por seu banco de dados. A cobrança por segundo não se aplica ao armazenamento de backup (medido em GB por mês).

  Para obter mais informações, consulte Backup, restauração e exportação de dados.
- Transferência de dados (por GB): transferência de dados para dentro e para fora de sua instância de banco de dados de ou para a Internet e outras regiões da AWS. Para ter exemplos úteis, consulte a postagem do blog da AWS Exploring Data Transfer Costs for AWS Managed Databases.

O Amazon RDS fornece as seguintes opções de compra para permitir otimizar os custos com base em suas necessidades:

- On-Demand Instances (Instâncias sob demanda): pague por hora pelas horas de instância de banco de dados que você usar. A definição de preço está listada em uma base por hora, mas é calculada em segundos e mostra o tempo no formato decimal. O uso do RDS agora é cobrado em incrementos de um segundo, com um mínimo de 10 minutos.
- Reserved Instances (Instâncias reservadas): reserve uma instância de banco de dados por um período de vigência de um ou três anos e receba um desconto significativo com relação à definição de preço das instâncias de banco de dados sob demanda. Com o uso de instâncias reservadas, você pode iniciar, excluir, iniciar ou parar várias instâncias dentro de uma hora e obter o benefício da instância reservada para todas as instâncias.

Para obter informações sobre a definição de preço do Amazon RDS, consulte a página Definição de preço do Amazon RDS.

Tópicos

- Instâncias de banco de dados sob demanda do Amazon RDS
- Instâncias de banco de dados reservadas para o Amazon RDS

Instâncias de banco de dados sob demanda do Amazon RDS

As Instâncias de banco de dados sob demanda do Amazon RDS são cobradas com base na classe da instância de banco de dados (por exemplo, db.t3.small ou db.m5.large). Para obter informações sobre a definição de preço do Amazon RDS, consulte a página de produto do Amazon RDS.

O faturamento é iniciado para uma instância de banco de dados assim que a instância de banco de dados se torna disponível. A definição de preço está listada em uma base por hora, mas é calculada em segundos e mostra o tempo no formato decimal. O uso do Amazon RDS é faturado em incrementos de um segundo, com um mínimo de dez minutos. No caso de uma alteração de configuração faturável, como a expansão da capacidade de computação ou armazenamento, você paga um mínimo de 10 minutos. O faturamento continua até que a instância de banco de dados é encerrada, o que ocorre quando você exclui a instância de banco de dados ou se ocorre uma falha na instância de banco de dados.

Se não desejar ser faturado por sua instância de banco de dados, você deverá encerrá-la ou excluí-la para evitar ser cobrado por horas adicionais da instância de banco de dados. Para obter mais informações sobre os estados de instâncias de banco de dados dos quais você é faturado, consulte Visualizar o status de uma instância de banco de dados Amazon RDS em um .

Instâncias de banco de dados interrompidas

Quando sua instância de banco de dados for interrompida, você será cobrado pelo armazenamento provisionado, incluindo IOPS provisionadas. Você também é faturado pelo armazenamento de backup, incluindo o armazenamento de snapshots manuais e backups automatizados durante a janela de retenção especificada. Não há cobrança por horas de instância de banco de dados.

Instâncias de banco de dados multi-AZ

Uma configuração multi-AZ aprimora a durabilidade e disponibilidade dos dados ao provisionar e manter automaticamente uma réplica em espera síncrona em outra zona de disponibilidade. Devido aos recursos adicionais e ao aumento da disponibilidade, as implantações multi-AZ têm um preço mais alto do que as implantações single-AZ e podem custar aproximadamente o dobro devido à instância em espera adicional e aos recursos associados.

Considere os seguintes detalhes importantes sobre o preço de multi-AZ:

- Custos de computação: cobrados por hora de instância de banco de dados para as instâncias primária e em espera.

- Custos de armazenamento: cobrados por GB-mês pelo armazenamento provisionado para as instâncias primária e em espera.
- Custos de transferência de dados: a replicação entre as instâncias primária e em espera está incluída no custo, mas outras taxas de transferência de dados podem ser aplicadas com base no uso.

Para estimar com precisão seus custos mensais com base em seu caso de uso específico e na Região da AWS, você pode usar a AWS Calculadora de Preços. Nessa ferramenta, você insere os detalhes da sua configuração e recebe um detalhamento abrangente dos custos.

> **Note**
>
> Os preços estão sujeitos a alterações. Consulte as informações mais atualizadas na página Preço do Amazon RDS.

Instâncias de banco de dados reservadas para o Amazon RDS

Usando instâncias de banco de dados reservadas, você pode reservar uma instância de banco de dados por um período de um ou três anos. As instâncias de banco de dados reservadas fornecem um desconto significativo em comparação com os preços de instâncias de banco de dados sob demanda. As instâncias de banco de dados reservadas não são instâncias físicas, mas um desconto na fatura aplicado na sua conta pelo uso de determinadas instâncias de banco de dados sob demanda. Os descontos para instâncias de bancos de dados reservadas são vinculados ao tipo de instância e à Região da AWS.

O processo geral de trabalho com instâncias de banco de dados reservadas é: primeiro, obtenha informações sobre ofertas de instâncias de banco de dados reservadas disponíveis; em seguida, compre uma oferta de instância de banco de dados reservada; e, por fim, obtenha informações sobre suas instâncias de banco de dados reservadas existentes.

Para ter informações sobre a compra de instâncias de banco de dados reservadas e a visualização de cobranças de instâncias de banco de dados reservadas, consulte as seções a seguir.

- Comprar instâncias de banco de dados reservadas para o Amazon RDS
- Visualizar a cobrança de instâncias de banco de dados reservadas para o Amazon RDS

Visão geral de instâncias de banco de dados reservadas

Ao comprar uma instância de banco de dados reservada no Amazon RDS, você adquire um compromisso de obter uma taxa com desconto sobre um tipo específico de instância de banco de dados pela duração dela. Para usar uma instância de banco de dados reservada do Amazon RDS, crie uma instância de banco de dados como faria para uma instância sob demanda.

A nova instância de banco de dados que você criar deve ter as mesmas especificações da instância de banco de dados reservada quanto ao seguinte:

- Região da AWS
- Mecanismo de banco de dados (não é preciso haver correspondência com o número da versão do mecanismo de banco de dados).
- Tipo de instância do banco de dados
- Tamanho da instância de banco de dados (licença do RDS para Microsoft SQL Server e Amazon RDS para Oracle incluída)

- Edição (RDS para SQL Server e RDS para Oracle)
- Tipo de licença (licença incluída ou traga sua própria licença)

Se as especificações da nova instância de banco de dados corresponderem às de uma instância de banco de dados reservada existente para a sua conta, será cobrada a taxa de desconto oferecida para a instância reservada. Caso contrário, uma taxa sob demanda será cobrada para a instância de banco de dados.

Você pode modificar uma instância de banco de dados que está usando como instância de banco de dados reservada. Se a modificação estiver dentro das especificações da instância de banco de dados reservada, parte ou todo o desconto ainda se aplicará à instância de banco de dados modificada. Se a modificação estiver fora das especificações, como alterar a classe de instância, o desconto não será mais aplicado. Para ter mais informações, consulte Instâncias de banco de dados reservadas de tamanho flexível.

Tópicos

- Tipos de oferta
- Instâncias de banco de dados reservadas de tamanho flexível
- Exemplo de faturamento de instância de banco de dados reservada
- Instâncias de banco de dados reservadas para um cluster de banco de dados multi-AZ
- Excluir uma instância de banco de dados reservada

Para ter mais informações sobre instâncias de banco de dados reservadas, inclusive definição de preço, consulte Instâncias reservadas do Amazon RDS.

Tipos de oferta

Instâncias de banco de dados reservadas estão disponíveis em três variedades: sem pagamento adiantado, com pagamento adiantado parcial e com pagamento adiantado integral. Esses tipos permitem otimizar os custos do Amazon RDS com base no uso esperado.

> **Note**
>
> Nem todas as classes de instância do RDS são compatíveis com todos os tipos de oferta de instância reservada. Por exemplo, algumas classes de instância podem não oferecer a opção Sem pagamento adiantado. Para confirmar a disponibilidade, analise as ofertas de instância

reservada no AWS Management Console ou use o comando `describe-reserved-db-instances-offerings` da AWS CLI.

Sem taxas iniciais

Essa opção fornece acesso a uma instância de banco de dados reservado sem a necessidade de pagamento adiantado. Sua instância de banco de dados reservada sem pagamento adiantado será cobrada de acordo com uma taxa horária com desconto por cada hora dentro do período de vigência, independentemente do uso, e nenhum pagamento adiantado é obrigatório. Essa opção só está disponível como uma reserva de um ano.

Adiantado parcial

Essa opção requer que uma parte da instância de banco de dados reservada seja paga antecipadamente. As horas restantes do período de vigência serão cobradas com base em uma taxa horária com desconto, independentemente do uso. Essa opção é a substituição da opção de Utilização pesada anterior.

Adiantado integral

O pagamento integral é feito no início do período de vigência, sem outros custos ou cobranças por hora incorridos pelo restante do período, independentemente do número de horas usadas.

Se você estiver usando faturamento consolidado, todas as contas da organização serão tratadas como se fossem uma só. Isso significa que todas as contas na organização podem receber o custo-benefício por hora das instâncias de banco de dados reservadas que são compradas por qualquer outra conta. Para ter mais informações sobre o faturamento consolidado, consulte Instâncias de bancos de dados reservadas do Amazon RDS no Guia do usuário do Gerenciamento de Faturamento e Custos da AWS.

Instâncias de banco de dados reservadas de tamanho flexível

Ao adquirir uma instância de banco de dados reservada, uma das especificações feitas é a classe da instância, por exemplo, db.r5.large. Para ter mais informações sobre classes de instância de banco de dados, consulte Classes de instâncias de banco de dados .

Se você tiver uma instância de banco de dados e precisar escalá-la para uma capacidade maior, sua instância de banco de dados reservada será automaticamente aplicada à sua instância de

banco de dados escalada. Ou seja, suas instâncias de banco de dados reservadas são aplicadas automaticamente em todos os tamanhos de classe de instância de banco de dados. As instâncias de banco de dados reservadas de tamanho flexível estão disponíveis para instâncias de bancos de dados com a mesma Região da AWS e mecanismo de banco de dados. As instâncias de banco de dados reservadas de tamanho flexível só reduzem o tipo de classe de instância. Por exemplo, uma instância de banco de dados reservada de db.r5.large pode se aplicar a uma db.r5.xlarge, mas não a uma db.r6g.large, porque db.r5 e db.r6g são tipos de classe de instância diferentes.

Os benefícios da instância de banco de dados reservada se aplicam às configurações multi-AZ e single-AZ. Isso significa que você pode se mover livremente entre configurações no mesmo tipo de classe de instância de banco de dados. Por exemplo, é possível passar de uma implantação de uma única zona de disponibilidade em execução em uma instância de banco de dados grande (quatro unidades normalizadas por hora) para uma implantação multi-AZ em execução em duas instâncias de banco de dados médias (2 + 2 = 4 unidades normalizadas por hora).

As instâncias de banco de dados reservadas de tamanho flexível estão disponíveis para os seguintes mecanismos de banco de dados do Amazon RDS:

- RDS para MariaDB
- RDS para MySQL
- RDS para Oracle, Traga sua própria licença
- RDS para PostgreSQL

A flexibilidade de tamanho não se aplica à licença do RDS para SQL Server e do RDS para Oracle incluída.

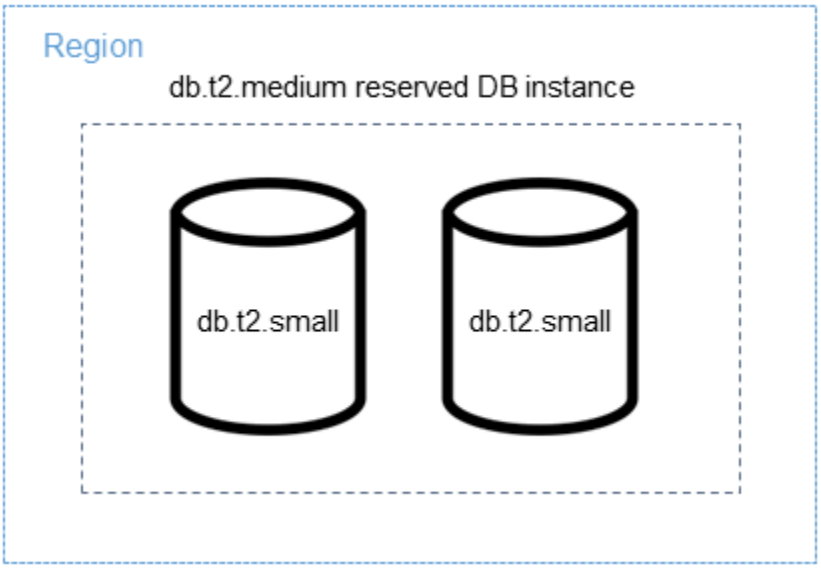
Para obter detalhes de como usar as instâncias reservadas de tamanho flexível com o Aurora, consulte Instâncias de banco de dados reservadas para o Aurora.

Você pode comparar o uso de tipos diferentes de instância de banco de dados reservada usando unidades normalizadas por hora. Por exemplo, uma unidade de uso em duas instâncias de banco de dados db.r3.large é equivalente a oito unidades normalizadas por hora de uso em uma db.r3.small. A tabela a seguir mostra o número de unidades normalizadas por hora para cada tamanho de instância de banco de dados.

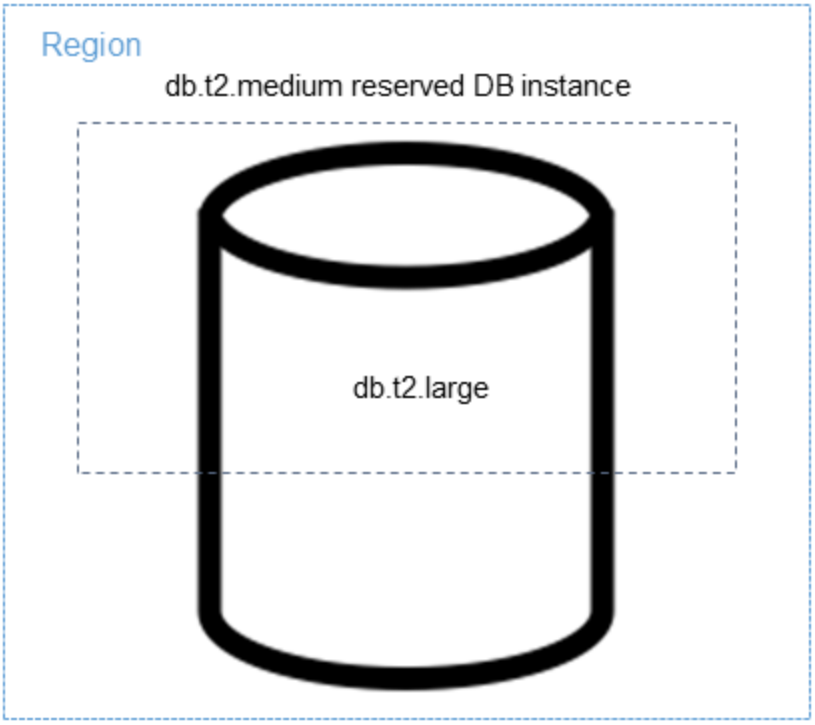
Tamanho da instância	Unidades normalizadas single-AZ por hora (implantação com uma instância de banco de dados)	Unidades normalizadas de instâncias de banco de dados multi-AZ por hora (implantação com uma instância de banco de dados e uma em espera)	Unidades normalizadas de cluster de banco de dados multi-AZ por hora (implantação com uma instância de banco de dados e duas em espera)
micro	0,5	1	1.5
pequeno	1	2	3
médio	2	4	6
grande	4	8	12
xlarge	8	16	24
2xlarge	16	32	48
4xlarge	32	64	96
6xlarge	48	96	144
8xlarge	64	128	192
10xlarge	80	160	240
12xlarge	96	192	288
16xlarge	128	256	384
24xlarge	192	384	576
32xlarge	256	512	768

Por exemplo, suponhamos que você compre uma instância de bancos de dados `db.t2.medium` reservada e tenha duas instâncias de banco de dados `db.t2.small` em execução na conta na

mesma Região da AWS. Nesse caso, o benefício de faturamento é aplicado integralmente a ambas as instâncias.



Ou, se você tiver uma instância `db.t2.large` em execução em sua conta na mesma Região da AWS, o benefício de faturamento será aplicado a 50% do uso da instância de banco de dados.



Exemplo de faturamento de instância de banco de dados reservada

O preço de uma instância de banco de dados reservada não oferece um desconto para os custos associados a armazenamento, backups e E/S. Ele oferece um desconto somente no uso de uma instância sob demanda por hora. O exemplo a seguir ilustra o custo total por mês para uma instância de banco de dados reservada:

- Uma classe de instância de banco de dados single-AZ db.r5.large reservada do RDS para MySQL no Leste dos EUA (N. da Virgínia) com a opção Sem adiantamento por um custo de USD 0,12 pela instância, ou USD 90,00 por mês
- 400 GiB de armazenamento SSD (gp2) de uso geral a um custo de 0,115 por GiB por mês, ou US$ 45,60 por mês
- 600 GiB de armazenamento de backup a US$ 0,095, ou US$ 19 por mês (400 GiB grátis)

Adicione todas essas cobranças (USD 90 + USD 45,60 + USD 19) com a instância de banco de dados reservada e o custo total por mês USD 154,60.

Se você optar por usar uma instância de banco de dados sob demanda em vez de uma instância de banco de dados reservada, uma classe de instância de banco de dados single-AZ db.r5.large do RDS para MySQL no Leste dos EUA (N. da Virgínia) custará USD 0,1386 por hora, ou USD 101,18 por mês. Dessa maneira, para uma instância de banco de dados sob demanda, adicione todas essas opções (US$ 101,18 + US$ 45,60 + US$ 19), e o custo total por mês é US$ 165,78. Você economiza um pouco mais de USD 11 por mês utilizando a instância de banco de dados reservada.

> **Note**
>
> Os preços neste exemplo são preços de amostra e talvez não correspondam aos preços reais. Para obter informações sobre a definição de preço do Amazon RDS, consulte Definição de preço do Amazon RDS.

Instâncias de banco de dados reservadas para um cluster de banco de dados multi-AZ

Para comprar as instâncias de banco de dados reservadas equivalentes para um cluster de banco de dados multi-AZ, faça o seguinte:

- Reserve três instâncias de banco de dados single-AZ que tenham o mesmo tamanho das instâncias no cluster.
- Reserve uma instância de banco de dados multi-AZ e uma instância de banco de dados single-AZ que tenham o mesmo tamanho das instâncias de banco de dados contidas no cluster.

Por exemplo, digamos que você tenha um cluster que consiste em três instâncias de banco de dados db.m6gd.large. Nesse caso, você pode comprar três instâncias de banco de dados reservadas db.m6gd.large single-AZ ou uma instância de banco de dados reservada db.m6gd.large multi-AZ e uma instância de banco de dados reservada db.m6gd.large single-AZ. Qualquer uma dessas opções reserva o desconto máximo de instância reservada para o cluster de banco de dados multi-AZ.

Como alternativa, você pode usar instâncias de banco de dados de tamanho flexível e comprar uma instância de banco de dados maior para cobrir instâncias de banco de dados menores em um ou mais clusters. Por exemplo, se você tiver dois clusters com seis instâncias de banco de dados db.m6gd.large no total, poderá comprar três instâncias de banco de dados reservadas db.m6gd.xl single-AZ. Isso reserva todas as seis instâncias de banco de dados nos dois clusters. Para ter mais informações, consulte Instâncias de banco de dados reservadas de tamanho flexível.

Você pode reservar instâncias de banco de dados do mesmo tamanho das instâncias de banco de dados no cluster, mas reservar menos instâncias de banco de dados do que o número total de instâncias de banco de dados no cluster. No entanto, se você fizer isso, o cluster será reservado apenas parcialmente. Por exemplo, suponha que você tenha um cluster com três instâncias de banco de dados db.m6gd.large e compre uma instância de banco de dados reservada db.m6gd.large multi-AZ. Nesse caso, o cluster é reservado apenas parcialmente, porque somente duas das três instâncias no cluster são cobertas por instâncias de banco de dados reservadas. A instância de banco de dados restante é cobrada de acordo com a taxa horária de db.m6gd.large sob demanda.

Para ter mais informações sobre clusters de banco de dados multi-AZ, consulte Implantações de cluster de banco de dados multi-AZ para o Amazon RDS.

Excluir uma instância de banco de dados reservada

Os períodos de vigência de uma instância de banco de dados reservada envolvem um compromisso de um ou três anos. Não é possível cancelar uma instância de banco de dados reservada. No entanto, você pode excluir uma instância de banco de dados coberta por um desconto de instância de banco de dados reservada. O processo de exclusão de uma instância de banco de dados coberta por um desconto de instância de banco de dados reservada é o mesmo que o de qualquer outra instância de banco de dados.

Você receberá uma cobrança pelos custos adiantados, independentemente do uso dos recursos.

Se você excluir uma instância de banco de dados coberta por um desconto de instância de banco de dados reservada, poderá iniciar outra instância de banco de dados com especificações compatíveis. Neste caso, você continua recebendo a taxa com desconto durante o período de vigência da reserva (um ou três anos).

Comprar instâncias de banco de dados reservadas para o Amazon RDS

Você pode usar o AWS Management Console, a AWS CLI e a API do RDS para trabalhar com instâncias de banco de dados reservadas.

Console

Você pode usar o AWS Management Console para trabalhar com instâncias de banco de dados reservadas conforme exibido nos procedimentos a seguir.

Para obter informações sobre preços e ofertas de instâncias de bancos de dados reservadas disponíveis

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Reserved instances (Instâncias reservadas).
3. Escolha Purchase Reserved DB Instance (Comprar instância de banco de dados reservada).
4. Em Product description (Descrição do produto), escolha o tipo de licenciamento e mecanismo de banco de dados.
5. Para DB instance class (Classe de instância do banco de dados), escolha a classe da instância de banco de dados.
6. Em Opção de implantação, selecione se deseja uma implantação de instância de banco de dados single-AZ ou multi-AZ.

> **Note**
>
> Para comprar as instâncias de banco de dados reservadas equivalentes para uma implantação de cluster de banco de dados multi-AZ, compre três instâncias de banco de dados reservadas single-AZ ou uma instância de banco de dados reservada multi-AZ e uma single-AZ. Para ter mais informações, consulte Instâncias de banco de dados reservadas para um cluster de banco de dados multi-AZ.

7. Em Prazo, selecione por quanto tempo a instância de banco de dados deve ser reservada.
8. Em Offering type (Tipo de oferta), escolha o tipo de oferta.

   Após selecionar o tipo de oferta, você pode visualizar as informações de preço.

   > **Important**
   >
   > Escolha Cancel (Cancelar) para evitar a compra da instância de banco de dados reservada e gerar quaisquer cobranças.

Assim que tiver informações sobre as ofertas de instâncias de banco de dados reservadas disponíveis, você poderá usá-las para comprar uma oferta, conforme mostrado no procedimento a seguir.

Para comprar uma instância de banco de dados reservada

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Reserved instances (Instâncias reservadas).
3. Selecione Purchase Reserved DB Instance (Comprar instância de banco de dados reservada).
4. Em Product description (Descrição do produto), escolha o tipo de licenciamento e mecanismo de banco de dados.
5. Para DB instance class (Classe de instância do banco de dados), escolha a classe da instância de banco de dados.
6. Em Implantação multi-AZ, selecione se deseja uma implantação de instância de banco de dados single-AZ ou multi-AZ.

   > **Note**
   >
   > Para comprar as instâncias de banco de dados reservadas equivalentes para uma implantação de cluster de banco de dados multi-AZ, compre três instâncias de banco de dados reservadas Single-AZ ou uma instância de banco de dados reservada multi-AZ e uma Single-AZ. Para ter mais informações, consulte Instâncias de banco de dados reservadas para um cluster de banco de dados multi-AZ.

7. Em Term (Prazo), escolha quanto tempo você deseja que a instância de banco de dados seja reservada.
8. Em Offering type (Tipo de oferta), escolha o tipo de oferta.

   Após selecionar o tipo de oferta, você pode visualizar as informações de definição de preço.
9. (Opcional) Você pode atribuir seu próprio identificador às instâncias de banco de dados reservadas adquiridas, a fim de ajudar a manter o controle delas. Em Reserved Id (ID da instância reservada), digite um identificador para a instância de banco de dados reservada.
10. Selecione Enviar.

    Sua instância de banco de dados reservada é comprada e, depois, exibida na lista Reserved instances (Instâncias reservadas).

Depois de adquirir instâncias de banco de dados reservadas, você poderá obter informações sobre elas, conforme mostrado no procedimento a seguir.

Para obter informações sobre instâncias de Bancos de Dados reservadas para a sua conta da AWS

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel Navigation (Navegação), escolha Reserved instances (Instâncias reservadas).

   As instâncias de banco de dados reservadas para sua conta são exibidas. Para ver informações detalhadas sobre uma instância de banco de dados reservada específica, escolha essa instância na lista. Você pode acabar vendo informações detalhadas sobre essa instância no painel de detalhes na parte inferior do console.

AWS CLI

Você pode usar a AWS CLI para trabalhar com instâncias de banco de dados reservadas, conforme mostrado nos exemplos a seguir.

Example obtenção de ofertas de instâncias de banco de dados reservadas disponíveis

Para obter informações sobre as ofertas disponíveis de instâncias de banco de dados reservadas, chame o comando AWS CLI da `describe-reserved-db-instances-offerings`.

```
aws rds describe-reserved-db-instances-offerings
```

Essa chamada retorna uma saída semelhante à seguinte:

```
OFFERING  OfferingId                            Class         Multi-AZ  Duration  Fixed
 Price  Usage Price  Description  Offering Type
OFFERING  438012d3-4052-4cc7-b2e3-8d3372e0e706  db.r3.large   y         1y
 1820.00 USD  0.368 USD    mysql        Partial  Upfront
OFFERING  649fd0c8-cf6d-47a0-bfa6-060f8e75e95f  db.r3.small   n         1y
 227.50 USD  0.046 USD    mysql        Partial  Upfront
OFFERING  123456cd-ab1c-47a0-bfa6-12345667232f  db.r3.small   n         1y
 162.00 USD   0.00 USD    mysql        All      Upfront
    Recurring Charges:   Amount  Currency  Frequency
    Recurring Charges:   0.123   USD       Hourly
OFFERING  123456cd-ab1c-37a0-bfa6-12345667232d  db.r3.large   y         1y
 700.00 USD   0.00 USD    mysql        All      Upfront
    Recurring Charges:   Amount  Currency  Frequency
    Recurring Charges:   1.25    USD       Hourly
OFFERING  123456cd-ab1c-17d0-bfa6-12345667234e  db.r3.xlarge  n         1y
 4242.00 USD   2.42 USD    mysql        No       Upfront
```

Assim que tiver informações sobre as ofertas de instâncias de banco de dados reservadas disponíveis, você poderá usá-las para comprar uma oferta, conforme mostrado no exemplo a seguir.

Para comprar uma instância de banco de dados reservada, use o comando AWS CLI da `purchase-reserved-db-instances-offering` com os seguintes parâmetros:

- `--reserved-db-instances-offering-id` – o ID da oferta que você deseja comprar. Consulte o exemplo anterior para obter o ID da oferta.
- `--reserved-db-instance-id` – você pode atribuir seu próprio identificador às instâncias de banco de dados reservadas adquiridas, a fim de ajudar a manter o controle delas.

Example compra de uma instância de banco de dados reservada

O exemplo a seguir compra a oferta de instância de banco de dados reservada com o ID *649fd0c8-cf6d-47a0-bfa6-060f8e75e95f* e atribui o identificador *MyReservation*.

Para Linux, macOS ou Unix:

```
aws rds purchase-reserved-db-instances-offering \
    --reserved-db-instances-offering-id 649fd0c8-cf6d-47a0-bfa6-060f8e75e95f \
    --reserved-db-instance-id MyReservation
```

Para Windows:

```
aws rds purchase-reserved-db-instances-offering ^
    --reserved-db-instances-offering-id 649fd0c8-cf6d-47a0-bfa6-060f8e75e95f ^
    --reserved-db-instance-id MyReservation
```

Esse comando retorna uma saída semelhante à seguinte:

```
RESERVATION  ReservationId      Class        Multi-AZ  Start Time
 Duration  Fixed Price  Usage Price  Count  State           Description  Offering Type
RESERVATION  MyReservation      db.r3.small  y         2011-12-19T00:30:23.247Z  1y
    455.00 USD   0.092 USD    1      payment-pending  mysql        Partial  Upfront
```

Depois de adquirir instâncias de banco de dados reservadas, você poderá obter informações sobre elas, conforme mostrado no exemplo a seguir.

Para obter informações sobre instâncias de Bancos de Dados reservadas para sua conta da AWS, chame o comando da AWS CLI `describe-reserved-db-instances`, como mostrado no exemplo a seguir.

Example obtenção de suas instâncias de banco de dados reservadas

```
aws rds describe-reserved-db-instances
```

Esse comando retorna uma saída semelhante à seguinte:

```
RESERVATION  ReservationId     Class        Multi-AZ  Start Time
 Duration  Fixed Price  Usage Price  Count  State    Description  Offering Type
RESERVATION  MyReservation     db.r3.small  y         2011-12-09T23:37:44.720Z  1y
   455.00 USD   0.092 USD    1      retired  mysql        Partial  Upfront
```

API do RDS

Você pode usar a API do RDS para trabalhar com instâncias de banco de dados reservadas:

- Para obter informações sobre as ofertas de instâncias de bancos de dados reservadas disponíveis, chame a operação da API do Amazon RDS `DescribeReservedDBInstancesOfferings`.
- Assim que tiver informações sobre as ofertas de instâncias de banco de dados reservadas disponíveis, você poderá usá-las para comprar uma oferta, conforme mostrado no exemplo a

seguir. Chame a operação da API do RDS `PurchaseReservedDBInstancesOffering` com os seguintes parâmetros:

- `--reserved-db-instances-offering-id` – o ID da oferta que você deseja comprar.
- `--reserved-db-instance-id` – você pode atribuir seu próprio identificador às instâncias de banco de dados reservadas adquiridas, a fim de ajudar a manter o controle delas.

- Depois de adquirir instâncias de banco de dados reservadas, você poderá obter informações sobre elas, conforme mostrado no exemplo a seguir. Chame a operação da API do RDS `DescribeReservedDBInstances`.

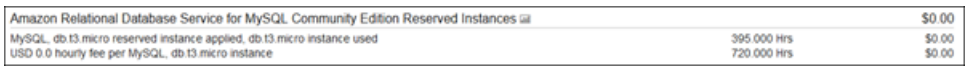
Visualizar a cobrança de instâncias de banco de dados reservadas para o Amazon RDS

É possível visualizar o faturamento das suas instâncias de banco de dados reservadas no Painel de cobrança do AWS Management Console.

Para visualizar a cobrança de instâncias de banco de dados reservadas

1. Faça login no AWS Management Console.
2. No menu da conta no canto superior direito, escolha Billing Dashboard (Painel de faturamento).
3. Escolha Bill Details (Detalhes da fatura) no canto superior direito do painel.
4. Em AWS Service Charges (Cobranças de serviços da), expanda Relational Database Service (Serviço de banco de dados relacional).
5. Expanda a Região da AWS na qual as suas instâncias de banco de dados reservadas se encontram, por exemplo Oeste dos EUA (Oregon).

   Suas instâncias de banco de dados reservadas e suas cobranças por hora no mês atual são mostradas em Instâncias reservadas do Amazon Relational Database Service for ***Mecanismo de banco de dados***.

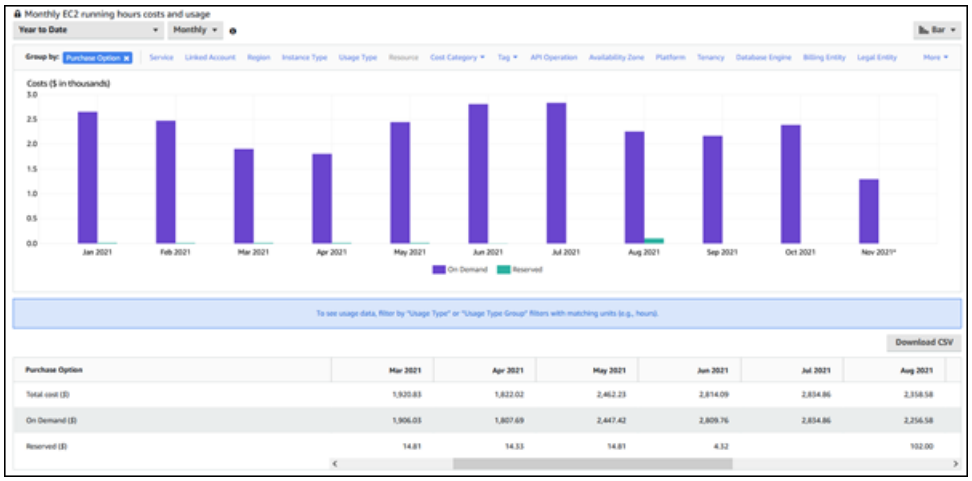
   

   A instância de banco de dados reservada neste exemplo foi comprada com Adiantamento integral e, portanto, não há cobranças por hora.
6. Escolha o ícone do Cost Explorer (gráfico de barras) ao lado do título Reserved Instances (Instâncias reservadas).

O Cost Explorer exibe o gráfico Monthly EC2 running hours costs and usage (Custos e uso por hora mensais de execução do EC2).

7. Limpe o filtro Usage Type Group (Grupo de tipos de uso) à direita do gráfico.
8. Escolha o período e a unidade de tempo para os quais você deseja examinar custos de uso.

O seguinte exemplo mostra os custos mensais acumulados no ano referentes ao uso de instâncias de banco de dados sob demanda e reservadas.



Os custos da instância de banco de dados reservada de janeiro até junho de 2021 referem-se a cobranças mensais para uma instância com Pagamento adiantado parcial, enquanto os custos em agosto de 2021 referem-se a uma cobrança única para uma instância com Adiantamento integral.

O desconto de instâncias reservadas para a instância com Pagamento adiantado parcial expirou em junho de 2021, mas a instância de banco de dados não foi excluída. Após a data de validade, ela foi simplesmente cobrada com base na taxa sob demanda.

Configuração do ambiente do Amazon RDS

Esta página oferece um guia abrangente para configurar o Amazon Relational Database Service, incluindo configuração de contas, segurança e gerenciamento de recursos. Ela orienta você pelas etapas essenciais para criar, gerenciar e proteger os ambientes de banco de dados com eficiência. Se você não conhece o Amazon RDS ou está se preparando para requisitos específicos, estas seções ajudam a garantir que sua configuração seja otimizada e compatível com as práticas recomendadas.

Tópicos

- Inscrever-se para uma Conta da AWS
- Criar um usuário com acesso administrativo
- Conceder acesso programático
- Determinar requisitos
- Fornecer acesso à instância de banco de dados na VPC criando um grupo de segurança

Se você já tem uma Conta da AWS, conhece os requisitos do Amazon RDS e prefere usar os padrões para grupos de segurança do IAM e da VPC, para Conceitos básicos do Amazon RDS.

Inscrever-se para uma Conta da AWS

Se você ainda não tem uma Conta da AWS, siga as etapas abaixo para criar uma.

Como cadastrar uma Conta da AWS

1. Abra https://portal.aws.amazon.com/billing/signup.
2. Siga as instruções online.

   Parte do procedimento de inscrição envolve receber uma chamada telefônica ou uma mensagem de texto e inserir um código de verificação pelo teclado do telefone.

   Quando você se inscreve para uma Conta da AWS, um Usuário raiz da conta da AWS é criado. O usuário-raiz tem acesso a todos os Serviços da AWS e recursos na conta. Como prática recomendada de segurança, atribua o acesso administrativo a um usuário e use somente o usuário-raiz para executar tarefas que exigem acesso de usuário-raiz.

A AWS envia um e-mail de confirmação depois que o processo de inscrição é concluído. A qualquer momento, é possível exibir as atividades da conta atual e gerenciar sua conta acessando https://aws.amazon.com/ e selecionando Minha conta.

Criar um usuário com acesso administrativo

Depois de se cadastrar em uma Conta da AWS, proteja seu Usuário raiz da conta da AWS, habilite o AWS IAM Identity Center e crie um usuário administrativo para não usar o usuário-raiz em tarefas cotidianas.

Proteger o Usuário raiz da conta da AWS

1. Faça login no AWS Management Console como o proprietário da conta ao escolher a opção Usuário-raiz e inserir o endereço de e-mail da Conta da AWS. Na próxima página, insira a senha.

   Para obter ajuda ao fazer login usando o usuário-raiz, consulte Fazer login como usuário-raiz no Guia do usuário do Início de Sessão da AWS.
2. Habilite a autenticação multifator (MFA) para o usuário-raiz.

   Para obter instruções, consulte Habilitar um dispositivo MFA virtual para sua Conta da AWS de usuário-raiz (console) no Guia do usuário do IAM.

Criar um usuário com acesso administrativo

1. Habilita o Centro de Identidade do IAM.

   Para obter instruções, consulte Habilitar o AWS IAM Identity Center no Guia do usuário do AWS IAM Identity Center.
2. No Centro de Identidade do IAM, conceda o acesso administrativo a um usuário.

   Para obter um tutorial sobre como usar o Diretório do Centro de Identidade do IAM como a fonte de identidade, consulte Configurar o acesso dos usuários com o Diretório do Centro de Identidade do IAM padrão no Guia do usuário do AWS IAM Identity Center.

Iniciar sessão como o usuário com acesso administrativo

- Para fazer login com o seu usuário do Centro de Identidade do IAM, use o URL de login enviado ao seu endereço de e-mail quando o usuário do Centro de Identidade do IAM foi criado.

  Para obter ajuda para fazer login usando um usuário do Centro de Identidade do IAM, consulte Fazer login no portal de acesso da AWS, no Guia do usuário do Início de Sessão da AWS.

Atribuir acesso a usuários adicionais

1. No Centro de Identidade do IAM, crie um conjunto de permissões que siga as práticas recomendadas de aplicação de permissões com privilégio mínimo.

   Para obter instruções, consulte Criar um conjunto de permissões no Guia do usuário do AWS IAM Identity Center.
2. Atribua usuários a um grupo e, em seguida, atribua o acesso de autenticação única ao grupo.

   Para obter instruções, consulte Adicionar grupos no Guia do usuário do AWS IAM Identity Center.

Conceder acesso programático

Os usuários precisam de acesso programático se quiserem interagir com a AWS de fora do AWS Management Console. A forma de conceder acesso programático depende do tipo de usuário que está acessando a AWS.

Para conceder acesso programático aos usuários, selecione uma das seguintes opções:

Qual usuário precisa de acesso programático?	Para	Por
Identidade da força de trabalho (Usuários gerenciados no Centro de Identidade do IAM)	Use credenciais temporárias para assinar solicitações programáticas para a AWS CLI, os SDKs da AWS ou as APIs da AWS.	Siga as instruções da interface que deseja utilizar. • Para a AWS CLI, consulte Configuração da AWS CLI para usar o AWS IAM Identity Center no Guia do

Qual usuário precisa de acesso programático?	Para	Por
		usuário da AWS Command Line Interface. • Para os SDKs da AWS, ferramentas e APIs da AWS, consulte Autenticação do Centro de Identidade do IAM no Guia de referência de ferramentas e SDKs da AWS.
IAM	Use credenciais temporárias para assinar solicitações programáticas para a AWS CLI, os SDKs da AWS ou as APIs da AWS.	Siga as instruções em Usar credenciais temporárias com recursos da AWS no Guia do usuário do IAM.

Qual usuário precisa de acesso programático?	Para	Por
IAM	(Não recomendado) Use credenciais de longo prazo para assinar solicitaç ões programáticas para a AWS CLI, os SDKs da AWS ou as APIs da AWS.	Siga as instruções da interface que deseja utilizar. • Para a AWS CLI, consulte Autenticação usando as credenciais de usuário do IAM no Guia do usuário da AWS Command Line Interface. • Para as ferramentas e SDKs da AWS, consulte Autenticação usando as credenciais de longo prazo no Guia de referência de ferramentas e SDKs da AWS. • Para as APIs da AWS, consulte Gerenciamento de chaves de acesso de usuários do IAM no Guia do usuário do IAM.

Determinar requisitos

O bloco de construção básico do Amazon RDS é a instância do banco de dados. Em uma instância de banco de dados, você cria seus bancos de dados. Uma instância de banco de dados fornece um endereço de rede chamado de endpoint. Seus aplicativos usam o endpoint para se conectar à instância de banco de dados. Ao criar a instância de banco de dados, você especifica detalhes como armazenamento, memória, mecanismo e versão de banco de dados, configuração de rede, segurança e períodos de manutenção. Você controla o acesso de rede a uma instância de banco de dados por meio de um grupo de segurança.

Antes de criar uma instância de banco de dados e um grupo de segurança, você precisa conhecer as necessidades de sua instância de banco de dados e de sua rede. Veja aqui alguns fatores importantes a considerar:

- Requisitos de recurso – quais são os requisitos de memória e processador de seu aplicativo ou serviço? Você usa essas configurações para ajudá-lo a determinar que classe de instância de banco de dados deve usar. Para conhecer especificações sobre as classes de instâncias de bancos de dados, consulte Classes de instâncias de banco de dados .
- VPC, sub-rede e grupo de segurança – é mais provável que sua instância de banco de dados esteja uma nuvem privada virtual (VPC). Para se conectar à sua instância de banco de dados, você precisa definir regras de grupo de segurança. Essas regras são definidas diferentemente, dependendo do tipo de VPC que você usa e como a usa. Por exemplo, você pode usar: uma VPC padrão ou uma VPC definida pelo usuário.

  A lista a seguir descreve as regras para cada opção da VPC:

  - VPC padrão: se sua conta da AWS tiver uma VPC padrão na região atual da AWS, essa VPC estará configurada para oferecer suporte a instâncias de bancos de dados. Se você especificar a VPC padrão ao criar a instância de banco de dados, faça o seguinte:
    - Você deve criar um grupo de segurança da VPC que autorize conexões da aplicação ou serviço para a instância de banco de dados do Amazon RDS. Use a opção Security Group (Grupo de segurança) no console da VPC ou na AWS CLI para criar grupos de segurança da VPC. Para obter mais informações, consulte Etapa 3: Criar um grupo de segurança da VPC.
    - Especifique o grupo de sub-redes de banco de dados padrão. Se esta for a primeira instância de banco de dados que você criou na região da AWS, o Amazon RDS criará o grupo de sub-redes de banco de dados padrão quando criar a instância de banco de dados.
  - VPC definida pelo usuário – se quiser especificar uma VPC definida pelo usuário ao criar uma instância de banco de dados, esteja ciente do seguinte:
    - Você deve criar um grupo de segurança da VPC que autorize conexões da aplicação ou serviço para a instância de banco de dados do Amazon RDS. Use a opção Security Group (Grupo de segurança) no console da VPC ou na AWS CLI para criar grupos de segurança da VPC. Para obter mais informações, consulte Etapa 3: Criar um grupo de segurança da VPC.
    - A VPC deve atender a certos requisitos para hospedar instâncias de bancos de dados, como ter pelo menos duas sub-redes, cada uma em uma zona de disponibilidade separada. Para obter mais informações, consulte Amazon VPC e Amazon RDS.
    - Você deve especificar um grupo de sub-redes de banco de dados que defina quais sub-redes nessa VPC podem ser usadas pela instância de banco de dados. Para obter informações,

consulte a seção sobre grupos de sub-redes de banco de dados, em Trabalhar com uma instância de banco de dados em uma VPC.

- Alta disponibilidade: você precisa de suporte a failover? No Amazon RDS, uma implantação multi-AZ cria uma instância de banco de dados primária e uma instância de banco de dados em espera (secundária) em outra zona de disponibilidade para suporte de failover. Para manter a alta disponibilidade, recomendamos as implantações multi-AZ para cargas de trabalho de produção. Para fins de desenvolvimento e teste, você pode usar uma implantação que não seja multi-AZ. Para obter mais informações, consulte Configurar e gerenciar uma implantação multi-AZ para o Amazon RDS.
- Políticas do IAM: sua conta da AWS tem políticas que concedem as permissões necessárias para executar operações do Amazon RDS? Quando você se conecta à AWS usando credenciais do IAM, sua conta do IAM deve ter políticas do IAM que concedam as permissões necessárias para realizar operações do Amazon RDS. Para obter mais informações, consulte Gerenciamento de identidade e acesso no Amazon RDS.
- Portas abertas: em que porta TCP/IP seu banco de dados ouve? O firewall de algumas empresas pode bloquear conexões com a porta padrão para o seu mecanismo de banco de dados. Se o firewall da sua empresa bloquear a porta padrão, escolha outra porta para a nova instância de banco de dados. Depois de criar uma instância de banco de dados que escuta em uma porta específica, você pode alterar essa porta modificando a instância de banco de dados.
- Região da AWS: em qual região da AWS você quer seu banco de dados? Ter o banco de dados próximo do aplicativo ou do serviço Web pode reduzir a latência da rede. Para obter mais informações, consulte Regiões, zonas de disponibilidade e Local Zones.
- Subsistema de disco de banco de dados: quais são suas necessidades de armazenamento? O Amazon RDS fornece três tipos de armazenamento:
  - Finalidade geral (SSD)
  - IOPS provisionadas (PIOPS)
  - Magnético (também conhecido como armazenamento padrão)

  Para obter mais informações sobre o armazenamento do Amazon RDS, consulte Armazenamento de instâncias de banco de dados do Amazon RDS.

Quando você tiver as informações necessárias para criar o grupo de segurança e a instância de banco de dados, continue na próxima etapa.

Fornecer acesso à instância de banco de dados na VPC criando um grupo de segurança

Os grupo de seguranças de VPC oferecem acesso a instância de banco de dados em uma VPC. Eles atuam como um firewall para a instância de banco de dados associada, controlando o tráfego de entrada e de saída no nível da instância de banco de dados. As instâncias de bancos de dados são criadas por padrão com um firewall e um grupo de segurança padrão que protege a instância de banco de dados.

Para conseguir se conectar à sua instância de banco de dados, você deve adicionar regras a um grupo de segurança que permita que você se conecte. Use suas informações de rede e configuração para criar regras e permitir acesso à sua instância de banco de dados.

Por exemplo, suponhamos que você tenha um aplicativo que acesse um banco de dados na sua instância de banco de dados em uma VPC. Neste caso, você deve adicionar uma regra TCP personalizada que especifique o intervalo da porta e os endereços IP que seu aplicativo usa para acessar o banco de dados. Se tiver um aplicativo uma instância do Amazon EC2, você poderá usar o grupo de segurança configurado para a instância do Amazon EC2.

Você pode configurar a conectividade entre uma instância do Amazon EC2 e uma instância de banco de dados ao criar a instância de banco de dados. Para obter mais informações, consulte Configurar a conectividade automática de rede com uma instância do EC2.

> **Tip**
>
> Você pode configurar a conectividade de rede entre uma instância do Amazon EC2 e uma instância de banco de dados automaticamente ao criar a instância de banco de dados. Para obter mais informações, consulte Configurar a conectividade automática de rede com uma instância do EC2.

Para obter informações sobre como conectar recursos no Amazon Lightsail a instâncias de banco de dados, consulte Conectar recursos do Lightsail aos Serviços da AWS usando peering de VPC.

Para obter informações sobre cenários comuns para acessar uma instância de banco de dados, consulte Cenários para acessar uma instância de banco de dados em uma VPC.

Para criar um grupo de segurança de VPC

1. Faça login no AWS Management Console e abra o console da Amazon VPC em https://console.aws.amazon.com/vpc.

   > **Note**
   >
   > Verifique se você está no console da VPC, não no console do RDS.

2. No canto superior direito do AWS Management Console, selecione a região da AWS na qual deseja criar o grupo de segurança da VPC e a instância de banco de dados. Na lista de recursos da Amazon VPC para essa região da AWS, você deve ver pelo menos uma VPC e várias sub-redes. Caso contrário, significa que não há uma VPC padrão na região da AWS em questão.
3. No painel de navegação, escolha Grupos de segurança.
4. Escolha Create grupo de segurança (Criar grupo de segurança).

   A página Create grupo de segurança (Criar grupo de segurança) é exibida.
5. Em Basic details (Detalhes básicos), insira o Security group name (Nome do grupo de segurança) e a Description (Descrição). Para VPC escolha a VPC na qual você deseja criar sua instância de banco de dados.
6. Em Inbound rules (Regras de entrada), escolha Add rule (Adicionar regra).

   a. Em Type (Tipo), escolha Custom TCP (TCP personalizada).

   b. Em Port Range (Intervalo de portas), digite o valor da porta a ser usada para sua instância de banco de dados.

   c. Em Source (Origem), selecione um nome de grupo de segurança ou digite o intervalo de endereços IP (valor CIDR) de onde você acessará a instância. Se você selecionar My IP (Meu IP), isso concederá acesso à instância de banco de dados do endereço IP detectado no navegador.
7. Se você precisar adicionar mais endereços IP ou intervalos de portas diferentes, escolha Add rule (Adicionar regra) e insira as informações para a regra.
8. (Opcional) Em Outbound rules (Regras de saída), adicione regras para o tráfego de saída. Por padrão, todo tráfego de saída é permitido.
9. Escolha Create grupo de segurança (Criar grupo de segurança).

Você pode usar o grupo de segurança de VPC que acabou de criar como o grupo de segurança de sua instância de banco de dados quando a criar.

> **Note**
>
> Se usar uma VPC padrão, será criada para você um grupo de sub-redes padrão distribuídas por todas as sub-redes da VPC. Quando cria uma instância de banco de dados, você pode selecionar a VPC padrão e usar default (padrão) em DB Subnet Group (Grupo de sub-redes de banco de dados).

Assim que completar os requisitos de configuração, você poderá criar uma instância de banco de dados usando seus requisitos e grupo de segurança. Para isso, siga as declarações em Criar uma instância de banco de dados do Amazon RDS. Para obter informações sobre como começar criando uma instância de banco de dados que usa um mecanismo de banco de dados específico, consulte a documentação relevante na tabela a seguir.

Mecanismo do banco de dados	Documentação
MariaDB	Criar e conectar uma instância de banco de dados MariaDB
Microsoft SQL Server	Criar uma instância de banco de dados do Microsoft SQL Server e conectar-se a ela
MySQL	Criar e conectar a uma instância de banco de dados MySQL
Oracle	Criar uma instância de banco de dados Oracle e conectar-se a ela
PostgreSQL	Criar uma instância de banco de dados PostgreSQL e conectar-se a ela

> **Note**
>
> Se você não conseguir se conectar a uma instância de banco de dados depois de criá-la, consulte as informações de solução de problemas em Não é possível conectar-se à instância de banco de dados do Amazon RDS.

Conceitos básicos do Amazon RDS

Nos exemplos a seguir, é possível descobrir como criar e se conectar a uma instância de banco de dados usando o Amazon Relational Database Service (Amazon RDS). É possível criar uma instância de banco de dados que use Db2, MariaDB, MySQL, Microsoft SQL Server, Oracle ou PostgreSQL.

> **Important**
>
> É necessário concluir as tarefas em Configuração do ambiente do Amazon RDS antes de criar ou se conectar a uma instância de banco de dados.

Criar uma instância de banco de dados e se conectar a um banco de dados em uma instância de banco de dados é um pouco diferente para cada um dos mecanismos de banco de dados. Escolha um dos seguintes mecanismo de banco de dados que você deseja usar para obter informações detalhadas sobre como criar e se conectar à instância de banco de dados. Depois de ter criado e se conectado à sua instância de banco de dados, há instruções para ajudá-lo a excluir essa instância de banco de dados.

Tópicos

- Criar e conectar uma instância de banco de dados MariaDB
- Criar uma instância de banco de dados do Microsoft SQL Server e conectar-se a ela
- Criar e conectar a uma instância de banco de dados MySQL
- Criar uma instância de banco de dados Oracle e conectar-se a ela
- Criar uma instância de banco de dados PostgreSQL e conectar-se a ela
- Tutorial: crie um servidor Web e uma instância de banco de dados do Amazon RDS
- Tutorial: Usar uma função do Lambda para acessar um banco de dados Amazon RDS

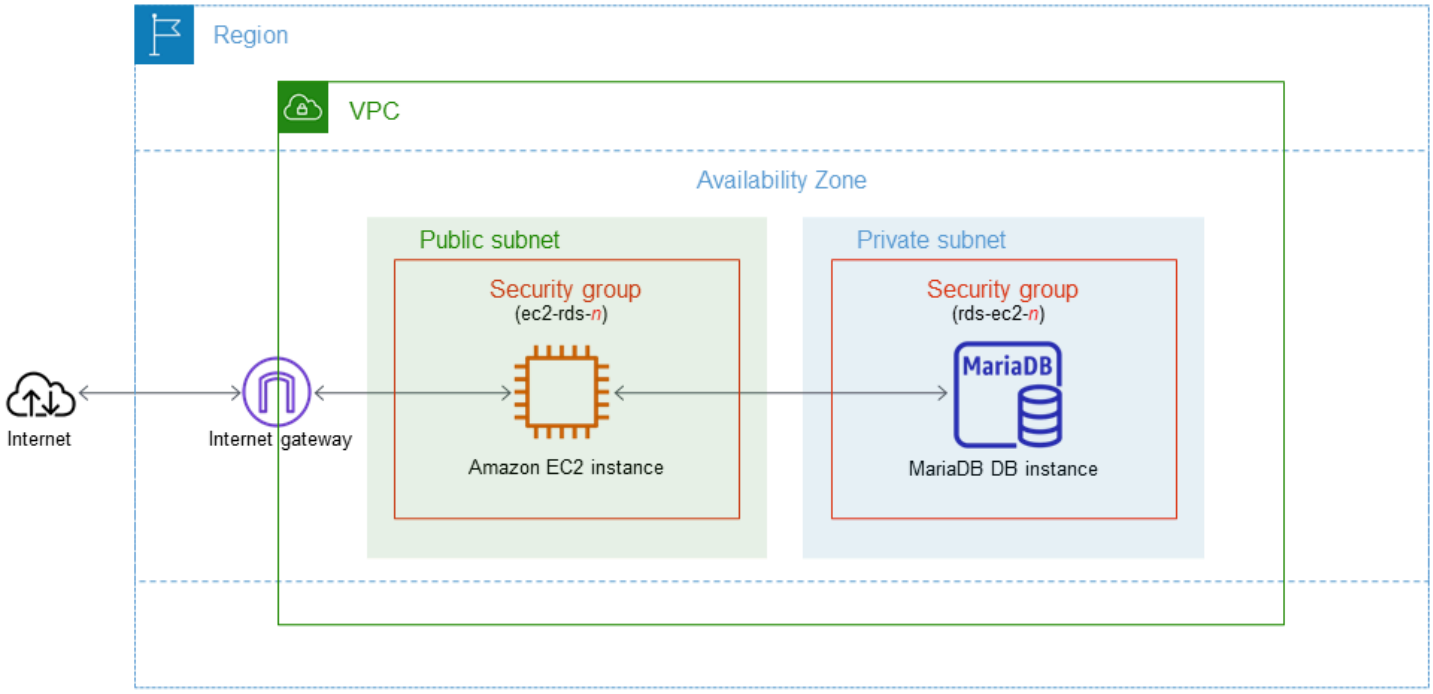
Criar e conectar uma instância de banco de dados MariaDB

Este tutorial cria uma instância do EC2 e uma instância de banco de dados do RDS para MariaDB. O tutorial mostra como acessar a instância de banco de dados pela instância do EC2 usando um cliente MySQL padrão. Como prática recomendada, este tutorial cria uma instância de banco de dados privada em uma nuvem privada virtual (VPC). Na maioria dos casos, outros recursos na mesma VPC, como instâncias do EC2, podem acessar a instância de banco de dados, mas recursos fora da VPC não podem acessá-la.

Depois de concluir o tutorial, haverá uma sub-rede pública e privada em cada zona de disponibilidade na sua VPC. Em uma zona de disponibilidade, a instância do EC2 está na sub-rede pública e a instância de banco de dados está na sub-rede privada.

> **Important**
>
> Não há cobrança para criar uma Conta da AWS. No entanto, ao concluir este tutorial, é possível gerar custos para os recursos da que você usa. Se esses recursos não forem mais necessários após a conclusão do tutorial, você poderá excluí-los.

O diagrama a seguir mostrará a configuração quando o tutorial estiver completo.



Esse tutorial permite criar recursos usando um dos seguintes métodos:

1. Use o AWS Management Console: Criar uma instância do EC2 e Criar uma instância de banco de dados do MariaDB
2. Use o AWS CloudFormation para criar a instância de banco de dados e a instância do EC2: (Opcional) Criar VPC, instância do EC2 e instância do MariaDB usando o AWS CloudFormation

O primeiro método usa Criação fácil para criar uma instância de banco de dados privada do MariaDB com o AWS Management Console. Aqui, você especifica somente o tipo de mecanismo de banco de dados, o tamanho da instância de banco de dados e o identificador da instância de banco de dados. A opção Easy create (Criação fácil) usa a configuração padrão para as outras opções de configuração.

Ao usar a opção Criação padrão, é possível especificar mais opções de configuração ao criar uma instância de banco de dados. Essas opções incluem configurações de disponibilidade, segurança, backups e manutenção. Para criar uma instância de banco de dados pública, você deve usar a Criação padrão. Para mais informações, consulte Criar uma instância de banco de dados do Amazon RDS.

Tópicos

- Pré-requisitos
- Criar uma instância do EC2
- Criar uma instância de banco de dados do MariaDB
- (Opcional) Criar VPC, instância do EC2 e instância do MariaDB usando o AWS CloudFormation
- Conectar-se a uma instância de banco de dados MariaDB
- Excluir a instância do EC2 e a instância de banco de dados
- (Opcional) Excluir a instância do EC2 e a instância de banco de dados criada com o CloudFormation
- (Opcional) Conectar a instância de banco de dados a uma função do Lambda

Pré-requisitos

Antes de começar, conclua as etapas nas seguintes seções:

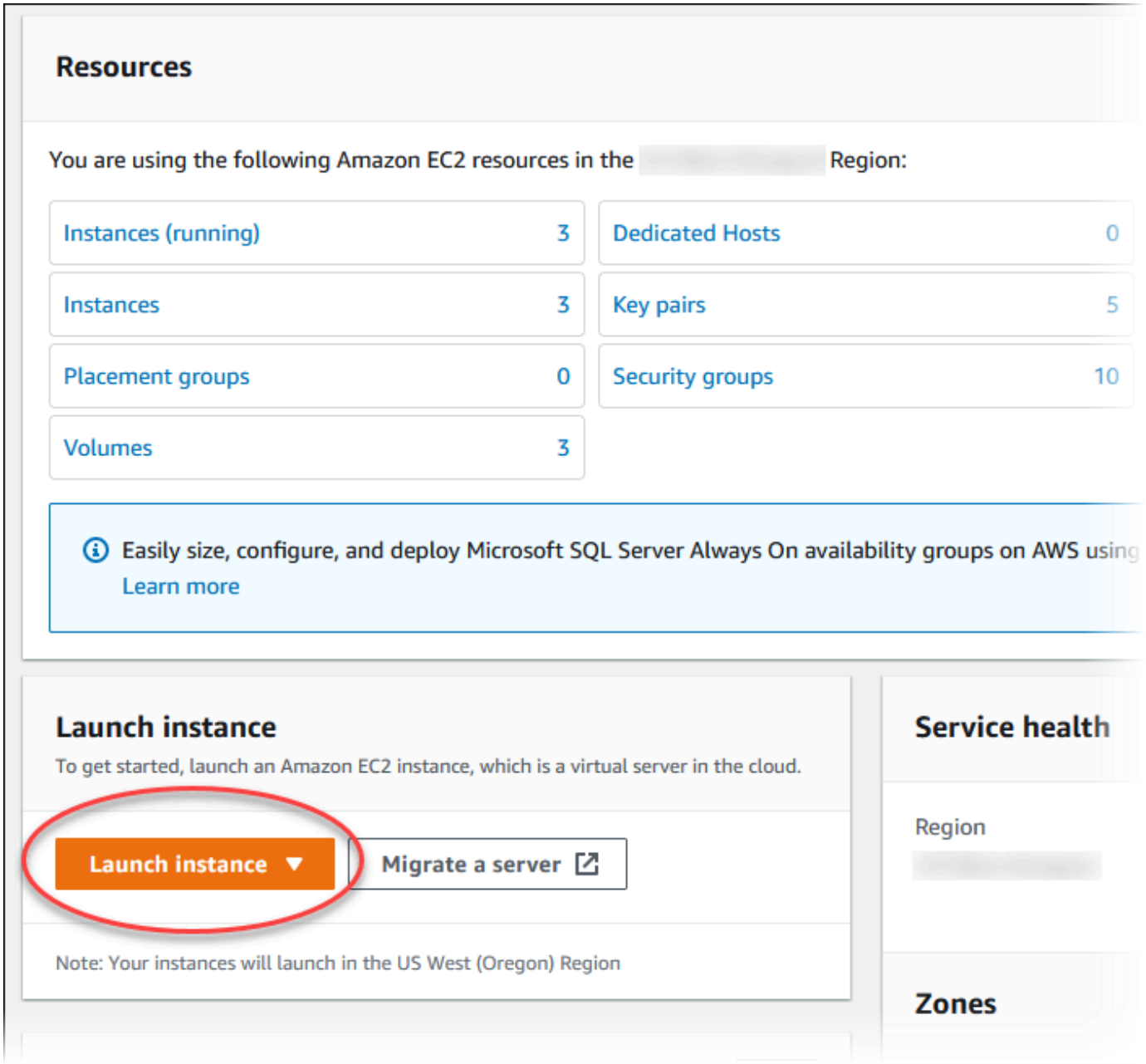
- Inscrever-se para uma Conta da AWS
- Criar um usuário com acesso administrativo

Criar uma instância do EC2

Crie uma instância do Amazon EC2 que você usará para se conectar ao banco de dados.

Para criar uma instância do EC2

1. Faça login no AWS Management Console e abra o console do Amazon EC2 em https://console.aws.amazon.com/ec2/.
2. No canto superior direito do AWS Management Console, selecione a Região da AWS em que você deseja criar a instância do EC2.
3. Escolha Painel do EC2 e Executar instância, conforme mostrado na imagem a seguir.

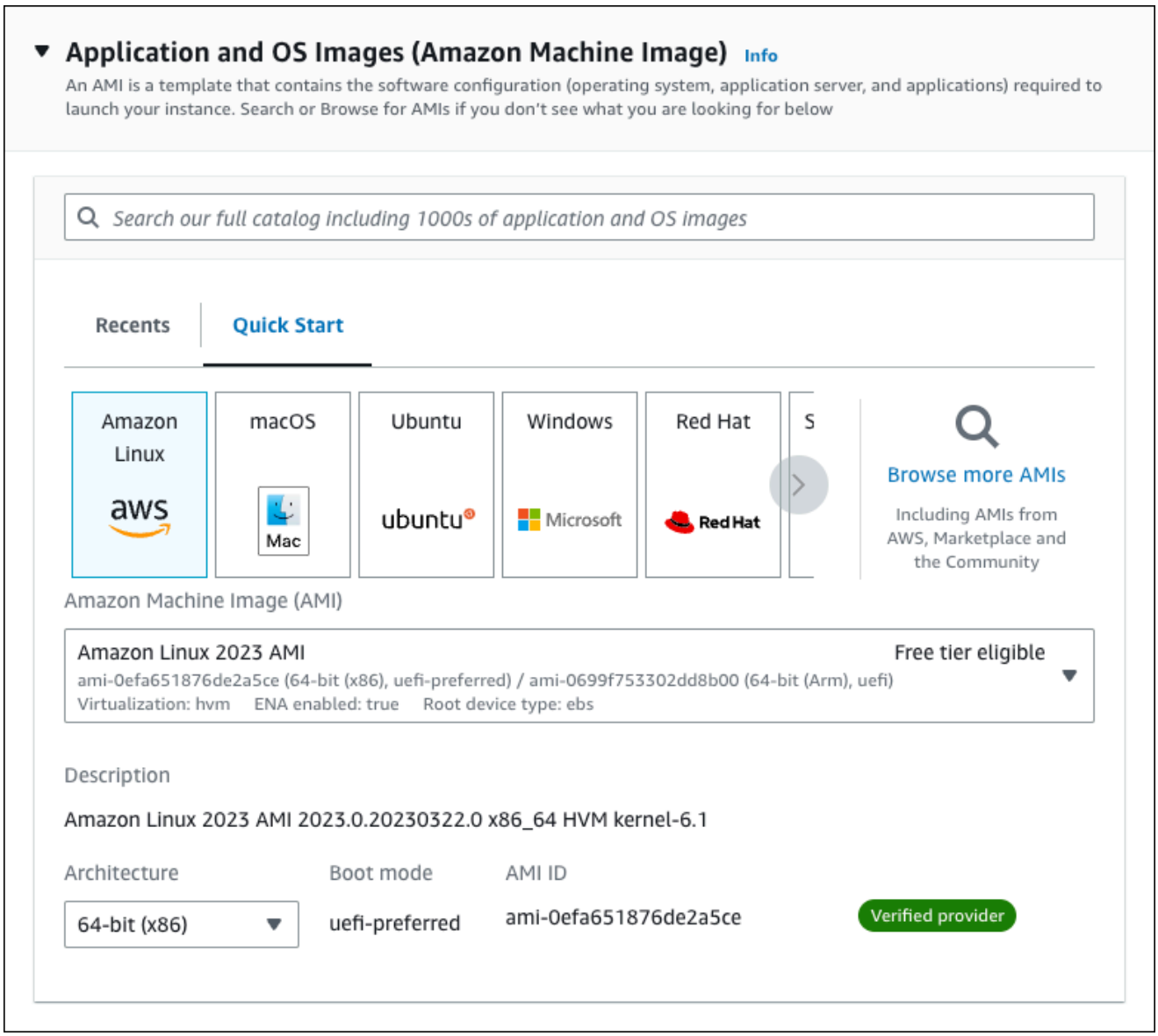


A página Iniciar uma instância é aberta.

4. Escolha as configurações a seguir na página Iniciar uma instância.

   a. Em Name and tags (Nome e etiquetas), em Name (Nome), insira **`ec2-database-connect`**.

   b. Em Imagens de aplicações e sistemas operacionais (imagem de máquina da Amazon), selecione Amazon Linux e, depois, AMI do Amazon Linux 2023. Mantenha as seleções padrão nas outras opções.



c. Em Instance type (Tipo de instância), escolha t2.micro.

d. Em Key pair (login) (Par de chaves (login)), escolha um Key pair name (Nome do par de chaves) para usar um par de chaves existente. Para criar um par de chaves para a instância do Amazon EC2, escolha Create new key pair (Criar um novo key pair), depois use a janela Create key pair (Criar par de chaves) para criá-lo.

   Consulte mais informações sobre como criar um par de chaves em Criar um par de chaves no Guia do usuário do Amazon EC2.

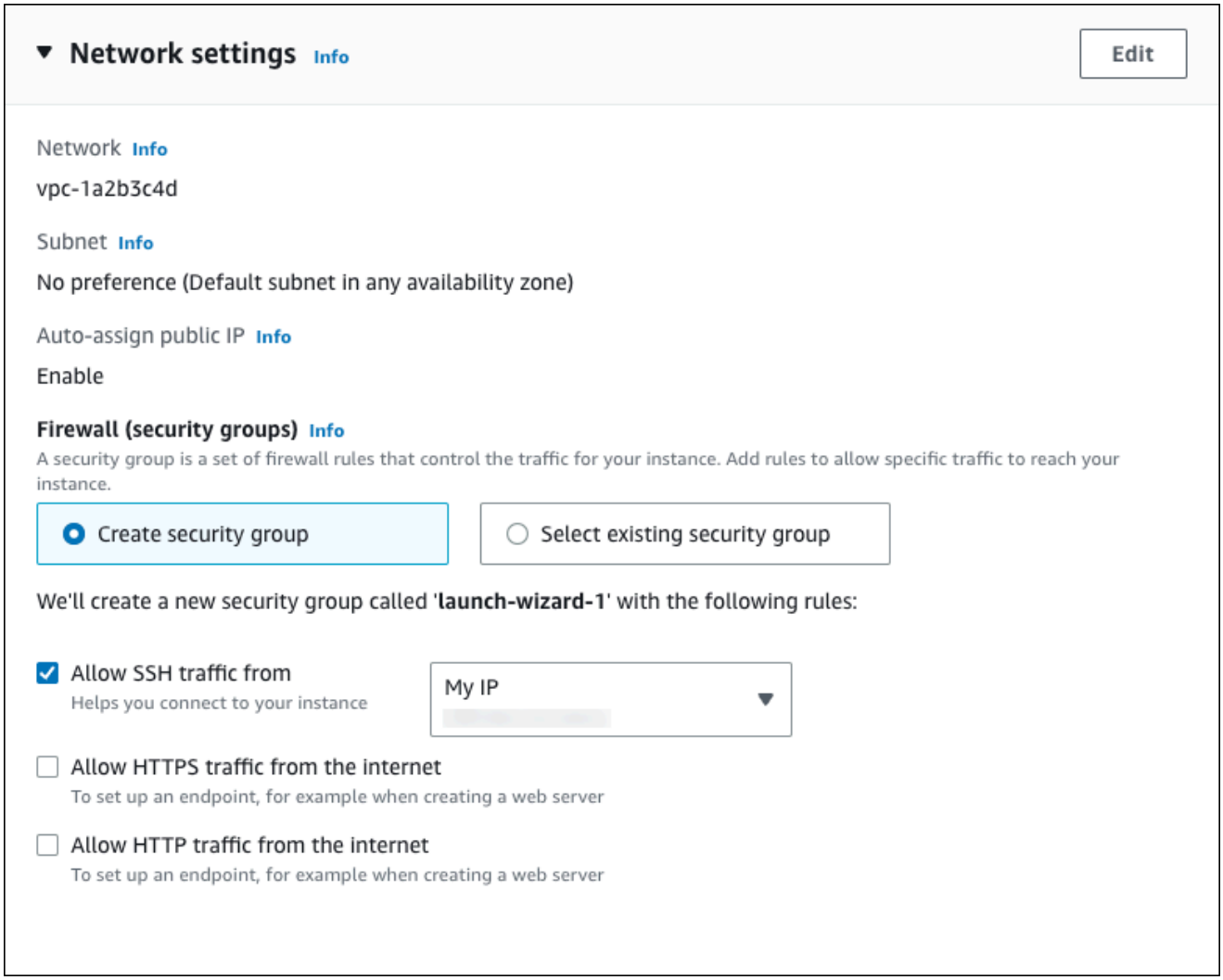
e. Para Permitir tráfego SSH, em Configurações de rede, selecione a origem das conexões SSH com a instância do EC2.

Você pode escolher My IP (Meu IP) se o endereço IP exibido estiver correto para conexões SSH. Caso contrário, determine o endereço IP a ser usado para se conectar a instâncias do EC2 em sua VPC usando Secure Shell (SSH). Para determinar seu endereço IP público, em uma janela ou guia diferente do navegador, é possível usar o serviço em https://checkip.amazonaws.com. Um exemplo de endereço IP é 192.0.2.1/32.

Em muitos casos, você pode se conectar por meio de um provedor de serviços de Internet (ISP) ou atrás de um firewall sem um endereço IP estático. Em caso afirmativo, determine o intervalo de endereços IP utilizado por computadores cliente.

> **Warning**
>
> Se usar `0.0.0.0/0` para acesso do SSH, você possibilitará que todos os endereços IP acessem suas instâncias públicas do EC2 usando SSH. Essa abordagem é aceitável por um período curto em um ambiente de teste, mas não é seguro em ambientes de produção. Em produção, autorize somente um endereço IP específico ou um intervalo de endereços para acessar suas instâncias do EC2 usando SSH.

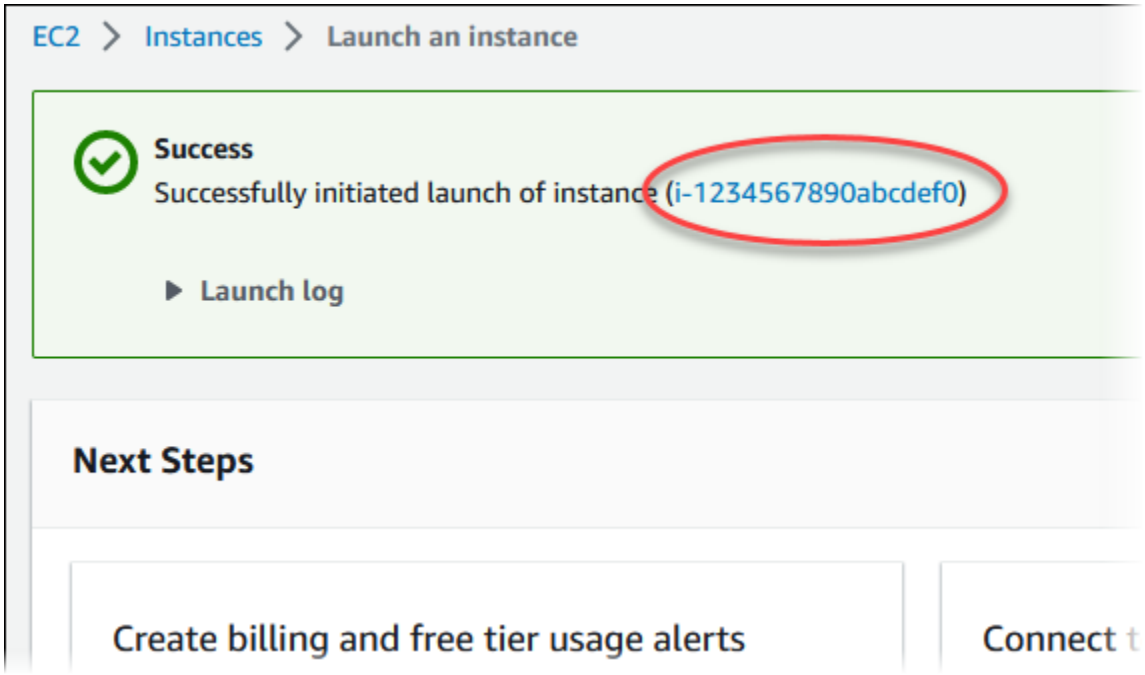
A imagem a seguir mostra um exemplo da seção Configurações de rede.



f. Mantenha os valores padrão para as seções restantes.

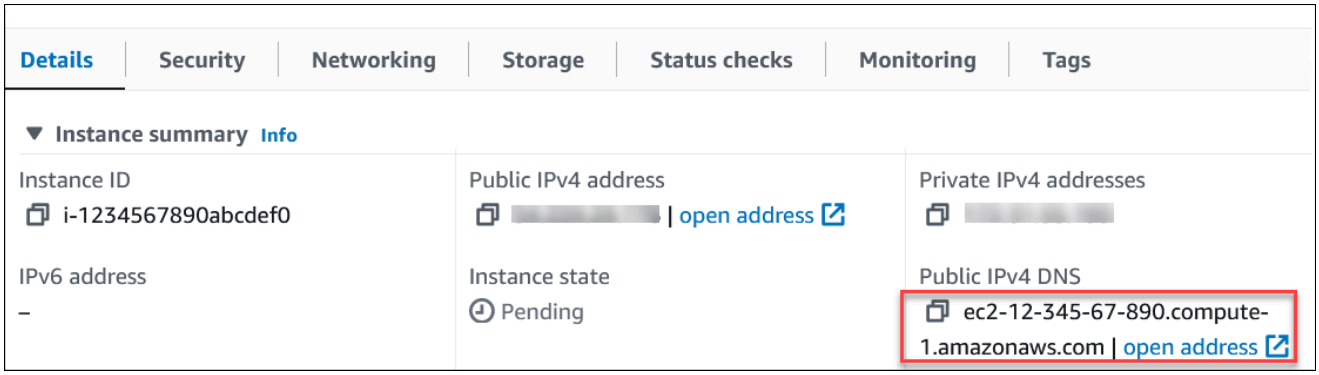
g. Revise um resumo da configuração da instância do EC2 no painel Resumo e, quando estiver com tudo pronto, escolha Executar instância.

5. Na página Status de inicialização, anote o identificador de sua nova instância do EC2, por exemplo: `i-1234567890abcdef0`.

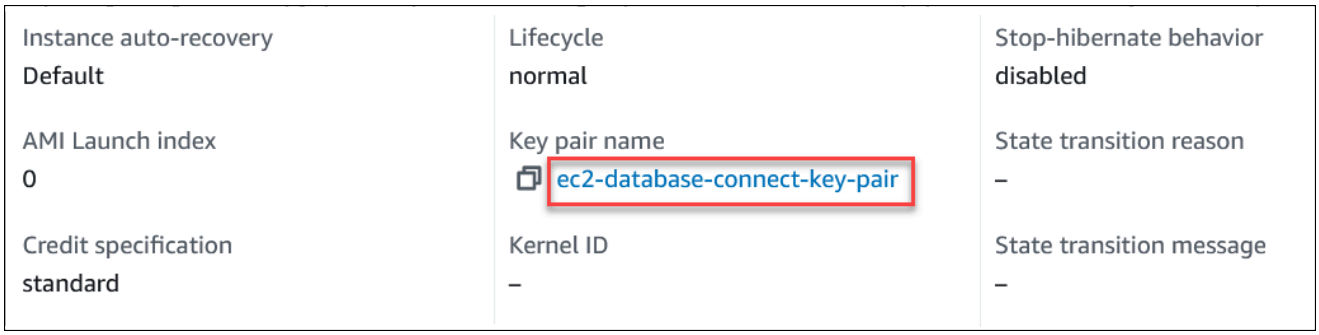


6. Selecione o identificador de instância do EC2 para abrir a lista de instâncias do EC2 e, depois, selecione sua instância do EC2.
7. Na guia Detalhes, observe os seguintes valores, necessários ao se conectar utilizando SSH:

   a. No Resumo da instância, observe o valor do DNS IPv4 público.



   b. Em Detalhes da instância, observe o valor do nome do par de chaves.



8. Espere até o Estado da instância para a instância do EC2 ficar como Executando antes de continuar.

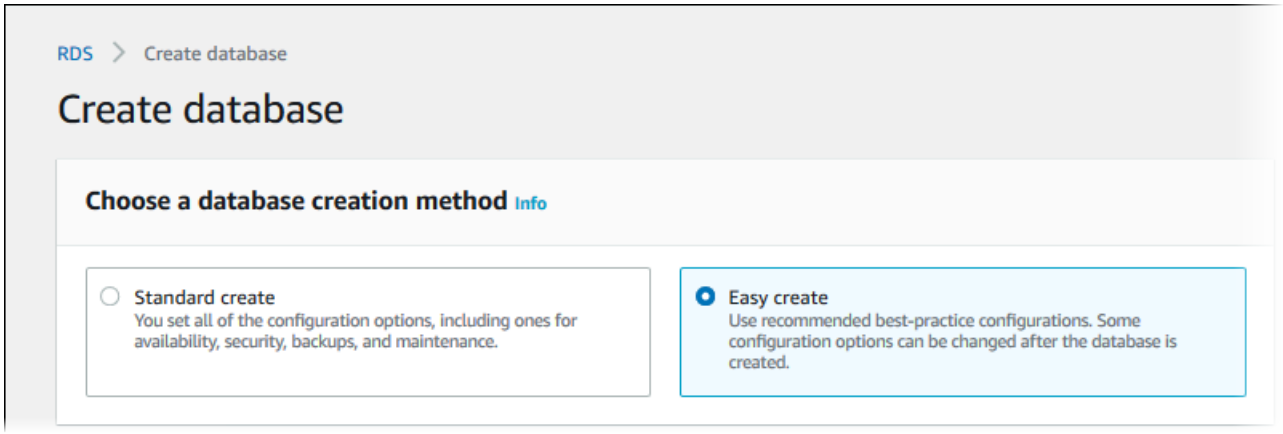
Criar uma instância de banco de dados do MariaDB

O bloco de construção básico do Amazon RDS é a instância do banco de dados. Este ambiente é onde você executa seus bancos de dados MariaDB.

Neste exemplo, use a opção Criação fácil para criar uma instância de banco de dados executando o mecanismo de banco de dados MariaDB com uma classe de instância de banco de dados db.t4g.micro.

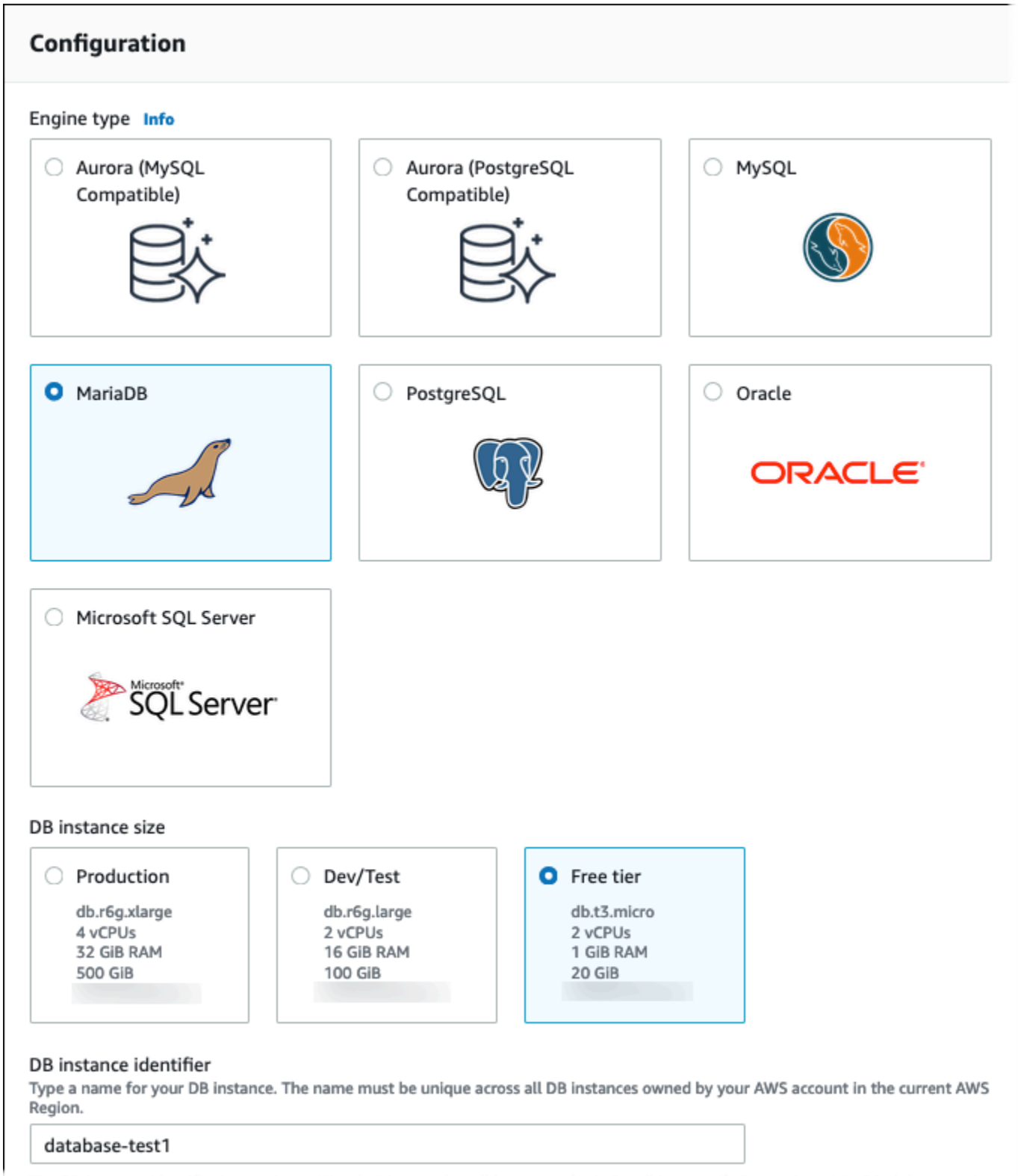
Para criar uma instância de banco de dados MariaDB com a criação fácil

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No canto superior direito do console do Amazon RDS, escolha a Região da AWS na qual você deseja criar a instância de banco de dados.
3. No painel de navegação, escolha Bancos de dados.
4. Selecione Criar banco de dados e escolha Criação fácil.



5. Em Configuration (Configuração), escolha MariaDB.
6. Em Tamanho da instância de banco de dados, escolha Nível gratuito ou Sandbox. Nível gratuito aparece para contas do plano gratuito. Sandbox aparece para contas do plano pago.
7. Em Identificador de instância de banco de dados, insira **`database-test1`**.
8. Em Nome do usuário principal, insira um nome para o usuário mestre ou deixe o nome padrão.

A página Create database (Criar banco de dados) deve ser semelhante à imagem a seguir. Para contas do plano gratuito, aparece Nível gratuito. Para contas do plano pago, aparece Sandbox.

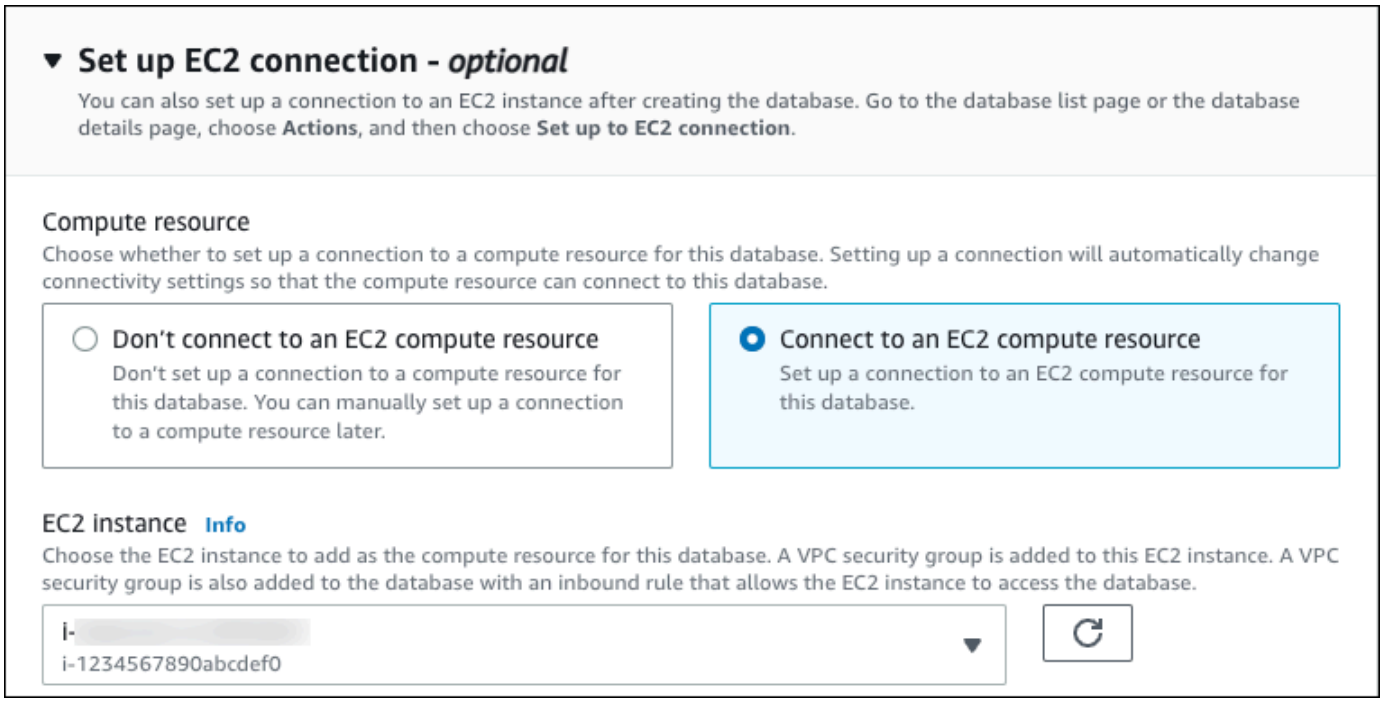


9. Para usar uma senha primária gerada automaticamente para a instância de banco de dados, marque a opção Gerar uma senha automaticamente.

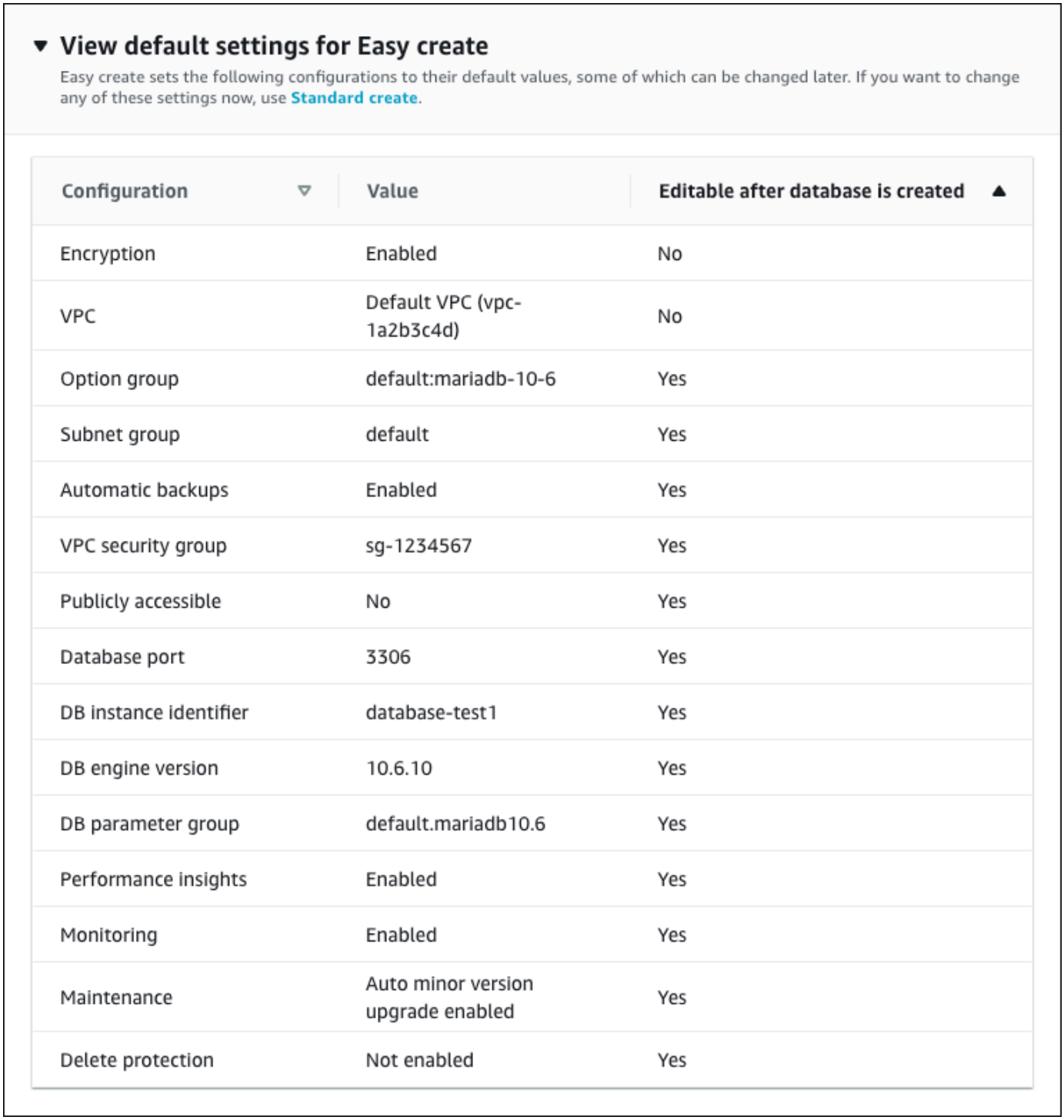
   Para inserir sua senha primária, desmarque a caixa de seleção Gerar uma senha automaticamente e insira a mesma senha em Senha primária e em Confirmar senha.

10. Para configurar uma conexão com a instância do EC2 que você criou anteriormente, expanda Configurar conexão do EC2: opcional.

    Selecione Conectar-se a um recurso computacional do EC2. Selecione a instância do EC2 que você criou anteriormente.



11. Expanda Visualizar as configurações padrão da criação fácil.



View default settings for Easy create

Easy create sets the following configurations to their default values, some of which can be changed later. If you want to change any of these settings now, use **Standard create**.

Configuration	Value	Editable after database is created
Encryption	Enabled	No
VPC	Default VPC (vpc-1a2b3c4d)	No
Option group	default:mariadb-10-6	Yes
Subnet group	default	Yes
Automatic backups	Enabled	Yes
VPC security group	sg-1234567	Yes
Publicly accessible	No	Yes
Database port	3306	Yes
DB instance identifier	database-test1	Yes
DB engine version	10.6.10	Yes
DB parameter group	default.mariadb10.6	Yes
Performance insights	Enabled	Yes
Monitoring	Enabled	Yes
Maintenance	Auto minor version upgrade enabled	Yes
Delete protection	Not enabled	Yes

Você pode examinar as configurações padrão usadas com a opção Easy Create (Criação fácil). A coluna Editável após a criação do banco de dados mostra quais opções podem ser alteradas após a criação do banco de dados.

- Se uma configuração tiver Não nessa coluna e você quiser uma configuração diferente, poderá usar a opção Criação padrão para criar a instância de banco de dados.
- Se uma configuração tiver Sim nessa coluna e você quiser uma configuração diferente, poderá usar a opção Criação padrão para criar a instância de banco de dados, ou modificar a instância de banco de dados depois de criá-la para alterar a configuração.

12. Selecione Criar banco de dados.

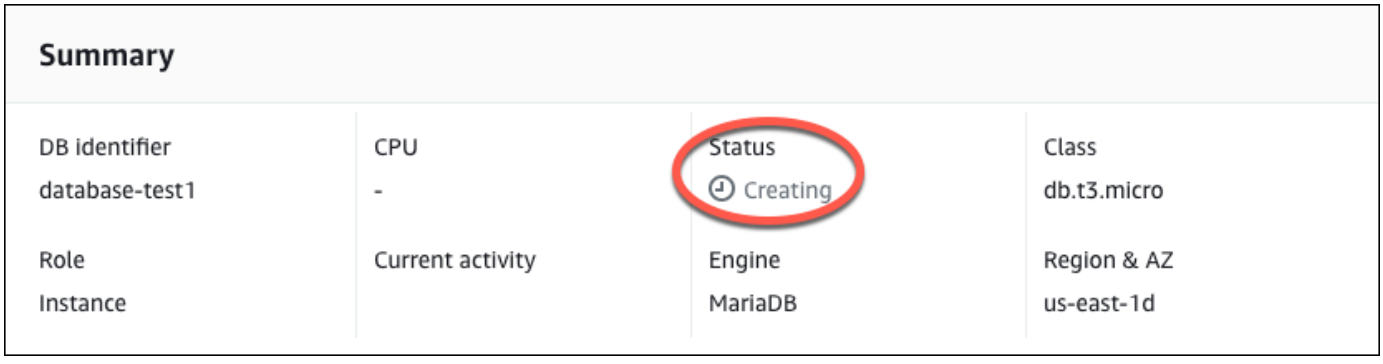
    Para exibir o nome de usuário mestre e a senha da instância de banco de dados, escolha View credential details (Ver detalhes da credencial).

    Use o nome de usuário e a senha que aparecem para se conectar à instância de banco de dados como o usuário mestre.

    > **Important**
    >
    > Você não pode visualizar a senha do usuário principal novamente. Caso você não a registre, talvez seja necessário alterá-la.
    > Se for necessário alterar a senha do usuário mestre depois que a instância de banco de dados estiver disponível, será possível modificar a instância de banco de dados para fazer isso. Para ter mais informações sobre a modificação de uma instância de banco de dados, consulte Modificar uma instância de banco de dados do Amazon RDS.

13. Na lista Bancos de dados, escolha o nome da nova instância de banco de dados MariaDB para mostrar seus detalhes.

    A instância de banco de dados permanecerá no status Criando até estar pronta para ser usada.



Quando o status muda para Available (Disponível), você pode se conectar à instância de banco de dados. Dependendo da classe da instância de banco de dados e da quantidade de armazenamento, pode levar até 20 minutos para que a nova instância esteja disponível.

(Opcional) Criar VPC, instância do EC2 e instância do MariaDB usando o AWS CloudFormation

Em vez de usar o console para criar a VPC, a instância do EC2 e a instância do MariaDB, você pode usar o AWS CloudFormation para provisionar recursos da AWS tratando a infraestrutura como código. Para ajudar a organizar os recursos da AWS em unidades menores e mais gerenciáveis, você pode usar a funcionalidade de pilha aninhada do AWS CloudFormation. Consulte mais informações em Criar uma pilha no console do AWS CloudFormation e Trabalhar com pilhas aninhadas.

> **Important**
>
> O AWS CloudFormation é gratuito, mas os recursos que o CloudFormation são ativos. Você incorre nas taxas de uso padrão para esses recursos até que os encerre. Para ter mais informações, consulte Preço do Amazon RDS para MariaDB.

- Baixar o modelo do CloudFormation
- Configurar recursos usando o CloudFormation

Baixar o modelo do CloudFormation

Um modelo do CloudFormation é um arquivo de texto JSON ou YAML que contém as informações da configuração dos recursos que você deseja criar na pilha. Esse modelo também cria uma VPC e um bastion host para você junto com a instância do RDS.

Para baixar o arquivo de modelo, abra o link MariaDB CloudFormation template.

Na página do Github, clique no botão Baixar arquivo bruto para salvar o arquivo YAML do modelo.

Configurar recursos usando o CloudFormation

> **Note**
>
> Antes de iniciar esse processo, verifique se você tem um par de chaves para uma instância do EC2 na Conta da AWS. Para obter mais informações, consulte Pares de chaves do Amazon EC2 e instâncias do Linux.

Ao usar o modelo do AWS CloudFormation, você deve selecionar os parâmetros certos para garantir que os recursos sejam criados corretamente. Siga as etapas abaixo:

1. Faça login no AWS Management Console e abra o console AWS CloudFormation em https://console.aws.amazon.com/cloudformation.
2. Selecione Create Stack (Criar pilha).
3. Na seção Especificar modelo, selecione Fazer upload de um arquivo de modelo do computador e escolha Próximo.
4. Na página Especificar detalhes da pilha, defina os seguintes parâmetros:
   a. Defina o Nome da pilha como MariaDBTestStack.
   b. Em Parâmetros, defina Zonas de disponibilidade selecionando três zonas de disponibilidade.
   c. Em Configuração do bastion host do Linux, em Nome da chave, selecione um par de chaves para fazer login na instância do EC2.
   d. Em Configurações do bastion host do Linux, defina o Intervalo de IP permitido para o endereço IP. Para conectar instâncias do EC2 à VPC usando o Secure Shell (SSH), determine o endereço IP público usando o serviço em https://checkip.amazonaws.com. Um exemplo de endereço IP é 192.0.2.1/32.

> **Warning**
>
> Se usar `0.0.0.0/0` para acesso do SSH, você possibilitará que todos os endereços IP acessem suas instâncias públicas do EC2 usando SSH. Essa abordagem é aceitável por um período curto em um ambiente de teste, mas não é seguro em ambientes de produção. Em produção, autorize somente um endereço IP específico ou um intervalo de endereços para acessar suas instâncias do EC2 usando SSH.

e. Em Configuração geral do banco de dados, defina a Classe da instância do banco de dados como db.t3.micro.

f. Defina o Nome do banco de dados como **`database-test1`**.

g. Em Nome de usuário principal do banco de dados, insira um nome para o usuário principal.

h. Defina a Gerenciar senha de usuário principal do banco de dados com o Secrets Manager como `false` para esse tutorial.

i. Em Senha do banco de dados, defina uma senha de sua escolha. Lembre-se dessa senha para as etapas seguintes do tutorial.

j. Em Configuração de armazenamento de banco de dados, defina o Tipo de armazenamento do banco de dados como gp2.

k. Em Configuração de monitoramento de banco de dados, defina Habilitar o Performance Insights do RDS como falso.

l. Deixe todas as outras configurações com os valores padrão. Clique em Próximo para continuar.

5. Na página Revisar pilha, selecione Enviar depois de verificar as opções do banco de dados e do bastion host do Linux.

Depois que o processo de criação da pilha for concluído, visualize as pilhas com os nomes BastionStack e RDSNS para anotar as informações necessárias para se conectar ao banco de dados. Consulte mais informações em Viewing AWS CloudFormation stack data and resources on the AWS Management Console.

Conectar-se a uma instância de banco de dados MariaDB

Você pode usar qualquer aplicação cliente padrão SQL para se conectar à instância de banco de dados. Neste exemplo, você se conecta a uma instância de banco de dados MariaDB usando o cliente da linha de comando mysql.

Como conectar-se a uma instância de banco de dados MariaDB

1. Encontre o endpoint (nome de DNS) e o número da porta para sua instância de banco de dados.

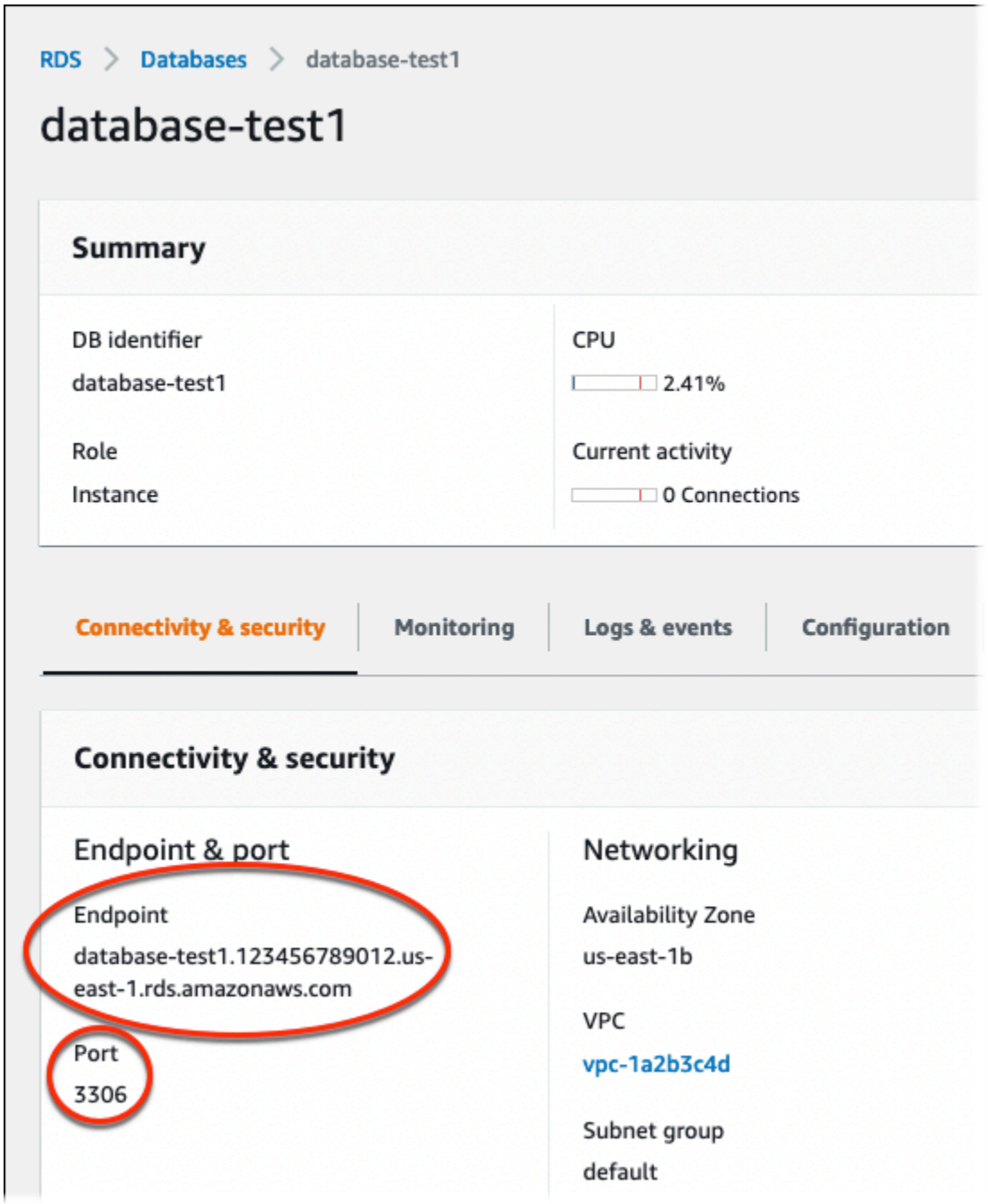
   a. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.

   b. No canto superior direito do console do Amazon RDS, escolha a Região da AWS da instância de banco de dados.

   c. No painel de navegação, escolha Bancos de dados.

d. Escolha o nome da instância de banco de dados MariaDB para exibir os detalhes.

e. Na guia Connectivity & security (Conectividade e segurança), copie o endpoint. Além disso, anote o número da porta. Você precisará do endpoint e do número da porta para conectar-se à instância de banco de dados.



2. Conecte-se à instância do EC2 que você criou anteriormente, seguindo as etapas em Conecte-se à sua instância do Linux no Guia do usuário do Amazon EC2.

Recomendamos que você se conecte à sua instância do EC2 utilizando SSH. Se o utilitário cliente SSH estiver instalado no Windows, Linux ou Mac, você poderá se conectar à instância utilizando o seguinte formato de comando:

```
ssh -i location_of_pem_file ec2-user@ec2-instance-public-dns-name
```

Por exemplo, suponha que `ec2-database-connect-key-pair.pem` esteja armazenado em `/dir1` no Linux e que o DNS IPv4 público para sua instância do EC2 seja `ec2-12-345-678-90.compute-1.amazonaws.com`. Seu comando SSH teria a seguinte aparência:

```
ssh -i /dir1/ec2-database-connect-key-pair.pem ec2-
user@ec2-12-345-678-90.compute-1.amazonaws.com
```

3. Obtenha as últimas correções de bugs e atualizações de segurança atualizando o software na instância do EC2. Para fazer isso, execute o seguinte comando.

> **Note**
>
> A opção `-y` instala as atualizações sem solicitar confirmação. Para examinar atualizações antes da instalação, omita essa opção.

```
sudo dnf update -y
```

4. Instale o cliente da linha de comando mysql do MariaDB.

   Para instalar o cliente da linha de comando do MariaDB no Amazon Linux 2023, execute o seguinte comando:

```
sudo dnf install mariadb105
```

5. Conecte-se à instância de banco de dados MariaDB. Por exemplo, insira o comando a seguir. Essa ação permite que você se conecte à instância de banco de dados MariaDB usando o cliente do MySQL.

Substitua o endpoint da instância de banco de dados (nome DNS) por *endpoint* e o nome do usuário principal usado por *admin*. Forneça a senha mestra usada quando for solicitada uma senha.

```
mysql -h endpoint -P 3306 -u admin -p
```

Depois de inserir a senha do usuário, você deverá ver uma saída semelhante à seguinte.

```
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 156
Server version: 10.6.10-MariaDB-log managed by https://aws.amazon.com/rds/

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>
```

Para ter mais informações sobre como se conectar à instância de banco de dados MariaDB, consulte Conectar-se à instância de banco de dados do MariaDB. Se você não conseguir se conectar à sua instância de banco de dados, consulte Não é possível conectar-se à instância de banco de dados do Amazon RDS.

Por questões de segurança, é uma prática recomendada usar conexões criptografadas. Use uma conexão MariaDB não criptografada apenas quando o cliente e o servidor estiverem na mesma VPC e a rede for confiável. Para obter informações sobre como usar conexões criptografadas, consulte Conectar-se à instância de banco de dados do MariaDB no Amazon RDS com SSL/TLS por meio do cliente de linha de comandos do MySQL (criptografado).

6. Executar comandos SQL.

   Por exemplo, o seguinte comando SQL mostra a data e a hora atuais:

   ```
   SELECT CURRENT_TIMESTAMP;
   ```

Excluir a instância do EC2 e a instância de banco de dados

Depois de se conectar e explorar a instância do EC2 e a instância de banco de dados criadas, exclua-as para não receber mais cobranças por elas.

Se você usou o AWS CloudFormation para criar recursos, ignore essa etapa e passe para a próxima.

Como excluir a instância do EC2

1. Faça login no AWS Management Console e abra o console do Amazon EC2 em https://console.aws.amazon.com/ec2/.
2. No painel de navegação, escolha Instâncias.
3. Selecione a instância do EC2 e escolha Estado da instância, Encerrar instância.
4. Quando a confirmação for solicitada, escolha Encerrar.

Consulte mais informações sobre como excluir uma instância do EC2 em Como encerrar uma instância no Guia do usuário do Amazon EC2.

Como excluir uma instância de banco de dados sem snapshot de banco de dados final

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Databases (Bancos de dados).
3. Escolha a instância de banco de dados que você deseja excluir.
4. Em Actions, selecione Delete.
5. Desmarque Criar snapshot final? e Reter backups automatizados.
6. Conclua a confirmação e escolha Excluir.

(Opcional) Excluir a instância do EC2 e a instância de banco de dados criada com o CloudFormation

Se você usou o AWS CloudFormation para criar recursos, exclua a pilha do CloudFormation depois de conectar e explorar a amostra de instância do EC2 e de instância de banco de dados para não receber mais cobranças por elas.

Para excluir os recursos do CloudFormation

1. Abra o console do AWS CloudFormation.
2. Na página Pilhas no console do CloudFormation, selecione a pilha raiz (a pilha sem o nome VPCStack, BastionStack ou RDSNS).
3. Escolha Excluir.
4. Selecione Excluir pilha quando a confirmação for solicitada.

Consulte mais informações de como excluir uma pilha no CloudFormation em Deleting a stack on the AWS CloudFormation console no Guia do usuário do AWS CloudFormation.

(Opcional) Conectar a instância de banco de dados a uma função do Lambda

Você também pode conectar a instância de banco de dados do RDS para MariaDB a um recurso de computação sem servidor do Lambda. As funções do Lambda permitem que você execute código sem provisionar nem gerenciar a infraestrutura. Uma função do Lambda também permite que você responda automaticamente a solicitações de execução de código em qualquer escala, de dezenas de eventos por dia a centenas por segundo. Para obter mais informações, consulte Conectar automaticamente uma função do Lambda e uma instância de banco de dados.

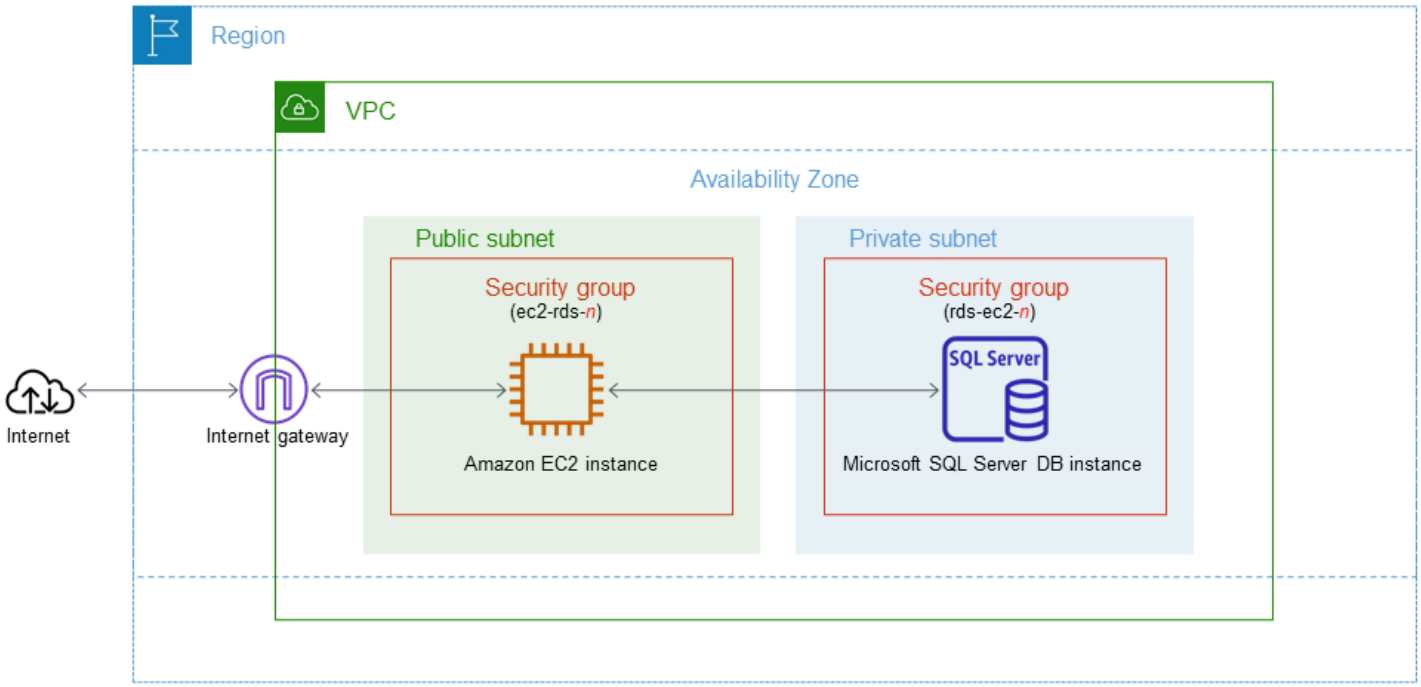
Criar uma instância de banco de dados do Microsoft SQL Server e conectar-se a ela

Este tutorial cria uma instância do EC2 e uma instância de banco de dados do RDS para Microsoft. O tutorial mostra como acessar a instância de banco de dados pela instância do EC2 usando um cliente do Microsoft SQL Server Management Studio. Como prática recomendada, este tutorial cria uma instância de banco de dados privada em uma nuvem privada virtual (VPC). Na maioria dos casos, outros recursos na mesma VPC, como instâncias do EC2, podem acessar a instância de banco de dados, mas recursos fora da VPC não podem acessá-la.

Depois de concluir o tutorial, haverá uma sub-rede pública e privada em cada zona de disponibilidade na sua VPC. Em uma zona de disponibilidade, a instância do EC2 está na sub-rede pública e a instância de banco de dados está na sub-rede privada.

> **Important**
>
> Não há cobrança para criar uma conta da AWS. No entanto, ao concluir este tutorial, é possível gerar custos para os recursos da AWS que você usa. Se esses recursos não forem mais necessários após a conclusão do tutorial, você poderá excluí-los.

O diagrama a seguir mostrará a configuração quando o tutorial estiver completo.



Esse tutorial permite criar recursos usando um dos seguintes métodos:

1. Use o AWS Management Console: Criar uma instância de banco de dados do SQL Server e Criar uma instância do EC2
2. Use o AWS CloudFormation para criar a instância de banco de dados e a instância do EC2: (Opcional) Criar VPC, instância do EC2 e instância do SQL Server usando o AWS CloudFormation

O primeiro método usa Criação fácil para criar uma instância de banco de dados privada do SQL Server com o AWS Management Console. Aqui, você especifica somente o tipo de mecanismo de banco de dados, o tamanho da instância de banco de dados e o identificador da instância de banco de dados. A opção Easy create (Criação fácil) usa a configuração padrão para as outras opções de configuração.

Ao usar a opção Criação padrão, é possível especificar mais opções de configuração ao criar uma instância de banco de dados. Essas opções incluem configurações de disponibilidade, segurança, backups e manutenção. Para criar uma instância de banco de dados pública, você deve usar a Criação padrão. Para mais informações, consulte Criar uma instância de banco de dados do Amazon RDS.

Tópicos

- Pré-requisitos
- Criar uma instância do EC2
- Criar uma instância de banco de dados do SQL Server
- (Opcional) Criar VPC, instância do EC2 e instância do SQL Server usando o AWS CloudFormation
- Conectar-se à instância de banco de dados do SQL Server
- Explorar a instância de banco de dados do SQL Server de amostra
- Excluir a instância do EC2 e a instância de banco de dados
- (Opcional) Excluir a instância do EC2 e a instância de banco de dados criada com o CloudFormation
- (Opcional) Conectar a instância de banco de dados a uma função do Lambda

Pré-requisitos

Antes de começar, conclua as etapas nas seguintes seções:

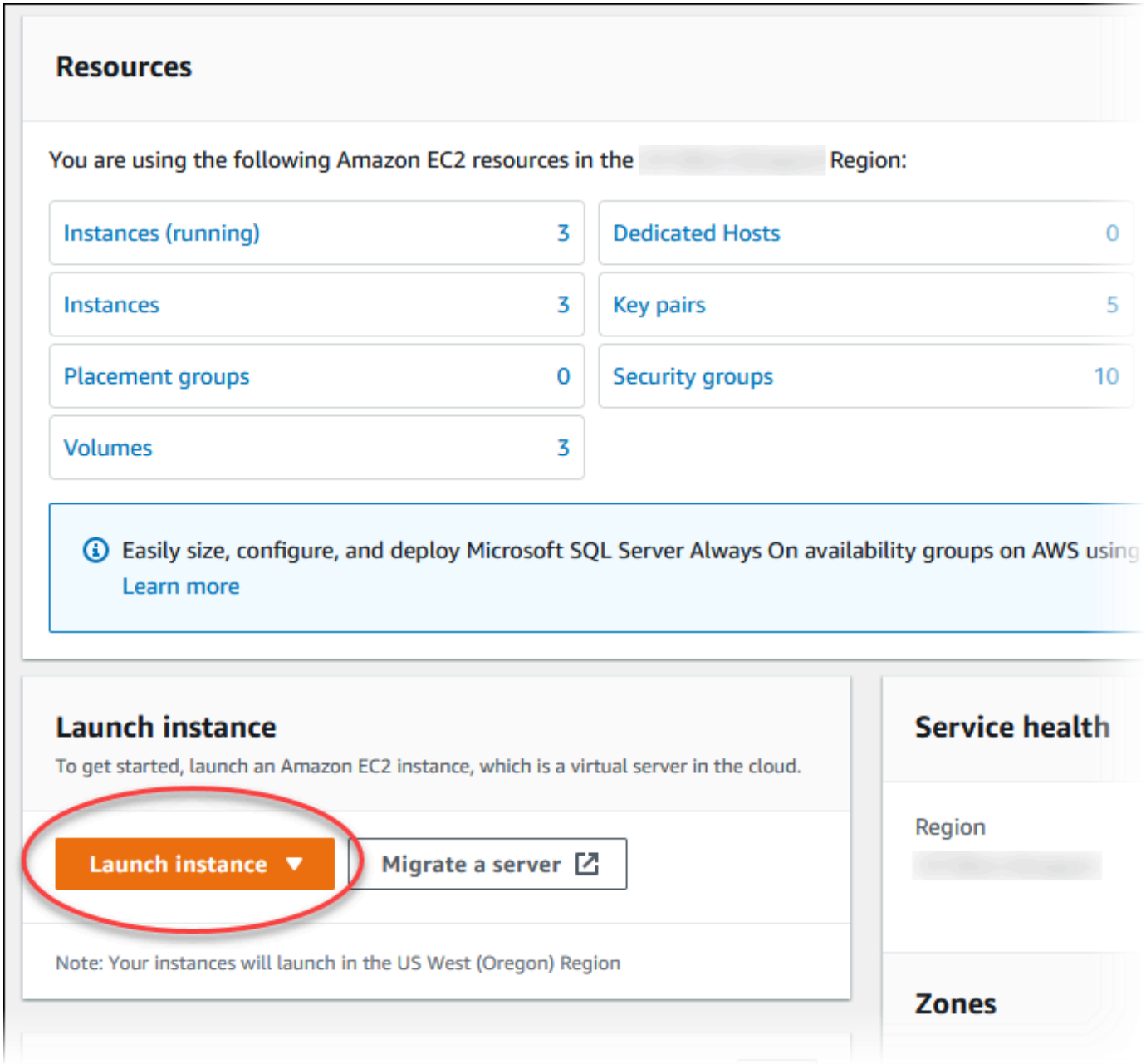
- Inscrever-se para uma Conta da AWS
- Criar um usuário com acesso administrativo

Criar uma instância do EC2

Crie uma instância do Amazon EC2 que você usará para se conectar ao banco de dados.

Para criar uma instância do EC2

1. Faça login no AWS Management Console e abra o console do Amazon EC2 em https://console.aws.amazon.com/ec2/.
2. No canto superior direito do AWS Management Console, escolha a Região da AWS que você usou para o banco de dados anteriormente.
3. Escolha Painel do EC2 e Executar instância, conforme mostrado na imagem a seguir.

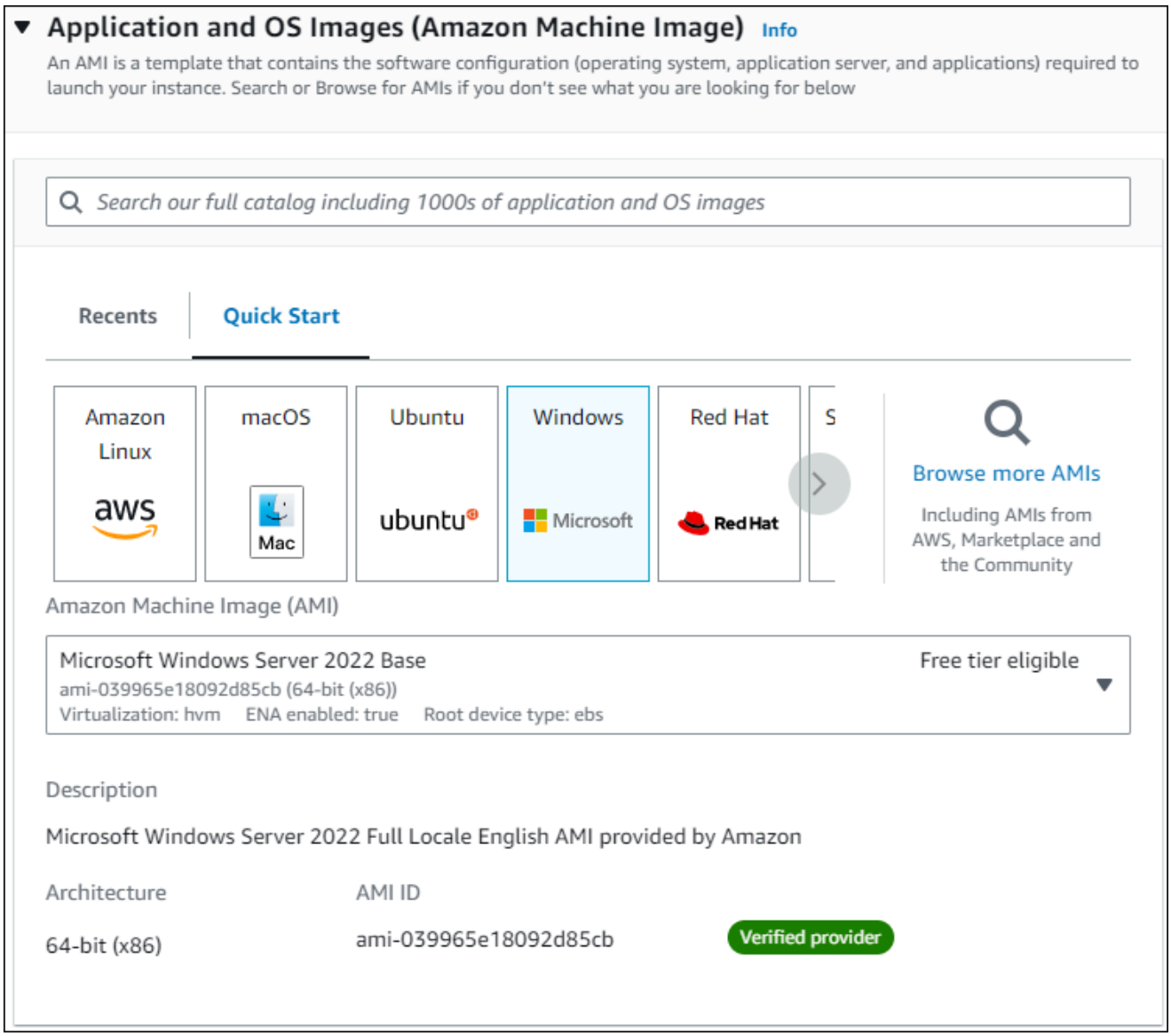


A página Iniciar uma instância é aberta.

4. Escolha as configurações a seguir na página Iniciar uma instância.

   a. Em Name and tags (Nome e etiquetas), em Name (Nome), insira **`ec2-database-connect`**.

   b. Em Imagens de aplicações e sistemas operacionais (imagem de máquina da Amazon), selecione Windows e, depois, selecione a Base do Microsoft Windows Server 2022. Mantenha as seleções padrão nas outras opções.



c. Em Instance type (Tipo de instância), escolha t2.micro.

d. Em Key pair (login) (Par de chaves (login)), escolha um Key pair name (Nome do par de chaves) para usar um par de chaves existente. Para criar um par de chaves para a instância do Amazon EC2, escolha Create new key pair (Criar um novo key pair), depois use a janela Create key pair (Criar par de chaves) para criá-lo.

   Para ter mais informações sobre como criar um par de chaves, consulte Criar um par de chaves no Guia do usuário do Amazon EC2 para instâncias do Windows.

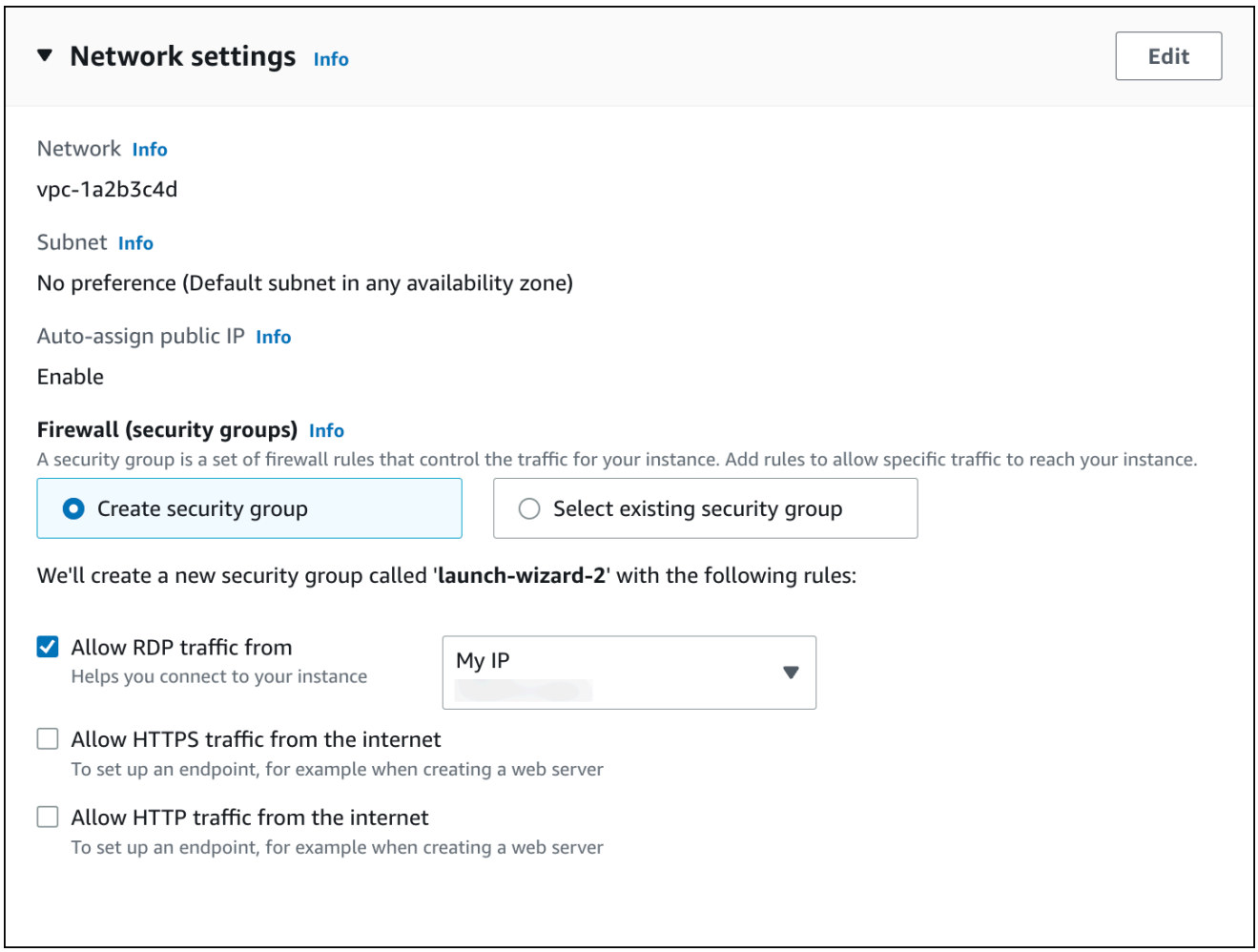
e. Para Firewall (grupos de segurança), em Configurações de rede, selecione Permitir tráfego RDP de para se conectar à instância do EC2.

Você poderá selecionar Meu IP se o endereço IP exibido estiver correto para conexões RDP. Caso contrário, determine o endereço IP a ser usado para se conectar a instâncias do EC2 em sua VPC usando RDP. Para determinar seu endereço IP público, em uma janela ou guia diferente do navegador, é possível usar o serviço em https://checkip.amazonaws.com. Um exemplo de endereço IP é 192.0.2.1/32.

Em muitos casos, você pode se conectar por meio de um provedor de serviços de Internet (ISP) ou atrás de um firewall sem um endereço IP estático. Em caso afirmativo, determine o intervalo de endereços IP utilizado por computadores cliente.

> **Warning**
>
> Se você usar `0.0.0.0/0` para acesso do RDP, você possibilitará que todos os endereços IP acessem suas instâncias públicas do EC2 usando RDP. Essa abordagem é aceitável por um período curto em um ambiente de teste, mas não é seguro em ambientes de produção. Em produção, autorize somente um endereço IP específico ou um intervalo de endereços para acessar suas instâncias do EC2 usando RDP.

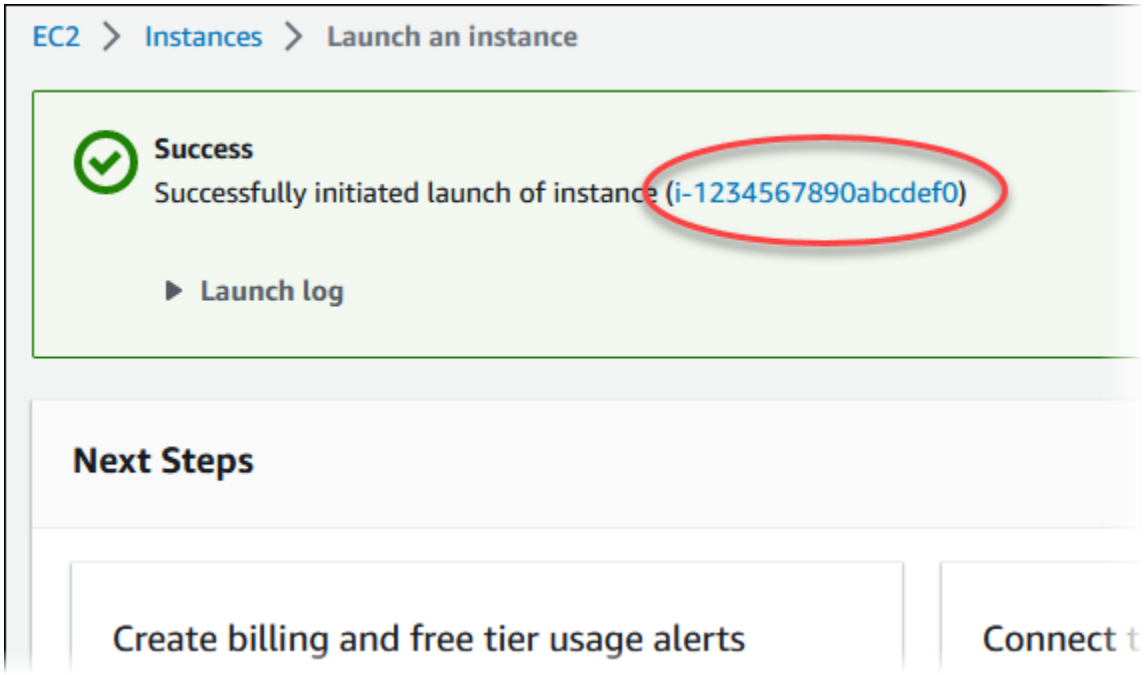
A imagem a seguir mostra um exemplo da seção Configurações de rede.



f. Mantenha os valores padrão para as seções restantes.

g. Revise um resumo da configuração da instância do EC2 no painel Resumo e, quando estiver com tudo pronto, escolha Executar instância.

5. Na página Status de inicialização, anote o identificador de sua nova instância do EC2, por exemplo: `i-1234567890abcdef0`.



6. Escolha o identificador de instância do EC2 para abrir a lista de instâncias do EC2.
7. Espere até o Estado da instância para a instância do EC2 ficar como Executando antes de continuar.

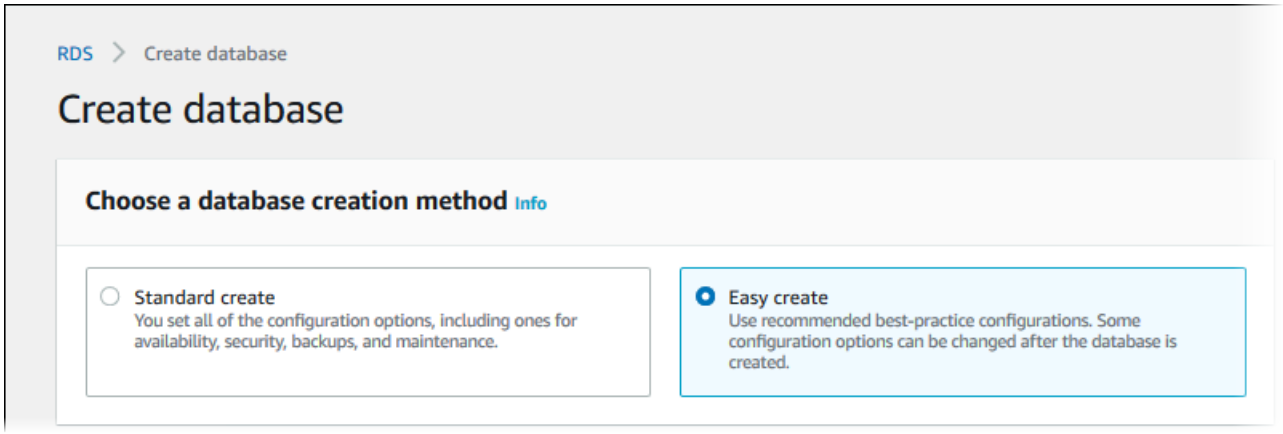
Criar uma instância de banco de dados do SQL Server

O bloco de construção básico do Amazon RDS é a instância do banco de dados. Este ambiente é onde você executa seus bancos de dados do SQL Server.

Neste exemplo, use a opção Criação fácil para criar uma instância de banco de dados que execute o mecanismo de banco de dados do SQL Server com uma classe de instância de banco de dados db.t2.micro.

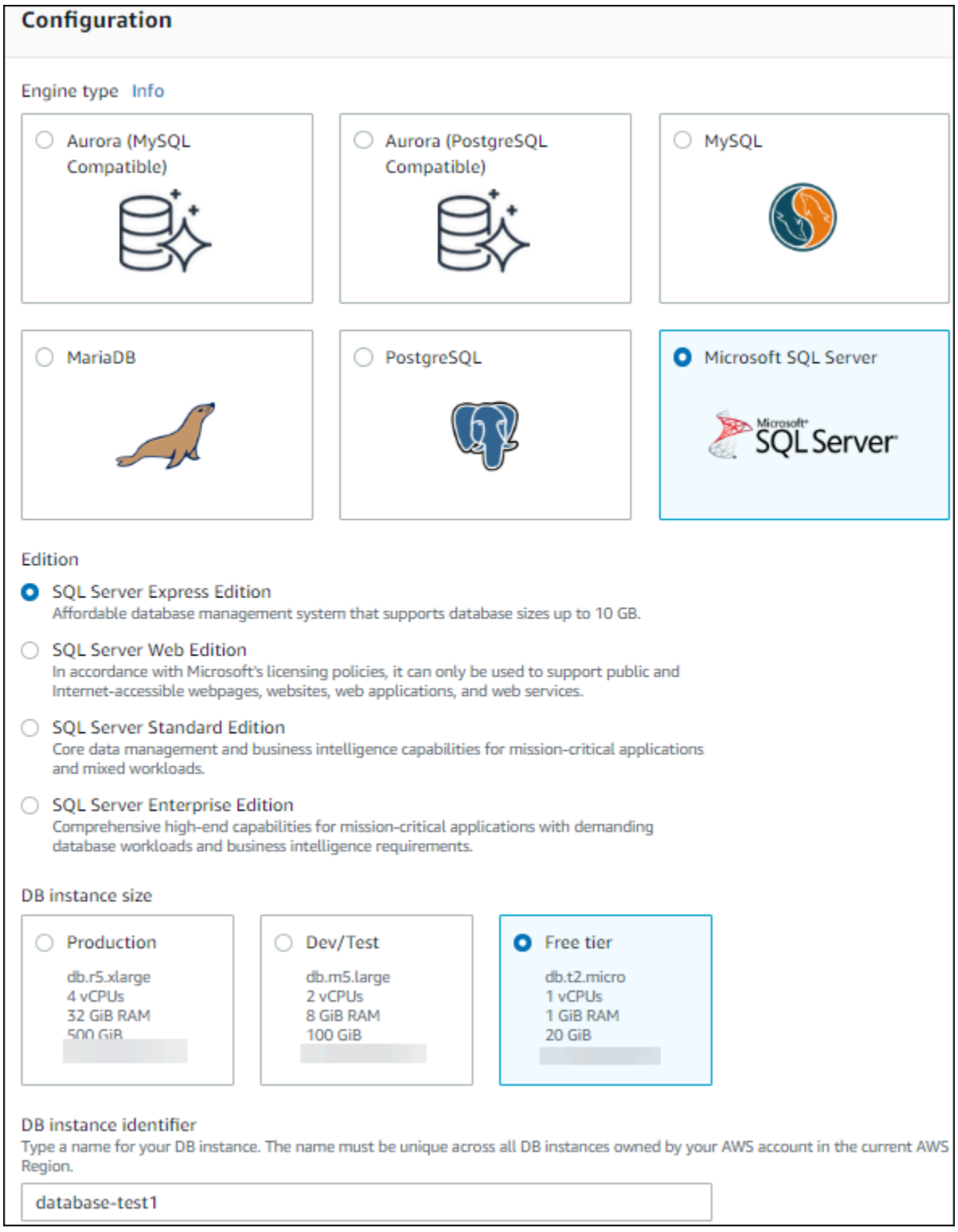
Como criar uma instância de banco de dados do Microsoft SQL Server com a Criação fácil

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No canto superior direito do console do Amazon RDS, escolha a Região da AWS na qual você deseja criar a instância de banco de dados.
3. No painel de navegação, escolha Databases (Bancos de dados).
4. Selecione a opção Create database (Criar banco de dados) e verifique se a opção Easy Create (Criação fácil) está habilitada.



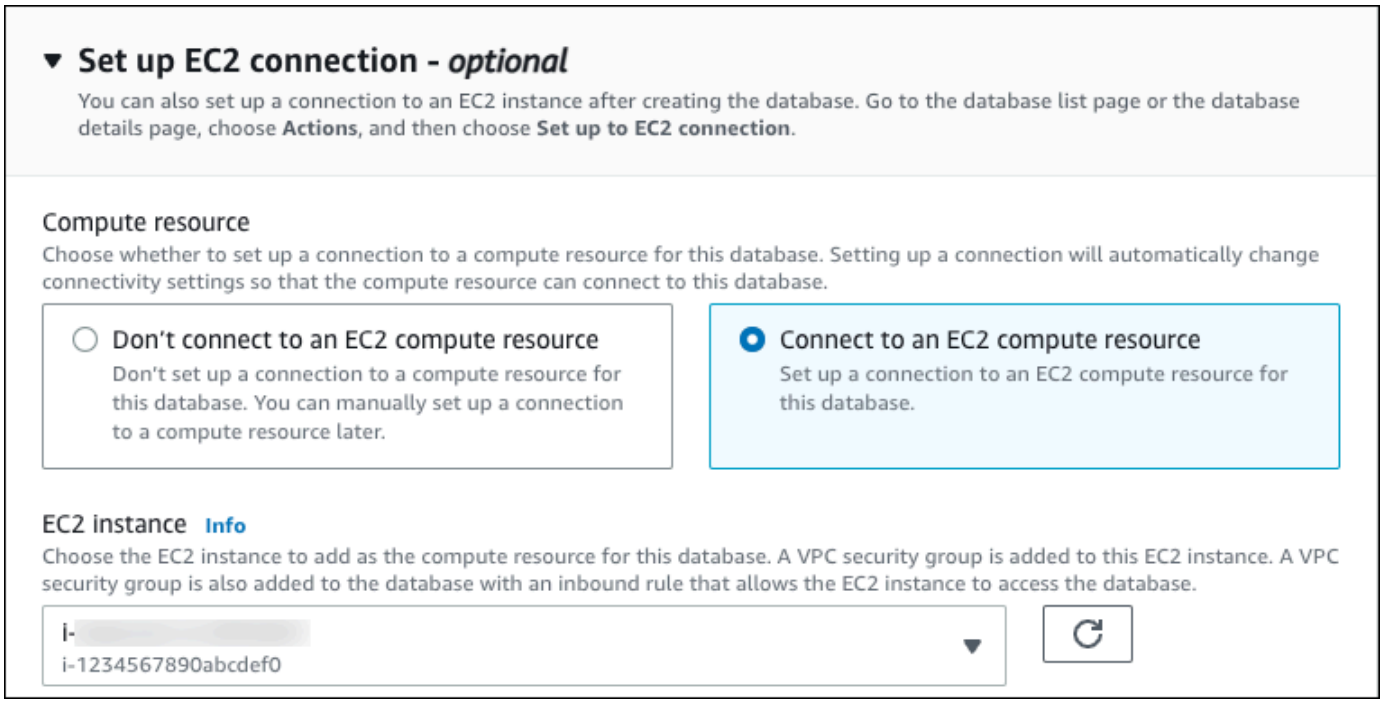
5. Em Configuration (Configuração), escolha Microsoft SQL Server.
6. Em Edição, selecione SQL Server Express Edition.
7. Em DB instance size (Tamanho da instância de banco de dados), escolha Free tier (Nível gratuito). Nível gratuito aparece para contas do plano gratuito. Sandbox aparece para contas do plano pago.
8. Em Identificador de instância de banco de dados, insira **`database-test1`**.

   A página Create database (Criar banco de dados) deve ser semelhante à imagem a seguir. Para contas do plano gratuito, aparece Nível gratuito. Para contas do plano pago, aparece Sandbox.



9. Em Nome do usuário principal, insira um nome para o usuário mestre ou deixe o nome padrão.
10. Para configurar uma conexão com a instância do EC2 que você criou anteriormente, abra Configurar conexão do EC2: opcional.

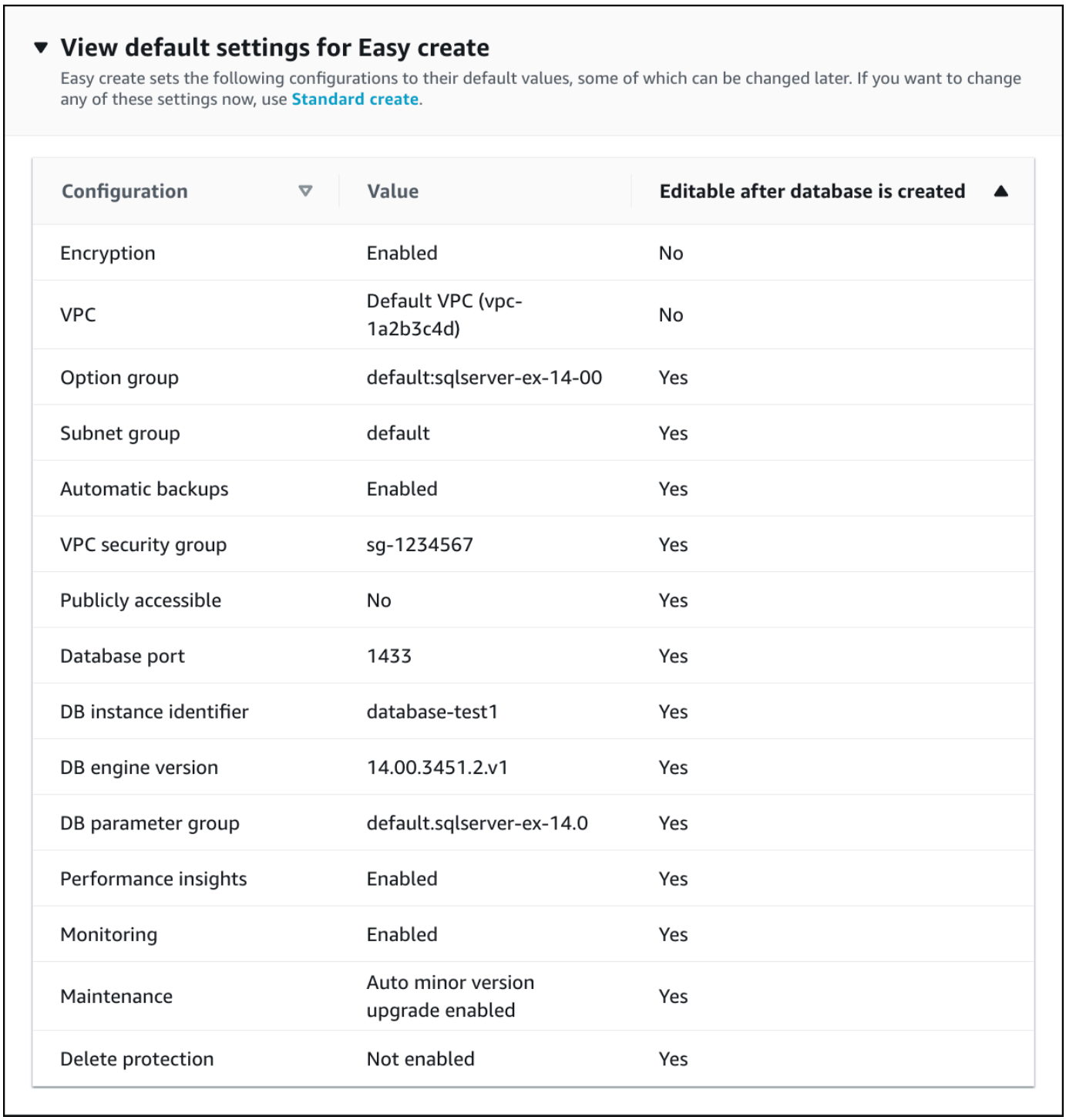
Selecione Conectar-se a um recurso computacional do EC2. Selecione a instância do EC2 que você criou anteriormente.



11. Para usar uma senha primária gerada automaticamente para a instância de banco de dados, selecione a caixa de seleção Auto generate a password (Gerar uma senha automaticamente).

    Para inserir sua senha primária, desmarque a caixa de seleção Auto generate a password (Gerar uma senha automaticamente) e insira a mesma senha em Master password (Senha primária) e em Confirm password (Confirmar senha).

12. Abra Visualizar as configurações padrão da criação fácil.



View default settings for Easy create

Easy create sets the following configurations to their default values, some of which can be changed later. If you want to change any of these settings now, use **Standard create**.

Configuration	Value	Editable after database is created
Encryption	Enabled	No
VPC	Default VPC (vpc-1a2b3c4d)	No
Option group	default:sqlserver-ex-14-00	Yes
Subnet group	default	Yes
Automatic backups	Enabled	Yes
VPC security group	sg-1234567	Yes
Publicly accessible	No	Yes
Database port	1433	Yes
DB instance identifier	database-test1	Yes
DB engine version	14.00.3451.2.v1	Yes
DB parameter group	default.sqlserver-ex-14.0	Yes
Performance insights	Enabled	Yes
Monitoring	Enabled	Yes
Maintenance	Auto minor version upgrade enabled	Yes
Delete protection	Not enabled	Yes

Você pode examinar as configurações padrão usadas com a opção Easy Create (Criação fácil). A coluna Editável após a criação do banco de dados mostra quais opções podem ser alteradas após a criação do banco de dados.

- Se uma configuração tiver Não nessa coluna e você quiser uma configuração diferente, poderá usar a opção Criação padrão para criar a instância de banco de dados.

- Se uma configuração tiver Sim nessa coluna e você quiser uma configuração diferente, poderá usar a opção Criação padrão para criar a instância de banco de dados, ou modificar a instância de banco de dados depois de criá-la para alterar a configuração.

13. Selecione Criar banco de dados.

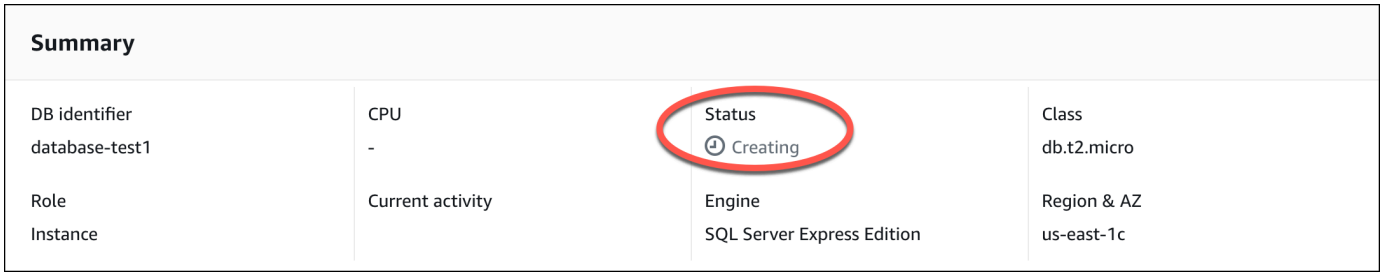
    Para exibir o nome de usuário mestre e a senha da instância de banco de dados, escolha View credential details (Ver detalhes da credencial).

    Use o nome de usuário e a senha que aparecem para se conectar à instância de banco de dados como o usuário mestre.

    > **Important**
    >
    > Você não pode visualizar a senha do usuário principal novamente. Caso você não a registre, talvez seja necessário alterá-la.
    > Se for necessário alterar a senha do usuário mestre depois que a instância de banco de dados estiver disponível, será possível modificar a instância de banco de dados para fazer isso. Para ter mais informações sobre a modificação de uma instância de banco de dados, consulte Modificar uma instância de banco de dados do Amazon RDS.

14. Na lista Bancos de dados, selecione o nome da nova instância de banco de dados do SQL Server para mostrar seus detalhes.

    A instância de banco de dados permanecerá no status Criando até estar pronta para ser usada.



    Quando o status muda para Available (Disponível), você pode se conectar à instância de banco de dados. Dependendo da classe da instância de banco de dados e da quantidade de armazenamento, pode levar até 20 minutos para que a nova instância esteja disponível.

(Opcional) Criar VPC, instância do EC2 e instância do SQL Server usando o AWS CloudFormation

Em vez de usar o console para criar a VPC, a instância do EC2 e a instância do SQL Server, você pode usar o AWS CloudFormation para provisionar recursos da AWS tratando a infraestrutura como código. Para ajudar a organizar os recursos da AWS em unidades menores e mais gerenciáveis, você pode usar a funcionalidade de pilha aninhada do AWS CloudFormation. Consulte mais informações em Criar uma pilha no console do AWS CloudFormation e Trabalhar com pilhas aninhadas.

> **Important**
>
> O AWS CloudFormation é gratuito, mas os recursos que o CloudFormation são ativos. Você incorre nas taxas de uso padrão para esses recursos até que os encerre. Para ter mais informações, consulte Definição de preço do Amazon RDS para SQL Server.

Para criar recursos usando o console do AWS CloudFormation, conclua as seguintes etapas:

- Baixar o modelo do CloudFormation
- Configurar recursos usando o CloudFormation

Baixar o modelo do CloudFormation

Um modelo do CloudFormation é um arquivo de texto JSON ou YAML que contém as informações da configuração dos recursos que você deseja criar na pilha. Esse modelo também cria uma VPC e um bastion host para você junto com a instância do RDS.

Para baixar o arquivo de modelo, abra o link SQL Server CloudFormation template.

Na página do Github, clique no botão Baixar arquivo bruto para salvar o arquivo YAML do modelo.

Configurar recursos usando o CloudFormation

> **Note**
>
> Antes de iniciar esse processo, verifique se você tem um par de chaves para uma instância do EC2 na Conta da AWS. Para obter mais informações, consulte Pares de chaves do Amazon EC2 e instâncias do Linux.

Ao usar o modelo do AWS CloudFormation, você deve selecionar os parâmetros certos para garantir que os recursos sejam criados corretamente. Siga as etapas abaixo:

1. Faça login no AWS Management Console e abra o console AWS CloudFormation em https://console.aws.amazon.com/cloudformation.
2. Selecione Create Stack (Criar pilha).
3. Na seção Especificar modelo, selecione Fazer upload de um arquivo de modelo do computador e escolha Próximo.
4. Na página Especificar detalhes da pilha, defina os seguintes parâmetros:
   a. Defina o Nome da pilha como SQLServerTestStack.
   b. Em Parâmetros, defina Zonas de disponibilidade selecionando três zonas de disponibilidade.
   c. Em Configuração do bastion host do Linux, em Nome da chave, selecione um par de chaves para fazer login na instância do EC2.
   d. Em Configurações do bastion host do Linux, defina o Intervalo de IP permitido para o endereço IP. Para conectar instâncias do EC2 à VPC usando o Secure Shell (SSH), determine o endereço IP público usando o serviço em https://checkip.amazonaws.com. Um exemplo de endereço IP é 192.0.2.1/32.

> **Warning**
>
> Se usar `0.0.0.0/0` para acesso do SSH, você possibilitará que todos os endereços IP acessem suas instâncias públicas do EC2 usando SSH. Essa abordagem é aceitável por um período curto em um ambiente de teste, mas não é seguro em ambientes de produção. Em produção, autorize somente um endereço IP específico ou um intervalo de endereços para acessar suas instâncias do EC2 usando SSH.

e. Em Configuração geral do banco de dados, defina a Classe da instância do banco de dados como db.t3.micro.

f. Defina o Nome do banco de dados como **`database-test1`**.

g. Em Nome de usuário principal do banco de dados, insira um nome para o usuário principal.

h. Defina a Gerenciar senha de usuário principal do banco de dados com o Secrets Manager como `false` para esse tutorial.

i. Em Senha do banco de dados, defina uma senha de sua escolha. Lembre-se dessa senha para as etapas seguintes do tutorial.

j. Em Configuração de armazenamento de banco de dados, defina o Tipo de armazenamento do banco de dados como gp2.

k. Em Configuração de monitoramento de banco de dados, defina Habilitar o Performance Insights do RDS como falso.

l. Deixe todas as outras configurações com os valores padrão. Clique em Próximo para continuar.

5. Na página Configurar opções de pilha, mantenha todas as opções padrão. Clique em Próximo para continuar.
6. Na página Revisar pilha, selecione Enviar depois de verificar as opções do banco de dados e do bastion host do Linux.

Depois que o processo de criação da pilha for concluído, visualize as pilhas com os nomes BastionStack e RDSNS para anotar as informações necessárias para se conectar ao banco de dados. Consulte mais informações em Viewing AWS CloudFormation stack data and resources on the AWS Management Console.

Conectar-se à instância de banco de dados do SQL Server

No procedimento a seguir, você se conectará à instância de banco de dados usando o Microsoft SQL Server Management Studio (SSMS).

Como se conectar a uma instância de banco de dados do RDS para SQL Server usando SSMS

1. Encontre o endpoint (nome de DNS) e o número da porta para sua instância de banco de dados.

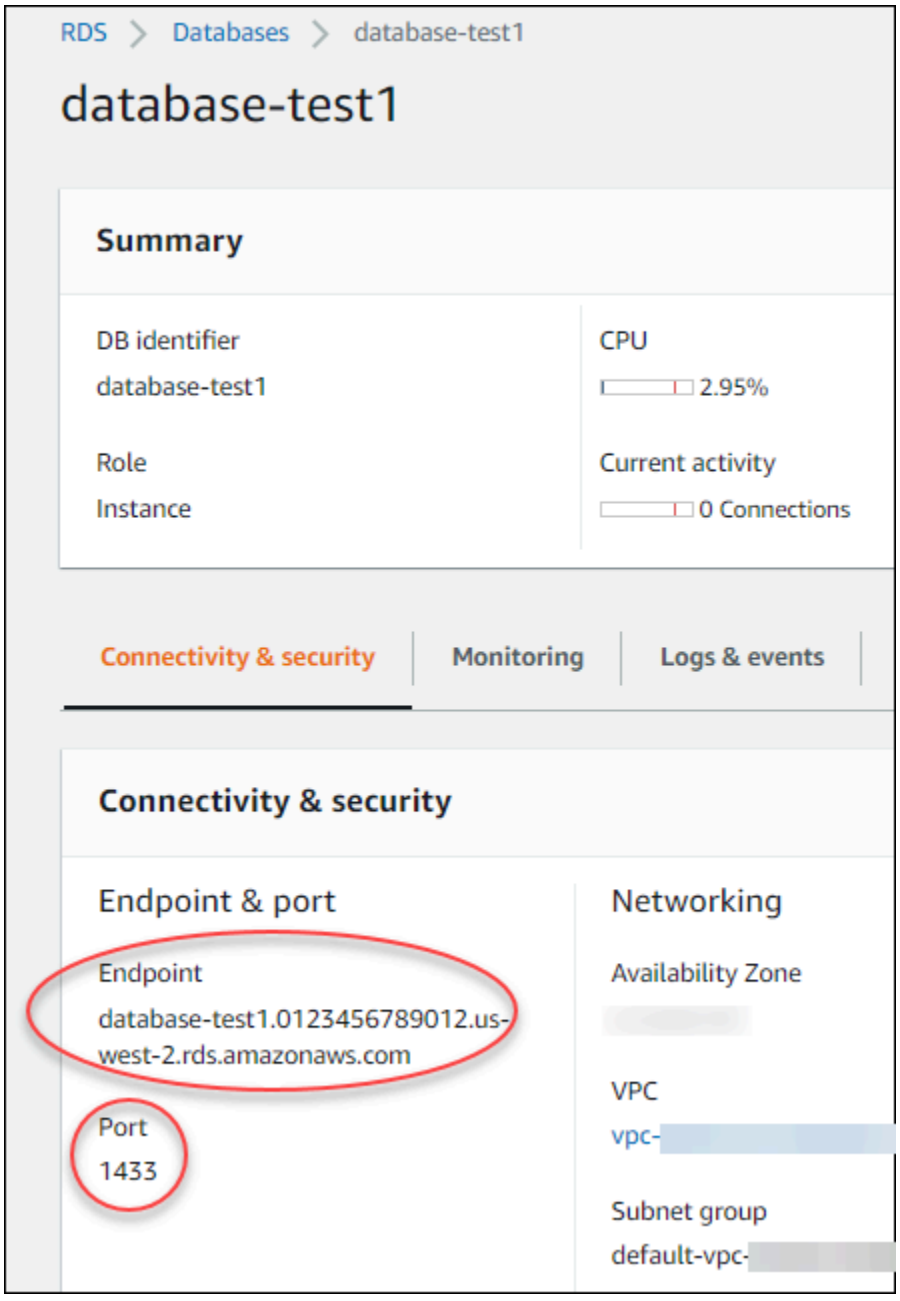
   a. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.

   b. No canto superior direito do console do Amazon RDS, escolha a Região da AWS da instância de banco de dados.

c. No painel de navegação, escolha Bancos de dados.

d. Escolha o nome da instância de banco de dados do SQL Server para exibir os detalhes.

e. Na guia Connectivity (Conectividade), copie o endpoint. Além disso, anote o número da porta. Você precisará do endpoint e do número da porta para conectar-se à instância de banco de dados.



2. Conecte-se à instância do EC2 que você criou anteriormente, seguindo as etapas em Conectar-se à sua instância do Microsoft Windows no Guia do usuário do Amazon EC2 para instâncias do Windows.

3. Instale o cliente do SQL Server Management Studio (SSMS) da Microsoft.

   Para baixar uma versão autônoma do SSMS para sua instância do EC2, consulte Baixar o SQL Server Management Studio (SSMS) na documentação da Microsoft.

   a. Use o menu Iniciar para abrir o Internet Explorer.

   b. Use o Internet Explorer para baixar e instalar uma versão autônoma do SSMS. Se você for avisado de que o site não é confiável, adicione-o à lista de sites confiáveis.

4. Inicie o SQL Server Management Studio (SSMS).

   A caixa de diálogo Connect to Server (Conectar-se ao servidor) é exibida.

5. Forneça as seguintes informações da sua instância de banco de dados de exemplo:

   a. Para Server type (Tipo de servidor), escolha Database Engine (Mecanismo do banco de dados).

   b. Em Server name (Nome do servidor), insira o nome DNS, seguido por uma vírgula e o número da porta (a porta padrão é 1433). Por exemplo, o nome do servidor deve ser semelhante ao seguinte:

   ```
   database-test1.0123456789012.us-west-2.rds.amazonaws.com,1433
   ```

   c. Para Authentication (Autenticação), escolha SQL Server Authentication (Autenticação do SQL Server).

   d. Em Login, digite o nome de usuário selecionado para sua instância de banco de dados de exemplo. Isso também é conhecido como o nome do usuário principal.

   e. Em Password (Senha), digite a senha escolhida anteriormente para a instância de banco de dados de exemplo. Isso também é conhecido como a senha do usuário mestre.

6. Selecione Conectar.

   Depois de alguns momentos, o SSMS conecta-se à sua instância de banco de dados. Por questões de segurança, é uma prática recomendada usar conexões criptografadas. Use uma conexão não criptografada do SQL Server apenas quando o cliente e o servidor estiverem na mesma VPC e a rede for confiável. Para ter informações sobre como usar conexões criptografadas, consulte ???.

Para ter mais informações sobre como se conectar a uma instância de banco de dados Microsoft SQL Server, consulte Conectar-se à instância de banco de dados do Microsoft SQL Server.

Para ter informações sobre problemas de conexão, consulte Não é possível conectar-se à instância de banco de dados do Amazon RDS.

Explorar a instância de banco de dados do SQL Server de amostra

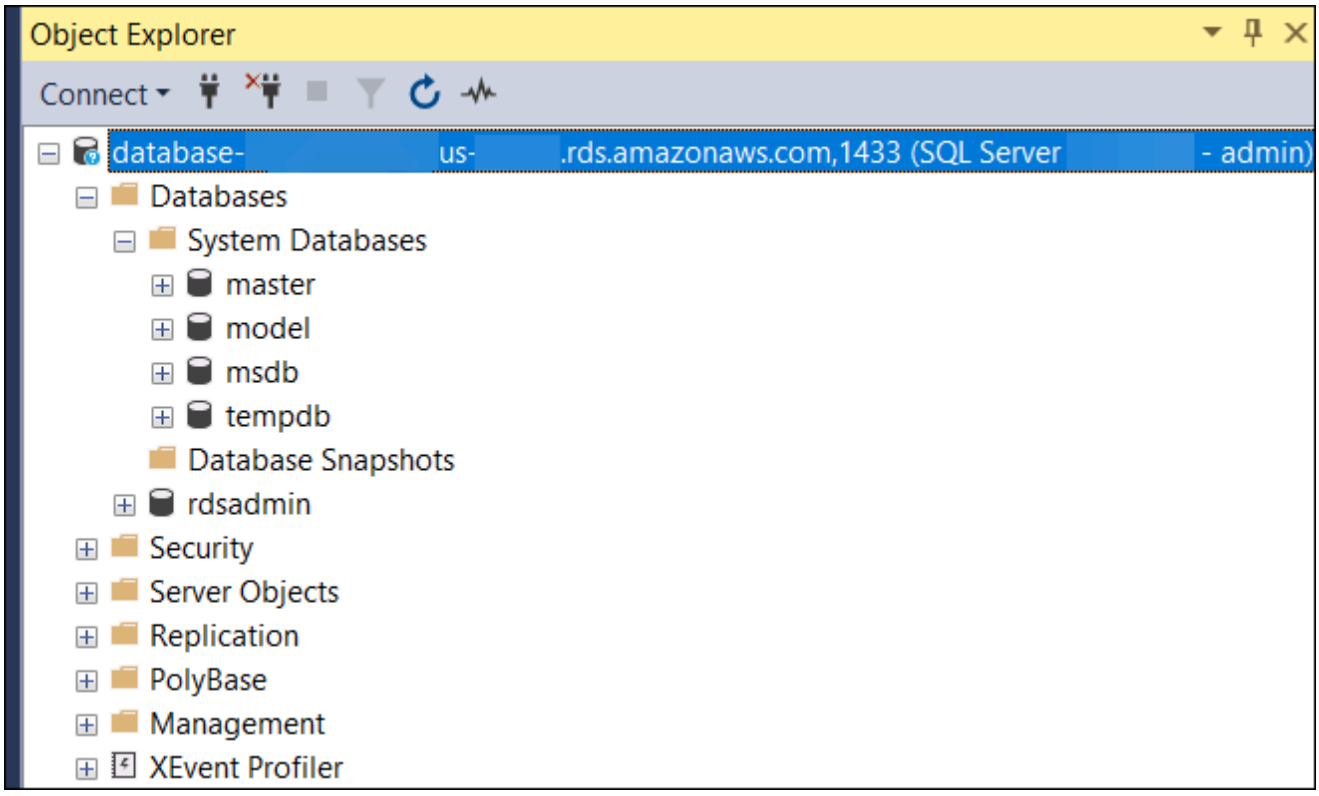
Você pode explorar sua instância de banco de dados de exemplo usando o Microsoft SQL Server Management Studio (SSMS).

Como explorar uma instância de banco de dados usando o SSMS

1. Sua instância de banco de dados SQL Server acompanha bancos de dados de sistema internos padrão do SQL Server (master, model, msdb e tempdb). Para explorar os bancos de dados do sistema, faça o seguinte:

   a. Em SSMS, no menu View (Exibir), escolha Object Explorer (Navegador de objetos).

   b. Expanda sua instância de banco de dados, expanda Databases (Bancos de dados) e depois expanda System Databases (Bancos de dados do sistema), conforme mostrado.



   Sua instância de banco de dados do SQL Server também conta com um banco de dados chamado `rdsadmin`. O Amazon RDS usa esse banco de dados para armazenar os objetos

usados para gerenciar o banco de dados. O banco de dados `rdsadmin` também inclui procedimentos armazenados que você pode executar para realizar tarefas avançadas.

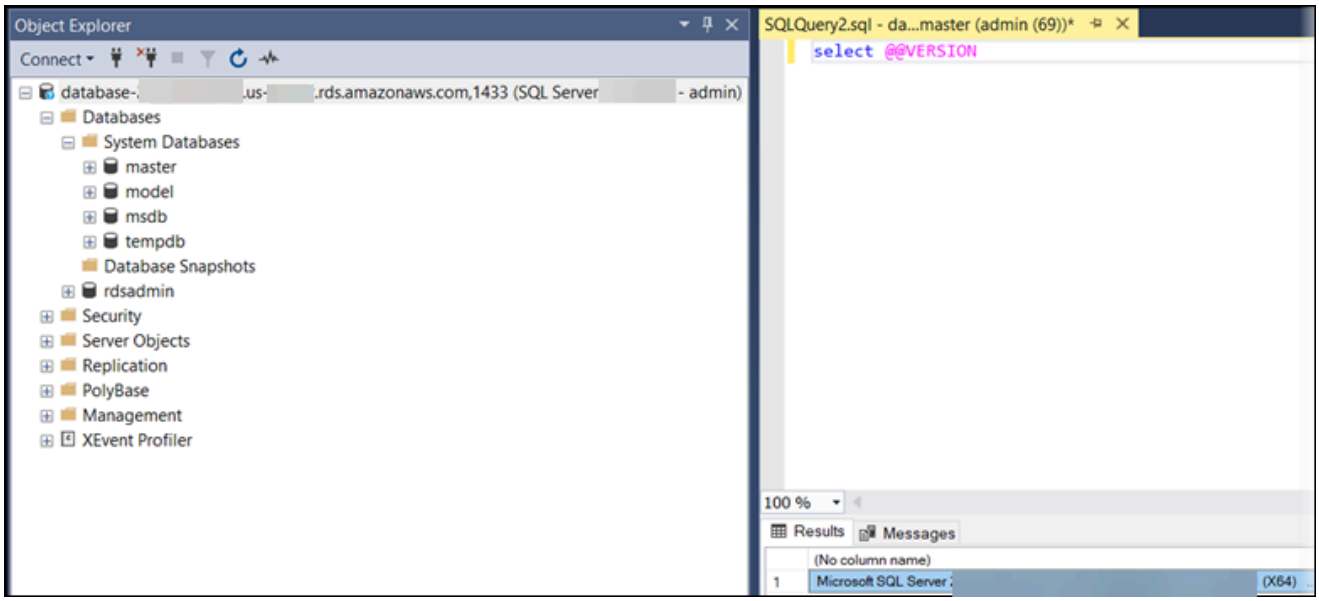
2. Comece a criar seus próprios bancos de dados e a executar consultas em instâncias de banco de dados e bancos de dados como de costume. Para executar uma consulta de teste na sua instância de banco de dados de exemplo, faça o seguinte:

   a. No SSMS, no menu Arquivo, aponte para Novo e selecione Consultar com a conexão atual.

   b. Digite a seguinte consulta SQL:

   ```
   select @@VERSION
   ```

   c. Execute a consulta. O SSMS retorna a versão do SQL Server da sua instância de banco de dados do Amazon RDS.



Excluir a instância do EC2 e a instância de banco de dados

Depois de se conectar e explorar a instância do EC2 e a instância de banco de dados criadas, exclua-as para não receber mais cobranças por elas.

Se você usou o AWS CloudFormation para criar recursos, ignore essa etapa e passe para a próxima.

Como excluir a instância do EC2

1. Faça login no AWS Management Console e abra o console do Amazon EC2 em https://console.aws.amazon.com/ec2/.

2. No painel de navegação, escolha Instâncias.
3. Selecione a instância do EC2 e escolha Estado da instância, Encerrar instância.
4. Quando a confirmação for solicitada, escolha Encerrar.

Para ter mais informações sobre como excluir uma instância do EC2, consulte Encerrar sua instância no Guia do usuário de instâncias do Windows.

Como excluir uma instância de banco de dados sem snapshot de banco de dados final

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Databases (Bancos de dados).
3. Escolha a instância de banco de dados que você deseja excluir.
4. Em Actions, selecione Delete.
5. Desmarque Criar snapshot final? e Reter backups automatizados.
6. Conclua a confirmação e escolha Excluir.

(Opcional) Excluir a instância do EC2 e a instância de banco de dados criada com o CloudFormation

Se você usou o AWS CloudFormation para criar recursos, exclua a pilha do CloudFormation depois de conectar e explorar a amostra de instância do EC2 e de instância de banco de dados para não receber mais cobranças por elas.

Para excluir os recursos do CloudFormation

1. Abra o console do AWS CloudFormation.
2. Na página Pilhas no console do CloudFormation, selecione a pilha raiz (a pilha sem o nome VPCStack, BastionStack ou RDSNS).
3. Escolha Excluir.
4. Selecione Excluir pilha quando a confirmação for solicitada.

Consulte mais informações de como excluir uma pilha no CloudFormation em Deleting a stack on the AWS CloudFormation console no Guia do usuário do AWS CloudFormation.

(Opcional) Conectar a instância de banco de dados a uma função do Lambda

Você também pode conectar a instância de banco de dados do RDS para SQL Server a um recurso de computação sem servidor do Lambda. As funções do Lambda permitem que você execute código sem provisionar nem gerenciar a infraestrutura. Uma função do Lambda também permite que você responda automaticamente a solicitações de execução de código em qualquer escala, de dezenas de eventos por dia a centenas por segundo. Para obter mais informações, consulte Conectar automaticamente uma função do Lambda e uma instância de banco de dados.

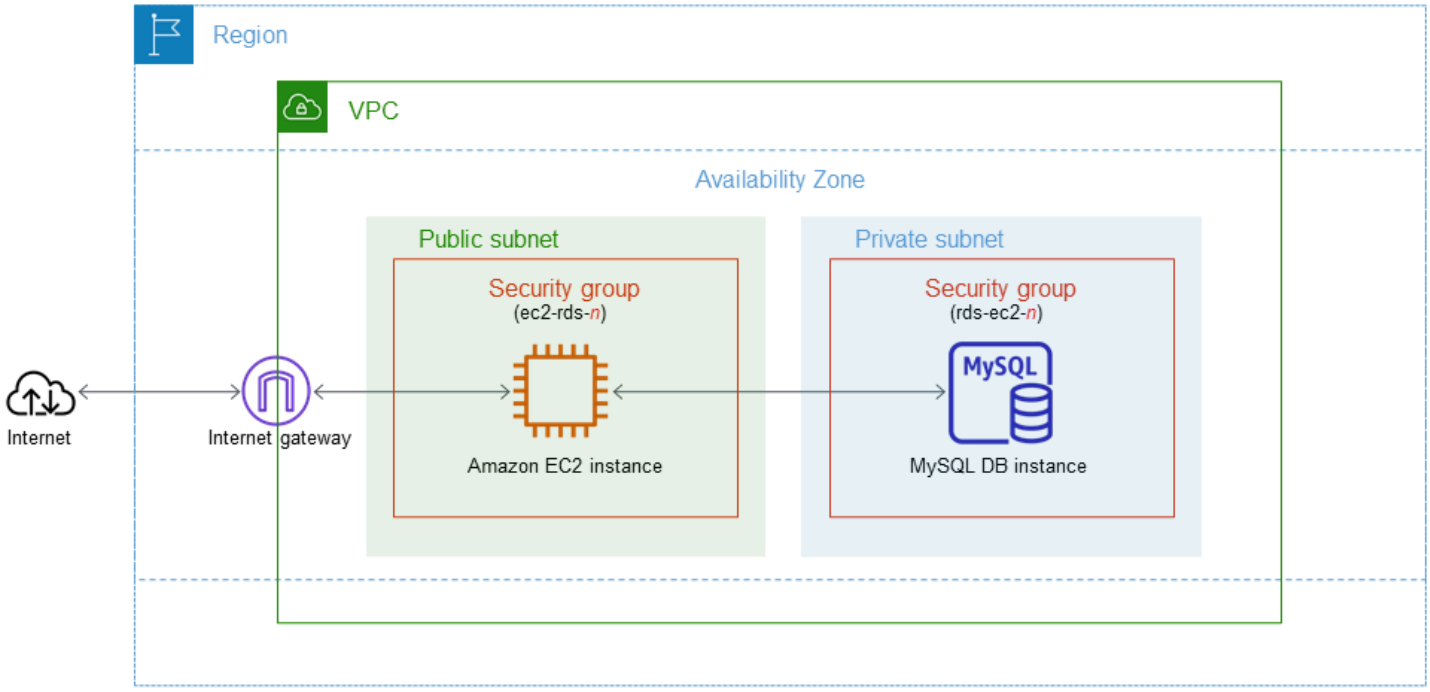
Criar e conectar a uma instância de banco de dados MySQL

Este tutorial cria uma instância do EC2 e uma instância de banco de dados do RDS para MySQL. O tutorial mostra como acessar a instância de banco de dados pela instância do EC2 usando um cliente MySQL padrão. Como prática recomendada, este tutorial cria uma instância de banco de dados privada em uma nuvem privada virtual (VPC). Na maioria dos casos, outros recursos na mesma VPC, como instâncias do EC2, podem acessar a instância de banco de dados, mas recursos fora da VPC não podem acessá-la.

Depois de concluir o tutorial, haverá uma sub-rede pública e privada em cada zona de disponibilidade na sua VPC. Em uma zona de disponibilidade, a instância do EC2 está na sub-rede pública e a instância de banco de dados está na sub-rede privada.

> **Important**
>
> Não há cobrança para criar uma conta da AWS. No entanto, ao concluir este tutorial, é possível gerar custos para os recursos da AWS que você usa. Se esses recursos não forem mais necessários após a conclusão do tutorial, você poderá excluí-los.

O diagrama a seguir mostrará a configuração quando o tutorial estiver completo.



Esse tutorial permite criar recursos usando um dos seguintes métodos:

1. Use o AWS Management Console: Criar uma instância de banco de dados do MySQL e Criar uma instância do EC2
2. Use o AWS CloudFormation para criar a instância de banco de dados e a instância do EC2: (Opcional) Criar VPC, instância do EC2 e instância do MySQL usando o AWS CloudFormation

O primeiro método usa Criação fácil para criar uma instância de banco de dados privada do MySQL com o AWS Management Console. Aqui, você especifica somente o tipo de mecanismo de banco de dados, o tamanho da instância de banco de dados e o identificador da instância de banco de dados. A opção Easy create (Criação fácil) usa a configuração padrão para as outras opções de configuração.

Ao usar a opção Criação padrão, é possível especificar mais opções de configuração ao criar uma instância de banco de dados. Essas opções incluem configurações de disponibilidade, segurança, backups e manutenção. Para criar uma instância de banco de dados pública, você deve usar a Criação padrão. Para mais informações, consulte Criar uma instância de banco de dados do Amazon RDS.

Tópicos

- Pré-requisitos
- Criar uma instância do EC2
- Criar uma instância de banco de dados do MySQL
- (Opcional) Criar VPC, instância do EC2 e instância do MySQL usando o AWS CloudFormation
- Conectar-se a uma instância de banco de dados MySQL
- Excluir a instância do EC2 e a instância de banco de dados
- (Opcional) Excluir a instância do EC2 e a instância de banco de dados criada com o CloudFormation
- (Opcional) Conectar a instância de banco de dados a uma função do Lambda

Pré-requisitos

Antes de começar, conclua as etapas nas seguintes seções:

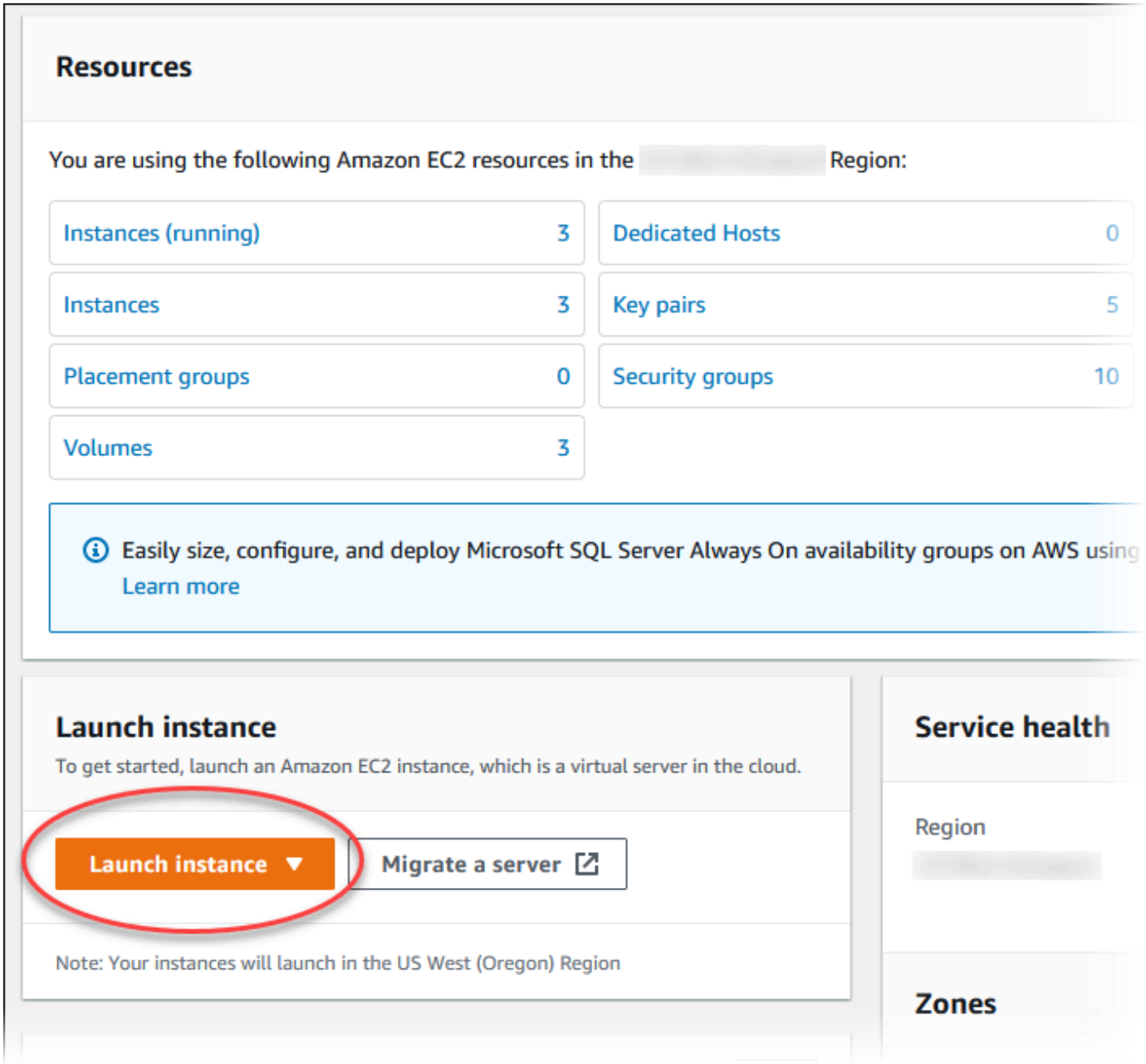
- Inscrever-se para uma Conta da AWS
- Criar um usuário com acesso administrativo

Criar uma instância do EC2

Crie uma instância do Amazon EC2 que você usará para se conectar ao banco de dados.

Para criar uma instância do EC2

1. Faça login no AWS Management Console e abra o console do Amazon EC2 em https://console.aws.amazon.com/ec2/.
2. No canto superior direito do AWS Management Console, selecione a Região da AWS em que você deseja criar a instância do EC2.
3. Escolha Painel do EC2 e Executar instância, conforme mostrado na imagem a seguir.

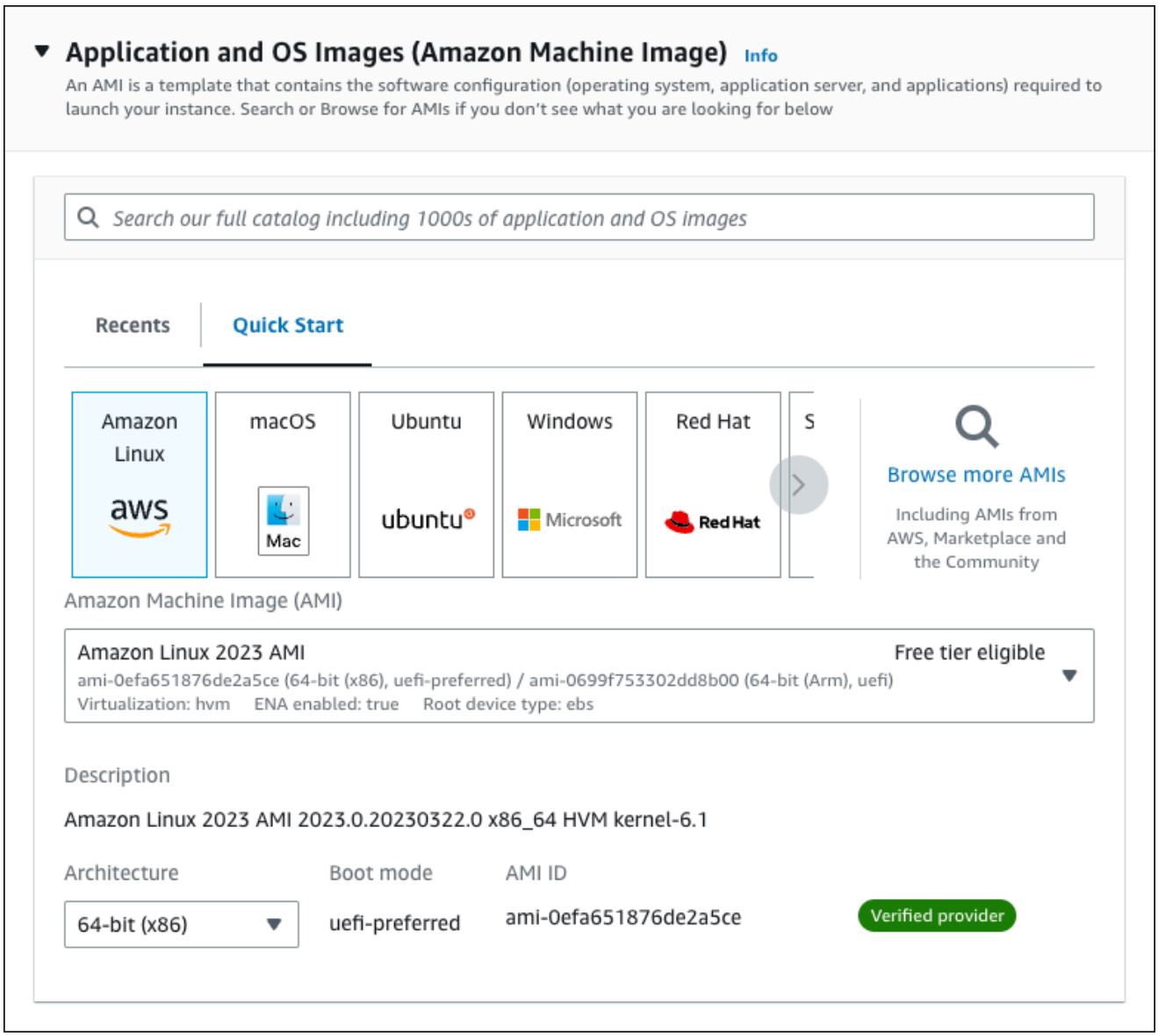


A página Iniciar uma instância é aberta.

4. Escolha as configurações a seguir na página Iniciar uma instância.

   a. Em Name and tags (Nome e etiquetas), em Name (Nome), insira **`ec2-database-connect`**.

   b. Em Imagens de aplicações e sistemas operacionais (imagem de máquina da Amazon), selecione Amazon Linux e, depois, AMI do Amazon Linux 2023. Mantenha as seleções padrão nas outras opções.



c. Em Instance type (Tipo de instância), escolha t2.micro.

d. Em Key pair (login) (Par de chaves (login)), escolha um Key pair name (Nome do par de chaves) para usar um par de chaves existente. Para criar um par de chaves para a instância do Amazon EC2, escolha Create new key pair (Criar um novo key pair), depois use a janela Create key pair (Criar par de chaves) para criá-lo.

   Consulte mais informações sobre como criar um par de chaves em Criar um par de chaves no Guia do usuário do Amazon EC2.

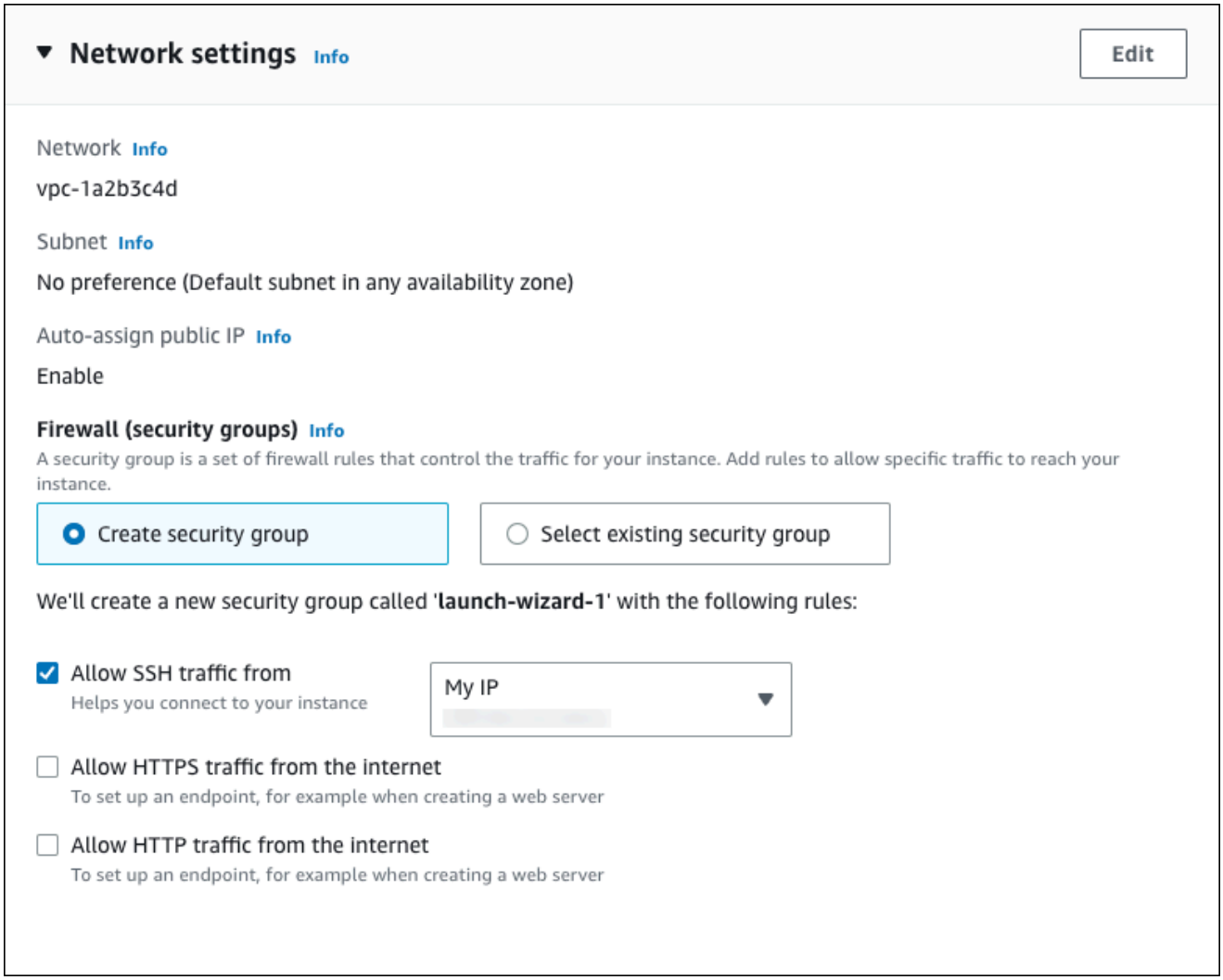
e. Para Permitir tráfego SSH, em Configurações de rede, selecione a origem das conexões SSH com a instância do EC2.

Você pode escolher My IP (Meu IP) se o endereço IP exibido estiver correto para conexões SSH. Caso contrário, determine o endereço IP a ser usado para se conectar a instâncias do EC2 em sua VPC usando Secure Shell (SSH). Para determinar seu endereço IP público, em uma janela ou guia diferente do navegador, é possível usar o serviço em https://checkip.amazonaws.com. Um exemplo de endereço IP é 192.0.2.1/32.

Em muitos casos, você pode se conectar por meio de um provedor de serviços de Internet (ISP) ou atrás de um firewall sem um endereço IP estático. Em caso afirmativo, determine o intervalo de endereços IP utilizado por computadores cliente.

> **Warning**
>
> Se usar `0.0.0.0/0` para acesso do SSH, você possibilitará que todos os endereços IP acessem suas instâncias públicas do EC2 usando SSH. Essa abordagem é aceitável por um período curto em um ambiente de teste, mas não é seguro em ambientes de produção. Em produção, autorize somente um endereço IP específico ou um intervalo de endereços para acessar suas instâncias do EC2 usando SSH.

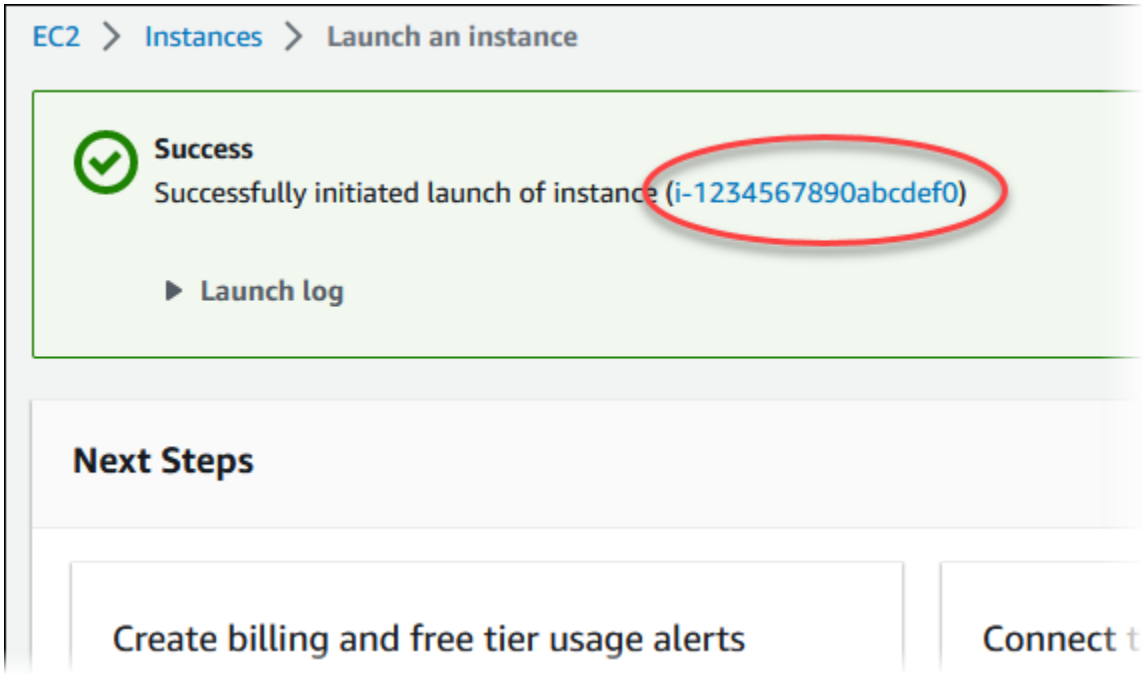
A imagem a seguir mostra um exemplo da seção Configurações de rede.



f. Mantenha os valores padrão para as seções restantes.

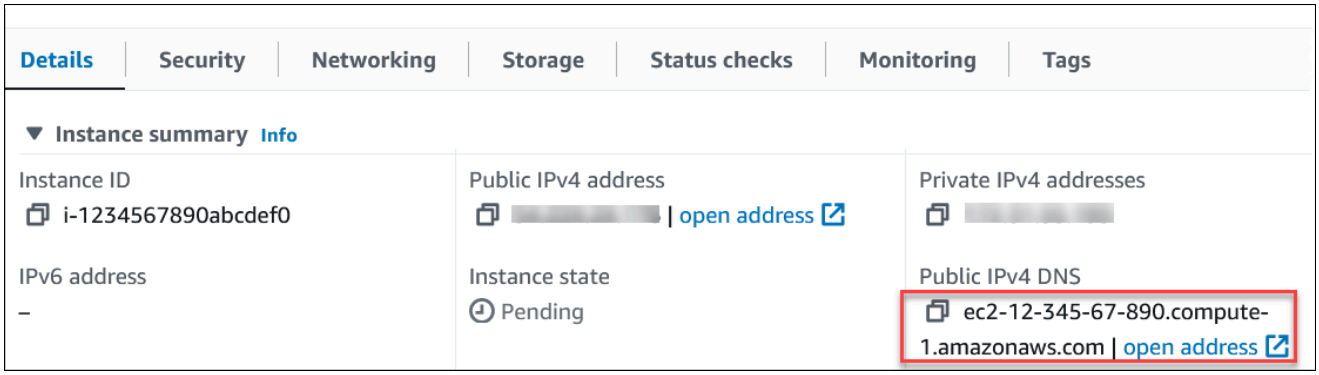
g. Revise um resumo da configuração da instância do EC2 no painel Resumo e, quando estiver com tudo pronto, escolha Executar instância.

5. Na página Status de inicialização, anote o identificador de sua nova instância do EC2, por exemplo: `i-1234567890abcdef0`.

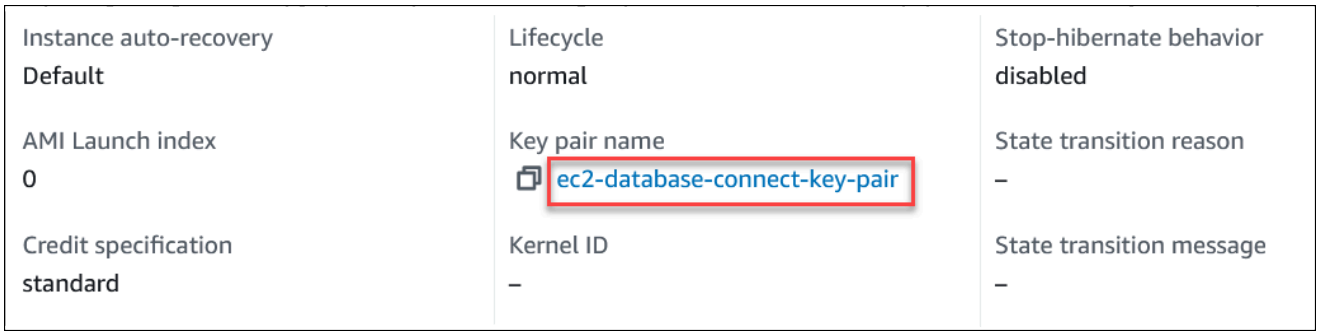


6. Selecione o identificador de instância do EC2 para abrir a lista de instâncias do EC2 e, depois, selecione sua instância do EC2.
7. Na guia Detalhes, observe os seguintes valores, necessários ao se conectar utilizando SSH:

   a. No Resumo da instância, observe o valor do DNS IPv4 público.



   b. Em Detalhes da instância, observe o valor do nome do par de chaves.



8. Espere até o Estado da instância para a instância do EC2 ficar como Executando antes de continuar.

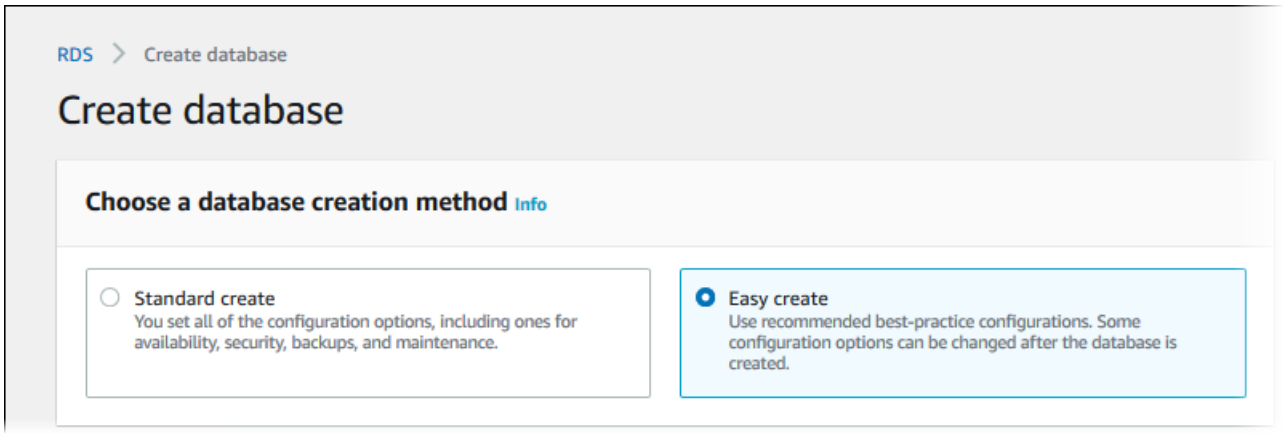
Criar uma instância de banco de dados do MySQL

O bloco de construção básico do Amazon RDS é a instância do banco de dados. Este ambiente é onde você executa seus bancos de dados MySQL.

Neste exemplo, use a opção Criação fácil para criar uma instância de banco de dados executando o mecanismo de banco de dados MySQL com uma classe de instâncias de banco de dados db.t3.micro.

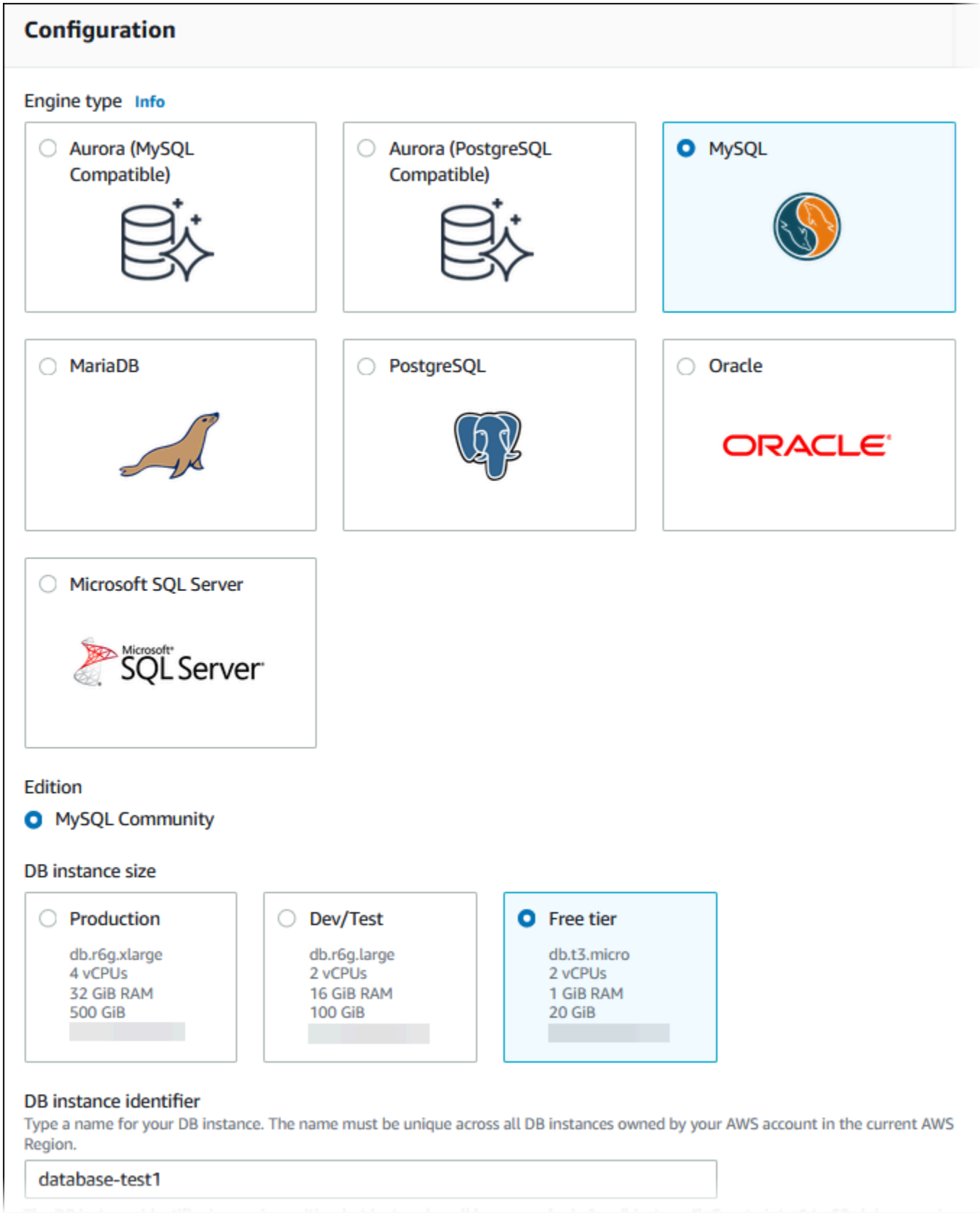
Como criar uma instância de banco de dados MySQL com a criação fácil habilitada

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No canto superior direito do console do Amazon RDS, selecione a Região da AWS que você utilizou para a instância do EC2 anteriormente.
3. No painel de navegação, escolha Databases (Bancos de dados).
4. Selecione a opção Create database (Criar banco de dados) e verifique se a opção Easy Create (Criação fácil) está habilitada.



5. Em Configuration (Configuração), escolha MySQL.
6. Em DB instance size (Tamanho da instância de banco de dados), escolha Free tier (Nível gratuito). Nível gratuito aparece para contas do plano gratuito. Sandbox aparece para contas do plano pago.
7. Em Identificador de instância de banco de dados, insira **`database-test1`**.
8. Em Nome do usuário principal, insira um nome para o usuário mestre ou deixe o nome padrão.

A página Create database (Criar banco de dados) deve ser semelhante à imagem a seguir. Para contas do plano gratuito, aparece Nível gratuito. Para contas do plano pago, aparece Sandbox.

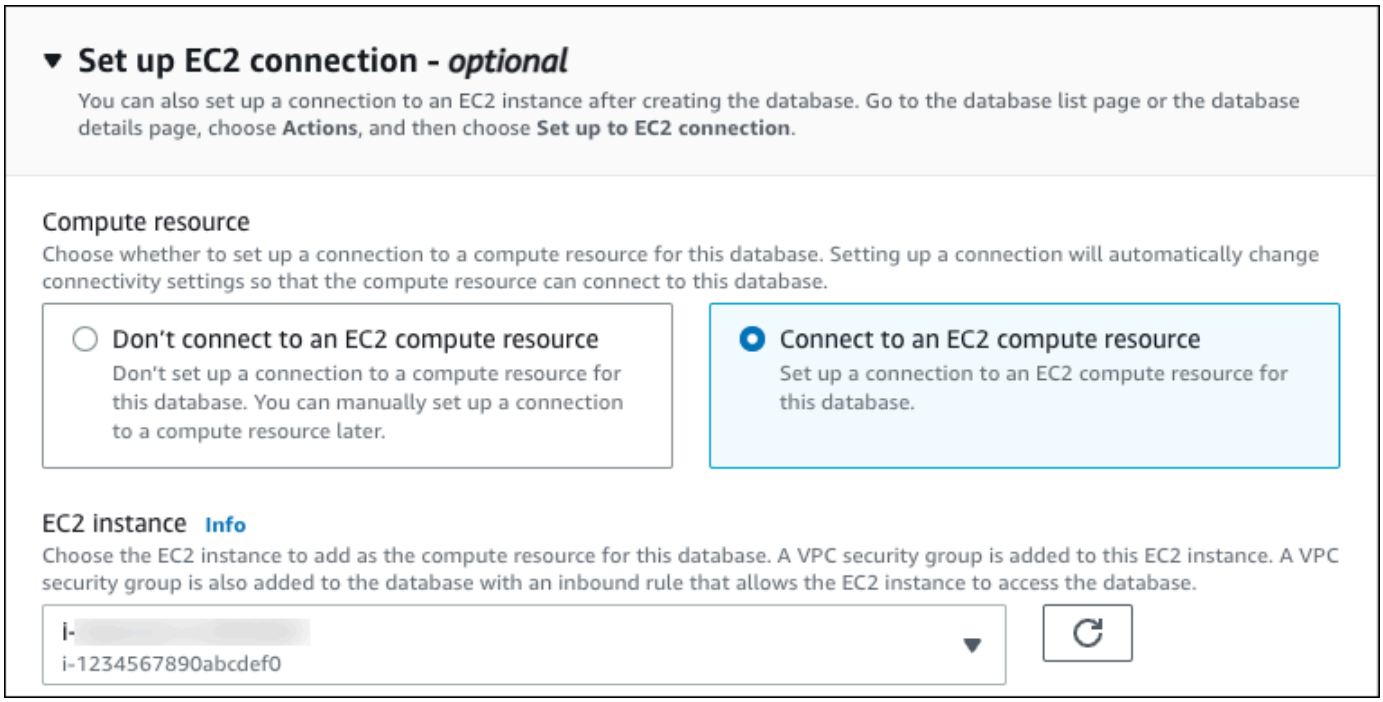

Configuration
Engine type Info
Aurora (MySQL Compatible)
Aurora (PostgreSQL Compatible)
MySQL
MariaDB
PostgreSQL
Oracle
ORACLE
Microsoft SQL Server
Microsoft SQL Server
Edition
MySQL Community
DB instance size
Production
db.r6g.xlarge
4 vCPUs
32 GiB RAM
500 GiB
Dev/Test
db.r6g.large
2 vCPUs
16 GiB RAM
100 GiB
Free tier
db.t3.micro
2 vCPUs
1 GiB RAM
20 GiB
DB instance identifier
Type a name for your DB instance. The name must be unique across all DB instances owned by your AWS account in the current AWS Region.
database-test1

9. Para usar uma senha primária gerada automaticamente para a instância de banco de dados, marque a opção Gerar uma senha automaticamente.

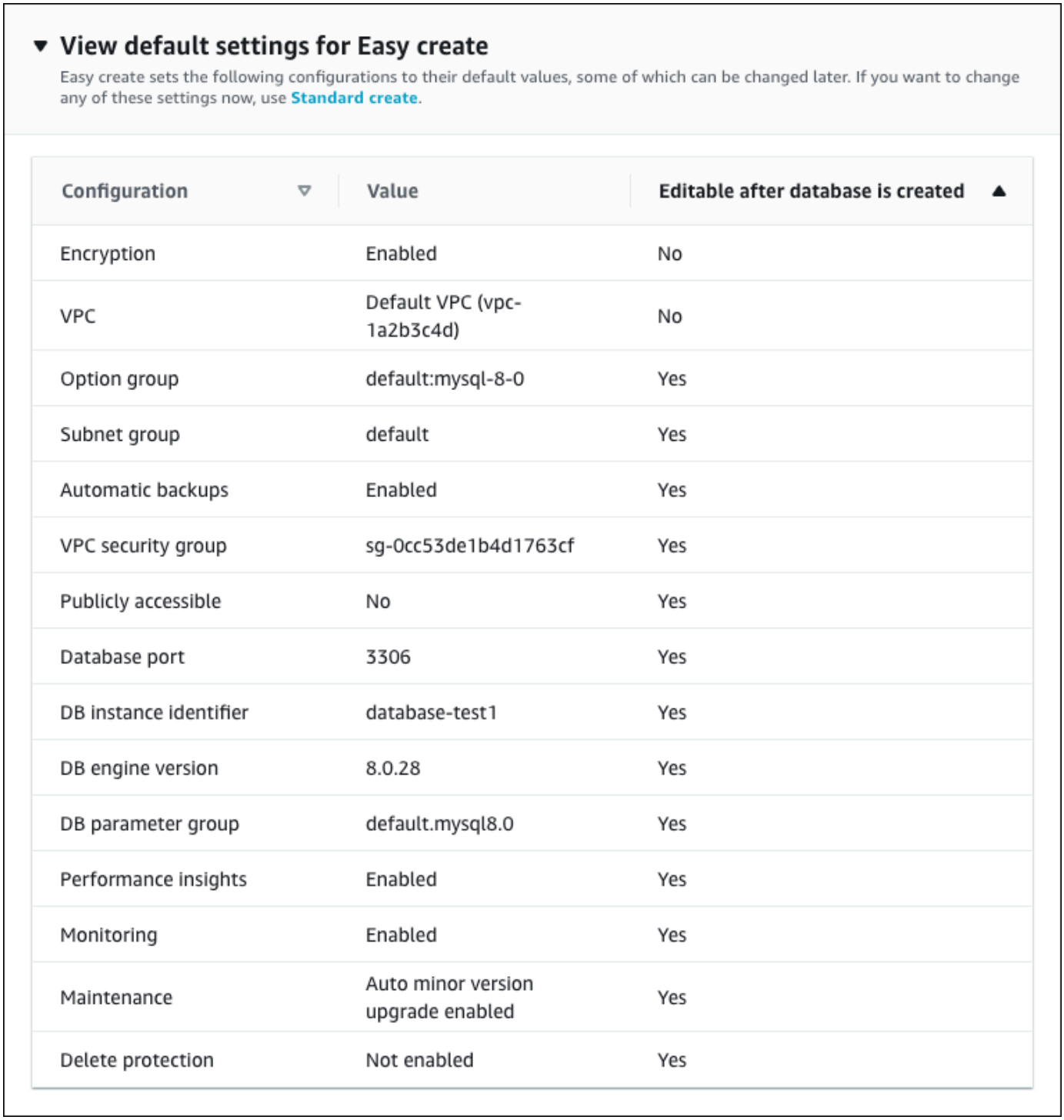
   Para inserir sua senha primária, desmarque a opção Gerar uma senha automaticamente, depois insira a mesma senha em Senha primária e em Confirmar senha.

10. Para configurar uma conexão com a instância do EC2 que você criou anteriormente, abra Configurar conexão do EC2: opcional.

    Selecione Conectar-se a um recurso computacional do EC2. Selecione a instância do EC2 que você criou anteriormente.



11. (Opcional) Abra View default settings for Easy create (Visualizar as configurações padrão da criação fácil).



View default settings for Easy create

Easy create sets the following configurations to their default values, some of which can be changed later. If you want to change any of these settings now, use **Standard create**.

Configuration	Value	Editable after database is created
Encryption	Enabled	No
VPC	Default VPC (vpc-1a2b3c4d)	No
Option group	default:mysql-8-0	Yes
Subnet group	default	Yes
Automatic backups	Enabled	Yes
VPC security group	sg-0cc53de1b4d1763cf	Yes
Publicly accessible	No	Yes
Database port	3306	Yes
DB instance identifier	database-test1	Yes
DB engine version	8.0.28	Yes
DB parameter group	default.mysql8.0	Yes
Performance insights	Enabled	Yes
Monitoring	Enabled	Yes
Maintenance	Auto minor version upgrade enabled	Yes
Delete protection	Not enabled	Yes

Você pode examinar as configurações padrão usadas com a opção Easy Create (Criação fácil). A coluna Editável após a criação do banco de dados mostra quais opções podem ser alteradas após a criação do banco de dados.

- Se uma configuração tiver Não nessa coluna e você quiser uma configuração diferente, poderá usar a opção Criação padrão para criar a instância de banco de dados.
- Se uma configuração tiver Sim nessa coluna e você quiser uma configuração diferente, poderá usar a opção Criação padrão para criar a instância de banco de dados, ou modificar a instância de banco de dados depois de criá-la para alterar a configuração.

12. Selecione Criar banco de dados.

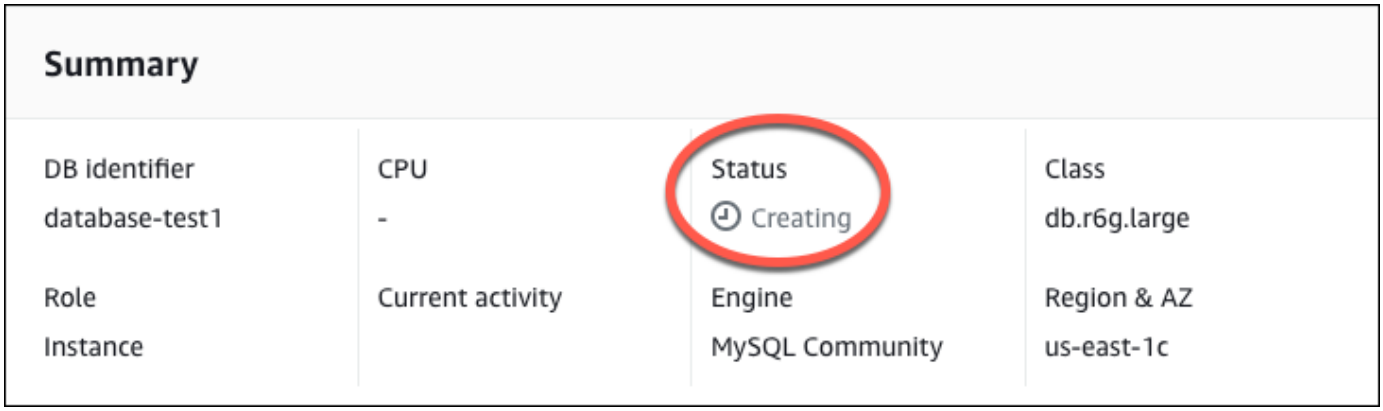
    Para exibir o nome de usuário mestre e a senha da instância de banco de dados, escolha View credential details (Ver detalhes da credencial).

    Use o nome de usuário e a senha que aparecem para se conectar à instância de banco de dados como o usuário mestre.

    > **Important**
    >
    > Você não pode visualizar a senha do usuário principal novamente. Caso você não a registre, talvez seja necessário alterá-la.
    > Se for necessário alterar a senha do usuário mestre depois que a instância de banco de dados estiver disponível, será possível modificar a instância de banco de dados para fazer isso. Para ter mais informações sobre a modificação de uma instância de banco de dados, consulte Modificar uma instância de banco de dados do Amazon RDS.

13. Na lista Bancos de dados, escolha o nome da nova instância de banco de dados MySQL para mostrar seus detalhes.

    A instância de banco de dados permanecerá no status Criando até estar pronta para ser usada.



Quando o status muda para Available (Disponível), você pode se conectar à instância de banco de dados. Dependendo da classe da instância de banco de dados e da quantidade de armazenamento, pode levar até 20 minutos para que a nova instância esteja disponível.

(Opcional) Criar VPC, instância do EC2 e instância do MySQL usando o AWS CloudFormation

Em vez de usar o console para criar a VPC, a instância do EC2 e a instância do MySQL, você pode usar o AWS CloudFormation para provisionar recursos da AWS tratando a infraestrutura como código. Para ajudar a organizar os recursos da AWS em unidades menores e mais gerenciáveis, você pode usar a funcionalidade de pilha aninhada do AWS CloudFormation. Consulte mais informações em Criar uma pilha no console do AWS CloudFormation e Trabalhar com pilhas aninhadas.

> **Important**
>
> O AWS CloudFormation é gratuito, mas os recursos que o CloudFormation são ativos. Você incorre nas taxas de uso padrão para esses recursos até que os encerre. Para ter mais informações, consulte Preço do Amazon RDS para MySQL.

Para criar recursos usando o console do AWS CloudFormation, conclua as seguintes etapas:

- Baixar o modelo do CloudFormation
- Configurar recursos usando o CloudFormation

Baixar o modelo do CloudFormation

Um modelo do CloudFormation é um arquivo de texto JSON ou YAML que contém as informações da configuração dos recursos que você deseja criar na pilha. Esse modelo também cria uma VPC e um bastion host para você junto com a instância do RDS.

Para baixar o arquivo de modelo, abra o link MySQL CloudFormation template.

Na página do Github, clique no botão Baixar arquivo bruto para salvar o arquivo YAML do modelo.

Configurar recursos usando o CloudFormation

> **Note**
>
> Antes de iniciar esse processo, verifique se você tem um par de chaves para uma instância do EC2 na Conta da AWS. Para obter mais informações, consulte Pares de chaves do Amazon EC2 e instâncias do Linux.

Ao usar o modelo do AWS CloudFormation, você deve selecionar os parâmetros certos para garantir que os recursos sejam criados corretamente. Siga as etapas abaixo:

1. Faça login no AWS Management Console e abra o console AWS CloudFormation em https://console.aws.amazon.com/cloudformation.
2. Selecione Create Stack (Criar pilha).
3. Na seção Especificar modelo, selecione Fazer upload de um arquivo de modelo do computador e escolha Próximo.
4. Na página Especificar detalhes da pilha, defina os seguintes parâmetros:
   a. Defina o Nome da pilha como MySQLTestStack.
   b. Em Parâmetros, defina Zonas de disponibilidade selecionando três zonas de disponibilidade.
   c. Em Configuração do bastion host do Linux, em Nome da chave, selecione um par de chaves para fazer login na instância do EC2.
   d. Em Configurações do bastion host do Linux, defina o Intervalo de IP permitido para o endereço IP. Para conectar instâncias do EC2 à VPC usando o Secure Shell (SSH), determine o endereço IP público usando o serviço em https://checkip.amazonaws.com. Um exemplo de endereço IP é 192.0.2.1/32.

> **Warning**
>
> Se usar `0.0.0.0/0` para acesso do SSH, você possibilitará que todos os endereços IP acessem suas instâncias públicas do EC2 usando SSH. Essa abordagem é aceitável por um período curto em um ambiente de teste, mas não é seguro em ambientes de produção. Em produção, autorize somente um endereço IP específico ou um intervalo de endereços para acessar suas instâncias do EC2 usando SSH.

e. Em Configuração geral do banco de dados, defina a Classe da instância do banco de dados como db.t3.micro.

f. Defina o Nome do banco de dados como **`database-test1`**.

g. Em Nome de usuário principal do banco de dados, insira um nome para o usuário principal.

h. Defina a Gerenciar senha de usuário principal do banco de dados com o Secrets Manager como `false` para esse tutorial.

i. Em Senha do banco de dados, defina uma senha de sua escolha. Lembre-se dessa senha para as etapas seguintes do tutorial.

j. Em Configuração de armazenamento de banco de dados, defina o Tipo de armazenamento do banco de dados como gp2.

k. Em Configuração de monitoramento de banco de dados, defina Habilitar o Performance Insights do RDS como falso.

l. Deixe todas as outras configurações com os valores padrão. Clique em Próximo para continuar.

5. Na página Configurar opções de pilha, mantenha todas as opções padrão. Clique em Próximo para continuar.
6. Na página Revisar pilha, selecione Enviar depois de verificar as opções do banco de dados e do bastion host do Linux.

Depois que o processo de criação da pilha for concluído, visualize as pilhas com os nomes BastionStack e RDSNS para anotar as informações necessárias para se conectar ao banco de dados. Consulte mais informações em Viewing AWS CloudFormation stack data and resources on the AWS Management Console.

Conectar-se a uma instância de banco de dados MySQL

Você pode usar qualquer aplicação cliente padrão SQL para se conectar à instância de banco de dados. Neste exemplo, você se conecta a uma instância de banco de dados MySQL usando o cliente da linha de comando mysql.

Como conectar-se a uma instância de banco de dados MySQL

1. Encontre o endpoint (nome de DNS) e o número da porta para sua instância de banco de dados.

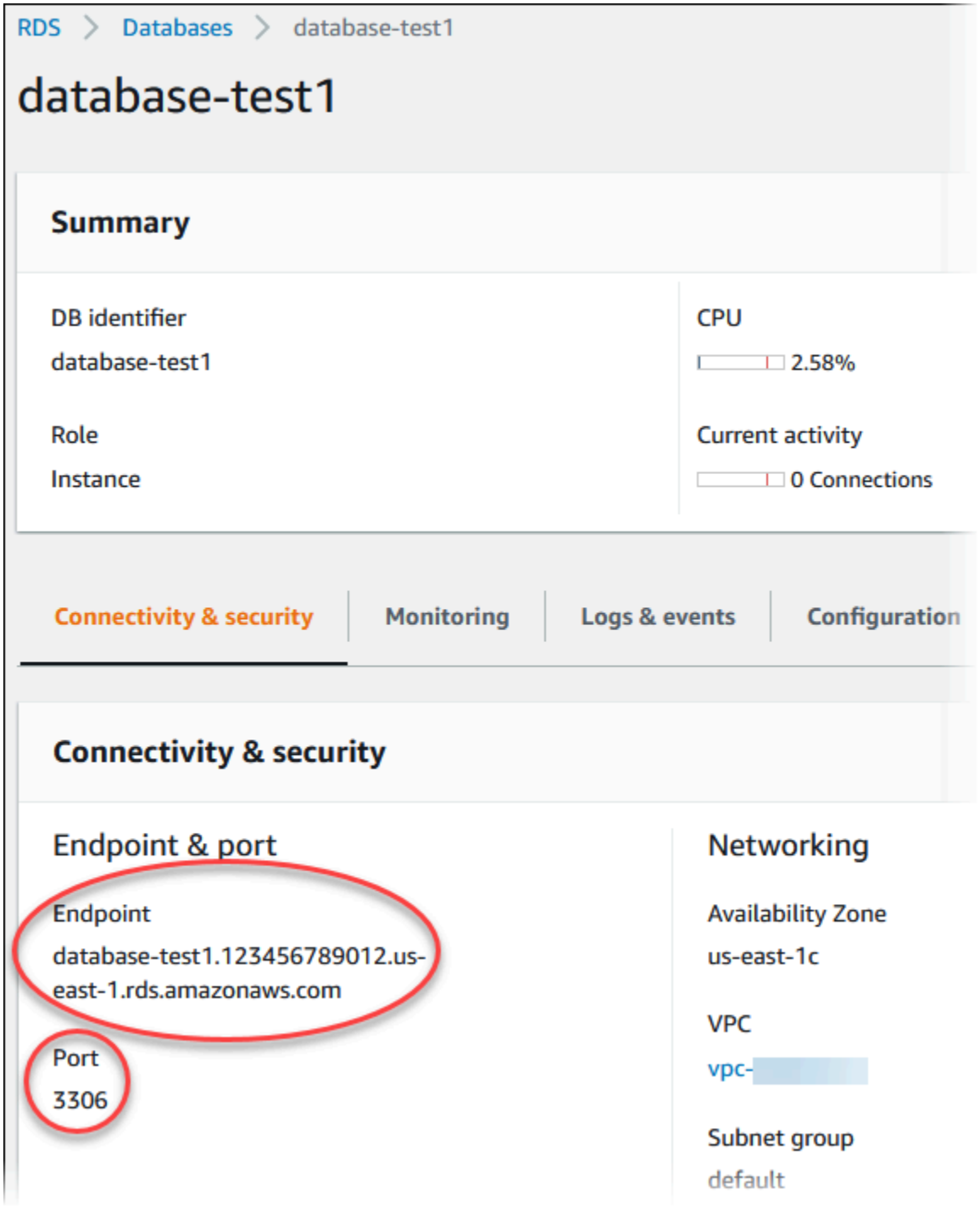
   a. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.

b. No canto superior direito do console do Amazon RDS, escolha a Região da AWS da instância de banco de dados.

c. No painel de navegação, escolha Databases (Bancos de dados).

d. Escolha o nome da instância de banco de dados MySQL para exibir os detalhes.

e. Na guia Connectivity & security (Conectividade e segurança), copie o endpoint. Além disso, anote o número da porta. Você precisará do endpoint e do número da porta para conectar-se à instância de banco de dados.



2. Conecte-se à instância do EC2 que você criou anteriormente, seguindo as etapas em Conecte-se à sua instância do Linux no Guia do usuário do Amazon EC2.

   Recomendamos que você se conecte à sua instância do EC2 utilizando SSH. Se o utilitário cliente SSH estiver instalado no Windows, Linux ou Mac, você poderá se conectar à instância utilizando o seguinte formato de comando:

```
ssh -i location_of_pem_file ec2-user@ec2-instance-public-dns-name
```

Por exemplo, suponha que `ec2-database-connect-key-pair.pem` esteja armazenado em `/dir1` no Linux e que o DNS IPv4 público para sua instância do EC2 seja `ec2-12-345-678-90.compute-1.amazonaws.com`. Seu comando SSH teria a seguinte aparência:

```
ssh -i /dir1/ec2-database-connect-key-pair.pem ec2-
user@ec2-12-345-678-90.compute-1.amazonaws.com
```

3. Obtenha as últimas correções de bugs e atualizações de segurança atualizando o software na instância do EC2. Para fazer isso, execute o seguinte comando.

> **Note**
>
> A opção `-y` instala as atualizações sem solicitar confirmação. Para examinar atualizações antes da instalação, omita essa opção.

```
sudo dnf update -y
```

4. Para instalar o cliente da linha de comando mysql do mysql no MariaDB no Amazon Linux 2023, execute o seguinte comando:

```
sudo dnf install mariadb105
```

5. Conecte-se à instância de banco de dados MySQL. Por exemplo, insira o comando a seguir. Essa ação permite que você se conecte à instância de banco de dados MySQL usando o cliente do MySQL.

   Substitua o endpoint da instância de banco de dados (nome DNS) por *`endpoint`* e o nome do usuário principal usado por *`admin`*. Forneça a senha mestra usada quando for solicitada uma senha.

```
mysql -h endpoint -P 3306 -u admin -p
```

   Depois de inserir a senha do usuário, você deverá ver uma saída semelhante à seguinte.

```
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 3082
Server version: 8.0.28 Source distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]>
```

Para ter mais informações sobre como se conectar à sua instância de banco de dados do MySQL, consulte Conectar-se à instância de banco de dados do MySQL. Se você não conseguir se conectar à sua instância de banco de dados, consulte Não é possível conectar-se à instância de banco de dados do Amazon RDS.

Por questões de segurança, é uma prática recomendada usar conexões criptografadas. Use uma conexão MySQL não criptografada apenas quando o cliente e o servidor estiverem na mesma VPC e a rede for confiável. Para obter informações sobre como usar conexões criptografadas, consulte Conectar-se à instância de banco de dados do MySQL no Amazon RDS com SSL/TLS por meio do cliente de linha de comandos do MySQL (criptografado).

6. Executar comandos SQL.

   Por exemplo, o seguinte comando SQL mostra a data e a hora atuais:

   ```
   SELECT CURRENT_TIMESTAMP;
   ```

Excluir a instância do EC2 e a instância de banco de dados

Depois de se conectar e explorar a instância do EC2 e a instância de banco de dados criadas, exclua-as para não receber mais cobranças por elas.

Se você usou o AWS CloudFormation para criar recursos, ignore essa etapa e passe para a próxima.

Como excluir a instância do EC2

1. Faça login no AWS Management Console e abra o console do Amazon EC2 em https://console.aws.amazon.com/ec2/.
2. No painel de navegação, escolha Instâncias.

3. Selecione a instância do EC2 e escolha Estado da instância, Encerrar instância.
4. Quando a confirmação for solicitada, escolha Encerrar.

Consulte mais informações sobre como excluir uma instância do EC2 em Como encerrar uma instância no Guia do usuário do Amazon EC2.

Como excluir uma instância de banco de dados sem snapshot de banco de dados final

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Databases (Bancos de dados).
3. Escolha a instância de banco de dados que você deseja excluir.
4. Em Actions, selecione Delete.
5. Desmarque Criar snapshot final? e Reter backups automatizados.
6. Conclua a confirmação e escolha Excluir.

(Opcional) Excluir a instância do EC2 e a instância de banco de dados criada com o CloudFormation

Se você usou o AWS CloudFormation para criar recursos, exclua a pilha do CloudFormation depois de conectar e explorar a amostra de instância do EC2 e de instância de banco de dados para não receber mais cobranças por elas.

Para excluir os recursos do CloudFormation

1. Abra o console do AWS CloudFormation.
2. Na página Pilhas no console do CloudFormation, selecione a pilha raiz (a pilha sem o nome VPCStack, BastionStack ou RDSNS).
3. Escolha Excluir.
4. Selecione Excluir pilha quando a confirmação for solicitada.

Consulte mais informações de como excluir uma pilha no CloudFormation em Deleting a stack on the AWS CloudFormation console no Guia do usuário do AWS CloudFormation.

(Opcional) Conectar a instância de banco de dados a uma função do Lambda

Você também pode conectar a instância de banco de dados do RDS para MySQL a um recurso de computação sem servidor do Lambda. As funções do Lambda permitem que você execute código sem provisionar nem gerenciar a infraestrutura. Uma função do Lambda também permite que você responda automaticamente a solicitações de execução de código em qualquer escala, de dezenas de eventos por dia a centenas por segundo. Para obter mais informações, consulte Conectar automaticamente uma função do Lambda e uma instância de banco de dados.

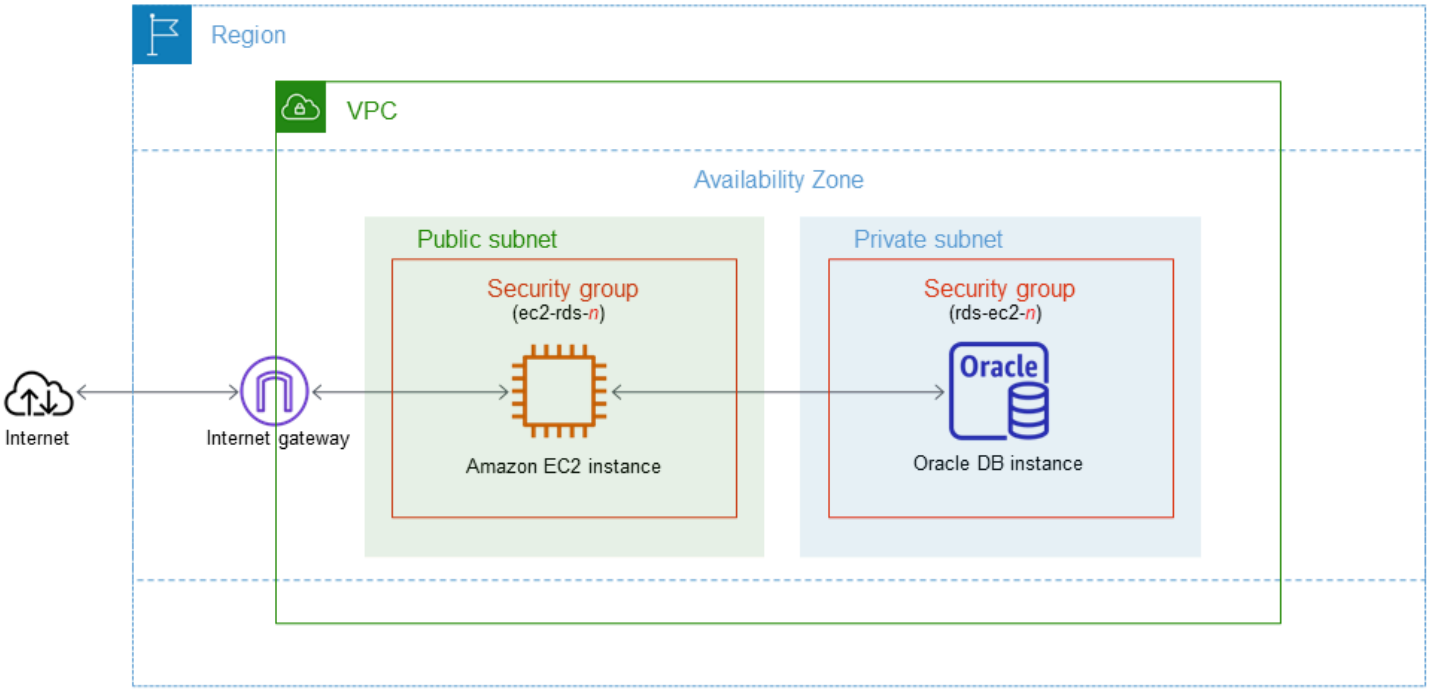
Criar uma instância de banco de dados Oracle e conectar-se a ela

Este tutorial cria uma instância do EC2 e uma instância de banco de dados Oracle. O tutorial mostra como acessar a instância de banco de dados pela instância do EC2 usando um cliente Oracle padrão. Como prática recomendada, este tutorial cria uma instância de banco de dados privada em uma nuvem privada virtual (VPC). Na maioria dos casos, outros recursos na mesma VPC, como instâncias do EC2, podem acessar a instância de banco de dados, mas recursos fora da VPC não podem acessá-la.

Depois de concluir o tutorial, haverá uma sub-rede pública e privada em cada zona de disponibilidade na sua VPC. Em uma zona de disponibilidade, a instância do EC2 está na sub-rede pública e a instância de banco de dados está na sub-rede privada.

> **Important**
>
> Não há cobrança para criar uma conta da AWS. No entanto, ao concluir este tutorial, é possível gerar custos para os recursos da AWS que você usa. Se esses recursos não forem mais necessários após a conclusão do tutorial, você poderá excluí-los.

O diagrama a seguir mostrará a configuração quando o tutorial estiver completo.



Esse tutorial permite criar recursos usando um dos seguintes métodos:

1. Use o AWS Management Console: Etapa 2: Criar uma instância de banco de dados Oracle e Etapa 1: Criar uma instância do EC2
2. Use o AWS CloudFormation para criar a instância de banco de dados e a instância do EC2: (Opcional) Criar VPC, instância do EC2 e instância de banco de dados do Oracle usando o AWS CloudFormation

O primeiro método usa Criação fácil para criar uma instância de banco de dados privada do Oracle com o AWS Management Console. Aqui, você especifica somente o tipo de mecanismo de banco de dados, o tamanho da instância de banco de dados e o identificador da instância de banco de dados. A opção Easy create (Criação fácil) usa a configuração padrão para as outras opções de configuração.

Ao usar a opção Criação padrão, é possível especificar mais opções de configuração ao criar uma instância de banco de dados. Essas opções incluem configurações de disponibilidade, segurança, backups e manutenção. Para criar uma instância de banco de dados pública, você deve usar a Criação padrão. Para mais informações, consulte Criar uma instância de banco de dados do Amazon RDS.

Tópicos

- Pré-requisitos
- Etapa 1: Criar uma instância do EC2
- Etapa 2: Criar uma instância de banco de dados Oracle
- (Opcional) Criar VPC, instância do EC2 e instância de banco de dados do Oracle usando o AWS CloudFormation
- Etapa 3: Conectar seu cliente SQL a uma instância de banco de dados Oracle.
- Etapa 4: Excluir a instância do EC2 e a instância de banco de dados
- (Opcional) Excluir a instância do EC2 e a instância de banco de dados criada com o CloudFormation
- (Opcional) Conectar a instância de banco de dados a uma função do Lambda

Pré-requisitos

Antes de começar, conclua as etapas nas seguintes seções:

- Inscrever-se para uma Conta da AWS

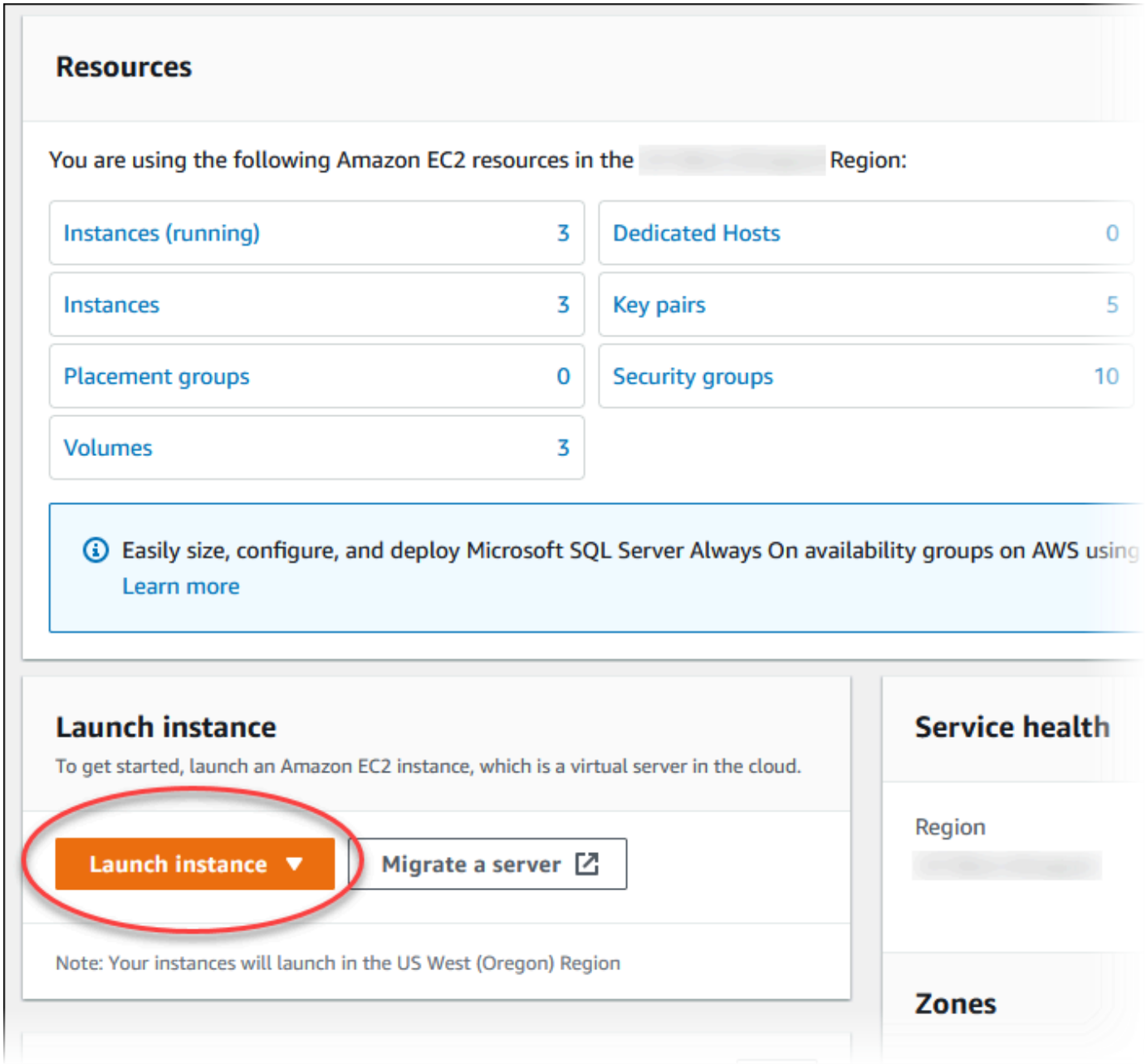
- Criar um usuário com acesso administrativo

Etapa 1: Criar uma instância do EC2

Crie uma instância do Amazon EC2 que você usará para se conectar ao banco de dados.

Para criar uma instância do EC2

1. Faça login no AWS Management Console e abra o console do Amazon EC2 em https://console.aws.amazon.com/ec2/.
2. No canto superior direito do AWS Management Console, selecione a Região da AWS em que você deseja criar a instância do EC2.
3. Escolha Painel do EC2 e Executar instância, conforme mostrado na imagem a seguir.

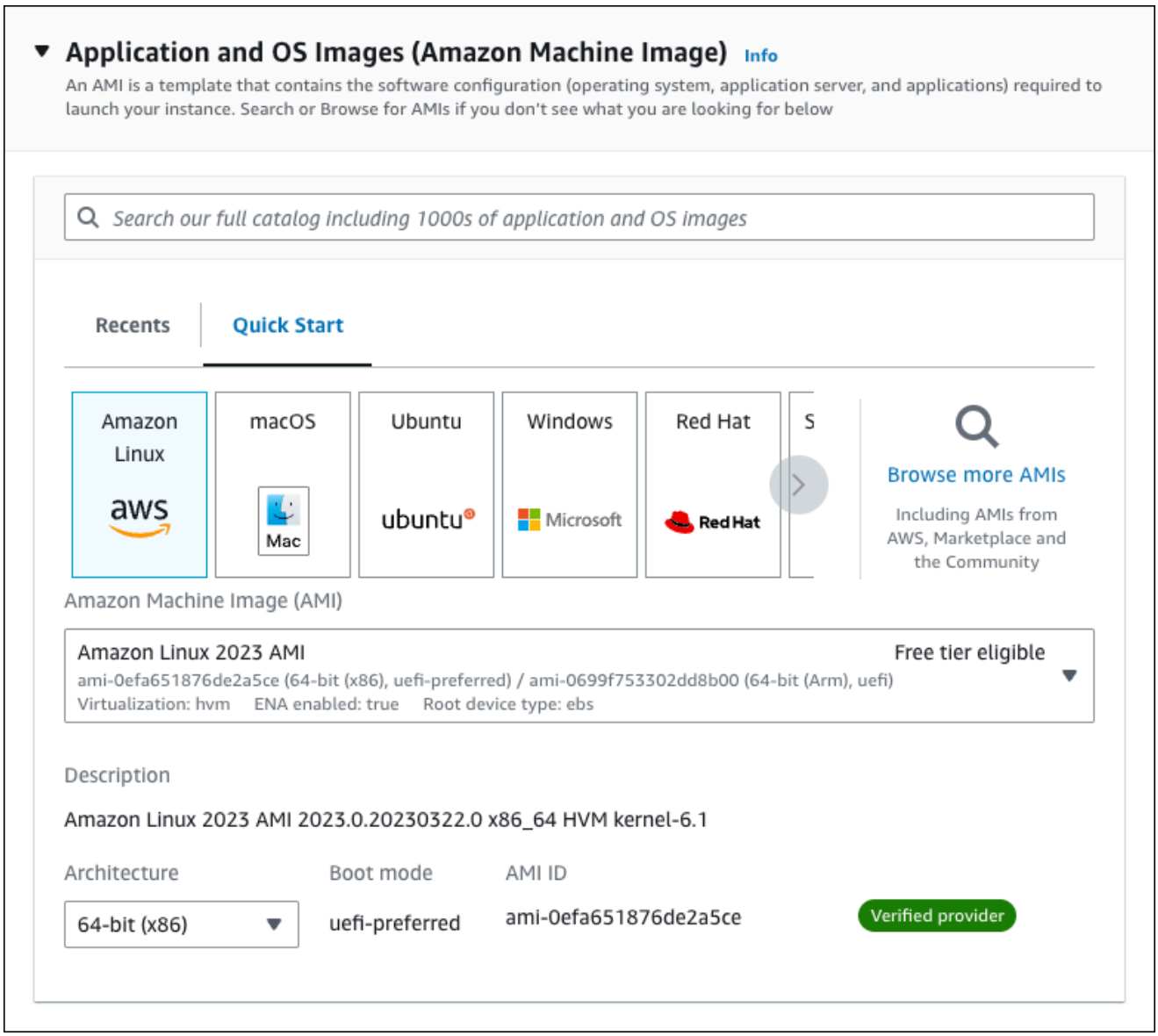


A página Iniciar uma instância é aberta.

4. Escolha as configurações a seguir na página Iniciar uma instância.

   a. Em Name and tags (Nome e etiquetas), em Name (Nome), insira **`ec2-database-connect`**.

   b. Em Imagens de aplicações e sistemas operacionais (imagem de máquina da Amazon), selecione Amazon Linux e, depois, AMI do Amazon Linux 2023. Mantenha as seleções padrão nas outras opções.



c. Em Instance type (Tipo de instância), escolha t2.micro.

d. Em Key pair (login) (Par de chaves (login)), escolha um Key pair name (Nome do par de chaves) para usar um par de chaves existente. Para criar um par de chaves para a instância do Amazon EC2, escolha Create new key pair (Criar um novo key pair), depois use a janela Create key pair (Criar par de chaves) para criá-lo.

   Consulte mais informações sobre como criar um par de chaves em Criar um par de chaves no Guia do usuário do Amazon EC2.

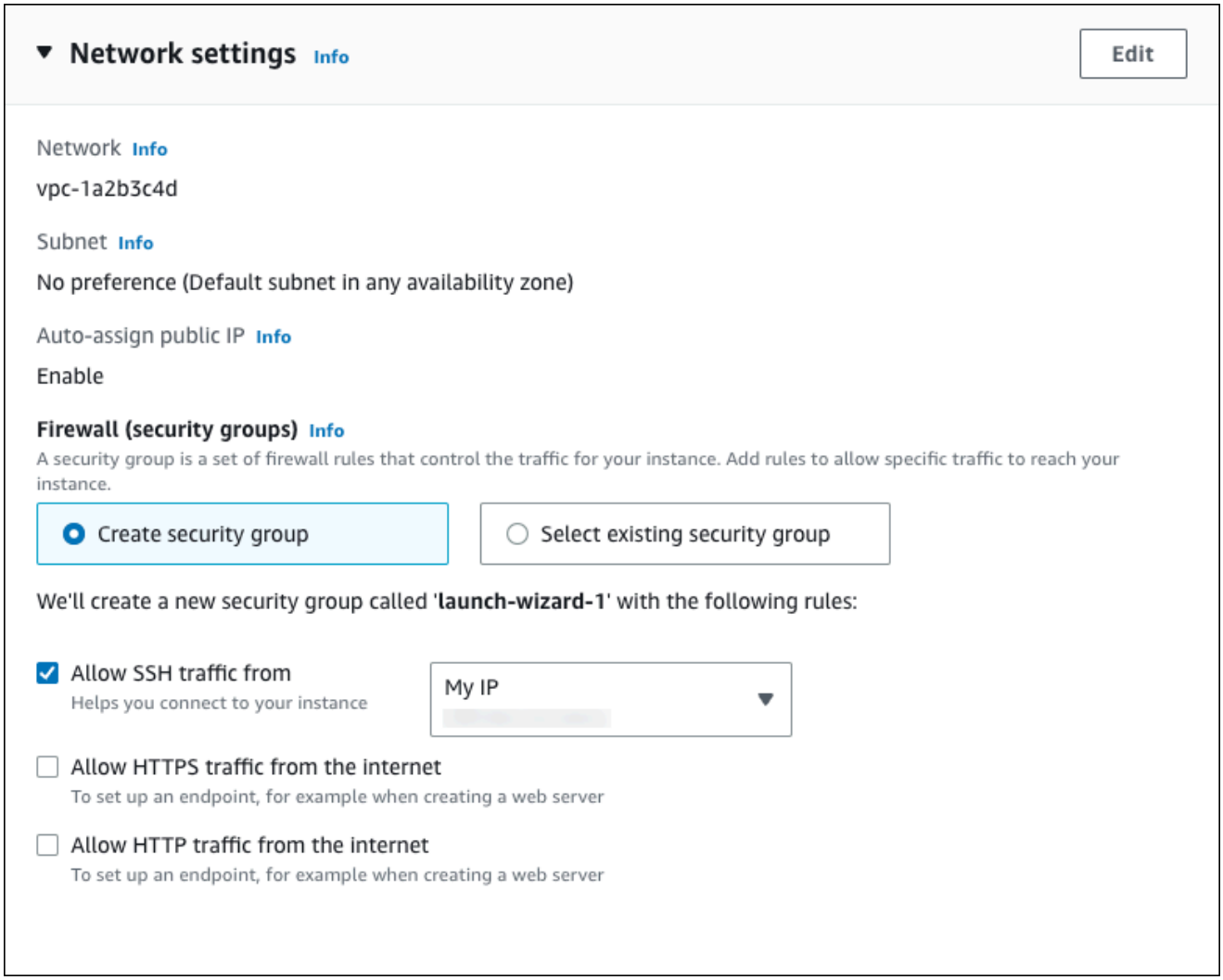
e. Para Permitir tráfego SSH, em Configurações de rede, selecione a origem das conexões SSH com a instância do EC2.

Você pode escolher My IP (Meu IP) se o endereço IP exibido estiver correto para conexões SSH. Caso contrário, determine o endereço IP a ser usado para se conectar a instâncias do EC2 em sua VPC usando Secure Shell (SSH). Para determinar seu endereço IP público, em uma janela ou guia diferente do navegador, é possível usar o serviço em https://checkip.amazonaws.com. Um exemplo de endereço IP é 192.0.2.1/32.

Em muitos casos, você pode se conectar por meio de um provedor de serviços de Internet (ISP) ou atrás de um firewall sem um endereço IP estático. Em caso afirmativo, determine o intervalo de endereços IP utilizado por computadores cliente.

> **Warning**
>
> Se usar `0.0.0.0/0` para acesso do SSH, você possibilitará que todos os endereços IP acessem suas instâncias públicas do EC2 usando SSH. Essa abordagem é aceitável por um período curto em um ambiente de teste, mas não é seguro em ambientes de produção. Em produção, autorize somente um endereço IP específico ou um intervalo de endereços para acessar suas instâncias do EC2 usando SSH.

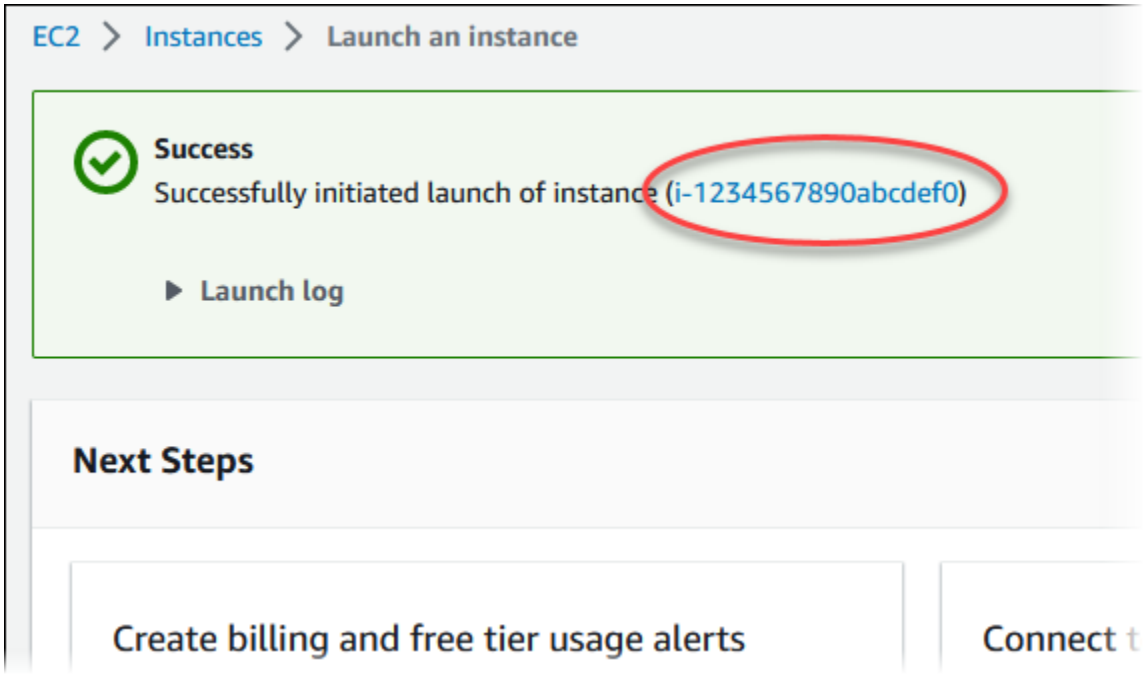
A imagem a seguir mostra um exemplo da seção Configurações de rede.



f. Mantenha os valores padrão para as seções restantes.

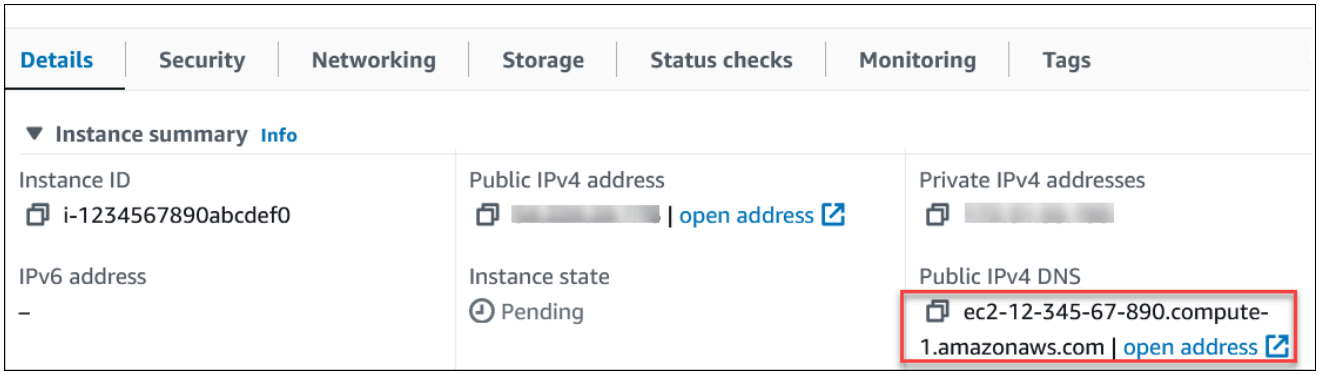
g. Revise um resumo da configuração da instância do EC2 no painel Resumo e, quando estiver com tudo pronto, escolha Executar instância.

5. Na página Status de inicialização, anote o identificador de sua nova instância do EC2, por exemplo: `i-1234567890abcdef0`.

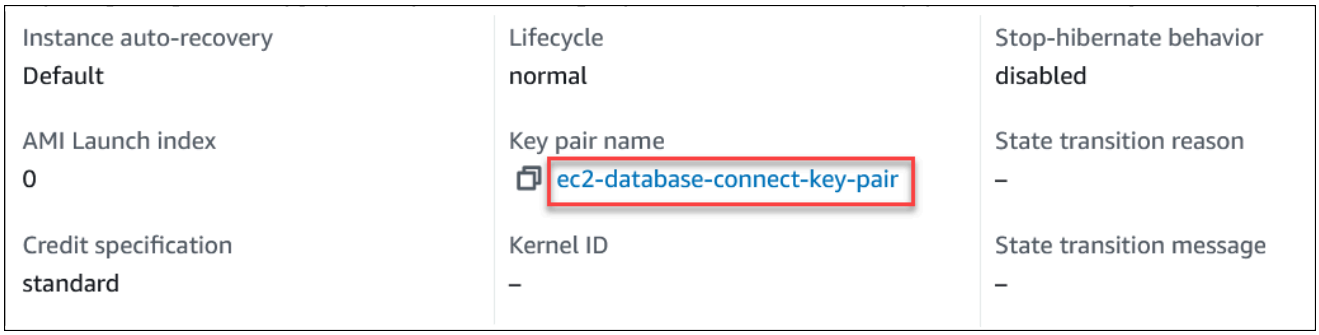


6. Selecione o identificador de instância do EC2 para abrir a lista de instâncias do EC2 e, depois, selecione sua instância do EC2.
7. Na guia Detalhes, observe os seguintes valores, necessários ao se conectar utilizando SSH:

   a. No Resumo da instância, observe o valor do DNS IPv4 público.



   b. Em Detalhes da instância, observe o valor do nome do par de chaves.



8. Espere até o Estado da instância para a instância do EC2 ficar como Executando antes de continuar.

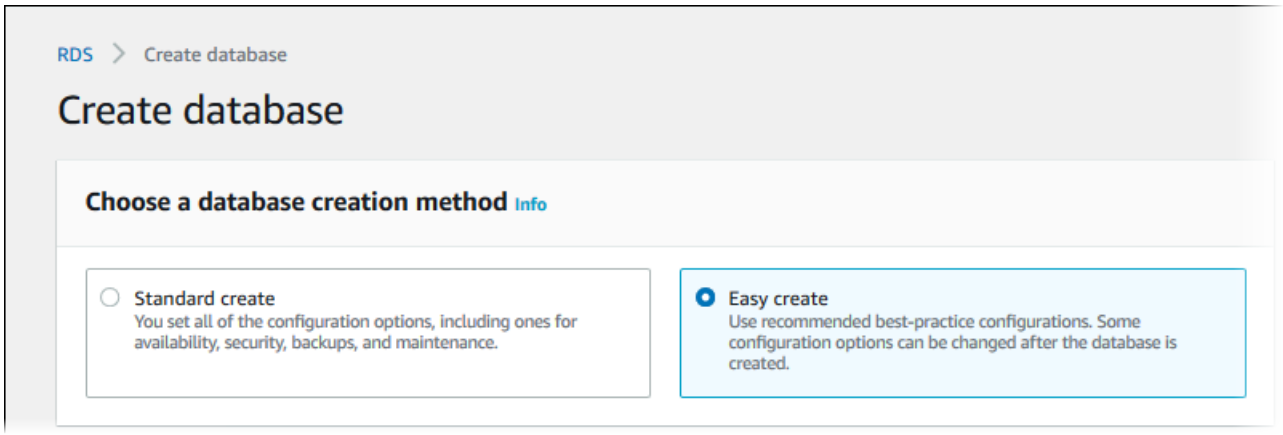
Etapa 2: Criar uma instância de banco de dados Oracle

O bloco de construção básico do Amazon RDS é a instância do banco de dados. Este ambiente é onde você executa seus bancos de dados Oracle.

Neste exemplo, use Criação fácil para criar uma instância de banco de dados executando o mecanismo de banco de dados Oracle com uma classe de instância de banco de dados db.m5.large.

Como criar uma instância de banco de dados Oracle com a criação fácil

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No canto superior direito do console do Amazon RDS, escolha a Região da AWS na qual você deseja criar a instância de banco de dados.
3. No painel de navegação, escolha Databases (Bancos de dados).
4. Selecione a opção Create database (Criar banco de dados) e verifique se a opção Easy Create (Criação fácil) está habilitada.



5. Em Configuration (Configuração), escolha Oracle.
6. Em DB instance size (Tamanho da instância de banco de dados), escolha Dev/Test (Desenvolvimento/teste).
7. Em Identificador de instância de banco de dados, insira **`database-test1`**.
8. Em Nome do usuário principal, insira um nome para o usuário mestre ou deixe o nome padrão.

   A página Create database (Criar banco de dados) deve ser semelhante à imagem a seguir.

Configuration

Engine type Info

- Aurora (MySQL Compatible)
- Aurora (PostgreSQL Compatible)
- MySQL
- MariaDB
- PostgreSQL
- Oracle
- Microsoft SQL Server

Edition

- Oracle Enterprise Edition
  Affordable and full-featured database management system supporting up to 16 vCPUs.
- Oracle Standard Edition Two
  Affordable and full-featured database management system supporting up to 16 vCPUs. Oracle Database Standard Edition Two is a replacement for Standard Edition and Standard Edition One.

DB instance size

- Production
  db.r5.large
  2 vCPUs
  16 GiB RAM
  500 GiB
- Dev/Test
  db.m5.large
  2 vCPUs
  8 GiB RAM
  100 GiB

DB instance identifier

Type a name for your DB instance. The name must be unique across all DB instances owned by your AWS account in the current AWS Region.

database-test1

The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 60 alphanumeric characters or hyphens. First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

Master username Info

Type a login ID for the master user of your DB instance.

admin

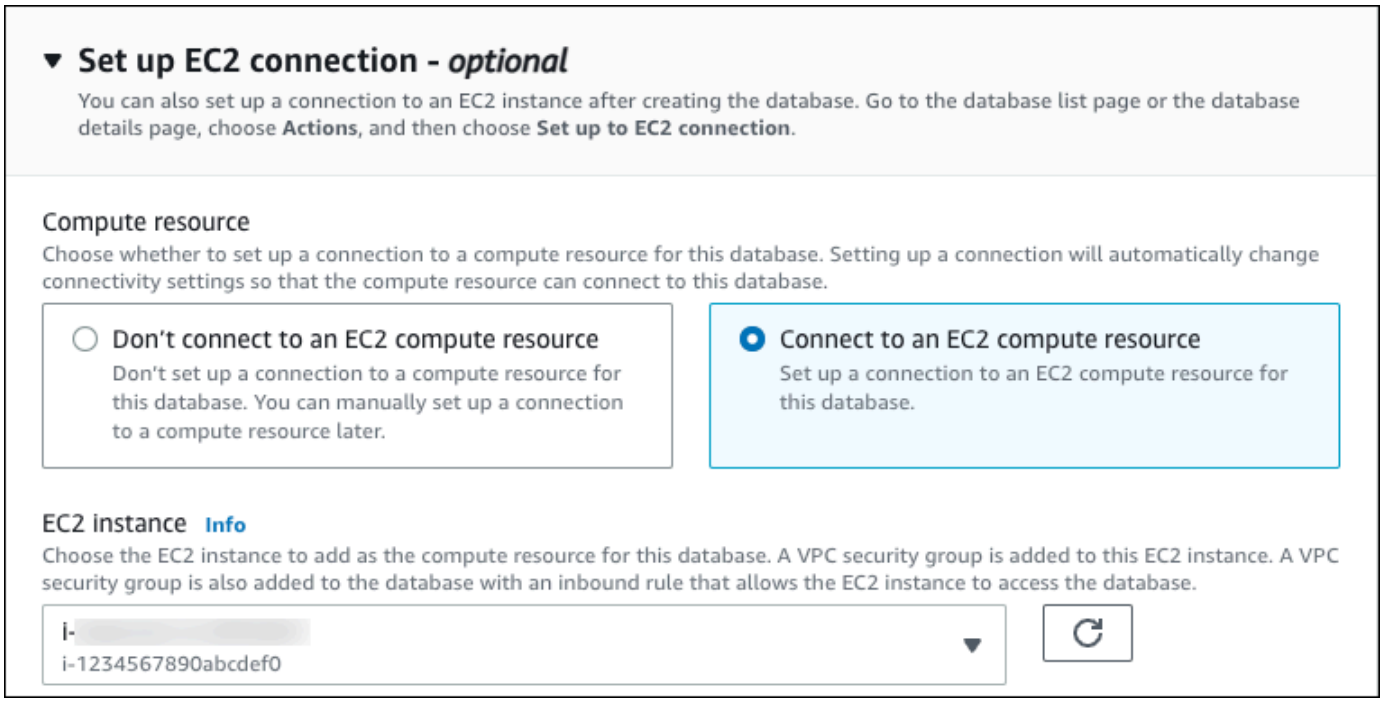
1 to 16 alphanumeric characters. First character must be a letter.

9. Para usar uma senha primária gerada automaticamente para a instância de banco de dados, marque a opção Gerar uma senha automaticamente.

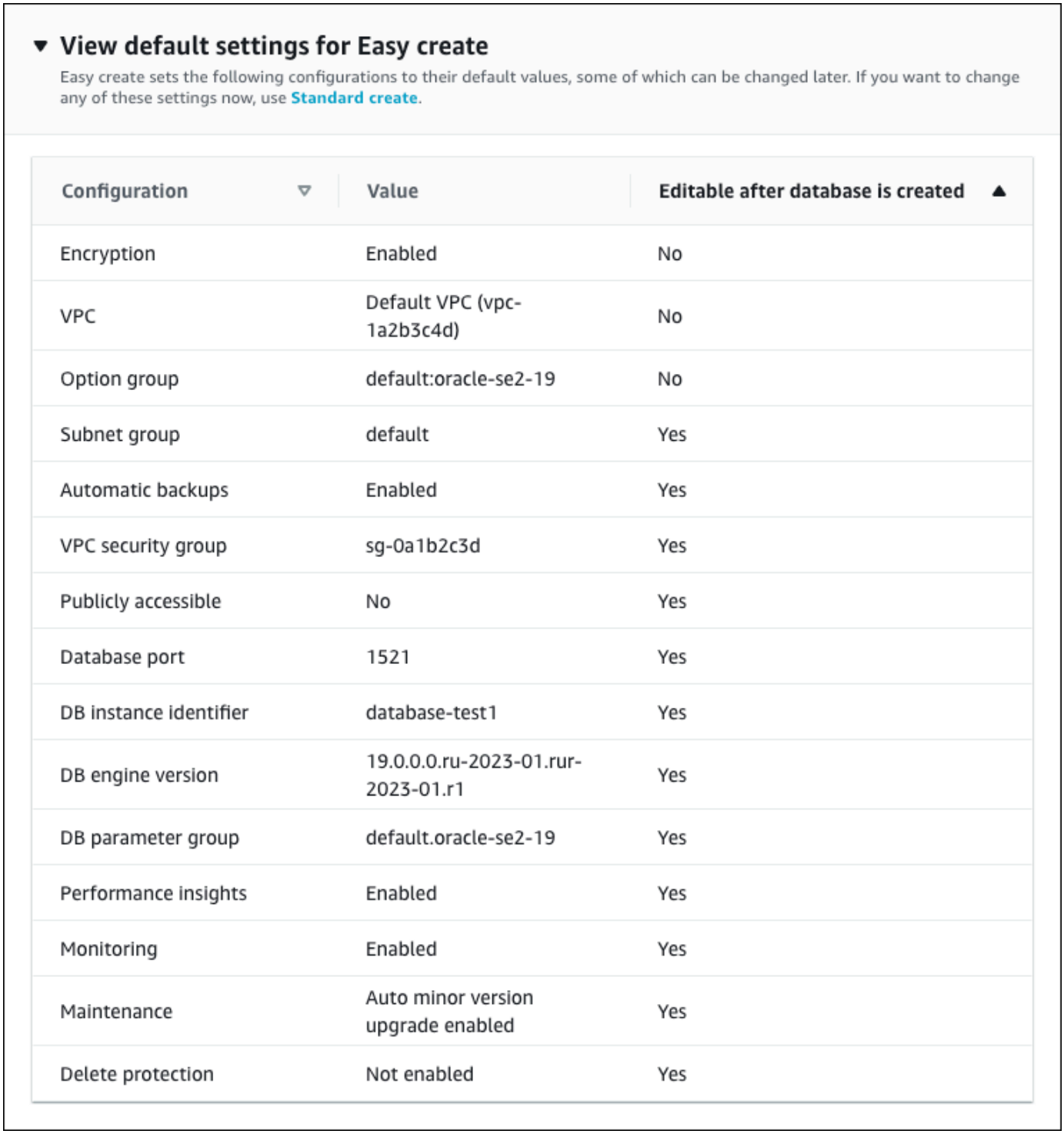
   Para inserir sua senha primária, desmarque a opção Gerar uma senha automaticamente, depois insira a mesma senha em Senha primária e em Confirmar senha.

10. Para configurar uma conexão com a instância do EC2 que você criou anteriormente, abra Configurar conexão do EC2: opcional.

    Selecione Conectar-se a um recurso computacional do EC2. Selecione a instância do EC2 que você criou anteriormente.



11. Abra Visualizar as configurações padrão da criação fácil.



View default settings for Easy create

Easy create sets the following configurations to their default values, some of which can be changed later. If you want to change any of these settings now, use **Standard create**.

Configuration	Value	Editable after database is created
Encryption	Enabled	No
VPC	Default VPC (vpc-1a2b3c4d)	No
Option group	default:oracle-se2-19	No
Subnet group	default	Yes
Automatic backups	Enabled	Yes
VPC security group	sg-0a1b2c3d	Yes
Publicly accessible	No	Yes
Database port	1521	Yes
DB instance identifier	database-test1	Yes
DB engine version	19.0.0.0.ru-2023-01.rur-2023-01.r1	Yes
DB parameter group	default.oracle-se2-19	Yes
Performance insights	Enabled	Yes
Monitoring	Enabled	Yes
Maintenance	Auto minor version upgrade enabled	Yes
Delete protection	Not enabled	Yes

Você pode examinar as configurações padrão usadas com a opção Easy Create (Criação fácil). A coluna Editável após a criação do banco de dados mostra quais opções podem ser alteradas após a criação do banco de dados.

- Se uma configuração tiver Não nessa coluna e você quiser uma configuração diferente, poderá usar a opção Criação padrão para criar a instância de banco de dados.
- Se uma configuração tiver Sim nessa coluna e você quiser uma configuração diferente, poderá usar a opção Criação padrão para criar a instância de banco de dados, ou modificar a instância de banco de dados depois de criá-la para alterar a configuração.

12. Selecione Criar banco de dados.

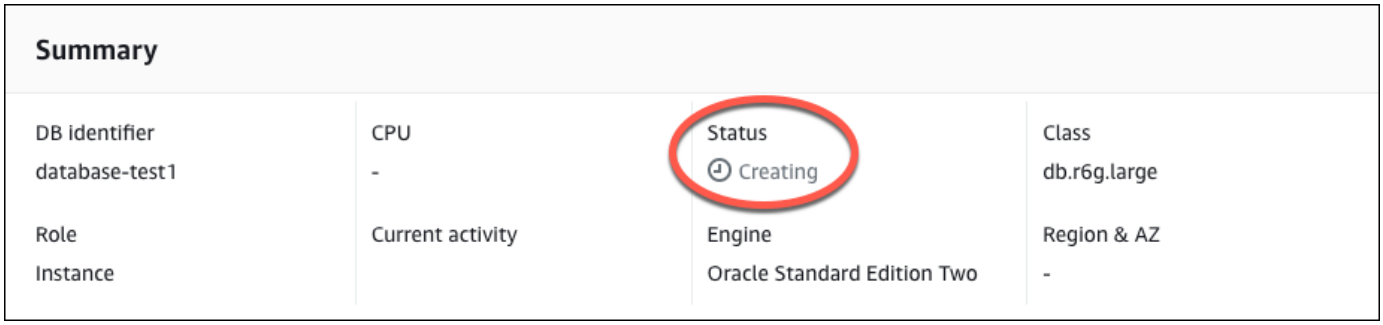
    Para exibir o nome de usuário mestre e a senha da instância de banco de dados, escolha View credential details (Ver detalhes da credencial).

    Use o nome de usuário e a senha que aparecem para se conectar à instância de banco de dados como o usuário mestre.

    > **Important**
    >
    > Você não pode visualizar a senha do usuário principal novamente. Caso você não a registre, talvez seja necessário alterá-la.
    > Se for necessário alterar a senha do usuário mestre depois que a instância de banco de dados estiver disponível, será possível modificar a instância de banco de dados para fazer isso. Para ter mais informações sobre a modificação de uma instância de banco de dados, consulte Modificar uma instância de banco de dados do Amazon RDS.

13. Na lista Bancos de dados, selecione o nome da nova instância de banco de dados Oracle para mostrar seus detalhes.

    A instância de banco de dados permanecerá no status Criando até estar pronta para ser usada.



    Quando o status muda para Available (Disponível), você pode se conectar à instância de banco de dados. Dependendo da classe da instância de banco de dados e da quantidade de armazenamento, pode levar até 20 minutos para que a nova instância esteja disponível.

Enquanto a instância de banco de dados está sendo criada, você pode passar para a próxima etapa e criar uma instância do EC2.

(Opcional) Criar VPC, instância do EC2 e instância de banco de dados do Oracle usando o AWS CloudFormation

Em vez de usar o console para criar a VPC, a instância do EC2 e a instância de banco de dados do Oracle, você pode usar o AWS CloudFormation para provisionar recursos da AWS tratando a infraestrutura como código. Para ajudar a organizar os recursos da AWS em unidades menores e mais gerenciáveis, você pode usar a funcionalidade de pilha aninhada do AWS CloudFormation. Consulte mais informações em Criar uma pilha no console do AWS CloudFormation e Trabalhar com pilhas aninhadas.

> **Important**
>
> O AWS CloudFormation é gratuito, mas os recursos que o CloudFormation são ativos. Você incorre nas taxas de uso padrão para esses recursos até que os encerre. Para ter mais informações, consulte Preços do Amazon RDS para Oracle.

Para criar recursos usando o console do AWS CloudFormation, conclua as seguintes etapas:

- Etapa 1: baixar o modelo do CloudFormation
- Etapa 2: configurar recursos usando o CloudFormation

Baixar o modelo do CloudFormation

Um modelo do CloudFormation é um arquivo de texto JSON ou YAML que contém as informações da configuração dos recursos que você deseja criar na pilha. Esse modelo também cria uma VPC e um bastion host para você junto com a instância do RDS.

Para baixar o arquivo de modelo, abra o link Oracle CloudFormation template.

Na página do Github, clique no botão Baixar arquivo bruto para salvar o arquivo YAML do modelo.

Configurar recursos usando o CloudFormation

> **Note**
>
> Antes de iniciar esse processo, verifique se você tem um par de chaves para uma instância do EC2 na Conta da AWS. Para obter mais informações, consulte Pares de chaves do Amazon EC2 e instâncias do Linux.

Ao usar o modelo do AWS CloudFormation, você deve selecionar os parâmetros certos para garantir que os recursos sejam criados corretamente. Siga as etapas abaixo:

1. Faça login no AWS Management Console e abra o console AWS CloudFormation em https://console.aws.amazon.com/cloudformation.
2. Selecione Create Stack (Criar pilha).
3. Na seção Especificar modelo, selecione Fazer upload de um arquivo de modelo do computador e escolha Próximo.
4. Na página Especificar detalhes da pilha, defina os seguintes parâmetros:
   a. Defina o Nome da pilha como OracleTestStack.
   b. Em Parâmetros, defina Zonas de disponibilidade selecionando três zonas de disponibilidade.
   c. Em Configuração do bastion host do Linux, em Nome da chave, selecione um par de chaves para fazer login na instância do EC2.
   d. Em Configurações do bastion host do Linux, defina o Intervalo de IP permitido para o endereço IP. Para conectar instâncias do EC2 à VPC usando o Secure Shell (SSH), determine o endereço IP público usando o serviço em https://checkip.amazonaws.com. Um exemplo de endereço IP é 192.0.2.1/32.

> **Warning**
>
> Se usar `0.0.0.0/0` para acesso do SSH, você possibilitará que todos os endereços IP acessem suas instâncias públicas do EC2 usando SSH. Essa abordagem é aceitável por um período curto em um ambiente de teste, mas não é seguro em ambientes de produção. Em produção, autorize somente um endereço IP específico ou um intervalo de endereços para acessar suas instâncias do EC2 usando SSH.

e. Em Configuração geral do banco de dados, defina a Classe da instância do banco de dados como db.t3.micro.

f. Defina o Nome do banco de dados como **`database-test1`**.

g. Em Nome de usuário principal do banco de dados, insira um nome para o usuário principal.

h. Defina a Gerenciar senha de usuário principal do banco de dados com o Secrets Manager como `false` para esse tutorial.

i. Em Senha do banco de dados, defina uma senha de sua escolha. Lembre-se dessa senha para as etapas seguintes do tutorial.

j. Em Configuração de armazenamento de banco de dados, defina o Tipo de armazenamento do banco de dados como gp2.

k. Em Configuração de monitoramento de banco de dados, defina Habilitar o Performance Insights do RDS como falso.

l. Deixe todas as outras configurações com os valores padrão. Clique em Próximo para continuar.

5. Na página Configurar opções de pilha, mantenha todas as opções padrão. Clique em Próximo para continuar.
6. Na página Revisar pilha, selecione Enviar depois de verificar as opções do banco de dados e do bastion host do Linux.

Depois que o processo de criação da pilha for concluído, visualize as pilhas com os nomes BastionStack e RDSNS para anotar as informações necessárias para se conectar ao banco de dados. Consulte mais informações em Viewing AWS CloudFormation stack data and resources on the AWS Management Console.

Etapa 3: Conectar seu cliente SQL a uma instância de banco de dados Oracle.

Você pode usar qualquer aplicação cliente padrão SQL para se conectar à instância de banco de dados. Neste exemplo, você se conecta a uma instância de banco de dados Oracle usando o cliente da linha de comando Oracle.

Como conectar-se a uma instância de banco de dados Oracle

1. Encontre o endpoint (nome de DNS) e o número da porta para sua instância de banco de dados.

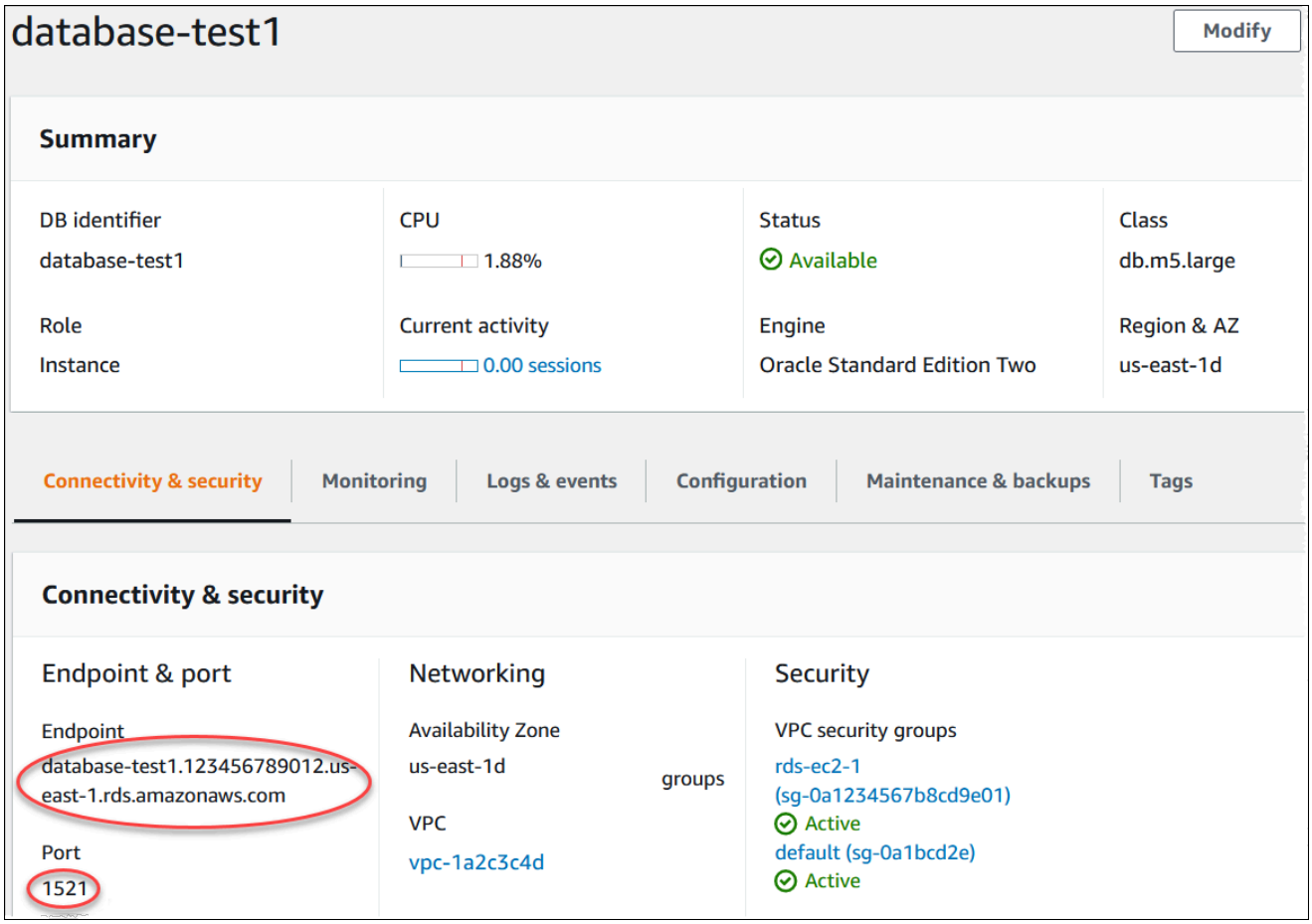
a. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.

b. No canto superior direito do console do Amazon RDS, escolha a Região da AWS da instância de banco de dados.

c. No painel de navegação, escolha Bancos de dados.

d. Escolha o nome da instância de banco de dados Oracle para exibir os detalhes.

e. Na guia Connectivity & security (Conectividade e segurança), copie o endpoint. Além disso, anote o número da porta. Você precisará do endpoint e do número da porta para conectar-se à instância de banco de dados.



2. Conecte-se à instância do EC2 que você criou anteriormente, seguindo as etapas em Conecte-se à sua instância do Linux no Guia do usuário do Amazon EC2.

   Recomendamos que você se conecte à sua instância do EC2 utilizando SSH. Se o utilitário cliente SSH estiver instalado no Windows, Linux ou Mac, você poderá se conectar à instância utilizando o seguinte formato de comando:

```
ssh -i location_of_pem_file ec2-user@ec2-instance-public-dns-name
```

Por exemplo, suponha que `ec2-database-connect-key-pair.pem` esteja armazenado em `/dir1` no Linux e que o DNS IPv4 público para sua instância do EC2 seja `ec2-12-345-678-90.compute-1.amazonaws.com`. Seu comando SSH teria a seguinte aparência:

```
ssh -i /dir1/ec2-database-connect-key-pair.pem ec2-
user@ec2-12-345-678-90.compute-1.amazonaws.com
```

3. Obtenha as últimas correções de bugs e atualizações de segurança atualizando o software na instância do EC2. Para fazer isso, use o comando a seguir.

> **Note**
>
> A opção `-y` instala as atualizações sem solicitar confirmação. Para examinar atualizações antes da instalação, omita essa opção.

```
sudo dnf update -y
```

4. Em um navegador da Web, acesse https://www.oracle.com/database/technologies/instant-client/linux-x86-64-downloads.html.
5. Para obter a versão mais recente do banco de dados que aparece na página da Web, copie os links .rpm (não os links .zip) para o Instant Client Basic Package e o SQL*Plus Package. Por exemplo, os links a seguir são para o Oracle Database versão 21.9:

   - https://download.oracle.com/otn_software/linux/instantclient/219000/oracle-instantclient-basic-21.9.0.0.0-1.el8.x86_64.rpm
   - https://download.oracle.com/otn_software/linux/instantclient/219000/oracle-instantclient-sqlplus-21.9.0.0.0-1.el8.x86_64.rpm

6. Em sua sessão SSH, execute o comando `wget` para baixar os arquivos .rpm dos links que você obteve na etapa anterior. O exemplo a seguir baixa os arquivos .rpm para a versão 21.9 do Oracle Database:

```
wget https://download.oracle.com/otn_software/linux/instantclient/219000/oracle-
instantclient-basic-21.9.0.0.0-1.el8.x86_64.rpm
```

```
wget https://download.oracle.com/otn_software/linux/instantclient/219000/oracle-
instantclient-sqlplus-21.9.0.0.0-1.el8.x86_64.rpm
```

7. Instale os pacotes executando o comando `dnf` da seguinte forma:

```
sudo dnf install oracle-instantclient-*.rpm
```

8. Inicie o SQL*Plus e conecte-se à instância de banco de dados do Oracle. Por exemplo, insira o comando a seguir.

   Substitua o endpoint da instância de banco de dados (nome DNS) por *`oracle-db-instance-endpoint`* e o nome do usuário principal usado por *`admin`*. Quando você usa a Criação fácil para Oracle, o nome do banco de dados é `DATABASE`. Forneça a senha mestra usada quando for solicitada uma senha.

```
sqlplus admin@oracle-db-instance-endpoint:1521/DATABASE
```

   Depois de inserir a senha do usuário, você deverá ver uma saída semelhante à seguinte.

```
SQL*Plus: Release 21.0.0.0.0 - Production on Wed Mar 1 16:41:28 2023
Version 21.9.0.0.0

Copyright (c) 1982, 2022, Oracle.  All rights reserved.

Enter password:
Last Successful login time: Wed Mar 01 2023 16:30:52 +00:00

Connected to:
Oracle Database 19c Standard Edition 2 Release 19.0.0.0.0 - Production
Version 19.18.0.0.0

SQL>
```

   Para ter mais informações sobre como se conectar à instância de banco de dados do RDS para Oracle, consulte Conectar-se a uma instância de banco de dados Oracle. Se você não conseguir se conectar à sua instância de banco de dados, consulte Não é possível conectar-se à instância de banco de dados do Amazon RDS.

   Por questões de segurança, é uma prática recomendada usar conexões criptografadas. Use uma conexão Oracle não criptografada apenas quando o cliente e o servidor estiverem

na mesma VPC e a rede for confiável. Para obter informações sobre como usar conexões criptografadas, consulte Proteção das conexões de instância de banco de dados Oracle.

9. Executar comandos SQL.

   Por exemplo, o seguinte comando SQL mostra a data atual:

```
SELECT SYSDATE FROM DUAL;
```

Etapa 4: Excluir a instância do EC2 e a instância de banco de dados

Depois de se conectar e explorar a instância do EC2 e a instância de banco de dados criadas, exclua-as para não receber mais cobranças por elas.

Se você usou o AWS CloudFormation para criar recursos, ignore essa etapa e passe para a próxima.

Como excluir a instância do EC2

1. Faça login no AWS Management Console e abra o console do Amazon EC2 em https://console.aws.amazon.com/ec2/.
2. No painel de navegação, escolha Instâncias.
3. Selecione a instância do EC2 e escolha Estado da instância, Encerrar instância.
4. Quando a confirmação for solicitada, escolha Encerrar.

Consulte mais informações sobre como excluir uma instância do EC2 em Como encerrar uma instância no Guia do usuário do Amazon EC2.

Como excluir uma instância de banco de dados sem snapshot de banco de dados final

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Databases (Bancos de dados).
3. Escolha a instância de banco de dados que você deseja excluir.
4. Em Actions, selecione Delete.
5. Desmarque Criar snapshot final? e Reter backups automatizados.
6. Conclua a confirmação e escolha Excluir.

(Opcional) Excluir a instância do EC2 e a instância de banco de dados criada com o CloudFormation

Se você usou o AWS CloudFormation para criar recursos, exclua a pilha do CloudFormation depois de conectar e explorar a amostra de instância do EC2 e de instância de banco de dados para não receber mais cobranças por elas.

Para excluir os recursos do CloudFormation

1. Abra o console do AWS CloudFormation.
2. Na página Pilhas no console do CloudFormation, selecione a pilha raiz (a pilha sem o nome VPCStack, BastionStack ou RDSNS).
3. Escolha Excluir.
4. Selecione Excluir pilha quando a confirmação for solicitada.

Consulte mais informações de como excluir uma pilha no CloudFormation em Deleting a stack on the AWS CloudFormation console no Guia do usuário do AWS CloudFormation.

(Opcional) Conectar a instância de banco de dados a uma função do Lambda

Você também pode conectar a instância de banco de dados do RDS para Oracle a um recurso de computação sem servidor do Lambda. As funções do Lambda permitem que você execute código sem provisionar nem gerenciar a infraestrutura. Uma função do Lambda também permite que você responda automaticamente a solicitações de execução de código em qualquer escala, de dezenas de eventos por dia a centenas por segundo. Para obter mais informações, consulte Conectar automaticamente uma função do Lambda e uma instância de banco de dados.

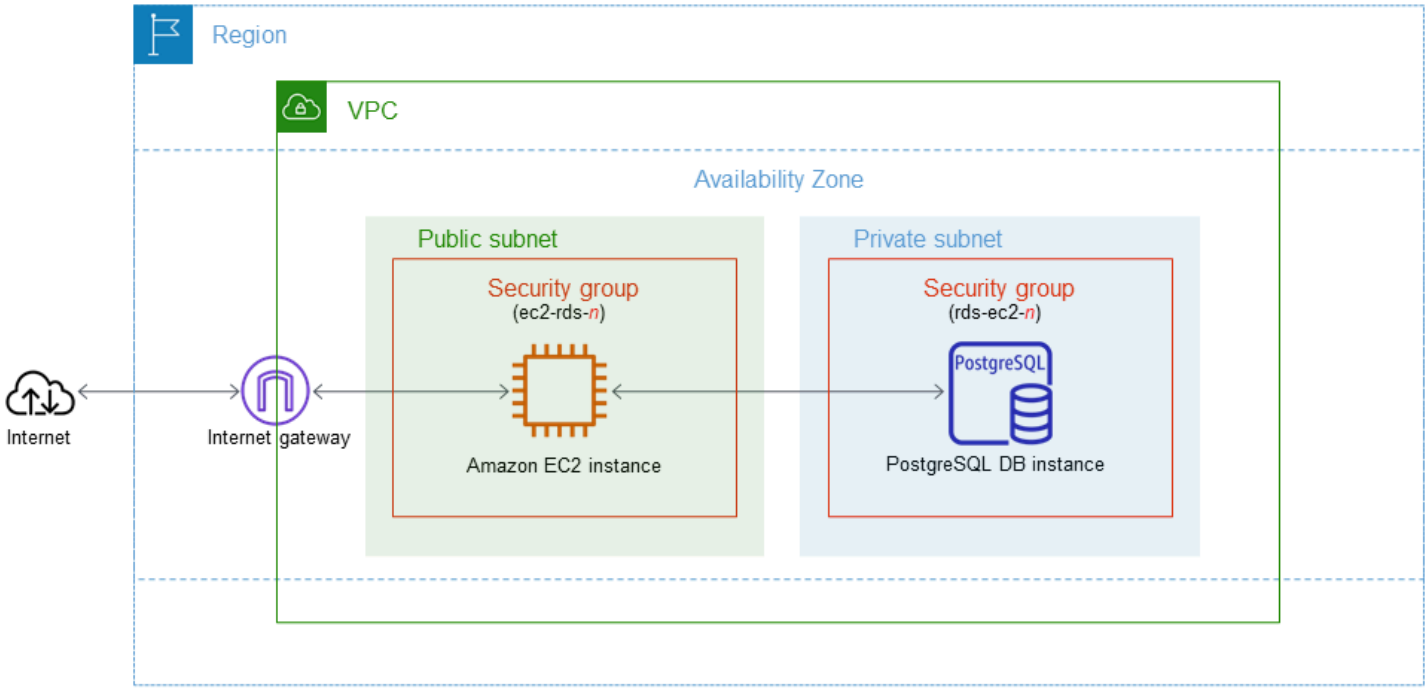
Criar uma instância de banco de dados PostgreSQL e conectar-se a ela

Este tutorial cria uma instância do EC2 e uma instância de banco de dados do RDS para PostgreSQL. O tutorial mostra como acessar a instância de banco de dados pela instância do EC2 usando um cliente PostgreSQL padrão. Como prática recomendada, este tutorial cria uma instância de banco de dados privada em uma nuvem privada virtual (VPC). Na maioria dos casos, outros recursos na mesma VPC, como instâncias do EC2, podem acessar a instância de banco de dados, mas recursos fora da VPC não podem acessá-la.

Depois de concluir o tutorial, haverá uma sub-rede pública e privada em cada zona de disponibilidade na sua VPC. Em uma zona de disponibilidade, a instância do EC2 está na sub-rede pública e a instância de banco de dados está na sub-rede privada.

> **Important**
>
> Não há cobrança para criar uma conta da AWS. No entanto, ao concluir este tutorial, é possível gerar custos para os recursos da AWS que você usa. Se esses recursos não forem mais necessários após a conclusão do tutorial, você poderá excluí-los.

O diagrama a seguir mostrará a configuração quando o tutorial estiver completo.



Esse tutorial permite criar recursos usando um dos seguintes métodos:

1. Use o AWS Management Console: Criar uma instância do EC2 e Criar uma instância de banco de dados do PostgreSQL
2. Use o AWS CloudFormation para criar a instância de banco de dados e a instância do EC2: (Opcional) Criar VPC, instância do EC2 e instância do PostgreSQL usando o AWS CloudFormation

O primeiro método usa Criação fácil para criar uma instância de banco de dados privada do PostgreSQL com o AWS Management Console. Aqui, você especifica somente o tipo de mecanismo de banco de dados, o tamanho da instância de banco de dados e o identificador da instância de banco de dados. A opção Easy create (Criação fácil) usa a configuração padrão para as outras opções de configuração.

Ao usar a opção Criação padrão, é possível especificar mais opções de configuração ao criar uma instância de banco de dados. Essas opções incluem configurações de disponibilidade, segurança, backups e manutenção. Para criar uma instância de banco de dados pública, você deve usar a Criação padrão. Para mais informações, consulte Criar uma instância de banco de dados do Amazon RDS.

Tópicos

- Pré-requisitos
- Criar uma instância do EC2
- Criar uma instância de banco de dados do PostgreSQL
- (Opcional) Criar VPC, instância do EC2 e instância do PostgreSQL usando o AWS CloudFormation
- Conectar-se à instância de banco de dados PostgreSQL
- Excluir a instância do EC2 e a instância de banco de dados
- (Opcional) Excluir a instância do EC2 e a instância de banco de dados criada com o CloudFormation
- (Opcional) Conectar a instância de banco de dados a uma função do Lambda

Pré-requisitos

Antes de começar, conclua as etapas nas seguintes seções:

- Inscrever-se para uma Conta da AWS

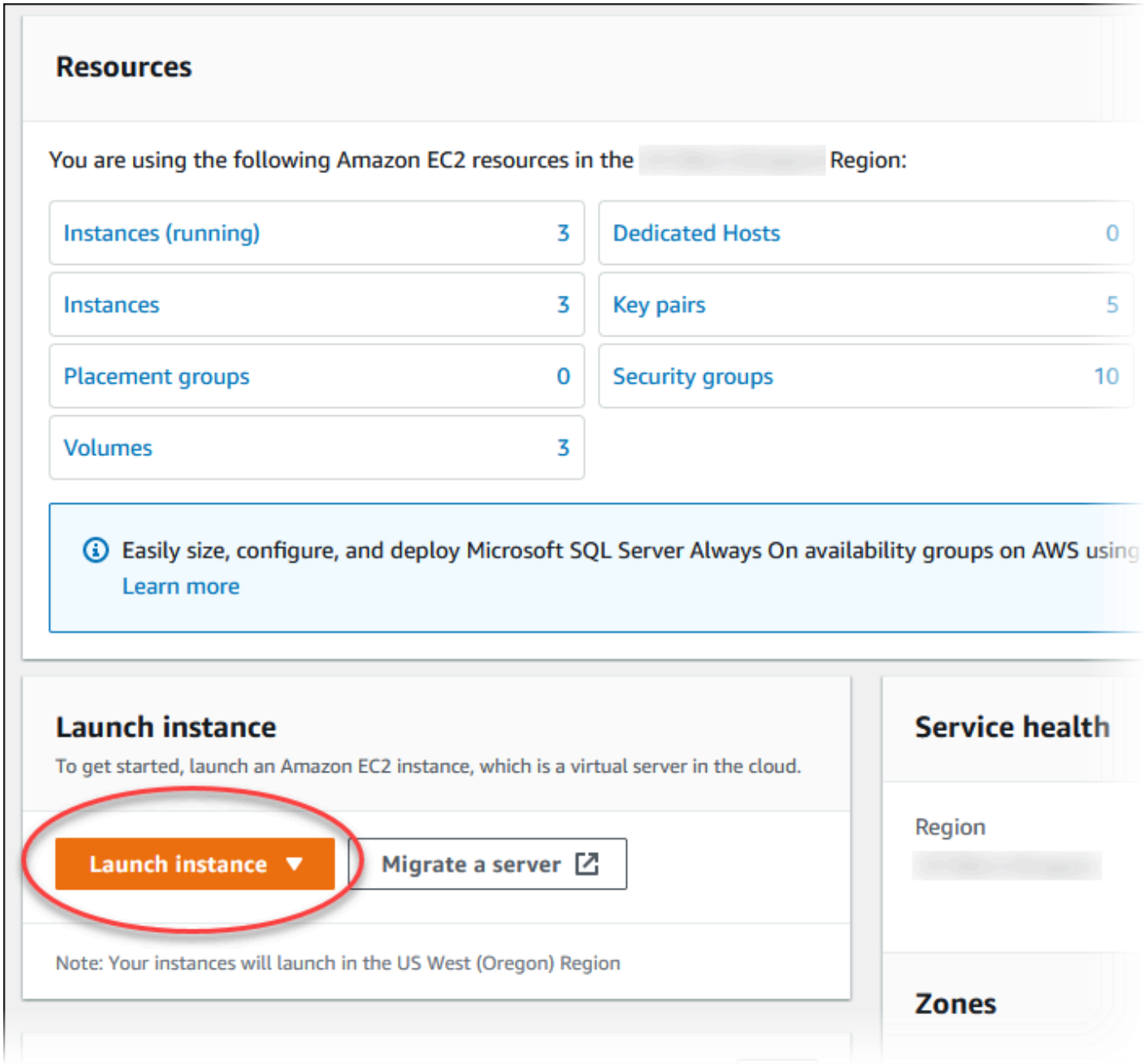
- Criar um usuário com acesso administrativo

Criar uma instância do EC2

Crie uma instância do Amazon EC2 que você usará para se conectar ao banco de dados.

Para criar uma instância do EC2

1. Faça login no AWS Management Console e abra o console do Amazon EC2 em https://console.aws.amazon.com/ec2/.
2. No canto superior direito do AWS Management Console, selecione a Região da AWS em que você deseja criar a instância do EC2.
3. Escolha Painel do EC2 e Executar instância, conforme mostrado na imagem a seguir.

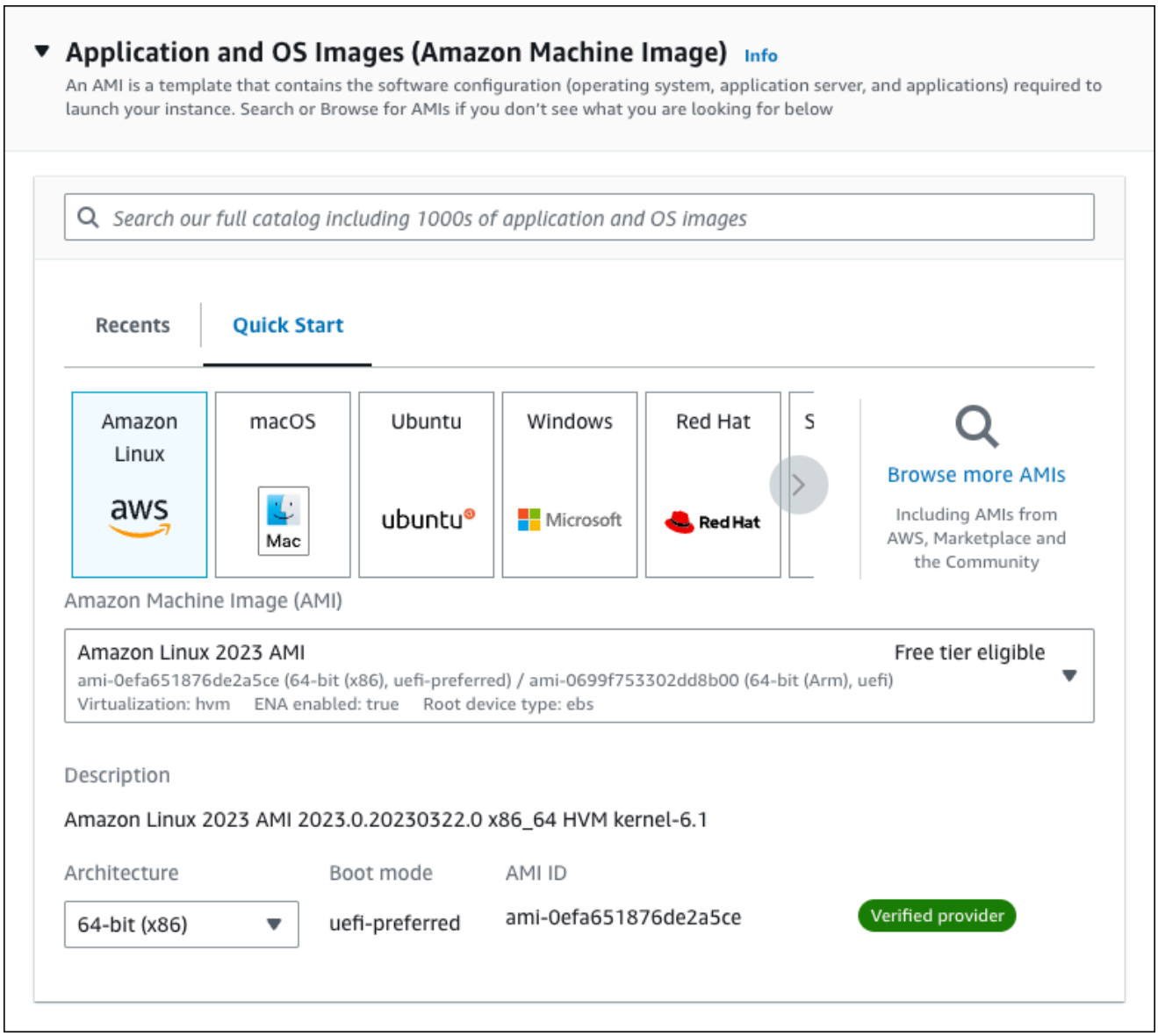


A página Iniciar uma instância é aberta.

4. Escolha as configurações a seguir na página Iniciar uma instância.

   a. Em Name and tags (Nome e etiquetas), em Name (Nome), insira **`ec2-database-connect`**.

   b. Em Imagens de aplicações e sistemas operacionais (imagem de máquina da Amazon), selecione Amazon Linux e, depois, AMI do Amazon Linux 2023. Mantenha as seleções padrão nas outras opções.



c. Em Instance type (Tipo de instância), escolha t2.micro.

d. Em Key pair (login) (Par de chaves (login)), escolha um Key pair name (Nome do par de chaves) para usar um par de chaves existente. Para criar um par de chaves para a instância do Amazon EC2, escolha Create new key pair (Criar um novo key pair), depois use a janela Create key pair (Criar par de chaves) para criá-lo.

   Consulte mais informações sobre como criar um par de chaves em Criar um par de chaves no Guia do usuário do Amazon EC2.

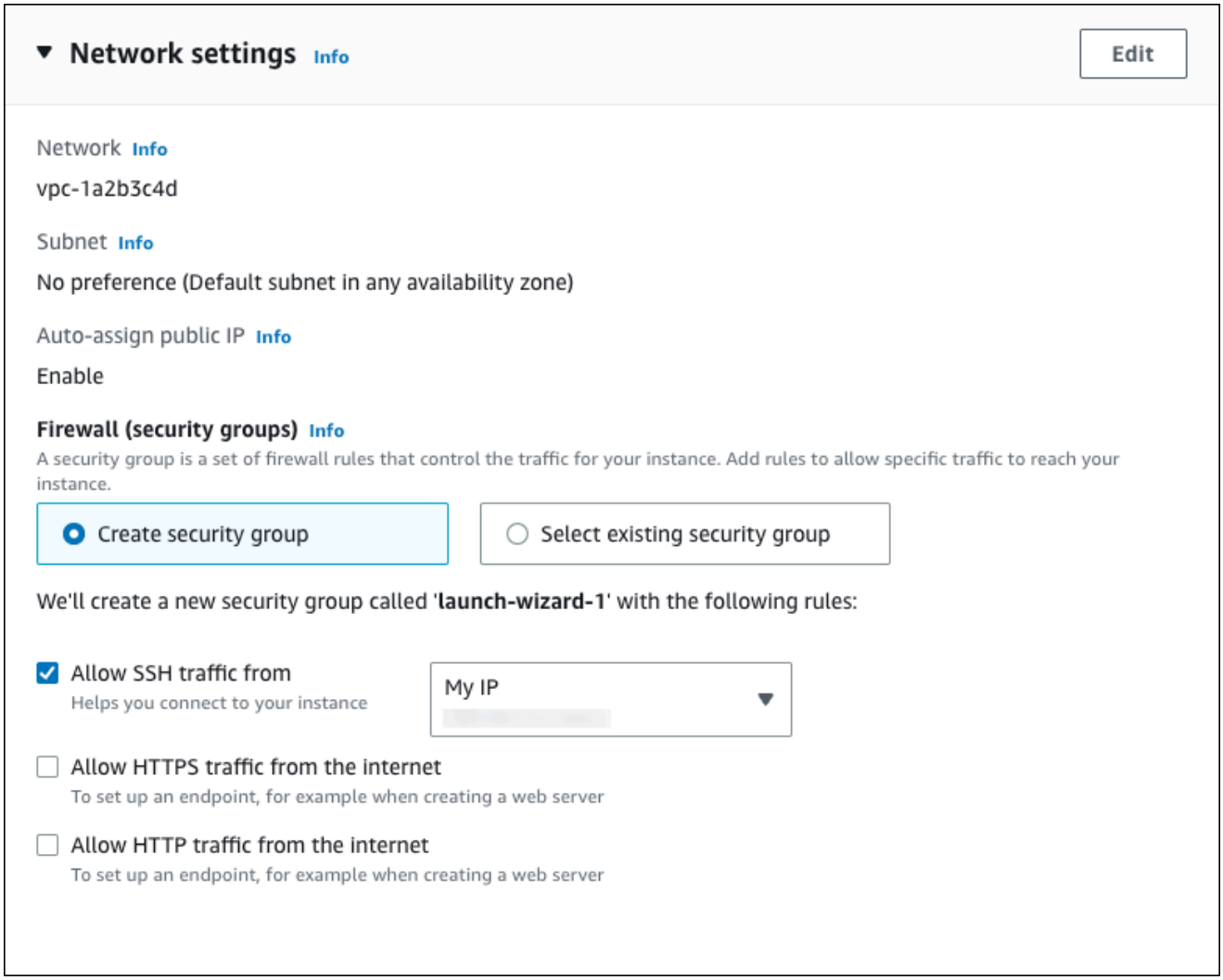
e. Para Permitir tráfego SSH, em Configurações de rede, selecione a origem das conexões SSH com a instância do EC2.

Você pode escolher My IP (Meu IP) se o endereço IP exibido estiver correto para conexões SSH. Caso contrário, determine o endereço IP a ser usado para se conectar a instâncias do EC2 em sua VPC usando Secure Shell (SSH). Para determinar seu endereço IP público, em uma janela ou guia diferente do navegador, é possível usar o serviço em https://checkip.amazonaws.com. Um exemplo de endereço IP é 192.0.2.1/32.

Em muitos casos, você pode se conectar por meio de um provedor de serviços de Internet (ISP) ou atrás de um firewall sem um endereço IP estático. Em caso afirmativo, determine o intervalo de endereços IP utilizado por computadores cliente.

> **Warning**
>
> Se usar `0.0.0.0/0` para acesso do SSH, você possibilitará que todos os endereços IP acessem suas instâncias públicas do EC2 usando SSH. Essa abordagem é aceitável por um período curto em um ambiente de teste, mas não é seguro em ambientes de produção. Em produção, autorize somente um endereço IP específico ou um intervalo de endereços para acessar suas instâncias do EC2 usando SSH.

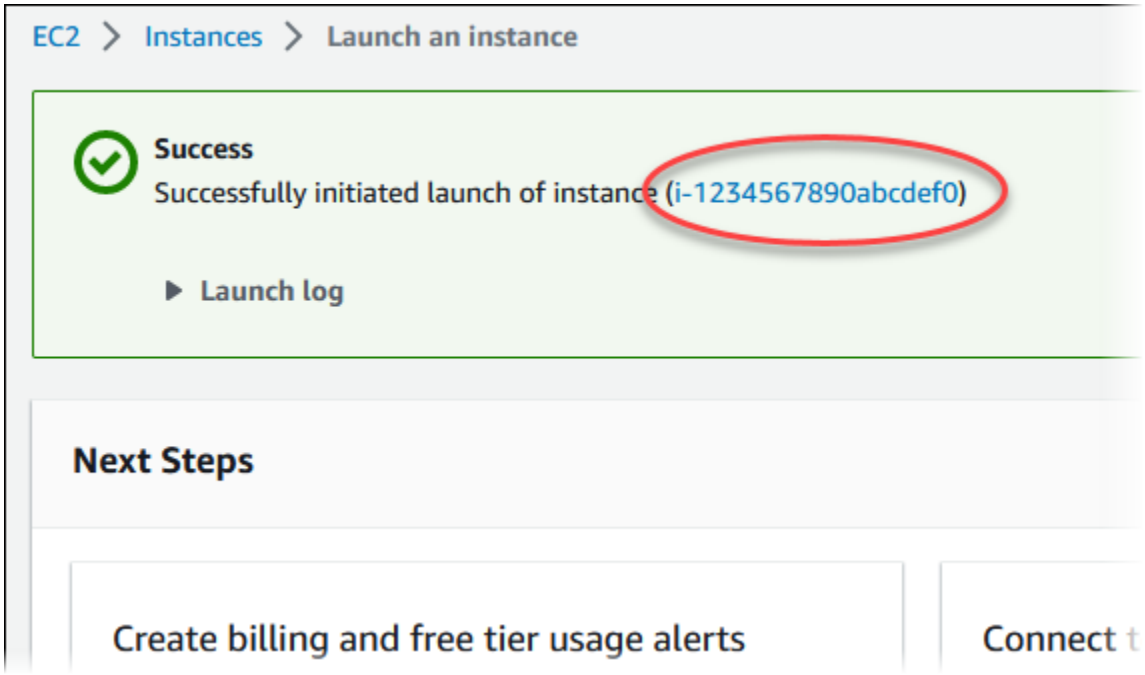
A imagem a seguir mostra um exemplo da seção Configurações de rede.



f. Mantenha os valores padrão para as seções restantes.

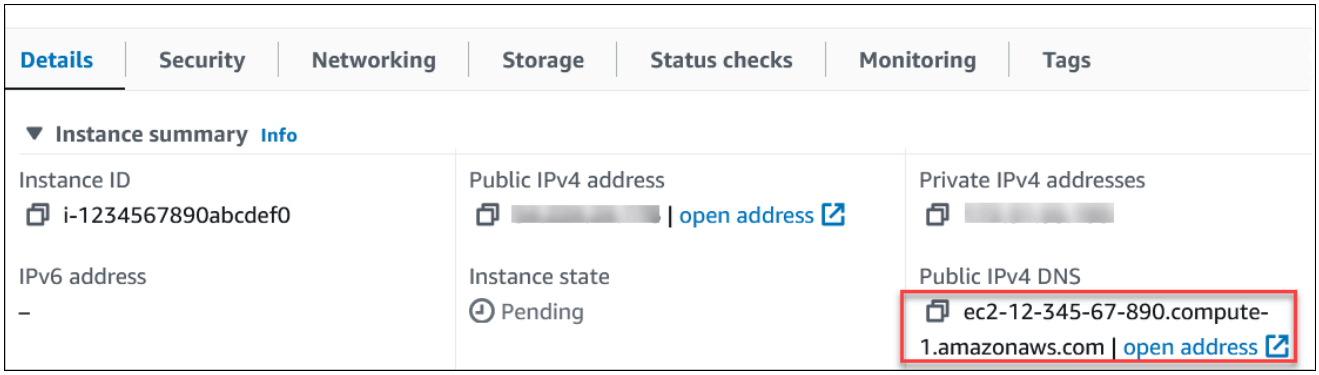
g. Revise um resumo da configuração da instância do EC2 no painel Resumo e, quando estiver com tudo pronto, escolha Executar instância.

5. Na página Status de inicialização, anote o identificador de sua nova instância do EC2, por exemplo: `i-1234567890abcdef0`.

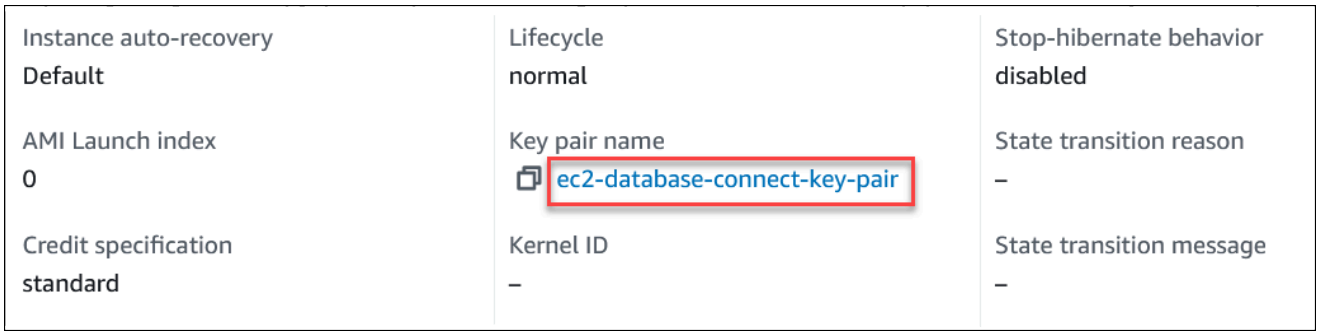


6. Selecione o identificador de instância do EC2 para abrir a lista de instâncias do EC2 e, depois, selecione sua instância do EC2.
7. Na guia Detalhes, observe os seguintes valores, necessários ao se conectar utilizando SSH:

   a. No Resumo da instância, observe o valor do DNS IPv4 público.



   b. Em Detalhes da instância, observe o valor do nome do par de chaves.



8. Espere até o Estado da instância para a instância do EC2 ficar como Executando antes de continuar.

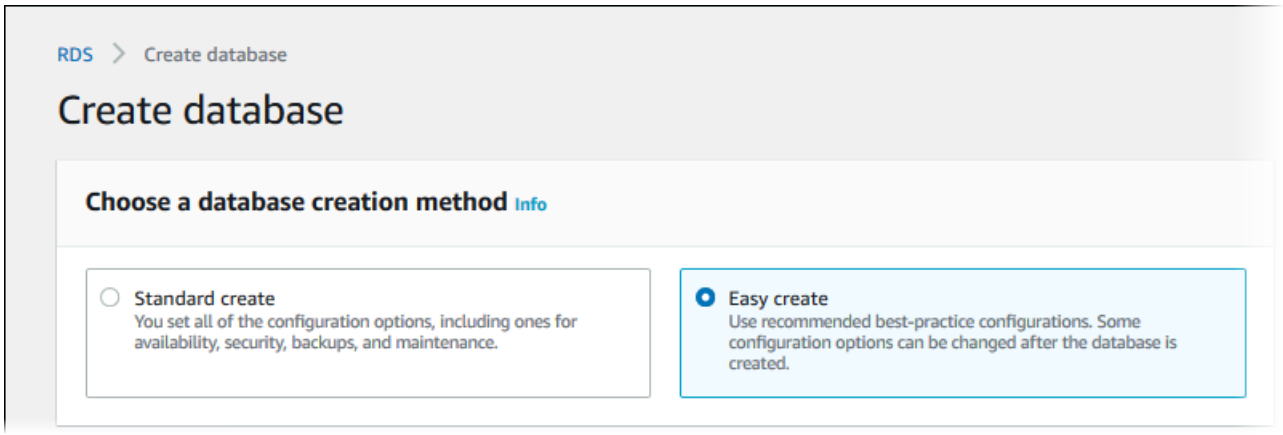
Criar uma instância de banco de dados do PostgreSQL

O bloco de construção básico do Amazon RDS é a instância do banco de dados. Este ambiente é onde você executa seus bancos de dados PostgreSQL.

Neste exemplo, use a Criação fácil para criar uma instância de banco de dados executando o mecanismo de banco de dados PostgreSQL com uma classe de instância de banco de dados db.t3.micro.

Para criar uma instância de banco de dados PostgreSQL com a criação fácil

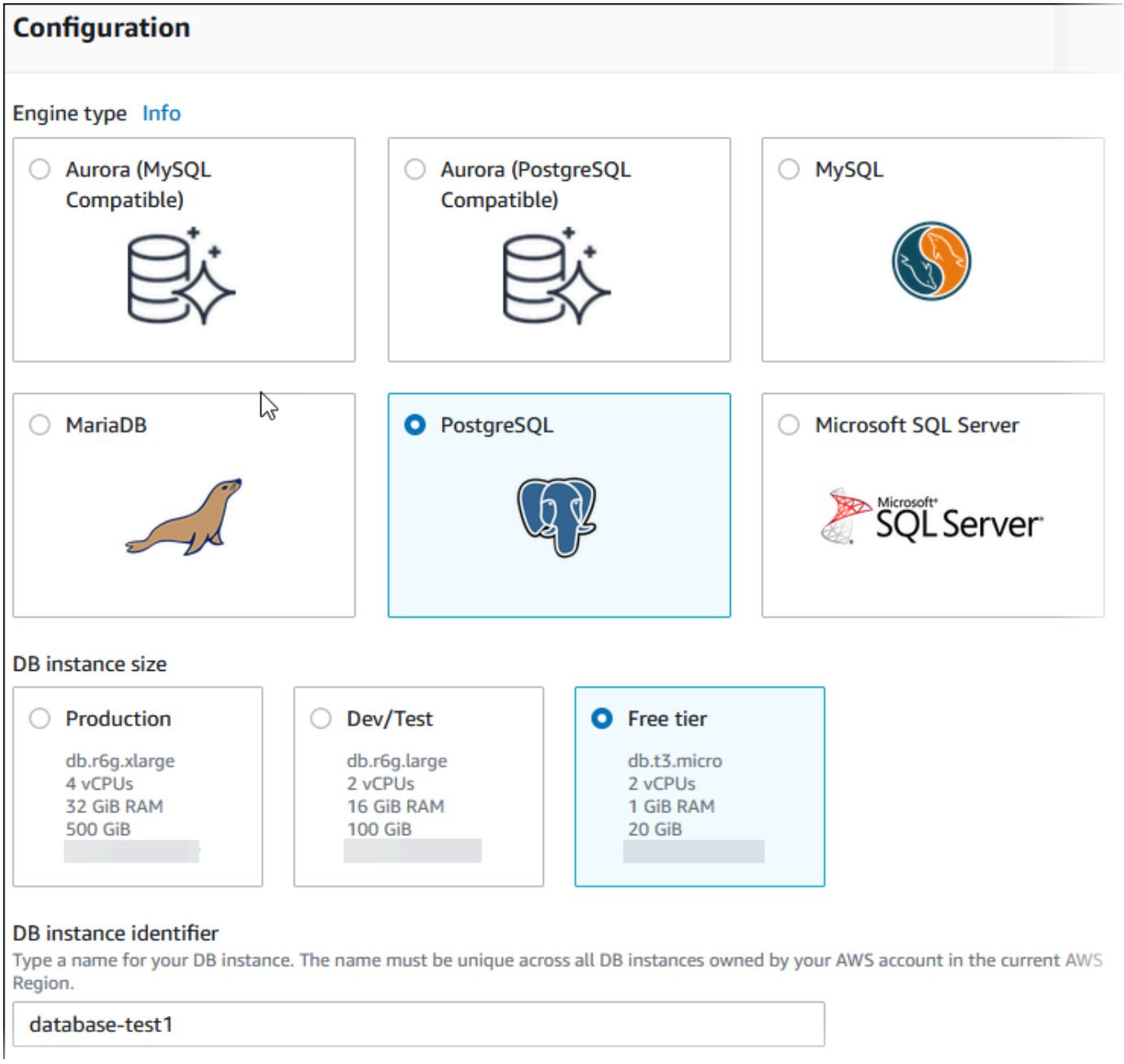
1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No canto superior direito do console do Amazon RDS, escolha a região da AWS na qual você deseja criar a instância de banco de dados.
3. No painel de navegação, escolha Databases (Bancos de dados).
4. Selecione a opção Create database (Criar banco de dados) e verifique se a opção Easy Create (Criação fácil) está habilitada.



5. Em Configuration (Configuração), escolha PostgreSQL.
6. Em DB instance size (Tamanho da instância de banco de dados), escolha Free tier (Nível gratuito). Nível gratuito aparece para contas do plano gratuito. Sandbox aparece para contas do plano pago.
7. Em Identificador de instância de banco de dados, insira **`database-test1`**.

8. Em Nome do usuário principal, insira um nome para o usuário principal ou deixe o nome padrão (**`postgres`**).

   A página Create database (Criar banco de dados) deve ser semelhante à imagem a seguir. Para contas do plano gratuito, aparece Nível gratuito. Para contas do plano pago, aparece Sandbox.

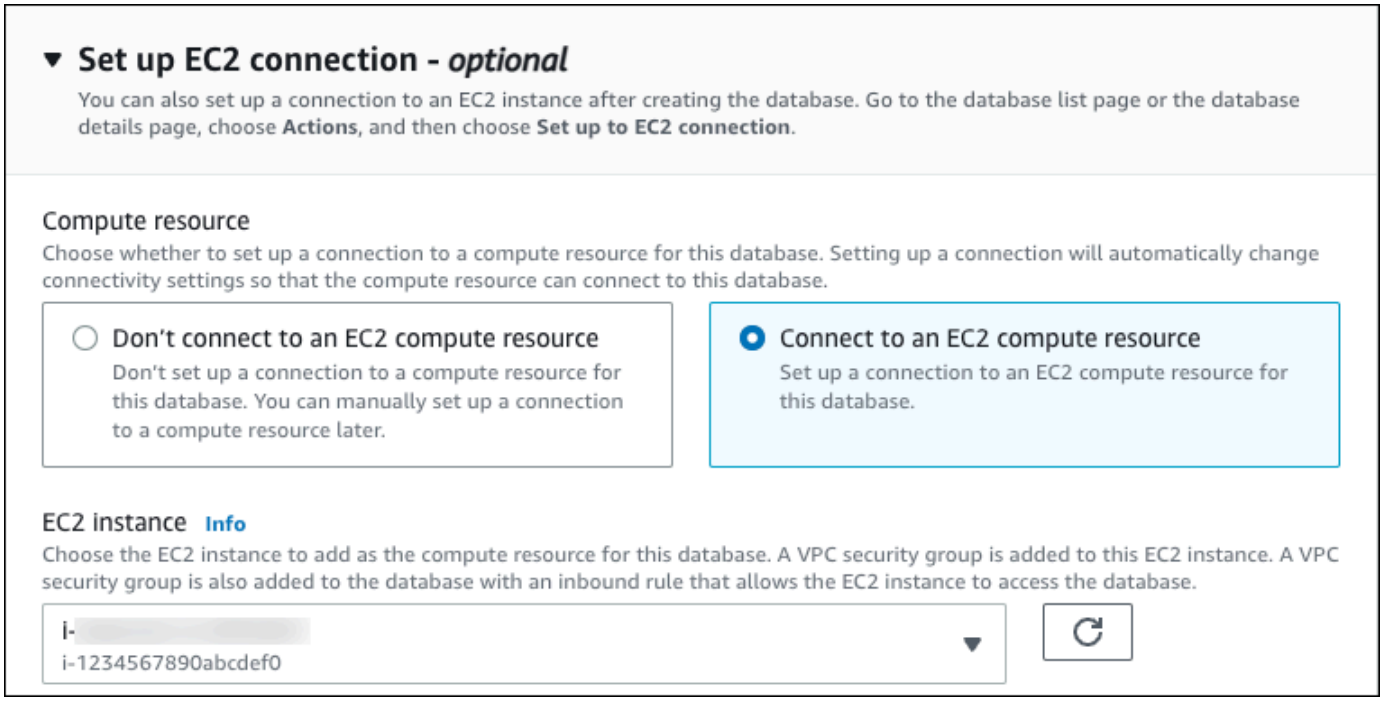


9. Para usar uma senha primária gerada automaticamente para a instância de banco de dados, marque a opção Gerar uma senha automaticamente.

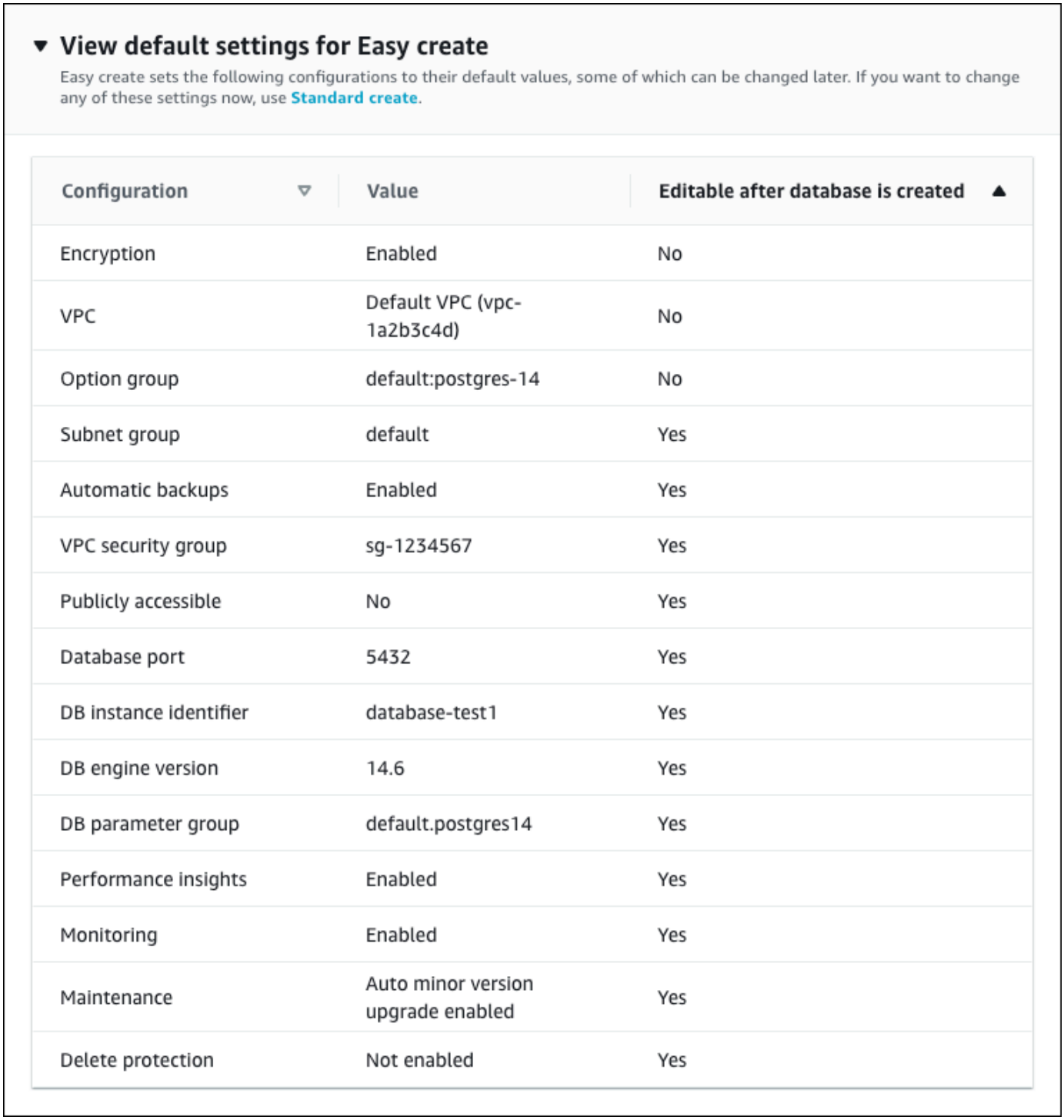
Para inserir sua senha primária, desmarque a opção Gerar uma senha automaticamente, depois insira a mesma senha em Senha primária e em Confirmar senha.

10. Para configurar uma conexão com a instância do EC2 que você criou anteriormente, abra Configurar conexão do EC2: opcional.

    Selecione Conectar-se a um recurso computacional do EC2. Selecione a instância do EC2 que você criou anteriormente.



11. Abra Visualizar as configurações padrão da criação fácil.



View default settings for Easy create

Easy create sets the following configurations to their default values, some of which can be changed later. If you want to change any of these settings now, use **Standard create**.

Configuration	Value	Editable after database is created
Encryption	Enabled	No
VPC	Default VPC (vpc-1a2b3c4d)	No
Option group	default:postgres-14	No
Subnet group	default	Yes
Automatic backups	Enabled	Yes
VPC security group	sg-1234567	Yes
Publicly accessible	No	Yes
Database port	5432	Yes
DB instance identifier	database-test1	Yes
DB engine version	14.6	Yes
DB parameter group	default.postgres14	Yes
Performance insights	Enabled	Yes
Monitoring	Enabled	Yes
Maintenance	Auto minor version upgrade enabled	Yes
Delete protection	Not enabled	Yes

Você pode examinar as configurações padrão usadas com a opção Easy Create (Criação fácil). A coluna Editável após a criação do banco de dados mostra quais opções podem ser alteradas após a criação do banco de dados.

- Se uma configuração tiver Não nessa coluna e você quiser uma configuração diferente, poderá usar a opção Criação padrão para criar a instância de banco de dados.
- Se uma configuração tiver Sim nessa coluna e você quiser uma configuração diferente, poderá usar a opção Criação padrão para criar a instância de banco de dados, ou modificar a instância de banco de dados depois de criá-la para alterar a configuração.

12. Selecione Criar banco de dados.

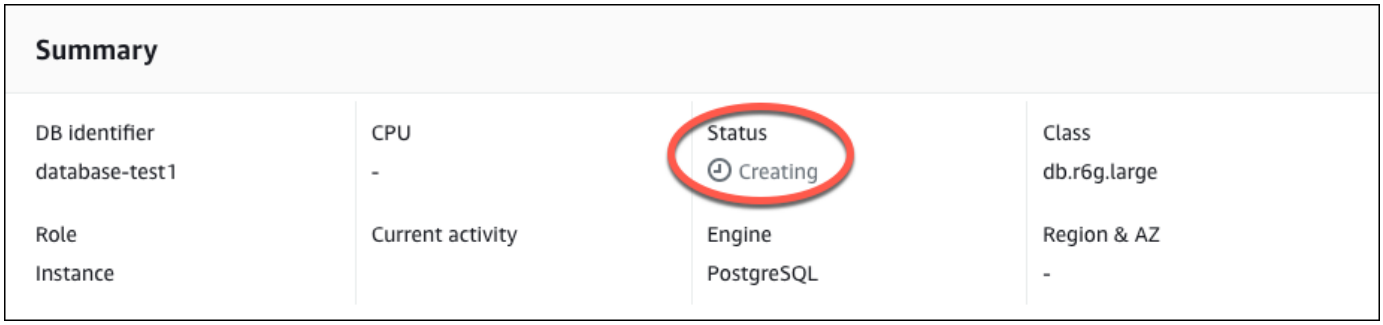
    Para exibir o nome de usuário mestre e a senha da instância de banco de dados, escolha View credential details (Ver detalhes da credencial).

    Use o nome de usuário e a senha que aparecem para se conectar à instância de banco de dados como o usuário mestre.

    > **Important**
    >
    > Você não pode visualizar a senha do usuário principal novamente. Caso você não a registre, talvez seja necessário alterá-la.
    > Se for necessário alterar a senha do usuário mestre depois que a instância de banco de dados estiver disponível, será possível modificar a instância de banco de dados para fazer isso. Para ter mais informações sobre a modificação de uma instância de banco de dados, consulte Modificar uma instância de banco de dados do Amazon RDS.

13. Na lista Bancos de dados, selecione o nome da nova instância de banco de dados PostgreSQL para mostrar seus detalhes.

    A instância de banco de dados permanecerá no status Criando até estar pronta para ser usada.



    Quando o status muda para Available (Disponível), você pode se conectar à instância de banco de dados. Dependendo da classe da instância de banco de dados e da quantidade de armazenamento, pode levar até 20 minutos para que a nova instância esteja disponível.

(Opcional) Criar VPC, instância do EC2 e instância do PostgreSQL usando o AWS CloudFormation

Em vez de usar o console para criar a VPC, a instância do EC2 e a instância do PostgreSQL, você pode usar o AWS CloudFormation para provisionar recursos da AWS tratando a infraestrutura como código. Para ajudar a organizar os recursos da AWS em unidades menores e mais gerenciáveis, você pode usar a funcionalidade de pilha aninhada do AWS CloudFormation. Consulte mais informações em Criar uma pilha no console do AWS CloudFormation e Trabalhar com pilhas aninhadas.

> **Important**
>
> O AWS CloudFormation é gratuito, mas os recursos que o CloudFormation são ativos. Você incorre nas taxas de uso padrão para esses recursos até que os encerre. Para ter mais informações, consulte Preço do Amazon RDS para PostgreSQL.

Para criar recursos usando o console do AWS CloudFormation, conclua as seguintes etapas:

- Baixar o modelo do CloudFormation
- Configurar recursos usando o CloudFormation

Baixar o modelo do CloudFormation

Um modelo do CloudFormation é um arquivo de texto JSON ou YAML que contém as informações da configuração dos recursos que você deseja criar na pilha. Esse modelo também cria uma VPC e um bastion host para você junto com a instância do RDS.

Para baixar o arquivo de modelo, abra o link PostgreSQL CloudFormation template.

Na página do Github, clique no botão Baixar arquivo bruto para salvar o arquivo YAML do modelo.

Configurar recursos usando o CloudFormation

> **Note**
>
> Antes de iniciar esse processo, verifique se você tem um par de chaves para uma instância do EC2 na Conta da AWS. Para obter mais informações, consulte Pares de chaves do Amazon EC2 e instâncias do Linux.

Ao usar o modelo do AWS CloudFormation, você deve selecionar os parâmetros certos para garantir que os recursos sejam criados corretamente. Siga as etapas abaixo:

1. Faça login no AWS Management Console e abra o console AWS CloudFormation em https://console.aws.amazon.com/cloudformation.
2. Selecione Create Stack (Criar pilha).
3. Na seção Especificar modelo, selecione Fazer upload de um arquivo de modelo do computador e escolha Próximo.
4. Na página Especificar detalhes da pilha, defina os seguintes parâmetros:
   a. Defina o Nome da pilha como PostgreSQLTestStack.
   b. Em Parâmetros, defina Zonas de disponibilidade selecionando três zonas de disponibilidade.
   c. Em Configuração do bastion host do Linux, em Nome da chave, selecione um par de chaves para fazer login na instância do EC2.
   d. Em Configurações do bastion host do Linux, defina o Intervalo de IP permitido para o endereço IP. Para conectar instâncias do EC2 à VPC usando o Secure Shell (SSH), determine o endereço IP público usando o serviço em https://checkip.amazonaws.com. Um exemplo de endereço IP é 192.0.2.1/32.

> **Warning**
>
> Se usar `0.0.0.0/0` para acesso do SSH, você possibilitará que todos os endereços IP acessem suas instâncias públicas do EC2 usando SSH. Essa abordagem é aceitável por um período curto em um ambiente de teste, mas não é seguro em ambientes de produção. Em produção, autorize somente um endereço IP específico ou um intervalo de endereços para acessar suas instâncias do EC2 usando SSH.

e. Em Configuração geral do banco de dados, defina a Classe da instância do banco de dados como db.t3.micro.

f. Defina o Nome do banco de dados como **`database-test1`**.

g. Em Nome de usuário principal do banco de dados, insira um nome para o usuário principal.

h. Defina a Gerenciar senha de usuário principal do banco de dados com o Secrets Manager como `false` para esse tutorial.

i. Em Senha do banco de dados, defina uma senha de sua escolha. Lembre-se dessa senha para as etapas seguintes do tutorial.

j. Em Configuração de armazenamento de banco de dados, defina o Tipo de armazenamento do banco de dados como gp2.

k. Em Configuração de monitoramento de banco de dados, defina Habilitar o Performance Insights do RDS como falso.

l. Deixe todas as outras configurações com os valores padrão. Clique em Próximo para continuar.

5. Na página Configurar opções de pilha, mantenha todas as opções padrão. Clique em Próximo para continuar.

6. Na página Revisar pilha, selecione Enviar depois de verificar as opções do banco de dados e do bastion host do Linux.

Depois que o processo de criação da pilha for concluído, visualize as pilhas com os nomes BastionStack e RDSNS para anotar as informações necessárias para se conectar ao banco de dados. Consulte mais informações em Viewing AWS CloudFormation stack data and resources on the AWS Management Console.

Conectar-se à instância de banco de dados PostgreSQL

Você pode se conectar à instância de banco de dados usando o pgadmin ou o psql. Este exemplo explica como se conectar a uma instância de banco de dados PostgreSQL usando o cliente da linha de comando psql.

Como se conectar a uma instância de banco de dados PostgreSQL usando psql

1. Encontre o endpoint (nome de DNS) e o número da porta para sua instância de banco de dados.

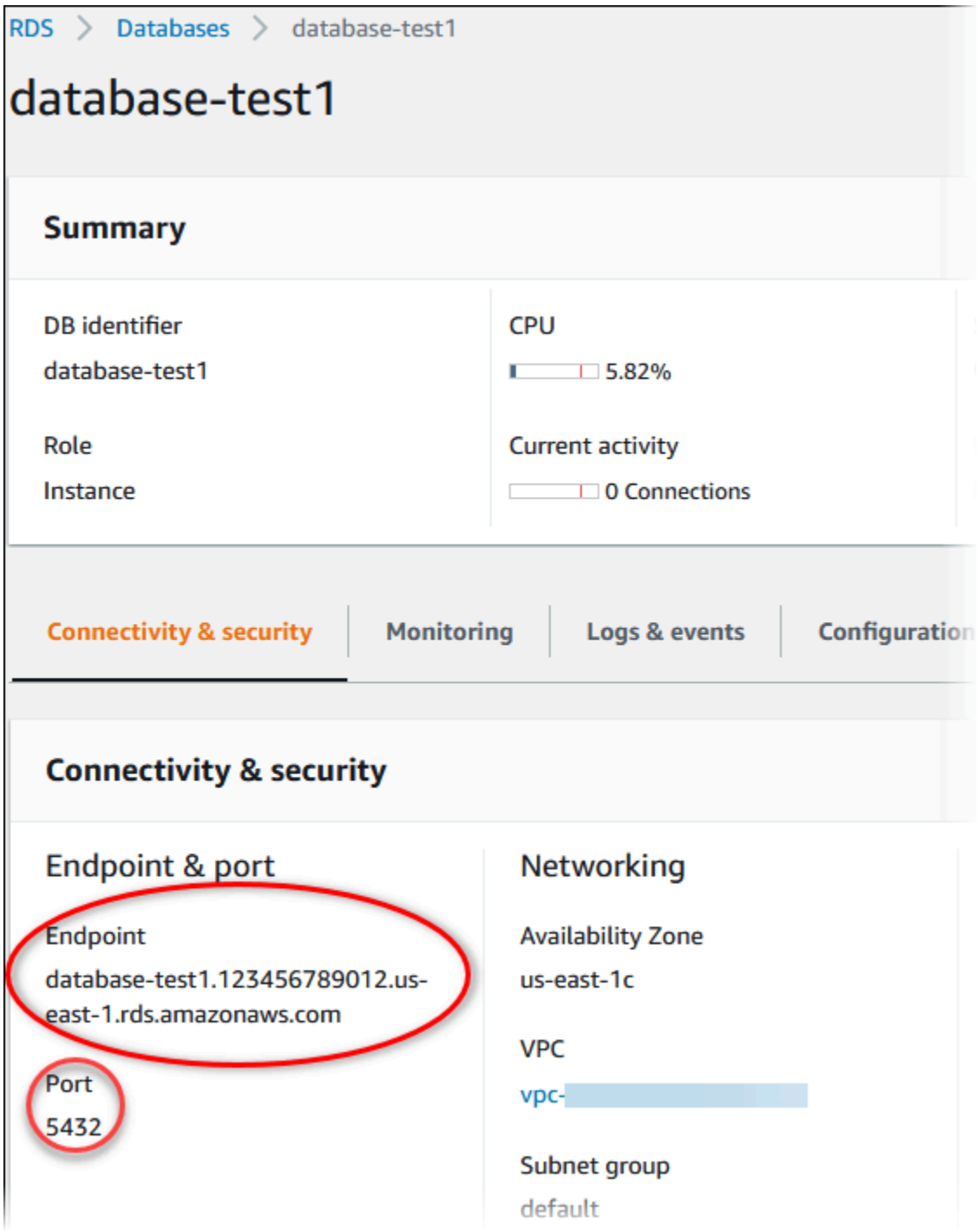
   a. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.

b. No canto superior direito do console do Amazon RDS, escolha a Região da AWS da instância de banco de dados.

c. No painel de navegação, escolha Bancos de dados.

d. Escolha o nome da instância de banco de dados do PostgreSQL para exibir os detalhes.

e. Na guia Connectivity & security (Conectividade e segurança), copie o endpoint. Além disso, anote o número da porta. Você precisará do endpoint e do número da porta para conectar-se à instância de banco de dados.



2. Conecte-se à instância do EC2 que você criou anteriormente, seguindo as etapas em Conecte-se à sua instância do Linux no Guia do usuário do Amazon EC2.

   Recomendamos que você se conecte à sua instância do EC2 utilizando SSH. Se o utilitário cliente SSH estiver instalado no Windows, Linux ou Mac, você poderá se conectar à instância utilizando o seguinte formato de comando:

```
ssh -i location_of_pem_file ec2-user@ec2-instance-public-dns-name
```

Por exemplo, suponha que `ec2-database-connect-key-pair.pem` esteja armazenado em `/dir1` no Linux e que o DNS IPv4 público para sua instância do EC2 seja `ec2-12-345-678-90.compute-1.amazonaws.com`. Seu comando SSH teria a seguinte aparência:

```
ssh -i /dir1/ec2-database-connect-key-pair.pem ec2-
user@ec2-12-345-678-90.compute-1.amazonaws.com
```

3. Obtenha as últimas correções de bugs e atualizações de segurança atualizando o software na instância do EC2. Para fazer isso, execute o seguinte comando.

> **Note**
>
> A opção `-y` instala as atualizações sem solicitar confirmação. Para examinar atualizações antes da instalação, omita essa opção.

```
sudo dnf update -y
```

4. Para instalar o cliente da linha de comando psql do PostgreSQL no Amazon Linux 2023, execute o seguinte comando:

```
sudo dnf install postgresql15
```

5. Conecte-se à instância de banco de dados PostgreSQL. Por exemplo, insira o comando a seguir em um prompt de comando em um computador cliente. Essa ação permite que você se conecte à instância de banco de dados PostgreSQL usando o cliente psql.

   Substitua o endpoint da instância de banco de dados (nome DNS) por *`endpoint`*, substitua o nome do banco de dados `--dbname` que você deseja acessar por *`postgres`* e substitua o nome do usuário principal usado por *`postgres`*. Forneça a senha mestra usada quando for solicitada uma senha.

```
psql --host=endpoint --port=5432 --dbname=postgres --username=postgres
```

Depois de inserir a senha do usuário, você deverá ver uma saída semelhante à seguinte:

```
psql (14.3, server 14.6)
SSL connection (protocol: TLSv1.2, cipher: ECDHE-RSA-AES256-GCM-SHA384, bits: 256,
 compression: off)
Type "help" for help.

postgres=>
```

Para ter mais informações sobre como se conectar à sua instância de banco de dados do PostgreSQL, consulte Conectar a uma instância de banco de dados executando o mecanismo de banco de dados do PostgreSQL. Se você não conseguir se conectar à sua instância de banco de dados, consulte Solucionar problemas de conexões com a sua instância RDS for PostgreSQL.

Por questões de segurança, é uma prática recomendada usar conexões criptografadas. Use uma conexão PostgreSQL não criptografada apenas quando o cliente e o servidor estiverem na mesma VPC e a rede for confiável. Para obter informações sobre como usar conexões criptografadas, consulte Conectar-se à instância de banco de dados PostgreSQL via SSL.

6. Executar comandos SQL.

   Por exemplo, o seguinte comando SQL mostra a data e a hora atuais:

   ```
   SELECT CURRENT_TIMESTAMP;
   ```

Excluir a instância do EC2 e a instância de banco de dados

Depois de se conectar e explorar a instância do EC2 e a instância de banco de dados criadas, exclua-as para não receber mais cobranças por elas.

Se você usou o AWS CloudFormation para criar recursos, ignore essa etapa e passe para a próxima.

Como excluir a instância do EC2

1. Faça login no AWS Management Console e abra o console do Amazon EC2 em https://console.aws.amazon.com/ec2/.
2. No painel de navegação, escolha Instâncias.

3. Selecione a instância do EC2 e escolha Estado da instância, Encerrar instância.
4. Quando a confirmação for solicitada, escolha Encerrar.

Consulte mais informações sobre como excluir uma instância do EC2 em Como encerrar uma instância no Guia do usuário do Amazon EC2.

Para excluir uma instância de banco de dados sem snapshot de banco de dados final

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Databases (Bancos de dados).
3. Escolha a instância de banco de dados que você deseja excluir.
4. Em Actions, selecione Delete.
5. Desmarque Criar snapshot final? e Reter backups automatizados.
6. Conclua a confirmação e escolha Excluir.

(Opcional) Excluir a instância do EC2 e a instância de banco de dados criada com o CloudFormation

Se você usou o AWS CloudFormation para criar recursos, exclua a pilha do CloudFormation depois de conectar e explorar a amostra de instância do EC2 e de instância de banco de dados para não receber mais cobranças por elas.

Para excluir os recursos do CloudFormation

1. Abra o console do AWS CloudFormation.
2. Na página Pilhas no console do CloudFormation, selecione a pilha raiz (a pilha sem o nome VPCStack, BastionStack ou RDSNS).
3. Escolha Excluir.
4. Selecione Excluir pilha quando a confirmação for solicitada.

Consulte mais informações de como excluir uma pilha no CloudFormation em Deleting a stack on the AWS CloudFormation console no Guia do usuário do AWS CloudFormation.

(Opcional) Conectar a instância de banco de dados a uma função do Lambda

Você também pode conectar a instância de banco de dados do RDS para PostgreSQL a um recurso de computação sem servidor do Lambda. As funções do Lambda permitem que você execute código sem provisionar nem gerenciar a infraestrutura. Uma função do Lambda também permite que você responda automaticamente a solicitações de execução de código em qualquer escala, de dezenas de eventos por dia a centenas por segundo. Para obter mais informações, consulte Conectar automaticamente uma função do Lambda e uma instância de banco de dados.

Tutorial: crie um servidor Web e uma instância de banco de dados do Amazon RDS

Este tutorial mostra como instalar um servidor web Apache com PHP e criar um banco de dados MariaDB, MySQL ou PostgreSQL. O servidor web é executado em uma instância do Amazon EC2 usando o Amazon Linux 2023, e você pode escolher entre uma instância de banco de dados do MySQL ou PostgreSQL. Tanto a instância do Amazon EC2 quanto a instância de banco de dados são executadas em uma virtual private cloud (VPC) com base no serviço da Amazon VPC.

Important

Não há cobrança para criar uma conta da AWS. No entanto, ao concluir este tutorial, é possível gerar custos para os recursos da AWS que você usa. Se esses recursos não forem mais necessários após a conclusão do tutorial, você poderá excluí-los.

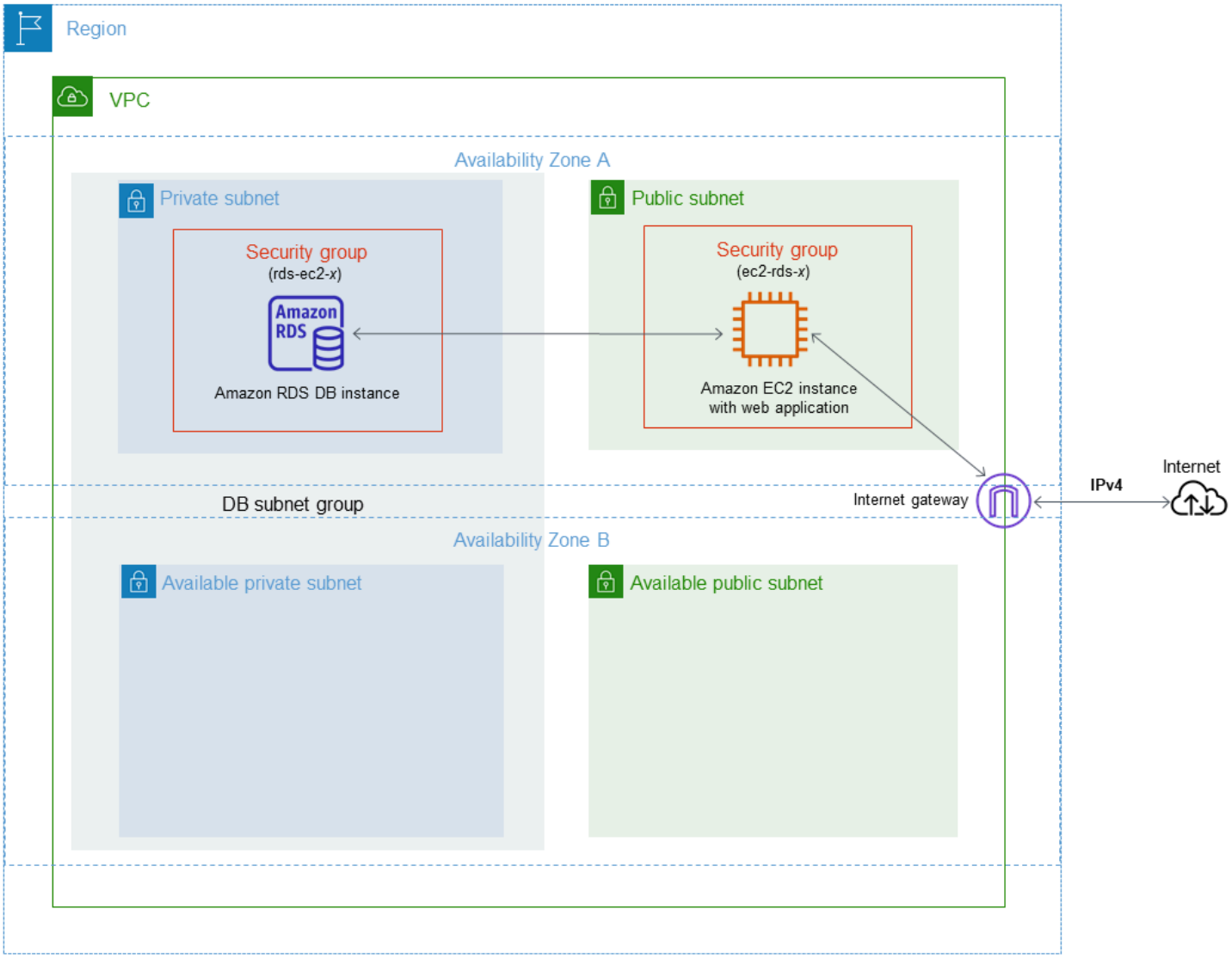
Note

Este tutorial funciona com Amazon Linux 2023 e pode não funcionar para outras versões do Linux.

No tutorial a seguir, crie uma instância do EC2 que usa a VPC, as sub-redes e o grupo de segurança padrão para a sua Conta da AWS. Este tutorial mostra como criar a instância de banco de dados e configurar automaticamente a conectividade com a instância do EC2 que você criou. Depois, o tutorial mostra como instalar o servidor Web na instância do EC2. Conecte o servidor Web à instância de banco de dados na VPC usando o endpoint da instância de banco de dados.

1. Iniciar uma instância do EC2 para se conectar à instância de banco de dados
2. Criar uma instância de banco de dados do Amazon RDS
3. Instalar um servidor Web na instância do EC2

O diagrama a seguir mostrará a configuração quando o tutorial estiver completo.



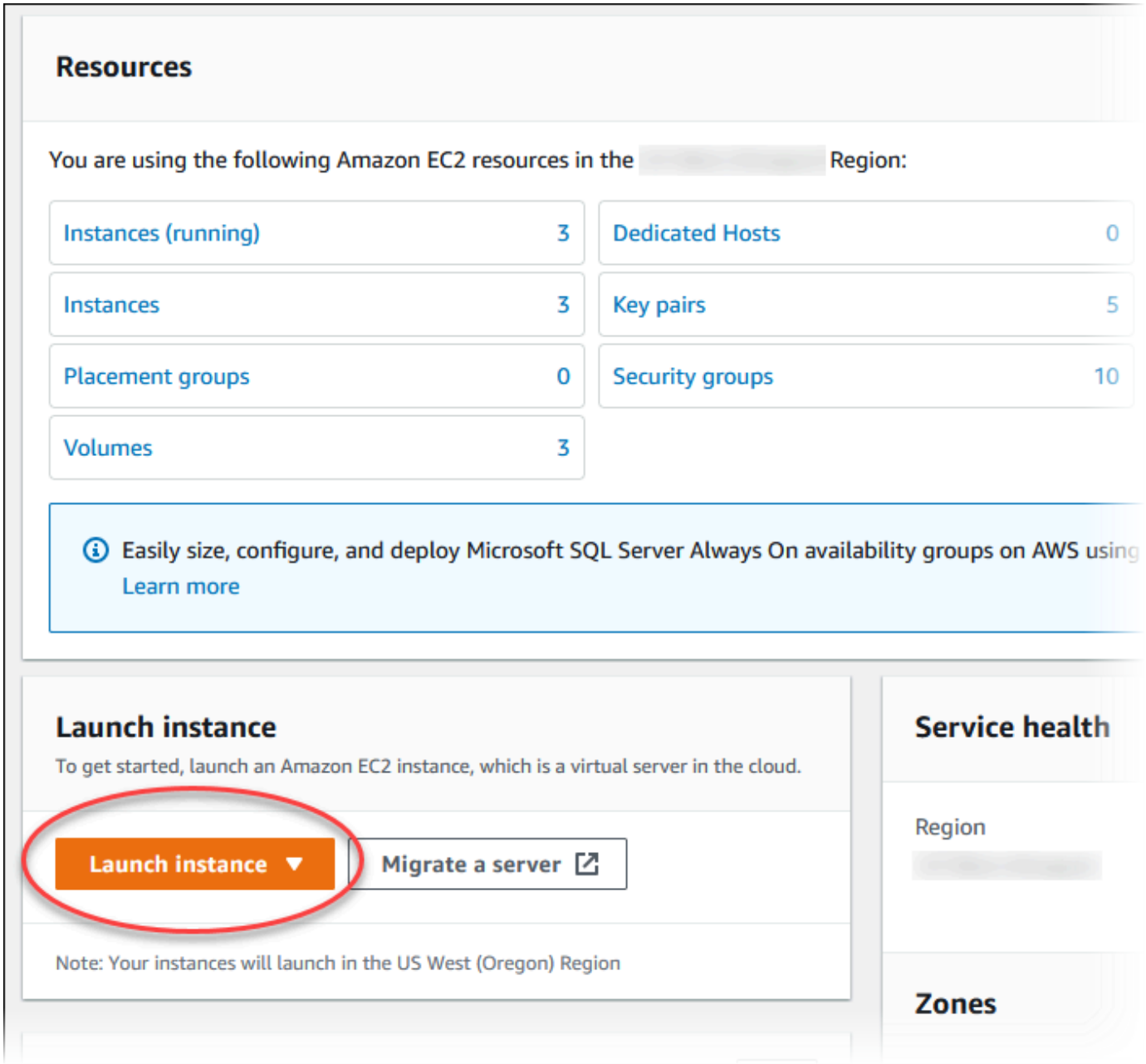
> **Note**
>
> Depois de concluir o tutorial, haverá uma sub-rede pública e privada em cada zona de disponibilidade na sua VPC. Este tutorial usa a VPC padrão para sua Conta da AWS e configura automaticamente a conectividade entre sua instância do EC2 e a instância de banco de dados. Se você preferir configurar uma nova VPC para esse cenário, conclua as tarefas em Tutorial: Criar uma VPC para usar com uma instância de banco de dados (somente IPv4).

Iniciar uma instância do EC2 para se conectar à instância de banco de dados

Crie uma instância do Amazon EC2 na sub-rede pública da VPC.

Para iniciar uma instância do EC2

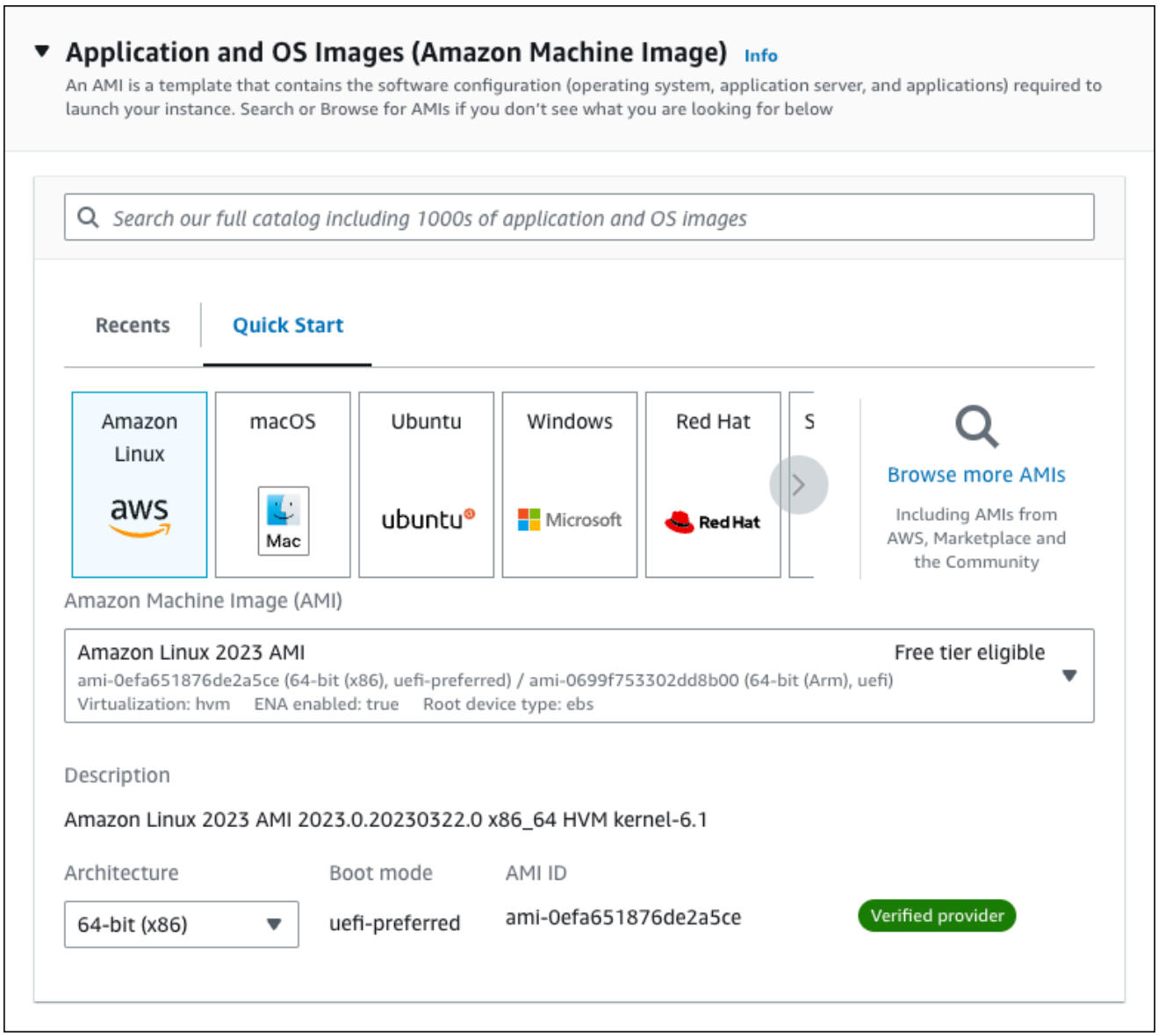
1. Faça login no AWS Management Console e abra o console do Amazon EC2 em https://console.aws.amazon.com/ec2/.
2. No canto superior direito do AWS Management Console, escolha a Região da AWS em que você deseja criar a instância do EC2.
3. Escolha Painel EC2 e Executar instância, conforme mostrado a seguir.



4. Escolha as configurações a seguir na página Iniciar uma instância.

   a. Em Name and tags (Nome e etiquetas), em Name (Nome), insira **`tutorial-ec2-instance-web-server`**.

   b. Em Imagens de aplicações e sistemas operacionais (imagem de máquina da Amazon), selecione Amazon Linux e, depois, AMI do Amazon Linux 2023. Mantenha os padrões nas outras opções.



c. Em Instance type (Tipo de instância), escolha t2.micro.

d. Em Key pair (login) (Par de chaves (login)), escolha um Key pair name (Nome do par de chaves) para usar um par de chaves existente. Para criar um par de chaves para a instância do Amazon EC2, escolha Create new key pair (Criar um novo key pair), depois use a janela Create key pair (Criar par de chaves) para criá-lo.

   Consulte mais informações sobre como criar um par de chaves em Criar um par de chaves no Guia do usuário do Amazon EC2.

e. Em Network settings (Configurações de rede), defina esses valores e mantenha os outros valores como padrão:

- Para Allow SSH traffic from (Permitir tráfego SSH de), escolha a origem das conexões SSH com a instância do EC2.

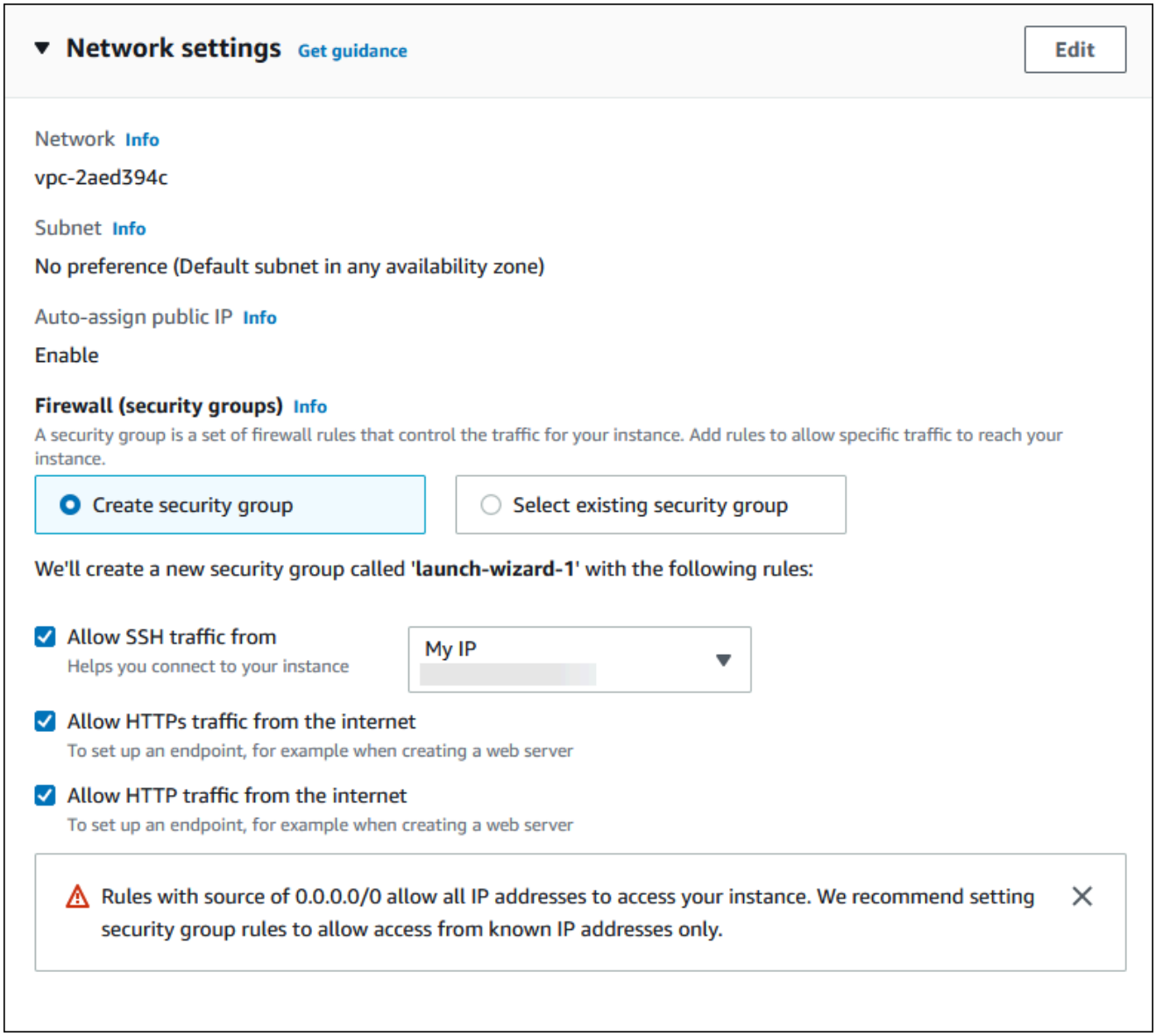
  Você pode escolher My IP (Meu IP) se o endereço IP exibido estiver correto para conexões SSH.

  Caso contrário, determine o endereço IP a ser usado para se conectar a instâncias do EC2 em sua VPC usando Secure Shell (SSH). Para determinar seu endereço IP público, em uma janela ou guia diferente do navegador, é possível usar o serviço em https://checkip.amazonaws.com. Um exemplo de endereço IP: `203.0.113.25/32`.

  Em muitos casos, você pode se conectar por meio de um provedor de serviços de Internet (ISP) ou atrás de um firewall sem um endereço IP estático. Em caso afirmativo, determine o intervalo de endereços IP utilizado por computadores cliente.

  > **Warning**
  >
  > Se usar `0.0.0.0/0` para acesso do SSH, você possibilitará que todos os endereços IP acessem suas instâncias públicas usando SSH. Essa abordagem é aceitável por um período curto em um ambiente de teste, mas não é seguro em ambientes de produção. Em produção, autorize somente um endereço IP específico ou um intervalo de endereços para acessar suas instâncias usando SSH.

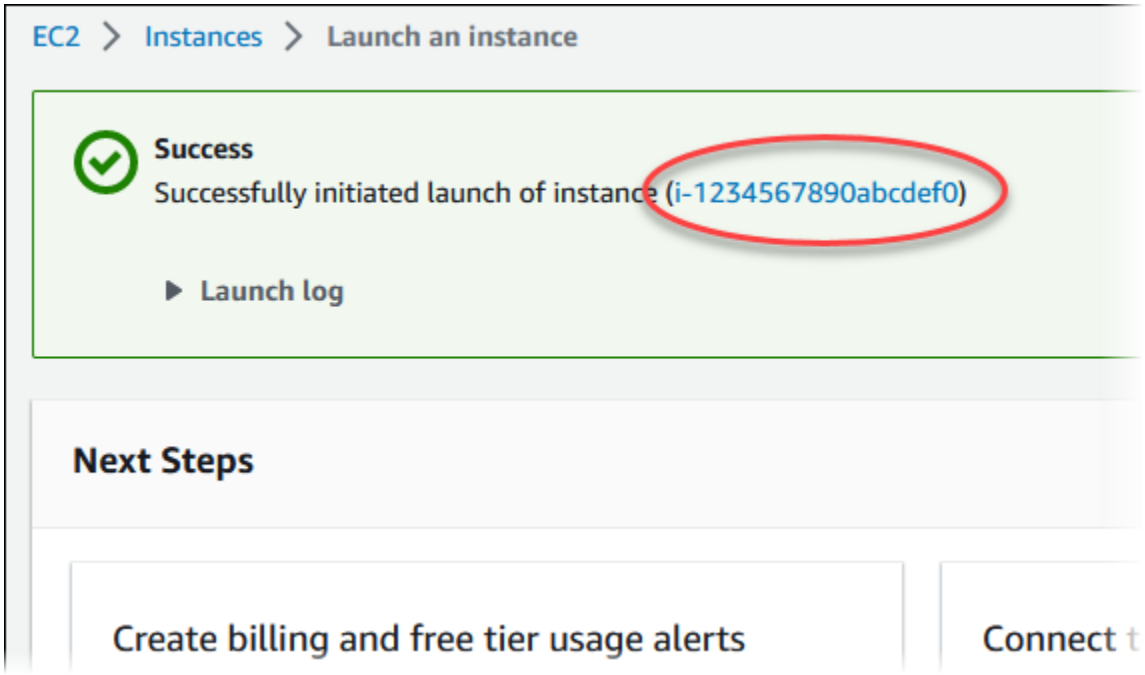
- Ative Allow HTTPs traffic from the internet (Permitir tráfego HTTPs da Internet).
- Ative Allow HTTP traffic from the internet (Permitir tráfego HTTP da Internet).



f. Mantenha os valores padrão para as seções restantes.

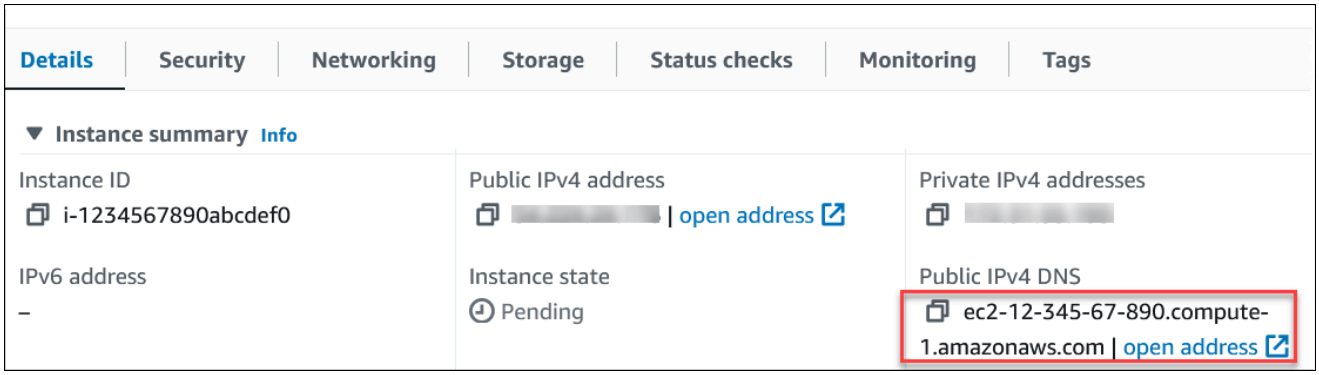
g. Revise um resumo da configuração da instância no painel Summary (Resumo) e, quando você estiver pronto, escolha Launch instance (Iniciar instância).

5. Na página Status de inicialização, anote o identificador de sua nova instância do EC2, por exemplo: `i-1234567890abcdef0`.

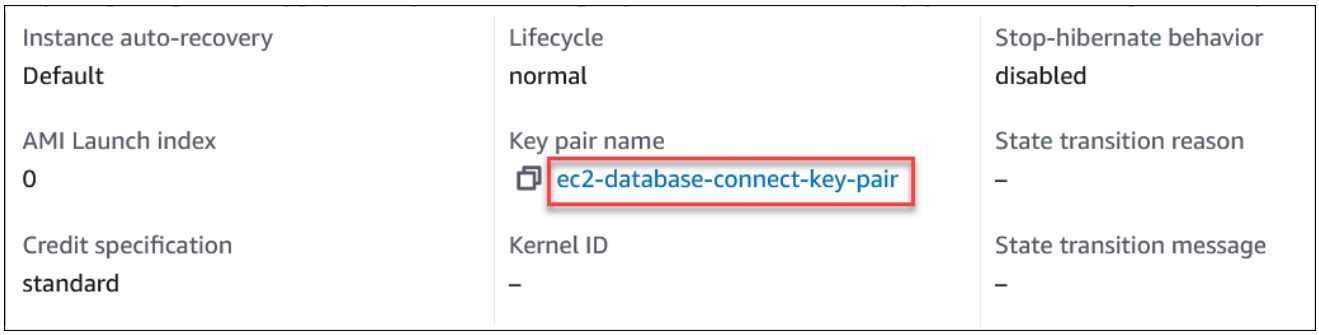


6. Selecione o identificador de instância do EC2 para abrir a lista de instâncias do EC2 e, depois, selecione sua instância do EC2.
7. Na guia Detalhes, observe os seguintes valores, necessários ao se conectar utilizando SSH:

   a. No Resumo da instância, observe o valor do DNS IPv4 público.



   b. Em Detalhes da instância, observe o valor do nome do par de chaves.



8. Aguarde até Instance state (Estado da instância) exibir Running (Em execução) para a instância antes de continuar.
9. Completa Criar uma instância de banco de dados do Amazon RDS.

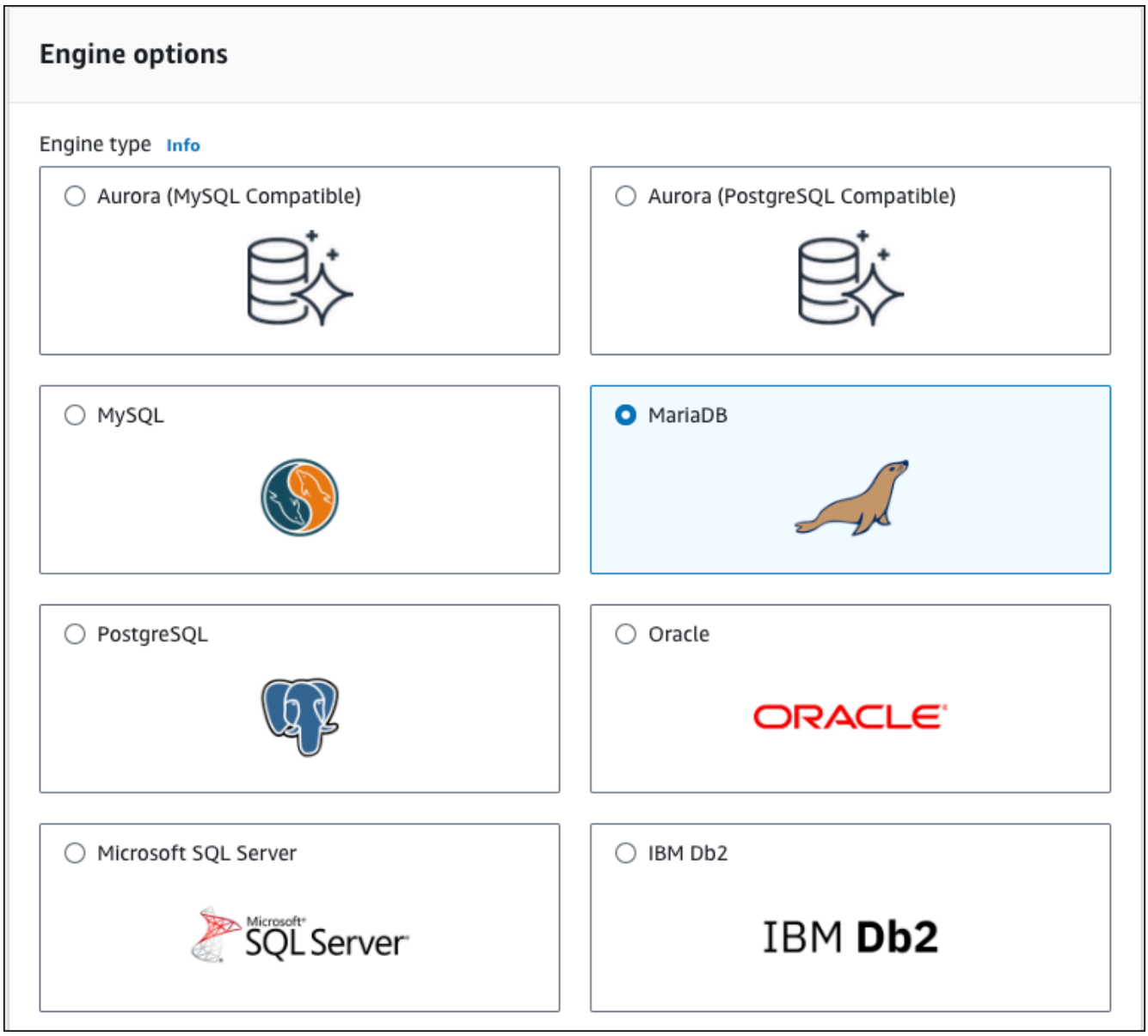
Criar uma instância de banco de dados do Amazon RDS

Crie uma instância de banco de dados do RDS para MariaDB, do RDS para MySQL ou do RDS para PostgreSQL que mantenha os dados usados por uma aplicação web.

RDS for MariaDB

Como criar uma instância do MariaDB

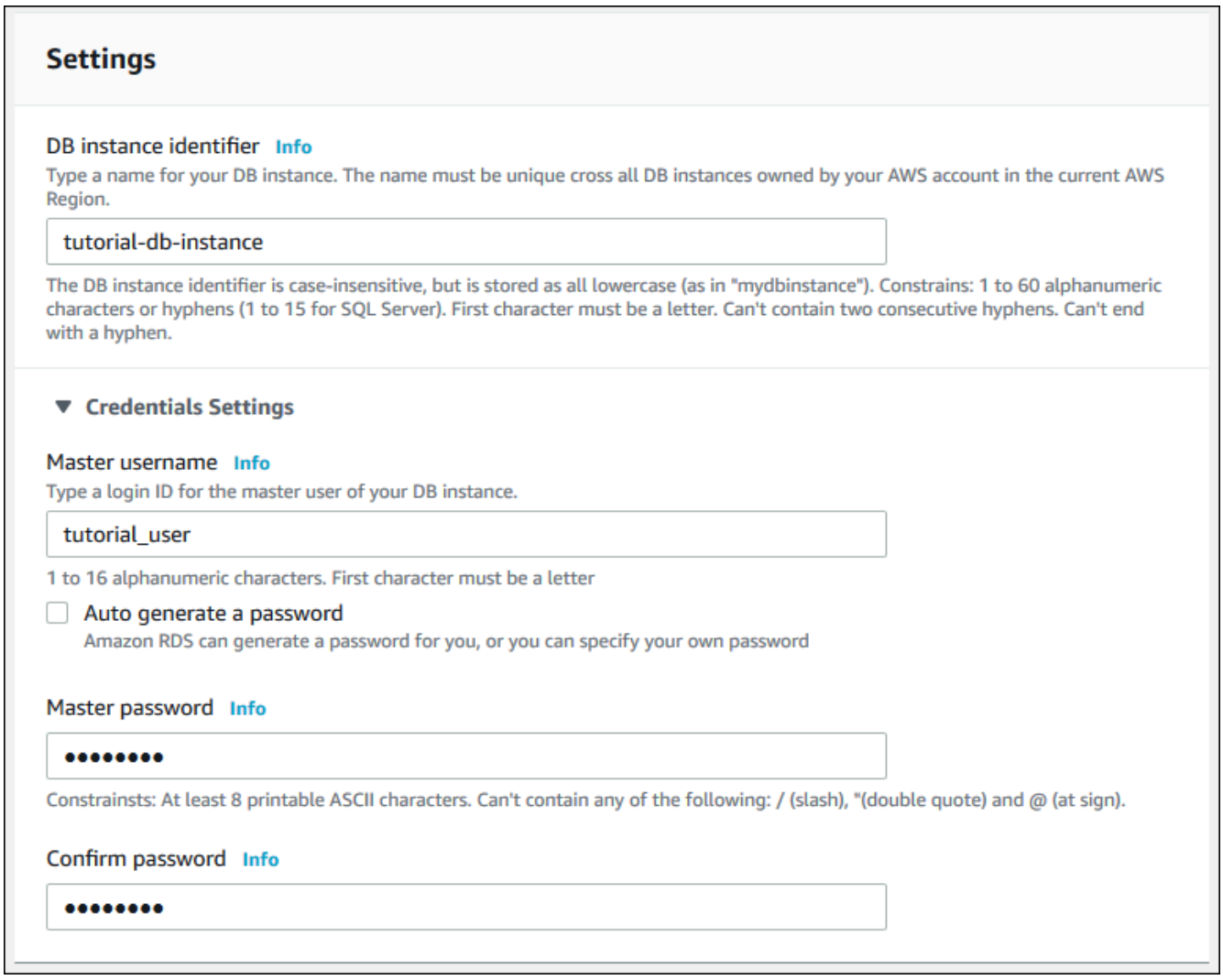
1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No canto superior direito do AWS Management Console, confira a Região da AWS. Deve ser a mesma em que você criou sua instância do EC2.
3. No painel de navegação, escolha Databases (Bancos de dados).
4. Selecione Criar banco de dados.
5. Na página Criar banco de dados, escolha Criação padrão.
6. Em Opções do mecanismo, escolha MariaDB.



7. Em Modelos, escolha Nível gratuito ou Sandbox. Nível gratuito aparece para contas do plano gratuito. Sandbox aparece para contas do plano pago.

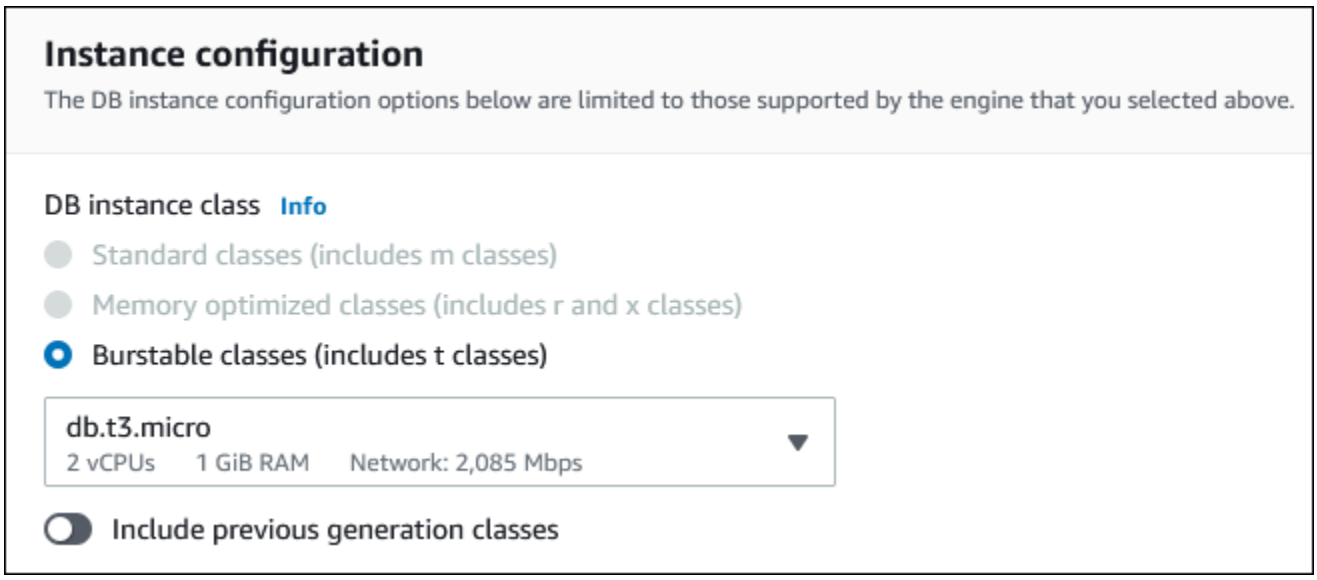


8. Na seção Availability and durability (Disponibilidade e durabilidade), mantenha as opções padrão.
9. Na seção Settings (Configurações), defina estes valores:
   - DB instance identifier (Identificador da instância de banco de dados): digite **`tutorial-db-instance`**.
   - Master username (Nome de usuário principal): digite **`tutorial_user`**.
   - Auto generate a password (Gerar uma senha automaticamente): desabilite a opção.
   - Master password (Senha principal): digite uma senha.
   - Confirm password (Confirmar senha) – digite novamente a senha.



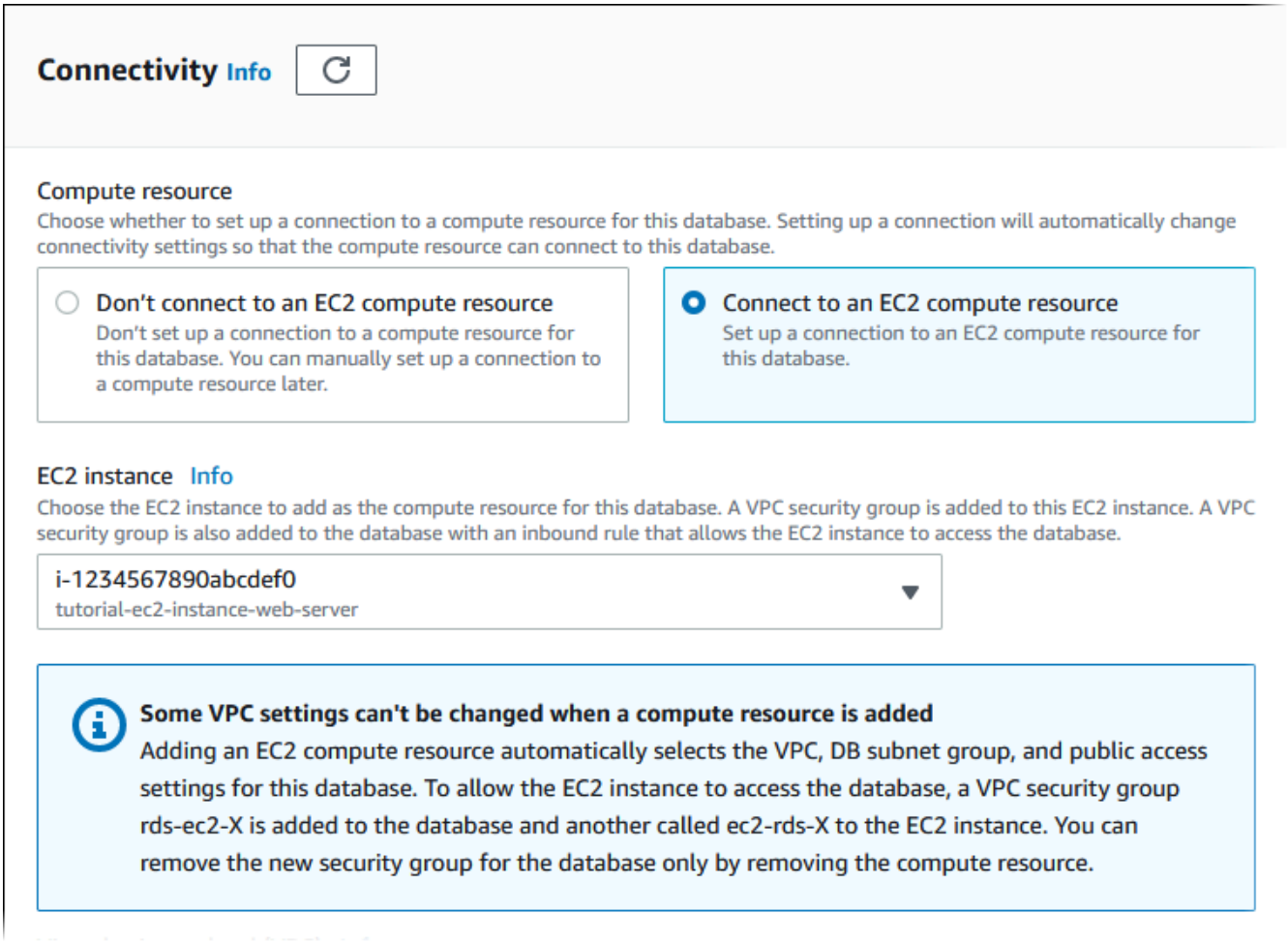
10. Na seção Instance configuration (Configuração da instância), defina estes valores:

    - Classes com capacidade de intermitência (inclui classes t)
    - db.t3.micro



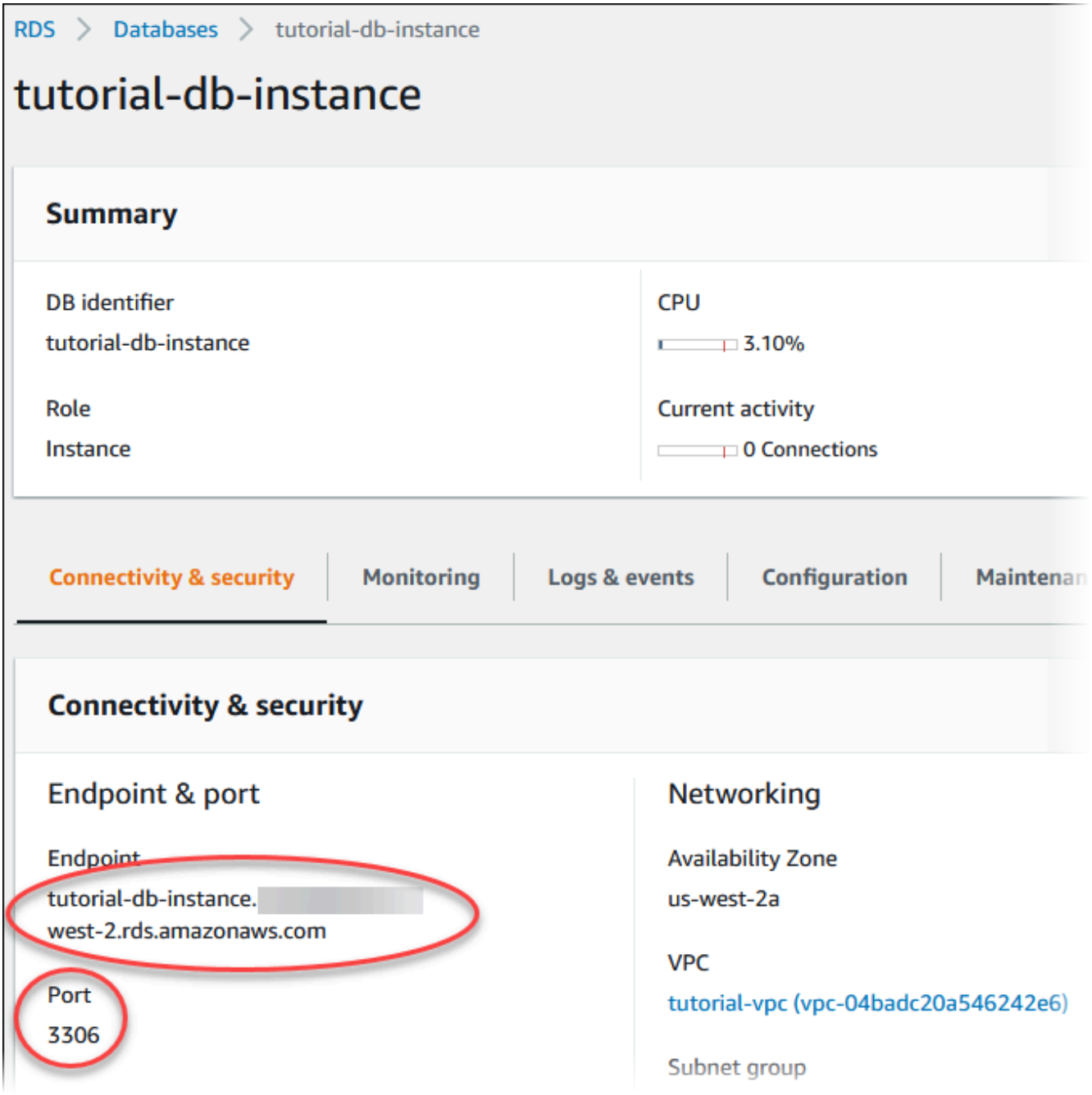
11. Na seção Storage (Armazenamento), mantenha os padrões.
12. Na seção Connectivity (Conectividade), defina esses valores e mantenha os outros valores como padrão:

    - Em Compute resource (Recurso de computação), escolha Connect to an EC2 compute resource (Conectar-se a um recurso de computação do EC2).
    - Em EC2 instance (Instância do EC2), escolha a instância do EC2 criada anteriormente, como tutorial-ec2-instance-web-server.



13. Na seção Autenticação de banco de dados, verifique se a autenticação de senha está selecionada.
14. Abra a seção Additional configuration (Configuração adicional) e insira **`sample`** em Initial database name (Nome do banco de dados inicial). Mantenha as configurações padrão para as outras opções.
15. Para criar a instância do MariaDB, escolha Criar banco de dados.

    Sua nova instância de banco de dados é exibida na lista Databases (Bancos de dados) com o status Creating (Criando).
16. Aguarde o Status de sua nova instância de banco de dados ser mostrado como Available (Disponível). Escolha o nome da instância de banco de dados para mostrar os detalhes.
17. Na seção Connectivity & security (Conectividade e segurança), visualize o Endpoint e a Port (Porta) da instância de banco de dados.



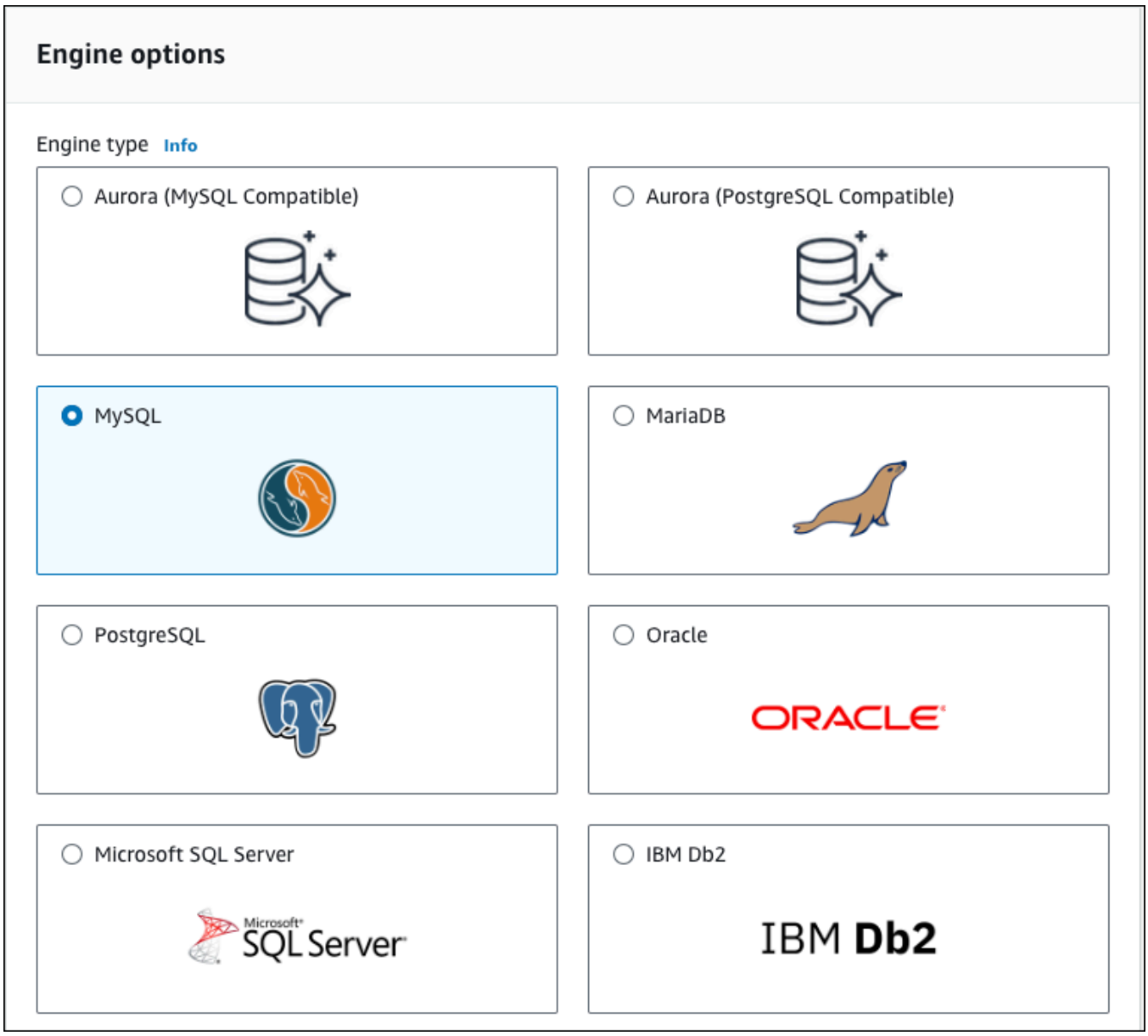
Anote o endpoint e a porta de sua instância de banco de dados. Use essas informações para conectar o servidor Web à instância de banco de dados.

18. Completa Instalar um servidor Web na instância do EC2.

RDS for MySQL

Para criar uma instância de banco de dados MySQL

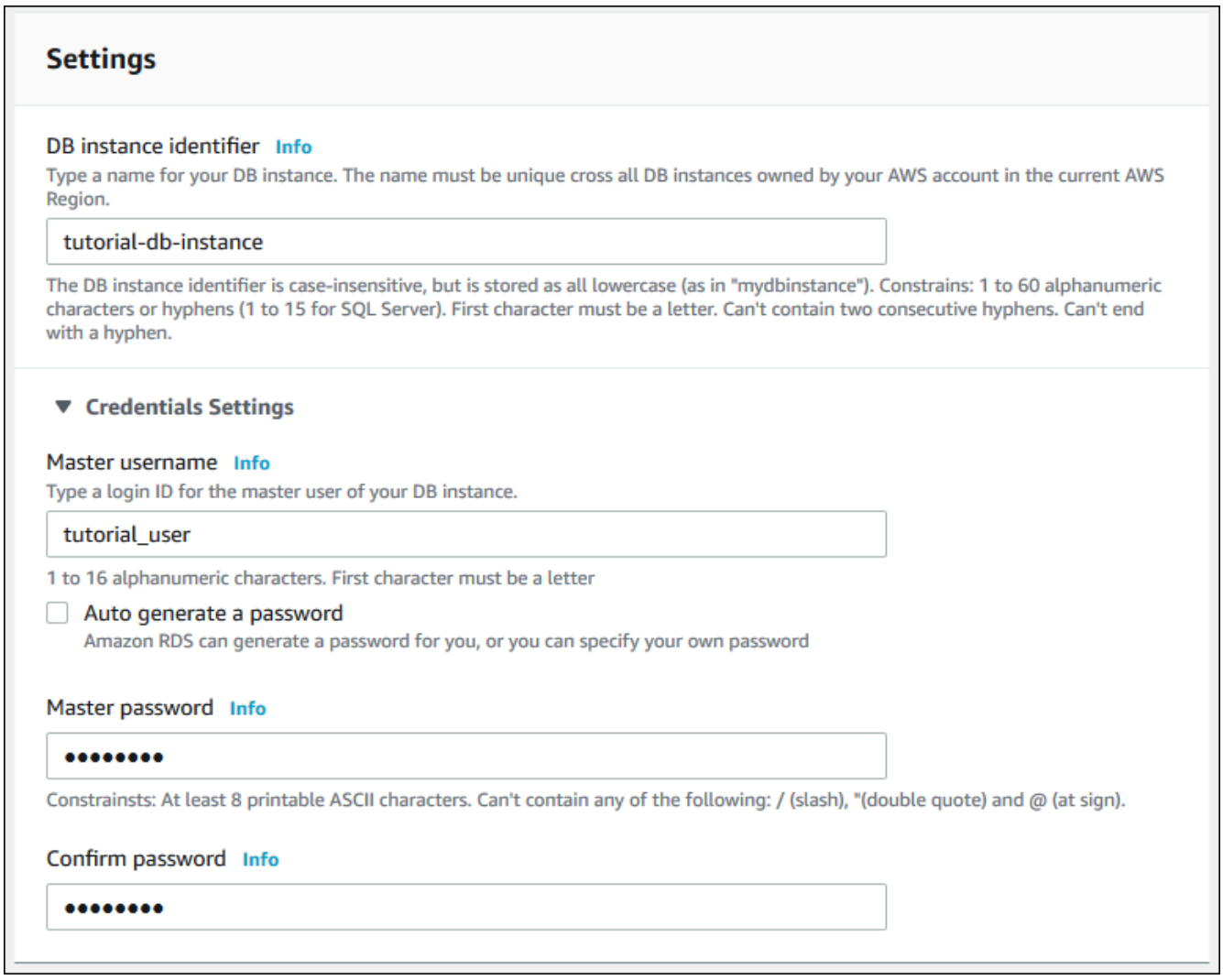
1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No canto superior direito do AWS Management Console, confira a Região da AWS. Deve ser a mesma em que você criou sua instância do EC2.
3. No painel de navegação, escolha Databases (Bancos de dados).
4. Selecione Criar banco de dados.
5. Na página Criar banco de dados, escolha Criação padrão.
6. Em Opções de mecanismo, selecione MySQL.



7. Em Modelos, escolha Nível gratuito ou Sandbox. Nível gratuito aparece para contas do plano gratuito. Sandbox aparece para contas do plano pago.

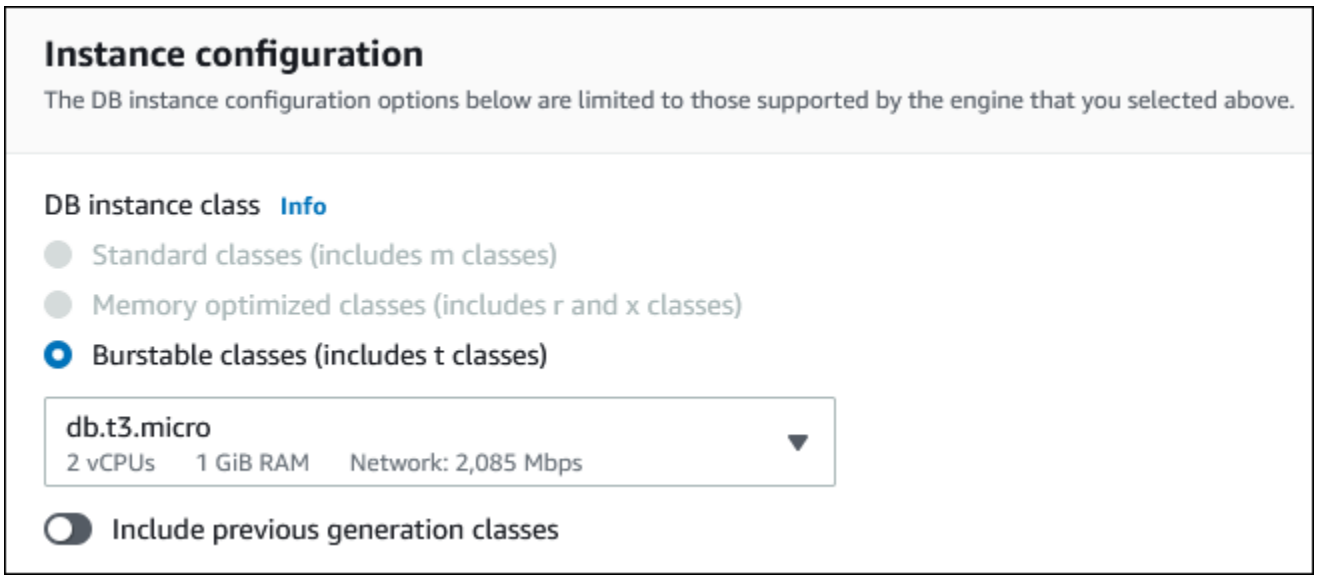


8. Na seção Availability and durability (Disponibilidade e durabilidade), mantenha as opções padrão.
9. Na seção Settings (Configurações), defina estes valores:
   - DB instance identifier (Identificador da instância de banco de dados): digite **`tutorial-db-instance`**.
   - Master username (Nome de usuário principal): digite **`tutorial_user`**.
   - Auto generate a password (Gerar uma senha automaticamente): desabilite a opção.
   - Master password (Senha principal): digite uma senha.
   - Confirm password (Confirmar senha) – digite novamente a senha.



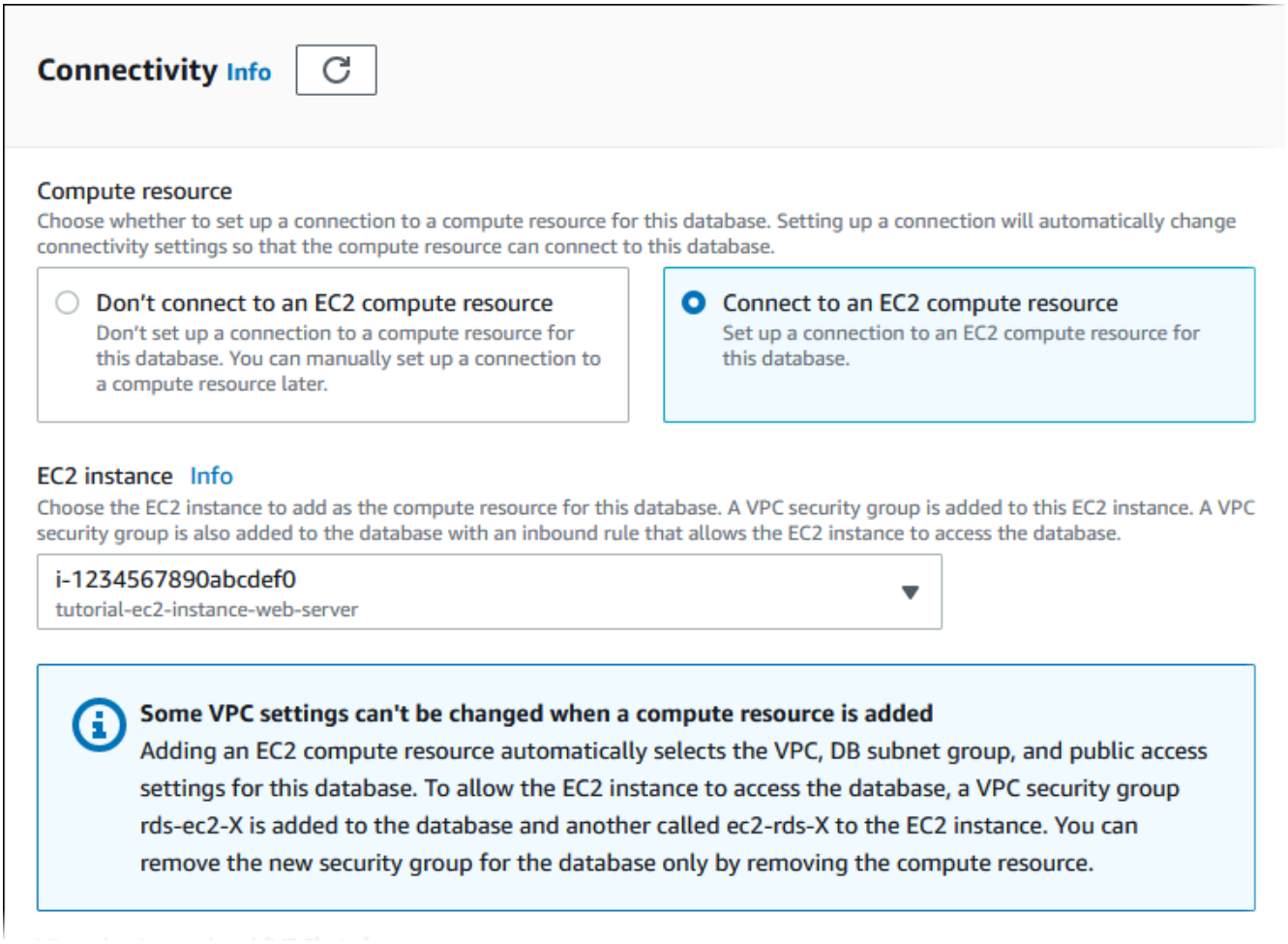
10. Na seção Instance configuration (Configuração da instância), defina estes valores:

    - Classes com capacidade de intermitência (inclui classes t)
    - db.t3.micro



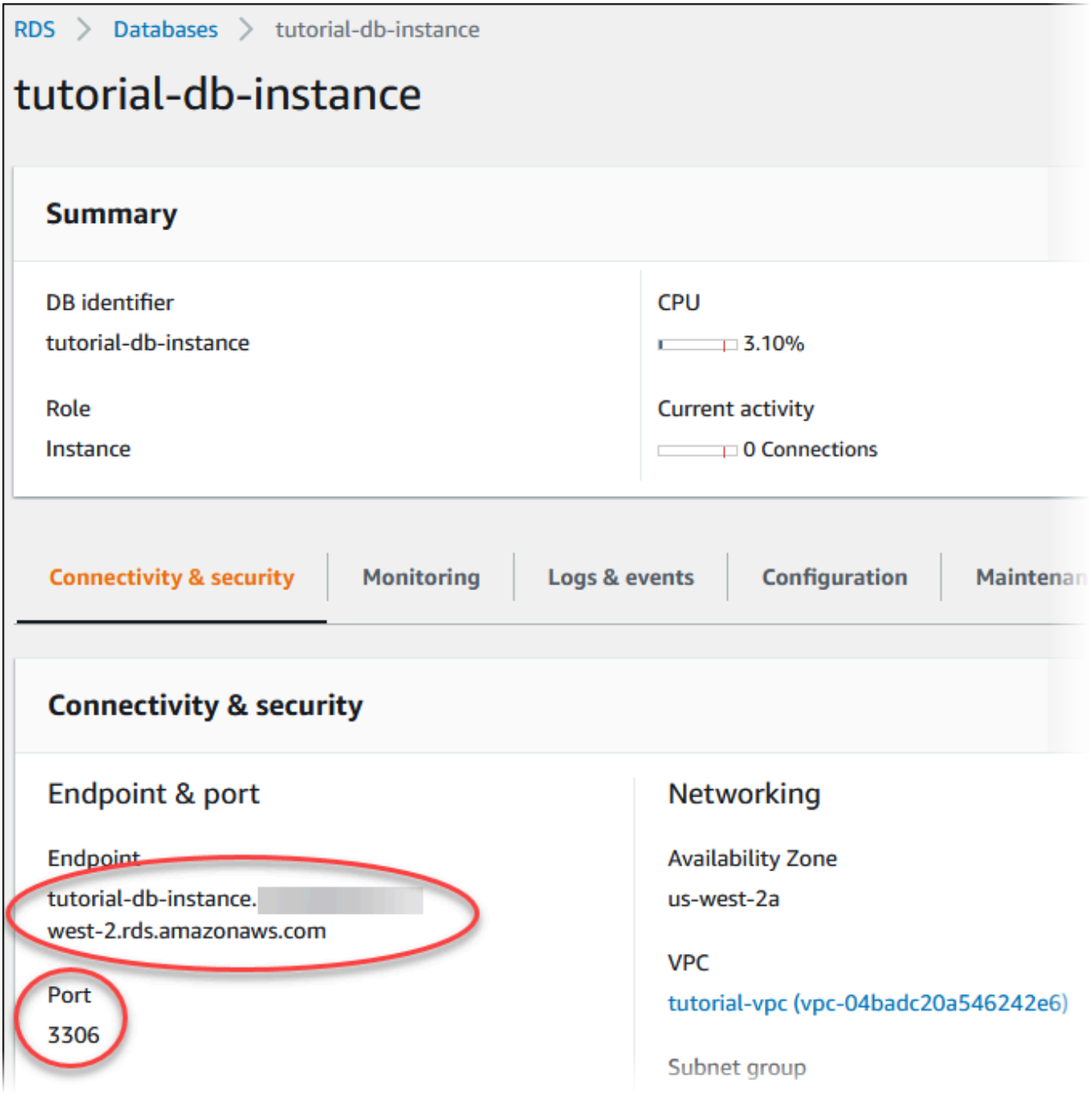
11. Na seção Storage (Armazenamento), mantenha os padrões.
12. Na seção Connectivity (Conectividade), defina esses valores e mantenha os outros valores como padrão:

    - Em Compute resource (Recurso de computação), escolha Connect to an EC2 compute resource (Conectar-se a um recurso de computação do EC2).
    - Em EC2 instance (Instância do EC2), escolha a instância do EC2 criada anteriormente, como tutorial-ec2-instance-web-server.



13. Na seção Autenticação de banco de dados, verifique se a autenticação de senha está selecionada.
14. Abra a seção Additional configuration (Configuração adicional) e insira **`sample`** em Initial database name (Nome do banco de dados inicial). Mantenha as configurações padrão para as outras opções.
15. Para criar sua instância de Bancos de Dados MySQL, escolha Create database (Criar banco de dados).

    Sua nova instância de banco de dados é exibida na lista Databases (Bancos de dados) com o status Creating (Criando).
16. Aguarde o Status de sua nova instância de banco de dados ser mostrado como Available (Disponível). Escolha o nome da instância de banco de dados para mostrar os detalhes.
17. Na seção Connectivity & security (Conectividade e segurança), visualize o Endpoint e a Port (Porta) da instância de banco de dados.



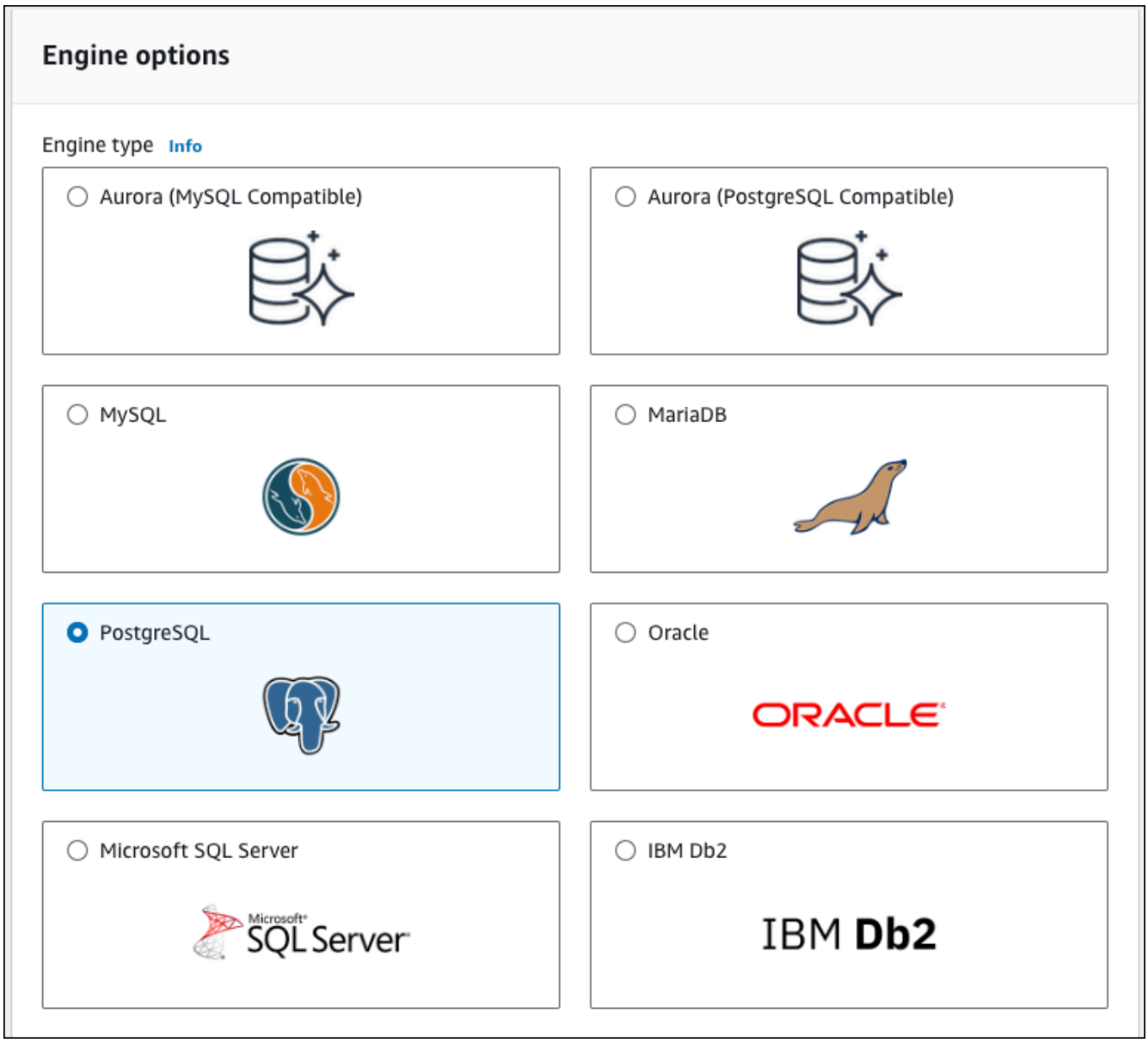
Anote o endpoint e a porta de sua instância de banco de dados. Use essas informações para conectar o servidor Web à instância de banco de dados.

18. Completa Instalar um servidor Web na instância do EC2.

RDS for PostgreSQL

Criar uma instância de banco de dados do PostgreSQL

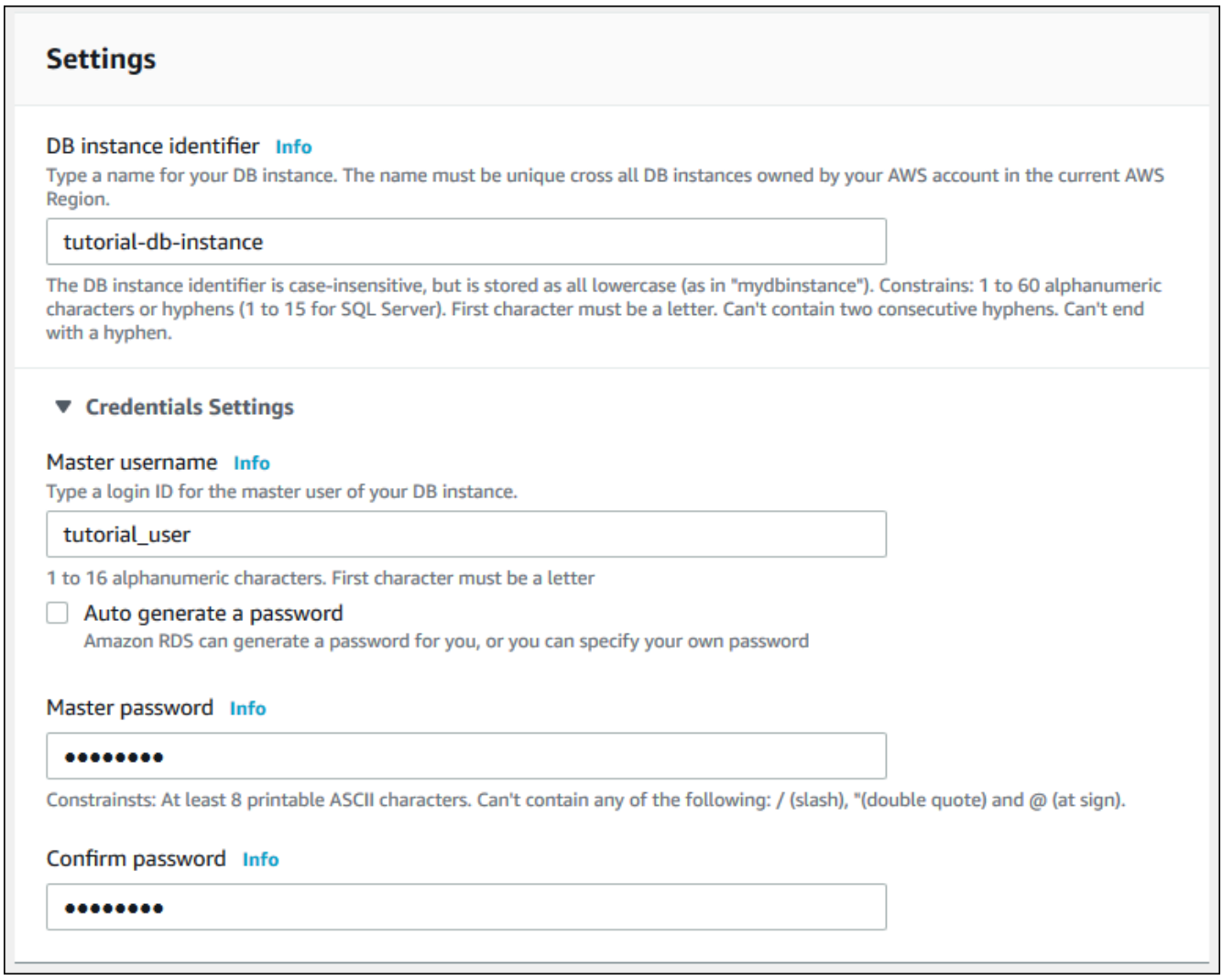
1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No canto superior direito do AWS Management Console, confira a Região da AWS. Deve ser a mesma em que você criou sua instância do EC2.
3. No painel de navegação, escolha Databases (Bancos de dados).
4. Selecione Criar banco de dados.
5. Na página Criar banco de dados, escolha Criação padrão.
6. Em Opções de mecanismo, escolha PostgreSQL.



7. Em Modelos, escolha Nível gratuito ou Sandbox. Nível gratuito aparece para contas do plano gratuito. Sandbox aparece para contas do plano pago.

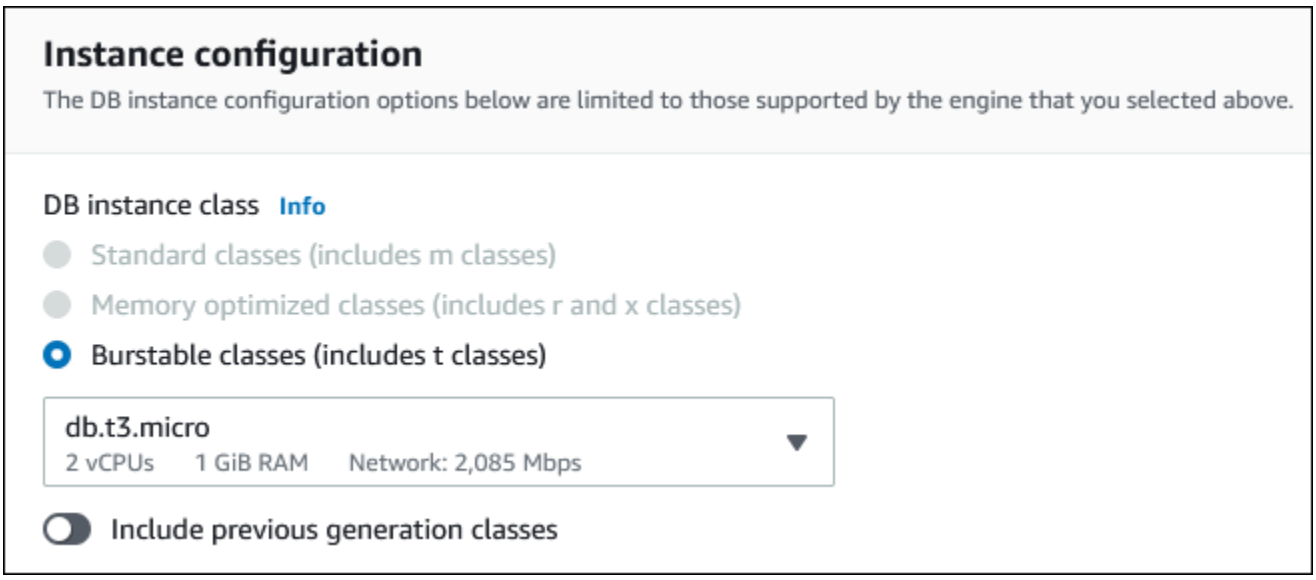


8. Na seção Availability and durability (Disponibilidade e durabilidade), mantenha as opções padrão.
9. Na seção Settings (Configurações), defina estes valores:
   - DB instance identifier (Identificador da instância de banco de dados): digite **`tutorial-db-instance`**.
   - Master username (Nome de usuário principal): digite **`tutorial_user`**.
   - Auto generate a password (Gerar uma senha automaticamente): desabilite a opção.
   - Master password (Senha principal): digite uma senha.
   - Confirm password (Confirmar senha) – digite novamente a senha.



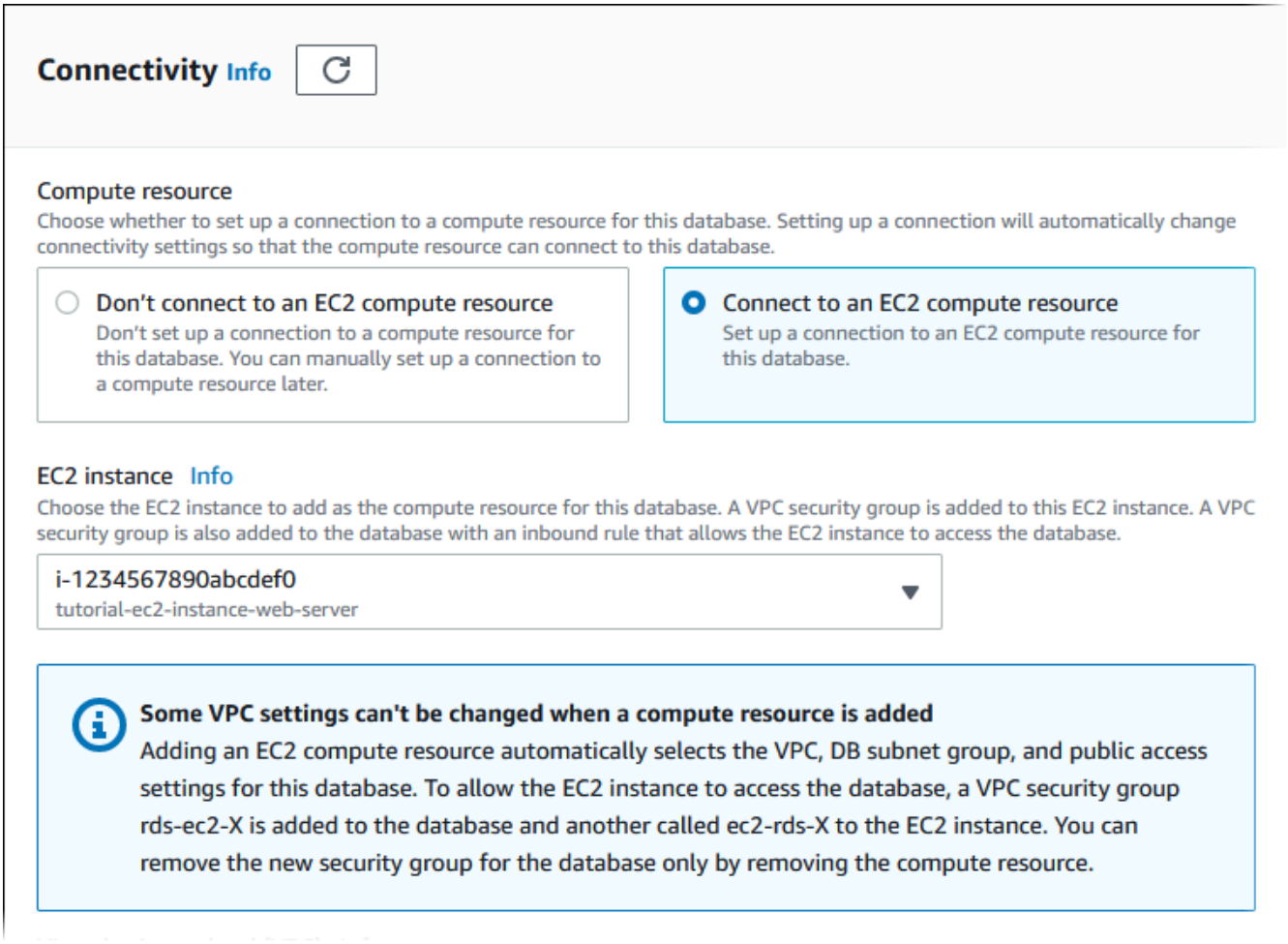
10. Na seção Instance configuration (Configuração da instância), defina estes valores:

    - Classes com capacidade de intermitência (inclui classes t)
    - db.t3.micro



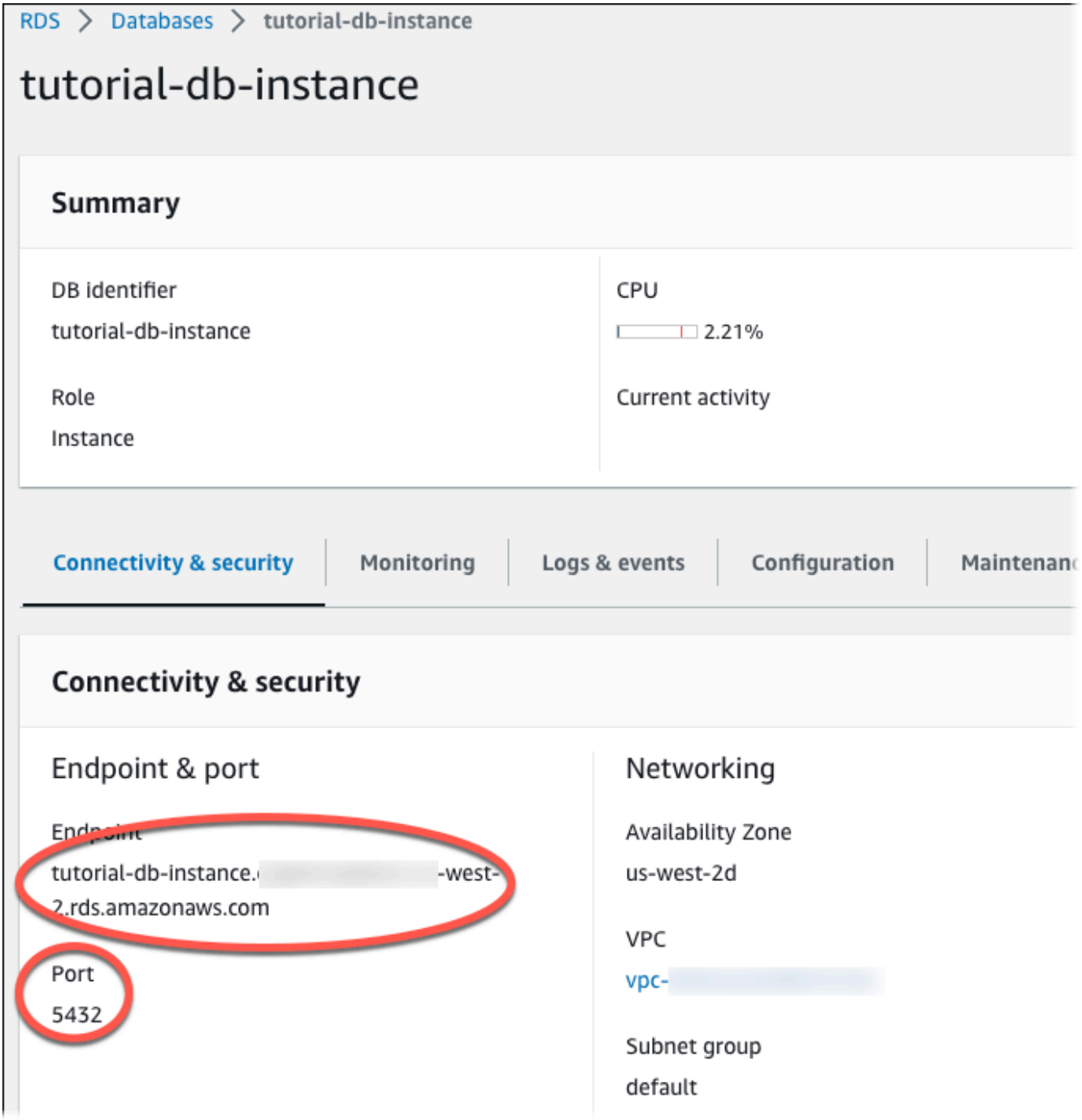
11. Na seção Storage (Armazenamento), mantenha os padrões.
12. Na seção Connectivity (Conectividade), defina esses valores e mantenha os outros valores como padrão:

    - Em Compute resource (Recurso de computação), escolha Connect to an EC2 compute resource (Conectar-se a um recurso de computação do EC2).
    - Em EC2 instance (Instância do EC2), escolha a instância do EC2 criada anteriormente, como tutorial-ec2-instance-web-server.



13. Na seção Autenticação de banco de dados, verifique se a autenticação de senha está selecionada.
14. Abra a seção Additional configuration (Configuração adicional) e insira **`sample`** em Initial database name (Nome do banco de dados inicial). Mantenha as configurações padrão para as outras opções.
15. Para criar uma instância de banco de dados do PostgreSQL, escolha Criar banco de dados.

    Sua nova instância de banco de dados é exibida na lista Databases (Bancos de dados) com o status Creating (Criando).
16. Aguarde o Status de sua nova instância de banco de dados ser mostrado como Available (Disponível). Escolha o nome da instância de banco de dados para mostrar os detalhes.
17. Na seção Connectivity & security (Conectividade e segurança), visualize o Endpoint e a Port (Porta) da instância de banco de dados.



Anote o endpoint e a porta de sua instância de banco de dados. Use essas informações para conectar o servidor Web à instância de banco de dados.

18. Completa Instalar um servidor Web na instância do EC2.

Instalar um servidor Web na instância do EC2

Instale um servidor Web na instância do EC2 criada em Iniciar uma instância do EC2 para se conectar à instância de banco de dados. O servidor Web se conecta à instância de banco de dados do Amazon RDS criada em Criar uma instância de banco de dados do Amazon RDS.

Instale um servidor Web do Apache com PHP e MariaDB

Conecte-se à sua instância do EC2 e instale o servidor Web.

Para conectar-se à sua instância do EC2 e instalar o servidor na web com PHP

1. Conecte-se à instância do EC2 que você criou anteriormente, seguindo as etapas em Conecte-se à sua instância do Linux no Guia do usuário do Amazon EC2.

   Recomendamos que você se conecte à sua instância do EC2 utilizando SSH. Se o utilitário cliente SSH estiver instalado no Windows, Linux ou Mac, você poderá se conectar à instância utilizando o seguinte formato de comando:

   ```
   ssh -i location_of_pem_file ec2-user@ec2-instance-public-dns-name
   ```

   Por exemplo, suponha que `ec2-database-connect-key-pair.pem` esteja armazenado em `/dir1` no Linux e que o DNS IPv4 público para sua instância do EC2 seja `ec2-12-345-678-90.compute-1.amazonaws.com`. Seu comando SSH teria a seguinte aparência:

   ```
   ssh -i /dir1/ec2-database-connect-key-pair.pem ec2-
   user@ec2-12-345-678-90.compute-1.amazonaws.com
   ```

2. Obtenha as últimas correções de bugs e atualizações de segurança atualizando o software na instância do EC2. Para fazer isso, execute o seguinte comando.

   > **Note**
   >
   > A opção `-y` instala as atualizações sem solicitar confirmação. Para examinar atualizações antes da instalação, omita essa opção.

   ```
   sudo dnf update -y
   ```

3. Após a conclusão das atualizações, instale o servidor web Apache, o PHP e o software MariaDB ou PostgreSQL utilizando os comandos a seguir. Este comando instala vários pacotes de software e dependências relacionadas ao mesmo tempo.

   MariaDB & MySQL

   ```
   sudo dnf install -y httpd php php-mysqli mariadb105
   ```

   PostgreSQL

   ```
   sudo dnf install -y httpd php php-pgsql postgresql15
   ```

   Se você receber um erro, isso significa que sua instância provavelmente não foi iniciada com uma AMI do Amazon Linux 2023. Em vez disso é, possível utilizar a AMI do Amazon Linux 2. Você pode visualizar sua versão do Amazon Linux usando o comando a seguir.

   ```
   cat /etc/system-release
   ```

   Para ter mais informações, consulte Atualização de software da instância.

4. Inicie o servidor Web com o comando mostrado a seguir.

   ```
   sudo systemctl start httpd
   ```

   É possível testar se o servidor Web está instalado e se foi iniciado corretamente. Para fazer isso, insira o nome de Domain Name System (DNS) público da instância do EC2 na barra de endereços de um navegador da web, por exemplo: `http://ec2-42-8-168-21.us-west-1.compute.amazonaws.com`. Se o seu servidor na web estiver em execução, você verá a página de teste do Apache.

   Se você não vir a página de teste do Apache, verifique as regras de entrada para o grupo de segurança da VPC criado no Tutorial: Criar uma VPC para usar com uma instância de banco de dados (somente IPv4). Verifique se as regras de entrada incluem uma regra que permita o acesso HTTP (porta 80) ao endereço IP para se conectar ao servidor da Web.

> **Note**
>
> A página de teste do Apache aparece apenas quando não há conteúdo no diretório raiz do documento, `/var/www/html`. Depois de adicionar o conteúdo ao diretório raiz de documentos, o conteúdo aparecerá no endereço DNS público da instância do EC2. Antes desse ponto, ele aparece na página de teste do Apache.

5. Configure o servidor Web para começar com cada inicialização do sistema usando o comando `systemctl`.

```
sudo systemctl enable httpd
```

Para permitir que `ec2-user` gerencie arquivos no diretório raiz padrão de servidor Web do Apache, modifique a propriedade e as permissões do diretório `/var/www`. Existem diversas maneiras de realizar essa tarefa. Neste tutorial, você adiciona o usuário `ec2-user` ao grupo `apache` para dar ao grupo `apache` a propriedade do diretório `/var/www` e atribuir permissões de gravação ao grupo.

Para definir as permissões de arquivos para o servidor na web Apache

1. Adicione o usuário `ec2-user` ao grupo `apache`.

```
sudo usermod -a -G apache ec2-user
```

2. Faça logout para atualizar as permissões e incluir o novo grupo `apache`.

```
exit
```

3. Faça login novamente e verifique se o grupo `apache` existe com o comando `groups`.

```
groups
```

   A saída será semelhante à seguinte:

```
ec2-user adm wheel apache systemd-journal
```

4. Altere a propriedade do grupo do diretório `/var/www` e o seu conteúdo para o grupo `apache`.

```
sudo chown -R ec2-user:apache /var/www
```

5. Altere as permissões do diretório do `/var/www` e dos subdiretórios para adicionar permissões de gravação do grupo e definir o ID do grupo em subdiretórios criados futuramente.

```
sudo chmod 2775 /var/www
find /var/www -type d -exec sudo chmod 2775 {} \;
```

6. Altere recursivamente as permissões de arquivos do diretório `/var/www` e os subdiretórios para adicionar permissões de gravação.

```
find /var/www -type f -exec sudo chmod 0664 {} \;
```

Agora, `ec2-user` (e todos os outros membros do grupo `apache`) pode adicionar, excluir e editar arquivos na raiz do documento Apache. Isso possibilita que você adicione conteúdo, como um site estático ou uma aplicação PHP.

> **Note**
>
> Um servidor web que executa o protocolo HTTP não fornece nenhuma segurança de transporte para os dados que envia ou recebe. Quando você se conecta a um servidor HTTP usando um navegador da Web, muitas informações ficam visíveis para os espiões em qualquer ponto da rede. Essas informações incluem os URLs que você acessa, o conteúdo de páginas da web recebido e o conteúdo (inclusive senhas) de todos os formulários HTML. A prática recomendada para proteger o servidor da Web é instalar suporte para HTTPS (HTTP seguro). Esse protocolo protege seus dados com criptografia SSL/TLS. Para ter mais informações, consulte Tutorial: Configurar o SSL/TLS com a AMI do Amazon Linux no Guia do Usuário do Amazon EC2.

Conectar o servidor Web Apache à instância de banco de dados

Depois, adicione o conteúdo ao servidor Web Apache que se conecta à instância de banco de dados do Amazon RDS.

Como adicionar o conteúdo ao servidor Web Apache que se conecta à instância de banco de dados

1. Enquanto estiver conectado à instância do EC2, altere o diretório para `/var/www` e crie um novo subdiretório chamado `inc`.

```
cd /var/www
mkdir inc
cd inc
```

2. Crie um novo arquivo no diretório `inc` chamado `dbinfo.inc` e edite o arquivo chamando nano (ou o editor de sua escolha).

```
>dbinfo.inc
nano dbinfo.inc
```

3. Adicione o conteúdo a seguir ao arquivo `dbinfo.inc`. Aqui, *`db_instance_endpoint`* é o endpoint de instância de banco de dados, sem a porta, para sua instância de banco de dados.

> **Note**
>
> Recomendamos colocar as informações de nome de usuário e senha em uma pasta que não faça parte da raiz do documento do servidor Web. Isso reduz a possibilidade de suas informações de segurança serem expostas.
> Altere `master password` para uma senha adequada em sua aplicação.

```
<?php

define('DB_SERVER', 'db_instance_endpoint');
define('DB_USERNAME', 'tutorial_user');
define('DB_PASSWORD', 'master password');
define('DB_DATABASE', 'sample');
?>
```

4. Salve e feche o arquivo `dbinfo.inc`. Se você estiver usando nano, salve e feche o arquivo usando Ctrl+S e Ctrl+X.

5. Altere o diretório para `/var/www/html`.

```
cd /var/www/html
```

6. Crie um novo arquivo no diretório `html` chamado `SamplePage.php` e edite o arquivo chamando nano (ou o editor de sua escolha).

```
>SamplePage.php
nano SamplePage.php
```

7. Adicione o conteúdo a seguir ao arquivo `SamplePage.php`:

   MariaDB & MySQL

```
<?php include "../inc/dbinfo.inc"; ?>
<html>
<body>
<h1>Sample page</h1>
<?php

  /* Connect to MySQL and select the database. */
  $connection = mysqli_connect(DB_SERVER, DB_USERNAME, DB_PASSWORD);

  if (mysqli_connect_errno()) echo "Failed to connect to MySQL: " .
 mysqli_connect_error();

  $database = mysqli_select_db($connection, DB_DATABASE);

  /* Ensure that the EMPLOYEES table exists. */
  VerifyEmployeesTable($connection, DB_DATABASE);

  /* If input fields are populated, add a row to the EMPLOYEES table. */
  $employee_name = htmlentities($_POST['NAME']);
  $employee_address = htmlentities($_POST['ADDRESS']);

  if (strlen($employee_name) || strlen($employee_address)) {
    AddEmployee($connection, $employee_name, $employee_address);
  }
?>

<!-- Input form -->
<form action="<?PHP echo $_SERVER['SCRIPT_NAME'] ?>" method="POST">
  <table border="0">
    <tr>
      <td>NAME</td>
      <td>ADDRESS</td>
    </tr>
```

```
    <tr>
      <td>
        <input type="text" name="NAME" maxlength="45" size="30" />
      </td>
      <td>
        <input type="text" name="ADDRESS" maxlength="90" size="60" />
      </td>
      <td>
        <input type="submit" value="Add Data" />
      </td>
    </tr>
  </table>
</form>

<!-- Display table data. -->
<table border="1" cellpadding="2" cellspacing="2">
  <tr>
    <td>ID</td>
    <td>NAME</td>
    <td>ADDRESS</td>
  </tr>

<?php

$result = mysqli_query($connection, "SELECT * FROM EMPLOYEES");

while($query_data = mysqli_fetch_row($result)) {
  echo "<tr>";
  echo "<td>",$query_data[0], "</td>",
       "<td>",$query_data[1], "</td>",
       "<td>",$query_data[2], "</td>";
  echo "</tr>";
}
?>

</table>

<!-- Clean up. -->
<?php

  mysqli_free_result($result);
  mysqli_close($connection);

?>
```

```
</body>
</html>


<?php

/* Add an employee to the table. */
function AddEmployee($connection, $name, $address) {
   $n = mysqli_real_escape_string($connection, $name);
   $a = mysqli_real_escape_string($connection, $address);

   $query = "INSERT INTO EMPLOYEES (NAME, ADDRESS) VALUES ('$n', '$a');";

   if(!mysqli_query($connection, $query)) echo("<p>Error adding employee data.</
p>");
}

/* Check whether the table exists and, if not, create it. */
function VerifyEmployeesTable($connection, $dbName) {
  if(!TableExists("EMPLOYEES", $connection, $dbName))
  {
     $query = "CREATE TABLE EMPLOYEES (
         ID int(11) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
         NAME VARCHAR(45),
         ADDRESS VARCHAR(90)
       )";

     if(!mysqli_query($connection, $query)) echo("<p>Error creating table.</
p>");
  }
}

/* Check for the existence of a table. */
function TableExists($tableName, $connection, $dbName) {
  $t = mysqli_real_escape_string($connection, $tableName);
  $d = mysqli_real_escape_string($connection, $dbName);

  $checktable = mysqli_query($connection,
      "SELECT TABLE_NAME FROM information_schema.TABLES WHERE TABLE_NAME = '$t'
 AND TABLE_SCHEMA = '$d'");

  if(mysqli_num_rows($checktable) > 0) return true;
```

```
  return false;
}
?>
```

PostgreSQL

```
<?php include "../inc/dbinfo.inc"; ?>

<html>
<body>
<h1>Sample page</h1>
<?php

/* Connect to PostgreSQL and select the database. */
$constring = "host=" . DB_SERVER . " dbname=" . DB_DATABASE . " user=" .
 DB_USERNAME . " password=" . DB_PASSWORD ;
$connection = pg_connect($constring);

if (!$connection){
 echo "Failed to connect to PostgreSQL";
 exit;
}

/* Ensure that the EMPLOYEES table exists. */
VerifyEmployeesTable($connection, DB_DATABASE);

/* If input fields are populated, add a row to the EMPLOYEES table. */
$employee_name = htmlentities($_POST['NAME']);
$employee_address = htmlentities($_POST['ADDRESS']);

if (strlen($employee_name) || strlen($employee_address)) {
  AddEmployee($connection, $employee_name, $employee_address);
}

?>

<!-- Input form -->
<form action="<?PHP echo $_SERVER['SCRIPT_NAME'] ?>" method="POST">
  <table border="0">
    <tr>
      <td>NAME</td>
      <td>ADDRESS</td>
```

```
    </tr>
    <tr>
      <td>
    <input type="text" name="NAME" maxlength="45" size="30" />
      </td>
      <td>
    <input type="text" name="ADDRESS" maxlength="90" size="60" />
      </td>
      <td>
    <input type="submit" value="Add Data" />
      </td>
    </tr>
  </table>
</form>
<!-- Display table data. -->
<table border="1" cellpadding="2" cellspacing="2">
  <tr>
    <td>ID</td>
    <td>NAME</td>
    <td>ADDRESS</td>
  </tr>

<?php

$result = pg_query($connection, "SELECT * FROM EMPLOYEES");

while($query_data = pg_fetch_row($result)) {
  echo "<tr>";
  echo "<td>",$query_data[0], "</td>",
       "<td>",$query_data[1], "</td>",
       "<td>",$query_data[2], "</td>";
  echo "</tr>";
}
?>
</table>

<!-- Clean up. -->
<?php

  pg_free_result($result);
  pg_close($connection);
?>
</body>
</html>
```

```
<?php

/* Add an employee to the table. */
function AddEmployee($connection, $name, $address) {
   $n = pg_escape_string($name);
   $a = pg_escape_string($address);
   echo "Forming Query";
   $query = "INSERT INTO EMPLOYEES (NAME, ADDRESS) VALUES ('$n', '$a');";

   if(!pg_query($connection, $query)) echo("<p>Error adding employee data.</
p>");
}

/* Check whether the table exists and, if not, create it. */
function VerifyEmployeesTable($connection, $dbName) {
  if(!TableExists("EMPLOYEES", $connection, $dbName))
  {
     $query = "CREATE TABLE EMPLOYEES (
         ID serial PRIMARY KEY,
         NAME VARCHAR(45),
         ADDRESS VARCHAR(90)
       )";

     if(!pg_query($connection, $query)) echo("<p>Error creating table.</p>");
  }
}
/* Check for the existence of a table. */
function TableExists($tableName, $connection, $dbName) {
  $t = strtolower(pg_escape_string($tableName)); //table name is case sensitive
  $d = pg_escape_string($dbName); //schema is 'public' instead of 'sample' db
 name so not using that

  $query = "SELECT TABLE_NAME FROM information_schema.TABLES WHERE TABLE_NAME =
 '$t';";
  $checktable = pg_query($connection, $query);

  if (pg_num_rows($checktable) >0) return true;
  return false;

}
?>
```

8. Salve e feche o arquivo `SamplePage.php`.
9. Verifique se o servidor Web se conecta com êxito à instância de banco de dados, abrindo um navegador da Web e navegando até `http://`*`EC2 instance endpoint`*`/SamplePage.php`, por exemplo: `http://ec2-12-345-67-890.us-west-2.compute.amazonaws.com/SamplePage.php`.

É possível usar `SamplePage.php` para adicionar dados à instância de banco de dados. Os dados que você adicionar serão exibidos na página. Para verificar se os dados foram inseridos na tabela, instale o cliente MySQL na instância do Amazon EC2. Depois, conecte-se à instância de banco de dados e consulte a tabela.

Para obter informações sobre como instalar o cliente MySQL e conectar-se a uma instância de banco de dados, consulte Conectar-se à instância de banco de dados do MySQL.

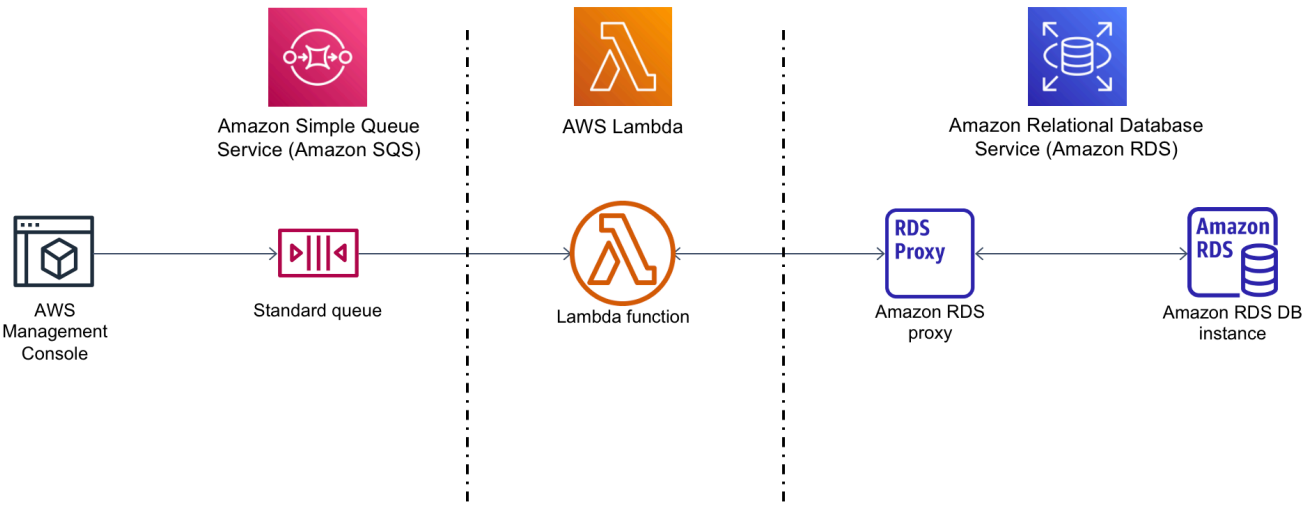
Para garantir que a instância de banco de dados seja o mais seguro possível, verifique se as fontes fora da VPC não podem se conectar à instância de banco de dados.

Depois de terminar de testar o servidor web e o banco de dados, é necessário excluir a instância de banco de dados e a instância do Amazon EC2.

- Para excluir uma instância de banco de dados, siga as instruções em Excluir uma instância de banco de dados. Não é necessário criar um snapshot final.
- Para encerrar uma instância do Amazon EC2, siga as instruções em Encerrar a instância no Guia do usuário do Amazon EC2.

Tutorial: Usar uma função do Lambda para acessar um banco de dados Amazon RDS

Neste tutorial, você usará uma função do Lambda para gravar dados em um banco de dados do Amazon Relational Database Service (Amazon RDS) por meio do RDS Proxy. Sua função do Lambda lê registros de uma fila do Amazon Simple Queue Service (Amazon SQS) e grava um novo item em uma tabela no banco de dados sempre que uma mensagem é adicionada. Neste exemplo, você usa o AWS Management Console para adicionar mensagens manualmente à sua fila. O diagrama a seguir mostrará os recursos da AWS que você usará para concluir o tutorial.



Com o Amazon RDS, você pode executar um banco de dados relacional gerenciado na nuvem usando produtos de banco de dados comuns, como Microsoft SQL Server, MariaDB, MySQL, Oracle Database e PostgresQL. Ao usar o Lambda para acessar seu banco de dados, você pode ler e gravar dados em resposta a eventos, como um novo cliente se registrando em seu site. Sua função, a instância de banco de dados e o proxy escalam automaticamente para atender a períodos de alta demanda.

Para concluir este tutorial, realize as seguintes tarefas:

1. Inicie uma instância de banco de dados do RDS para MySQL e um proxy na VPC padrão de sua Conta da AWS.

2. Crie e teste uma função do Lambda que cria uma nova tabela em seu banco de dados e grava dados nela.
3. Crie uma fila do Amazon SQS e configure-a para invocar sua função do Lambda sempre que uma nova mensagem for adicionada.
4. Teste a configuração completa adicionando mensagens à fila por meio do AWS Management Console e monitorando os resultados com o CloudWatch Logs.

Ao concluir essas etapas, você aprende a:

- Como usar o Amazon RDS para criar uma instância de banco de dados e um proxy e conectar uma função do Lambda ao proxy.
- Como usar o Lambda para realizar operações de criação e leitura em um banco de dados do Amazon RDS.
- Como usar o Amazon SQS para invocar uma função do Lambda.

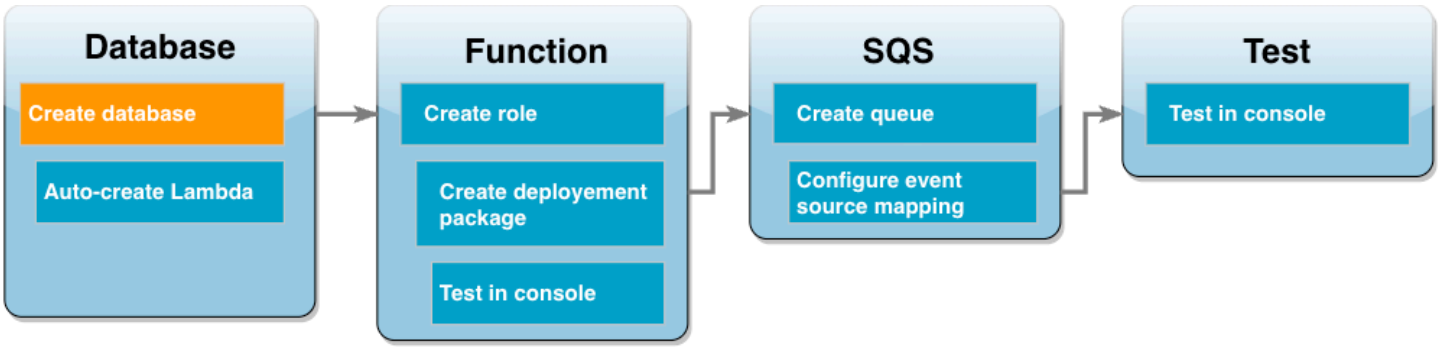
Você pode concluir este tutorial usando o AWS Management Console ou a AWS Command Line Interface (AWS CLI).

Pré-requisitos

Antes de começar, conclua as etapas nas seguintes seções:

- Inscrever-se para uma Conta da AWS
- Criar um usuário com acesso administrativo

Criar uma instância de banco de dados do Amazon RDS



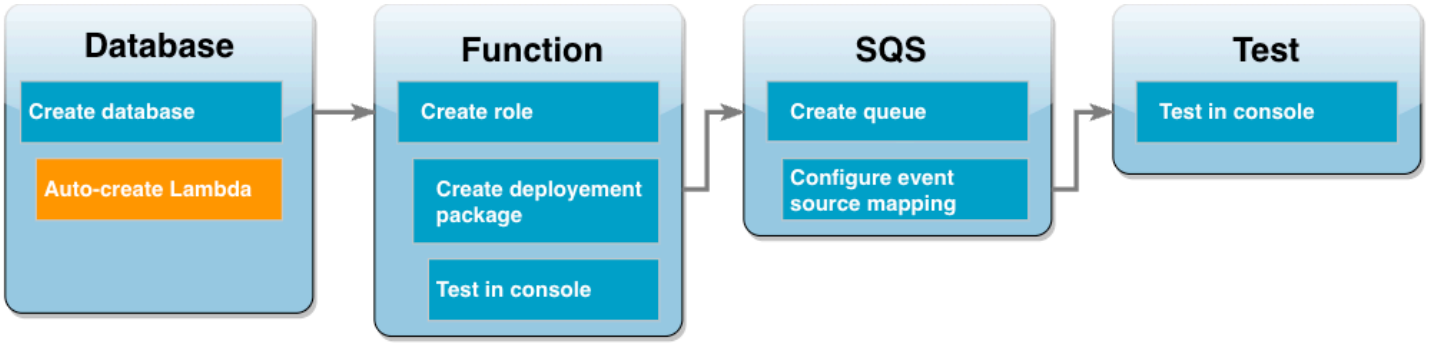
Uma instância de banco de dados do Amazon RDS é um ambiente de banco de dados isolado em execução na Nuvem AWS. Uma instância pode conter um ou mais bancos de dados criados pelo usuário. A menos que você especifique o contrário, o Amazon RDS cria instâncias de banco de dados na VPC padrão incluída em sua Conta da AWS. Para receber mais informações sobre o Amazon VPC, consulte o Guia do usuário do Amazon Virtual Private Cloud.

Neste tutorial, você cria uma nova instância na VPC padrão de sua Conta da AWS e cria um banco de dados com o nome `ExampleDB` nessa instância. Você pode criar o banco de dados e a respectiva instância usando o AWS Management Console ou a AWS CLI.

Como criar uma instância de banco de dados

1. No console do Amazon RDS, selecione Criar banco de dados.
2. Deixe a opção Criação padrão selecionada e, em Opções do mecanismo, escolha MySQL.
3. Em Modelos, escolha Nível gratuito ou Sandbox. O nível gratuito aparece para contas do nível gratuito. Sandbox aparece para contas do plano pago.
4. Em Configurações, para o identificador da instância de banco de dados, digite **`MySQLForLambda`**.
5. Para definir seu nome de usuário e senha, faça o seguinte:
   a. Em Configurações de credenciais, deixe o Nome do usuário principal definido como `admin`.
   b. Para a Senha mestra, digite e confirme uma senha para acessar seu banco de dados.
6. Para especificar o nome do banco de dados, faça o seguinte:
   - Deixe todas as opções padrão restantes selecionadas e role para baixo até a seção Configuração adicional.
   - Expanda essa seção e insira **`ExampleDB`** como o Nome do banco de dados inicial.
7. Deixe todas as opções padrão restantes selecionadas e escolha Criar banco de dados.

Criar a função do Lambda e o proxy



É possível usar o console do RDS para criar uma função do Lambda e um proxy na mesma VPC do banco de dados.

> **Note**
>
> Você só pode criar esses recursos associados quando o banco de dados tiver concluído a criação e estiver no status Disponível.

Como criar uma função e um proxy associados

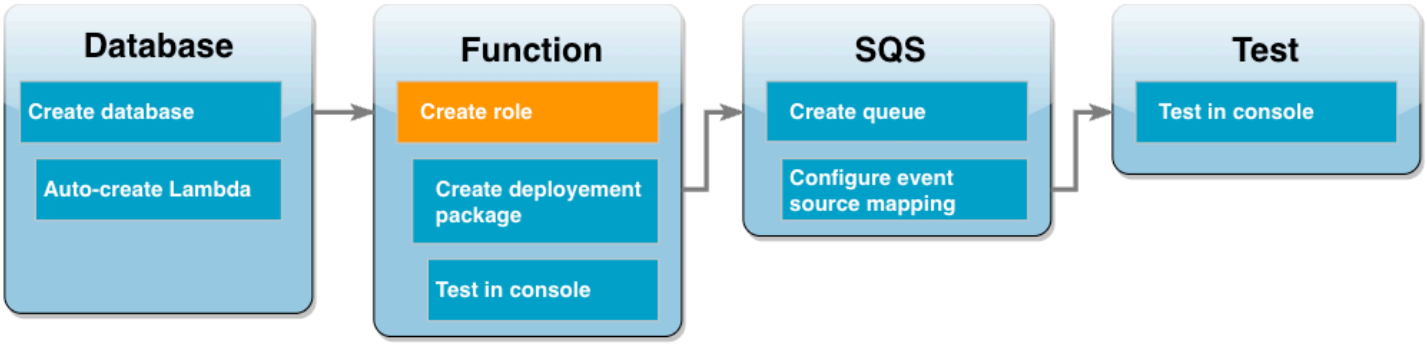
1. Na página Bancos de dados, confira se o banco de dados está no status Disponível. Se estiver, vá para a próxima etapa. Caso contrário, espere até que o banco de dados esteja disponível.
2. Selecione o banco de dados e escolha Configurar conexão do Lambda em Ações.
3. Na página Configurar conexão do Lambda, selecione Criar nova função.

   Defina o Nome da nova função do Lambda como **`LambdaFunctionWithRDS`**.
4. Na seção RDS Proxy, selecione a opção Conectar usando o RDS Proxy. Selecione ainda Criar proxy.

   - Em Credenciais do banco de dados, selecione Nome do usuário e senha do banco de dados.
   - Em Nome do usuário, especifique `admin`.
   - Em Senha, digite a senha que você criou para a instância de banco de dados.
5. Selecione Configurar para concluir a criação da função do Lambda e do proxy.

O assistente conclui a configuração e fornece um link para o console do Lambda para você examinar a nova função. Observe o endpoint do proxy antes de mudar para o console do Lambda.

Criar um perfil de execução de função



Antes de criar sua função do Lambda, você cria um perfil de execução para dar à sua função as permissões necessárias. Neste tutorial, o Lambda precisa de permissão para gerenciar a conexão de rede com a VPC que contém a instância de banco de dados e para pesquisar mensagens de uma fila do Amazon SQS.

Para dar à sua função do Lambda as permissões necessárias, este tutorial usa políticas gerenciadas pelo IAM. Essas são políticas que concedem permissões para muitos casos de uso comuns e estão disponíveis na sua Conta da AWS. Para obter mais informações sobre políticas gerenciadas, consulte Práticas recomendadas de política.

Para criar o perfil de execução do Lambda

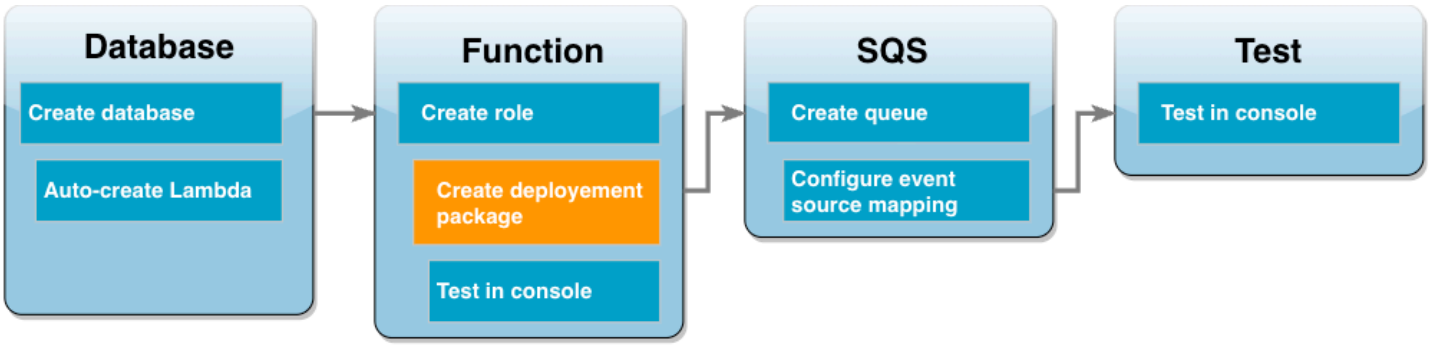
1. Abra a página Perfis do console do IAM e escolha Criar perfil.
2. Em Tipo de entidade confiável, selecione Serviço da AWS e, em Caso de uso, escolha Lambda.
3. Escolha Próximo.
4. Para adicionar as políticas gerenciadas do IAM, faça o seguinte:
    a. Usando a caixa de pesquisa de políticas, pesquise **`AWSLambdaSQSQueueExecutionRole`**.
    b. Na lista de resultados, marque a caixa de seleção ao lado do perfil e escolha Limpar filtros.
    c. Usando a caixa de pesquisa de políticas, pesquise **`AWSLambdaVPCAccessExecutionRole`**.
    d. Na lista de resultados, marque a caixa de seleção ao lado do perfil e escolha Avançar.
5. Em Nome do perfil, digite **`lambda-vpc-sqs-role`** e escolha Criar perfil.

Posteriormente no tutorial, você precisará do nome do recurso da Amazon (ARN) do perfil de execução que acabou de criar.

Para encontrar o ARN do perfil de execução

1. Abra a página Perfis do console do IAM e escolha o perfil (`lambda-vpc-sqs-role`).
2. Copie o ARN exibido na seção Resumo.

Criar o pacote de implantação do Lambda



O exemplo de código Python a seguir usa o pacote PyMySQL para abrir uma conexão com seu banco de dados. Na primeira vez que você invoca sua função, ela também cria uma nova tabela chamada `Customer`. A tabela usa o seguinte esquema, em que `CustID` é a chave primária:

```
Customer(CustID, Name)
```

A função também usa a PyMySQL para adicionar registros a essa tabela. A função adiciona registros usando os IDs de clientes e nomes especificados nas mensagens que você adicionará à sua fila do Amazon SQS.

O código cria a conexão com seu banco de dados fora da função do manipulador. A criação da conexão no código de inicialização permite que a conexão seja reutilizada por invocações subsequentes de sua função e melhora o desempenho. Em um aplicativo de produção, você também pode usar a simultaneidade provisionada para inicializar um número solicitado de conexões de banco de dados. Essas conexões estão disponíveis assim que sua função é invocada.

```
import sys
import logging
import pymysql
import json
import os

# rds settings
user_name = os.environ['USER_NAME']
```

```
password = os.environ['PASSWORD']
rds_proxy_host = os.environ['RDS_PROXY_HOST']
db_name = os.environ['DB_NAME']

logger = logging.getLogger()
logger.setLevel(logging.INFO)

# create the database connection outside of the handler to allow connections to be
# re-used by subsequent function invocations.
try:
        conn = pymysql.connect(host=rds_proxy_host, user=user_name, passwd=password,
 db=db_name, connect_timeout=5)
except pymysql.MySQLError as e:
    logger.error("ERROR: Unexpected error: Could not connect to MySQL instance.")
    logger.error(e)
    sys.exit(1)

logger.info("SUCCESS: Connection to RDS for MySQL instance succeeded")

def lambda_handler(event, context):
    """
    This function creates a new RDS database table and writes records to it
    """
    message = event['Records'][0]['body']
    data = json.loads(message)
    CustID = data['CustID']
    Name = data['Name']

    item_count = 0
    sql_string = f"insert into Customer (CustID, Name) values(%s, %s)"

    with conn.cursor() as cur:
        cur.execute("create table if not exists Customer ( CustID  int NOT NULL, Name
 varchar(255) NOT NULL, PRIMARY KEY (CustID))")
        cur.execute(sql_string, (CustID, Name))
        conn.commit()
        cur.execute("select * from Customer")
        logger.info("The following items have been added to the database:")
        for row in cur:
            item_count += 1
            logger.info(row)
    conn.commit()

    return "Added %d items to RDS for MySQL table" %(item_count)
```

> **Note**
>
> Neste exemplo, suas credenciais de acesso ao banco de dados são armazenadas como variáveis de ambiente. Em aplicações de produção, recomendamos que você use o AWS Secrets Manager como uma opção mais segura. Observe que, se sua função do Lambda estiver em uma VPC, para se conectar ao Secrets Manager, será necessário criar um endpoint da VPC. Consulte Como se conectar ao serviço Secrets Manager em uma nuvem privada virtual para saber mais.

Para incluir a dependência da PyMySQL em seu código de função, crie um pacote de implantação .zip. Os seguintes comandos funcionam para Linux, macOS ou Unix:

Para criar um pacote de implantação .zip

1. Salve o exemplo de código como um arquivo denominado `lambda_function.py`.
2. No mesmo diretório em que você criou seu arquivo `lambda_function.py`, crie um novo diretório chamado `package` e instale a biblioteca PyMySQL.

   ```
   mkdir package
   pip install --target package pymysql
   ```

3. Crie um arquivo zip contendo o código do seu aplicativo e a biblioteca PyMySQL. No Linux ou no macOS, execute os comandos da CLI a seguir. No Windows, use sua ferramenta de zip preferida para criar o arquivo `lambda_function.zip`. O arquivo de código-fonte `lambda_function.py` e as pastas que contêm as dependências devem ser instaladas na raiz do arquivo .zip.

   ```
   cd package
   zip -r ../lambda_function.zip .
   cd ..
   zip lambda_function.zip lambda_function.py
   ```

   Você também pode criar seu pacote de implantação usando um ambiente virtual Python. Consulte Implementar funções do Lambda em Python com arquivos .zip.

Atualizar a função do Lambda

Com o pacote .zip que acabou de criar, você atualizará a função do Lambda usando o console do Lambda. Para permitir que a função acesse o banco de dados, você também precisa configurar variáveis de ambiente com as credenciais de acesso.

Como atualizar a função do Lambda

1. Abra a página Funções do console do Lambda e selecione sua função `LambdaFunctionWithRDS`.
2. Na guia Configurações de tempo de execução, selecione Editar para alterar o Tempo de execução da função para Python 3.10.
3. Altere o Manipulador para `lambda_function.lambda_handler`.
4. Na guia Código, selecione Fazer upload de e, depois, arquivo .zip.
5. Selecione o arquivo `lambda_function.zip` que você criou no estágio anterior e escolha Salvar.

Agora configure a função com o perfil de execução que você criou anteriormente. Isso concede à função as permissões necessárias para acessar sua instância de banco de dados e pesquisar uma fila do Amazon SQS.

Como configurar o perfil de execução da função.

1. Na página Funções do console do Lambda, selecione a guia Configuração e escolha Permissões.
2. No painel Papel de execução, selecione Editar.
3. Em Função existente, selecione a função de execução (`lambda-vpc-sqs-role`).
4. Escolha Salvar.

Como configurar as variáveis de ambiente da função

1. Na página Funções do console do Lambda, selecione a guia Configuração e, em seguida, escolha Variáveis de ambiente.
2. Escolha Editar.
3. Para adicionar as credenciais de acesso ao banco de dados, faça o seguinte:

a. Escolha Adicionar variável de ambiente e, em seguida, para Chave, insira **`USER_NAME`**, e para Valor, insira **`admin`**.

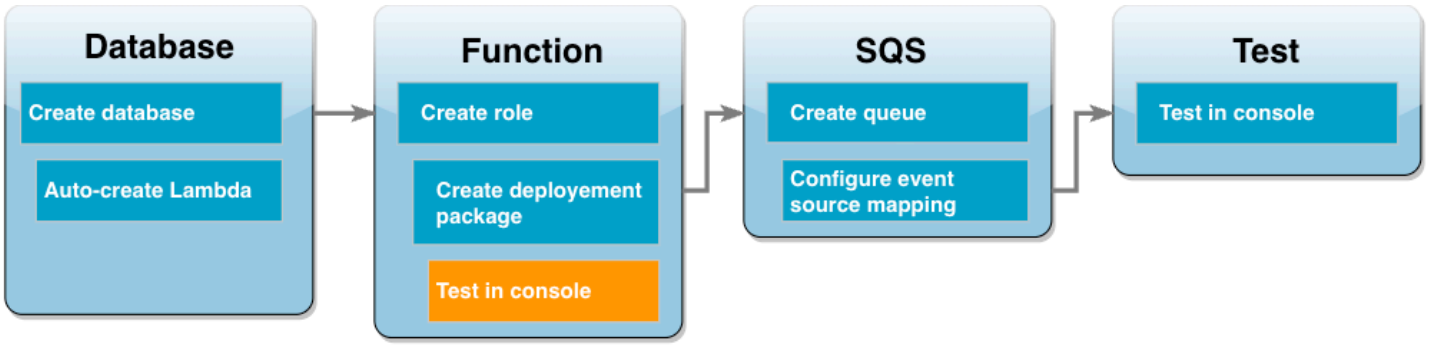
b. Escolha Adicionar variável de ambiente e, em seguida, para Chave, insira **`DB_NAME`**, e para Valor, insira **`ExampleDB`**.

c. Escolha Adicionar variável de ambiente e, em seguida, para Chave, insira **`PASSWORD`**, e para Valor, insira a senha escolhida ao criar seu banco de dados.

d. Selecione Adicionar variáveis de ambiente. Em Chave, insira **`RDS_PROXY_HOST`** e, em Valor, insira o endpoint do RDS Proxy que você anotou anteriormente.

e. Escolha Salvar.

Teste sua função do Lambda no console



Agora você pode usar o console do Lambda para testar sua função. Você cria um evento de teste que imita os dados que sua função receberá ao invocá-la usando o Amazon SQS na fase final do tutorial. Seu evento de teste contém um objeto JSON especificando um ID de cliente e um nome de cliente para adicionar à tabela `Customer` que sua função cria.

Para testar a função do Lambda

1. Abra a página Funções do console do Lambda e escolha sua função.
2. Selecione a seção Testar.
3. Selecione Criar novo evento e insira **`myTestEvent`** como nome do evento.
4. Copie o código a seguir no JSON do evento e escolha Salvar.

```
{
  "Records": [
    {
      "messageId": "059f36b4-87a3-44ab-83d2-661975830a7d",
```

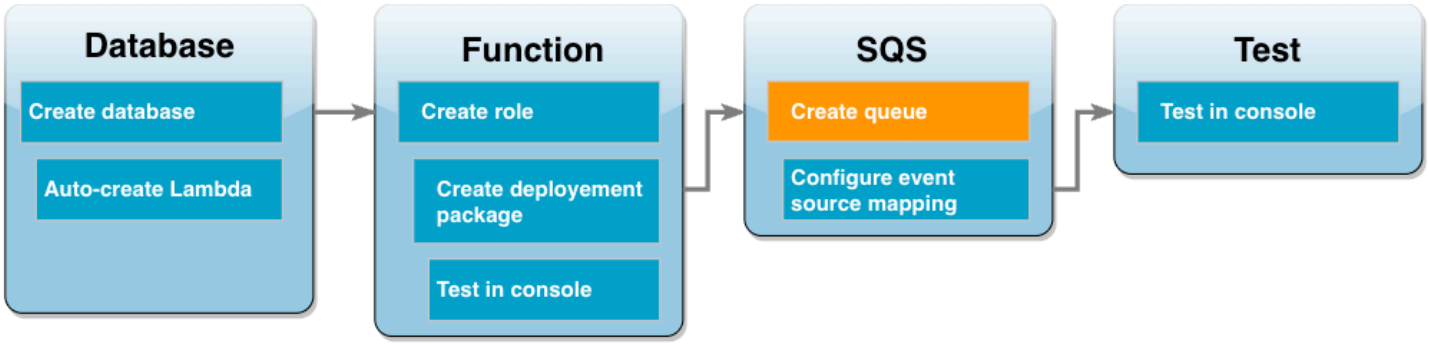
```
      "receiptHandle": "AQEBwJnKyrHigUMZj6rYigCgxlaS3SLy0a...",
      "body": "{\n     \"CustID\": 1021,\n     \"Name\": \"Martha Rivera\"\n}",
      "attributes": {
        "ApproximateReceiveCount": "1",
        "SentTimestamp": "1545082649183",
        "SenderId": "AIDAIENQZJOLO23YVJ4VO",
        "ApproximateFirstReceiveTimestamp": "1545082649185"
      },
      "messageAttributes": {},
      "md5OfBody": "e4e68fb7bd0e697a0ae8f1bb342846b3",
      "eventSource": "aws:sqs",
      "eventSourceARN": "arn:aws:sqs:us-west-2:123456789012:my-queue",
      "awsRegion": "us-west-2"
    }
  ]
}
```

5. Escolha Testar.

Na guia Resultados da execução, você verá resultados semelhantes aos seguintes exibidos nos Logs da função:

```
[INFO] 2023-02-14T19:31:35.149Z bdd06682-00c7-4d6f-9abb-89f4bbb4a27f The following
 items have been added to the database:
[INFO] 2023-02-14T19:31:35.149Z bdd06682-00c7-4d6f-9abb-89f4bbb4a27f (1021, 'Martha
 Rivera')
```

Criar uma fila do Amazon SQS

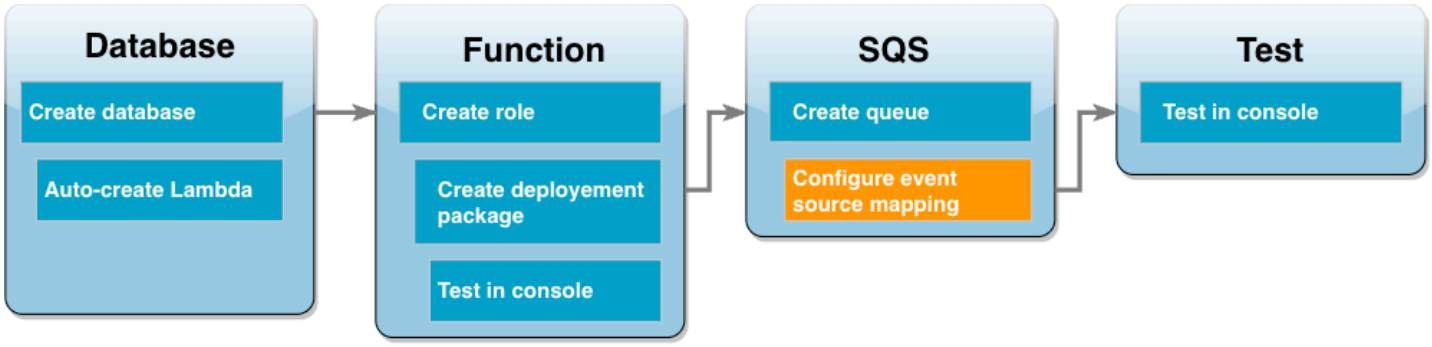


Você testou com sucesso a integração da sua função do Lambda e da instância do banco de dados do Amazon RDS. Agora você cria a fila do Amazon SQS que usará para invocar sua função do Lambda na fase final do tutorial.

Para criar uma fila do Amazon SQS (console)

1. Abra a página Filas do console do Amazon SQS e selecione Criar fila.
2. Deixe o Tipo como Padrão e insira **`LambdaRDSQueue`** como o nome da sua fila.
3. Mantenha todas as opções padrão selecionadas e escolha Criar fila.

Crie um mapeamento da origem do evento para invocar sua função do Lambda



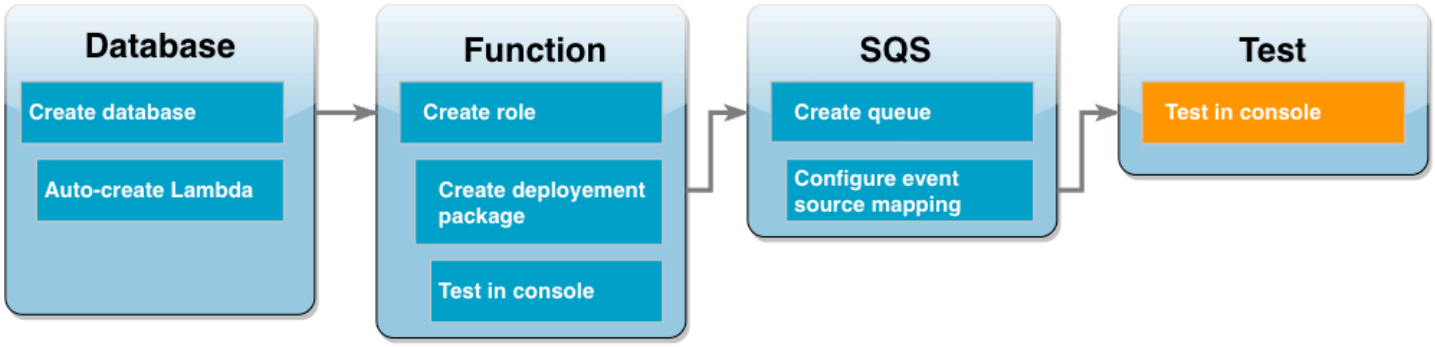
Um mapeamento da origem do evento é um recurso no Lambda que lê itens de um fluxo ou de uma fila e invoca uma função do Lambda. Ao configurar um mapeamento da origem do evento, você pode especificar um tamanho de lote para que os registros do seu fluxo ou da sua fila sejam agrupados em uma única carga útil. Neste exemplo, você define o tamanho do lote como 1 para que sua função do Lambda seja invocada toda vez que você enviar uma mensagem para sua fila. Você pode configurar o mapeamento da origem do evento usando a AWS CLI ou o console do Lambda.

Crie um mapeamento da origem do evento (console)

1. Abra a página Funções do console do Lambda e escolha sua função (`LambdaFunctionWithRDS`).
2. Na seção Visão geral da função, selecione Adicionar gatilho.
3. Para a origem, selecione Amazon SQS e, depois, o nome da fila (`LambdaRDSQueue`).
4. Em Tamanho do lote, insira **1**.
5. Deixe todas as outras opções definidas com os valores padrão e escolha Adicionar.

Agora você está pronto para testar sua configuração completa adicionando uma mensagem à sua fila do Amazon SQS.

Teste e monitore sua configuração



Para testar sua configuração completa, adicione mensagens à sua fila do Amazon SQS usando o console. Em seguida, você usa o CloudWatch Logs para confirmar que sua função do Lambda está gravando registros em seu banco de dados conforme o esperado.

Para testar e monitorar sua configuração

1. Abra a página Filas do console do Amazon SQS e selecione sua fila (`LambdaRDSQueue`).
2. Selecione Enviar e receber mensagens e cole o JSON a seguir no Corpo da mensagem na seção Enviar mensagem.

   ```
   {
       "CustID": 1054,
       "Name": "Richard Roe"
   }
   ```

3. Escolha Send Message (Enviar mensagem).

   Enviar sua mensagem para a fila fará com que o Lambda invoque sua função por meio do mapeamento da origem do evento. Para confirmar que o Lambda invocou sua função conforme o esperado, use o CloudWatch Logs para verificar se a função gravou o nome e o ID do cliente na tabela do banco de dados:

4. Abra a página Grupos de logs do console do CloudWatch e selecione o grupo de logs para sua função (`/aws/lambda/LambdaFunctionWithRDS`).
5. No painel Fluxos de logs, selecione o fluxo de logs mais recente.

   Sua tabela deve conter dois registros de clientes, um de cada invocação de sua função. No fluxo de logs, você deverá ver mensagens semelhantes às seguintes:

```
[INFO] 2023-02-14T19:06:43.873Z 45368126-3eee-47f7-88ca-3086ae6d3a77 The following
 items have been added to the database:
[INFO] 2023-02-14T19:06:43.873Z 45368126-3eee-47f7-88ca-3086ae6d3a77 (1021, 'Martha
 Rivera')
[INFO] 2023-02-14T19:06:43.873Z 45368126-3eee-47f7-88ca-3086ae6d3a77 (1054,
 'Richard Roe')
```

Limpe os recursos

Agora você pode excluir os recursos criados para este tutorial, a menos que queira mantê-los. Excluindo os recursos da AWS que você não está mais usando, você evita cobranças desnecessárias em sua conta da AWS.

Como excluir a função do Lambda

1. Abra a página Functions (Funções) no console do Lambda.
2. Selecione a função que você criou.
3. Selecione Ações, Excluir.
4. Escolha Excluir.

Para excluir a função de execução

1. Abra a página Roles (Funções) no console do IAM.
2. Selecione a função de execução que você criou.
3. Clique em Excluir função.
4. Depois em Sim, excluir.

Para excluir a instância de banco de dados MySQL do

1. Abrir oPágina Bancos de dadosdo console do Amazon RDS.
2. Selecione o banco de dados que você criou.
3. Selecione Ações, Excluir.
4. Desmarque a caixa de seleção Create final snapshot (Criar snapshot final).
5. Digite **`delete me`** na caixa de texto.

6. Escolha Excluir.

Para excluir a fila do Amazon SQS

1. Faça login no AWS Management Console e abra o console do Amazon SQS em https://console.aws.amazon.com/sqs/.
2. Selecione a fila que você criou.
3. Escolha Excluir.
4. Digite **`delete`** na caixa de texto.
5. Escolha Excluir.

Tutoriais do Amazon RDS e código de exemplo

A documentação da AWS contém vários tutoriais que guiarão você por meio de casos de uso comuns do Amazon RDS. Muitos desses tutoriais mostram como usar o Amazon RDS com outros serviços da AWS. Além disso, é possível acessar o código de exemplo no GitHub.

> **Note**
>
> Você pode encontrar mais tutoriais no AWS Database Blog. Para ter mais informações sobre treinamento, consulte AWS Training and Certification.

Tópicos

- Tutoriais neste guia
- Tutoriais em outros guias daAWS
- Portal de conteúdo de workshops e laboratório da AWS para Amazon RDS PostgreSQL
- Portal de conteúdo de workshops e laboratório da AWS para Amazon RDS MySQL
- Tutoriais e código de exemplo no GitHub
- Usar este serviço com um AWS SDK

Tutoriais neste guia

Os seguintes tutoriais neste guia mostram como realizar tarefas comuns ao usar o Amazon RDS:

- Tutorial: Criar uma VPC para usar com uma instância de banco de dados (somente IPv4)

  Saiba como incluir uma instância de banco de dados em uma nuvem privada virtual (VPC) com base no serviço Amazon VPC. Nesse caso, a VPC compartilha dados com um servidor da Web em execução em uma instância do Amazon EC2 na mesma VPC.
- Tutorial: Criar uma VPC para uso com uma instância de banco de dados (modo de pilha dupla)

  Saiba como incluir uma instância de banco de dados em uma nuvem privada virtual (VPC) com base no serviço Amazon VPC. Nesse caso, a VPC compartilha dados com uma instância do Amazon EC2 na mesma VPC. Neste tutorial, você criará a VPC para esse cenário que funciona com um banco de dados em execução no modo de pilha dupla.

- Tutorial: crie um servidor Web e uma instância de banco de dados do Amazon RDS

  Este tutorial ajuda você a instalar um servidor Web Apache com PHP e a criar um banco de dados MySQL. O servidor da web é executado em uma instância do Amazon EC2 usando o Amazon Linux, e o banco de dados MySQL é uma instância de banco de dados MySQL. Tanto a instância do Amazon EC2 quanto o da instância de banco de dados são executados em uma Amazon VPC.
- Tutorial: restaurar uma instância de banco de dados do Amazon RDS de um snapshot do banco de dados

  Para restaurar uma instância de banco de dados de um snapshot de banco de dados.
- Tutorial: Usar uma função do Lambda para acessar um banco de dados Amazon RDS

  Saiba como criar uma função do Lambda por meio do console do RDS para acessar um banco de dados usando um proxy, criar uma tabela, adicionar alguns registros e recuperá-los da tabela. Você também aprenderá a invocar a função do Lambda e verificar os resultados da consulta.
- Tutorial: especificar quais instâncias de banco de dados devem ser interrompidas usando tags

  Saiba como usar tags para especificar quais instâncias de banco de dados devem ser interrompidas.
- Tutorial: Registro de alterações de estado de uma instância de banco de dados usando o Amazon EventBridge

  Saiba como registrar uma alteração de estágio de instância de banco de dados usando o Amazon EventBridge e AWS Lambda.
- Tutorial: criar um alarme do Amazon CloudWatch para atraso de réplica do cluster de banco de dados multi-AZ para o Amazon RDS

  Aprenda a criar um alarme do CloudWatch que envia uma mensagem do Amazon SNS quando o atraso de réplica para um cluster de banco de dados multi-AZ excede um limite. Um alarme observa uma métrica `ReplicaLag` ao longo do período que você especificar. A ação é uma notificação enviada para um tópico do Amazon SNS ou por uma política do Amazon EC2 Auto Scaling.

Tutoriais em outros guias daAWS

Os seguintes tutoriais em outros guias da AWS mostram como realizar tarefas comuns com o Amazon RDS:

- Tutorial: Alternar um segredo para um banco de dados da AWS no Manual do usuário do AWS Secrets Manager

  Neste tutorial, você criará um segredo para um banco de dados da AWS e configurará o banco de dados para alternar em um cronograma. Você acionará uma rotação manualmente e confirmará se a nova versão do segredo continua fornecendo acesso.

- Tutoriais e amostras no Guia do desenvolvedor do AWS Elastic Beanstalk

  Saiba como implantar aplicações que usam bancos de dados do Amazon RDS com o AWS Elastic Beanstalk.

- Uso dos dados de um banco de dados Amazon RDS para criar uma fonte de dados do Amazon ML no Amazon Machine Learning Developer Guide

  Aprenda a criar um objeto de fonte de dados do Amazon Machine Learning (Amazon ML) a partir de dados armazenados em uma instância de banco de dados MySQL.

- Habilitação manual de acesso a uma instância do Amazon RDS em uma VPC no Manual do usuário do Amazon QuickSight

  Saiba como habilitar o acesso do QuickSight a uma instância de banco de dados do Amazon RDS em uma VPC.

Portal de conteúdo de workshops e laboratório da AWS para Amazon RDS PostgreSQL

A seguinte coleção de workshops e outros conteúdos práticos ajudam você a entender os recursos e as capacidades do Amazon RDS PostgreSQL:

- Criar uma instância de banco de dados

  Saiba como criar a instância de banco de dados.

- Monitoramento de performance com ferramentas do RDS

  Saiba como utilizar a AWS e as ferramentas de SQL (Cloudwatch, Enhanced Monitoring, Slow Query Logs, Performance Insights, PostgreSQL Catalog Views) para entender os problemas de performance e identificar formas de melhorar a performance de seu banco de dados.

Portal de conteúdo de workshops e laboratório da AWS para Amazon RDS MySQL

A seguinte coleção de workshops e outros conteúdos práticos ajudam você a entender os recursos e as capacidades do Amazon RDS MySQL:

- Criar uma instância de banco de dados

  Saiba como criar a instância de banco de dados.

- Utilizar o Performance Insights

  Saiba como monitorar e ajustar sua instância de banco de dados utilizando o Performance Insights.

Tutoriais e código de exemplo no GitHub

Os seguintes tutoriais e código de exemplo do GitHub mostram como realizar tarefas comuns ao usar o Amazon RDS:

- Como criar o rastreador de item do Amazon Relational Database Service

  Saiba como criar uma aplicação que monitora e gera relatórios sobre itens de trabalho. Essa aplicação usa o Amazon RDS, o Amazon Simple Email Service, o Elastic Beanstalk e o SDK for Java 2.x.

Usar este serviço com um AWS SDK

Os kits de desenvolvimento de software (SDKs) da AWS estão disponíveis para muitas linguagens de programação populares. Cada SDK fornece uma API, exemplos de código e documentação que permitem que os desenvolvedores criem facilmente aplicações em seu idioma de preferência.

Documentação do SDK	Exemplos de código
AWS SDK para C++	Exemplos de código do AWS SDK para C++
AWS CLI	Exemplos de código do AWS CLI

Documentação do SDK	Exemplos de código
AWS SDK para Go	Exemplos de código do AWS SDK para Go
AWS SDK para Java	Exemplos de código do AWS SDK para Java
AWS SDK para JavaScript	Exemplos de código do AWS SDK para JavaScript
AWS SDK para Kotlin	Exemplos de código do AWS SDK para Kotlin
AWS SDK para .NET	Exemplos de código do AWS SDK para .NET
AWS SDK para PHP	Exemplos de código do AWS SDK para PHP
Ferramentas da AWS para PowerShell	Exemplos de código do Ferramentas da AWS para PowerShell
AWS SDK para Python (Boto3)	Exemplos de código do AWS SDK para Python (Boto3)
AWS SDK para Ruby	Exemplos de código do AWS SDK para Ruby
AWS SDK para Rust	Exemplos de código do AWS SDK para Rust
SDK da AWS para SAP ABAP	Exemplos de código do SDK da AWS para SAP ABAP
AWS SDK for Swift	Exemplos de código do AWS SDK for Swift

Para obter exemplos específicos deste serviço, consulte Exemplos de código para o Amazon RDS usando AWS SDKs.

> **Exemplo de disponibilidade**
>
> Não consegue encontrar o que precisa? Solicite um exemplo de código usando o link Fornecer feedback na parte inferior desta página.

Práticas recomendadas do Amazon RDS

Conheça as práticas recomendadas para trabalhar com o Amazon RDS. Conforme são identificadas novas práticas recomendadas, manteremos esta seção atualizada.

Tópicos

- Diretrizes operacionais básicas do Amazon RDS
- Recomendações de RAM para a instância de banco de dados
- Mantenha as versões do mecanismo de banco de dados atualizadas
- Drivers de banco de dados da AWS
- Uso do monitoramento avançado para identificar problemas do sistema operacional
- Uso de métricas para identificar problemas de performance
- Ajuste das consultas
- Práticas recomendadas para trabalhar com o MySQL
- Práticas recomendadas para trabalhar com o MariaDB
- Práticas recomendadas para trabalhar com o Oracle
- Práticas recomendadas para trabalhar com PostgreSQL
- Práticas recomendadas para trabalhar com o SQL Server
- Trabalhar com grupos de parâmetros de banco de dados
- Práticas recomendadas para automatizar a criação de instâncias de banco de dados
- Vídeo de novos recursos do Amazon RDS

> **Note**
>
> Para reconhecer as recomendações comuns do Amazon RDS, consulte Recomendações do Amazon RDS.

Diretrizes operacionais básicas do Amazon RDS

As diretrizes operacionais básicas a seguir devem ser seguidas por todos ao trabalhar com o Amazon RDS. Observe que o Acordo de Nível de Serviço do Amazon RDS exige que você siga essas diretrizes:

- Use métricas para monitorar sua memória, CPU, atraso de réplica e uso de armazenamento. É possível configurar o Amazon CloudWatch para notificar você quando os padrões de uso mudam ou quando a implantação se aproxima dos limites de capacidade. Isso permite manter a disponibilidade e o desempenho do sistema.
- Escale sua instância de banco de dados quando estiver se aproximando dos limites de capacidade de armazenamento. É preciso ter algum buffer de armazenamento e memória para acomodar aumentos imprevistos na demanda de seus aplicativos.
- Habilite backups automáticos e configure a janela de backup para ocorrer durante a baixa diária em IOPS de gravação. É quando um backup causa menos interrupções ao uso do banco de dados.
- Se a workload do banco de dados exigir mais E/S do que você provisionou, a recuperação após um failover ou a falha no banco de dados será lenta. Para aumentar a capacidade de E/S de uma instância de banco de dados, realize uma ou todas as ações a seguir:
  - Migre para uma classe de instância de banco de dados diferente, com alta capacidade de E/S.
  - Converta o armazenamento magnético de uso geral ou armazenamento de IOPS provisionadas, dependendo de quanto é necessário aumentar. Para obter informações sobre os tipos de armazenamento disponíveis, consulte Tipos de armazenamento do Amazon RDS.

    Se você converter para o armazenamento de IOPS provisionadas, certifique-se usar também uma classe de instância de banco de dados otimizada para IOPS provisionadas. Para obter informações sobre IOPS provisionadas, consulte Armazenamento SSD de IOPS provisionadas.
  - Se você já estiver usando armazenamento de IOPS provisionadas, forneça a capacidade de transferência adicional.
- Se o seu aplicativo cliente estiver armazenando em cache os dados do Serviço de Nome de Domínio (DNS) de suas instâncias de banco de dados, defina um valor de tempo de vida (TTL) de menos de 30 segundos. O endereço IP subjacente de uma instância de banco de dados pode ser alterado após um failover. Dessa forma, armazenar os dados do DNS em cache por um longo período pode ocasionar falhas de conexão. Sua aplicação pode tentar se conectar a um endereço IP que não está mais em serviço.
- Teste o failover da instância de banco de dados para entender quanto tempo o processo leva para seu caso de uso específico. Além disso, teste o failover para garantir que a aplicação que acessa sua instância de banco de dados possa se conectar automaticamente à nova instância de banco de dados após o failover.

Recomendações de RAM para a instância de banco de dados

Uma das práticas recomendadas de performance do Amazon RDS é alocar RAM suficiente para que seu conjunto de trabalho resida quase que completamente na memória. O conjunto de trabalho é composto de dados e índices que são usados frequentemente em sua instância. Quanto mais você usar a instância de banco de dados, mais o conjunto de trabalho crescerá.

Para saber se o seu conjunto de trabalho está quase todo na memória, verifique a métrica ReadIOPS (usando o Amazon CloudWatch) enquanto a instância de banco de dados estiver sob carga. O valor de ReadIOPS deve ser pequeno e estável. Em alguns casos, aumentar a escala verticalmente da classe de instância de banco de dados para uma classe com mais RAM ocasiona uma queda dramática no ReadIOPS. Nesses casos, seu conjunto de trabalho não estava totalmente na memória. Continue a escalar até que a ReadIOPS não caia mais drasticamente após uma operação de escalabilidade, ou a ReadIOPS será reduzida a uma quantidade muito pequena. Para obter informações sobre como monitorar as métricas da instância de banco de dados, consulte Visualizar métricas no console do Amazon RDS.

Mantenha as versões do mecanismo de banco de dados atualizadas

Atualize regularmente a versão do mecanismo de banco de dados para manter a segurança, o desempenho e a conformidade. O Amazon RDS lança novas versões secundárias e principais que incluem patches de segurança, aprimoramentos de desempenho e novos recursos. A execução de um mecanismo de banco de dados desatualizado pode expor suas workloads a vulnerabilidades conhecidas, problemas de compatibilidade e suporte reduzido da AWS e dos fornecedores de bancos de dados.

Para minimizar a interrupção, considere o seguinte ao planejar as atualizações:

- Teste em um ambiente de preparação: valide a nova versão em relação à sua workload antes de atualizar os bancos de dados de produção.
- Use as atualizações gerenciadas pelo Amazon RDS: habilite atualizações automáticas de versões secundárias para facilitar a aplicação de patches.
- Programe as atualizações de versão principal: examine as notas de lançamento, teste a compatibilidade das aplicações e planeje uma janela de atualização controlada.

Atualizações regulares ajudam a garantir que seu banco de dados permaneça seguro, otimizado e alinhado com as práticas recomendadas da AWS.

Drivers de banco de dados da AWS

Recomendamos o pacote de drivers da AWS para conectividade de aplicações. Os drivers foram projetados para comportar tempos mais rápidos de transição e failover, além de autenticação com o AWS Secrets Manager, o AWS Identity and Access Management (IAM) e identidades federadas. Os drivers da AWS dependem do monitoramento do status da instância de banco de dados e do conhecimento da topologia da instância para determinar o novo gravador. Essa abordagem reduz os tempos de transição e de failover para segundos de um dígito, em comparação com dezenas de segundos para drivers de código aberto.

Como novos recursos do serviço são introduzidos, o objetivo do pacote de drivers da AWS é ter suporte integrado para esses recursos do serviço.

Para obter mais informações, consulte Conectar-se a instâncias de banco de dados com os drivers da AWS.

Uso do monitoramento avançado para identificar problemas do sistema operacional

Quando o monitoramento avançado está habilitado, o Amazon RDS fornece métricas em tempo real para o sistema operacional (SO) no qual a instância de banco de dados é executada. É possível visualizar as métricas para sua instância de banco de dados usando o console. Além disso, é possível consumir o resultado do JSON de monitoramento avançado do Amazon CloudWatch Logs em um sistema de monitoramento de sua escolha. Para obter mais informações sobre o monitoramento avançado, consulte Monitorar métricas do SO com o monitoramento avançado.

Uso de métricas para identificar problemas de performance

Para identificar problemas de performance causados por recursos insuficientes e outros gargalos comuns, você pode monitorar as métricas disponíveis para a instância de banco de dados do Amazon RDS.

Visualização de métricas de performance

Você deve monitorar as métricas de performance regularmente para ver os valores médio, máximo e mínimo de uma série de intervalos de tempo. Fazendo isso, você pode identificar quando a performance está degradado. Você também pode definir alarmes do Amazon CloudWatch para limites métricos específicos para que você seja alertado se eles forem atingidos.

Para solucionar problemas de performance, é importante entender a performance de linha de base do sistema. Ao configurar uma instância de banco de dados e executá-la com uma workload típica, capture os valores médio, máximo e mínimo de todas as métricas de performance. Faça isso em vários intervalos diferentes (por exemplo, uma hora, 24 horas, uma semana, duas semanas). Isso poder dar a você uma ideia do que é normal. Isso ajuda a obter comparações para as horas de operação de pico e fora de pico. Você pode usar essas informações para identificar quando a performance está ficando abaixo dos níveis padrão.

Se você usar clusters de banco de dados multi-AZ, monitore a diferença de tempo entre a transação mais recente na instância de banco de dados do gravador e a transação aplicada mais recente em uma instância de banco de dados do leitor. Essa diferença é chamada de atraso de réplica. Para obter mais informações, consulte Atraso de réplica e clusters de banco de dados multi-AZ.

É possível visualizar as métricas combinadas do Insights de Performance e do CloudWatch no painel do Insights de Performance e monitorar a instância de banco de dados. Para usar essa visualização de monitoramento, o Insights de Performance deve estar ativado para a instância de banco de dados. Para obter mais informações sobre essa visualização de monitoramento, consulte Visualizar métricas combinadas com o painel do Insights de Performance.

Você pode criar um relatório de análise de performance para um período específico e visualizar os insights identificados e as recomendações para resolver os problemas. Consulte mais informações em Criar um relatório de análise de performance no Insights de Performance.

Para visualizar as métricas de performance

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Databases (Bancos de dados) e uma instância de banco de dados.
3. Escolha Monitoring.

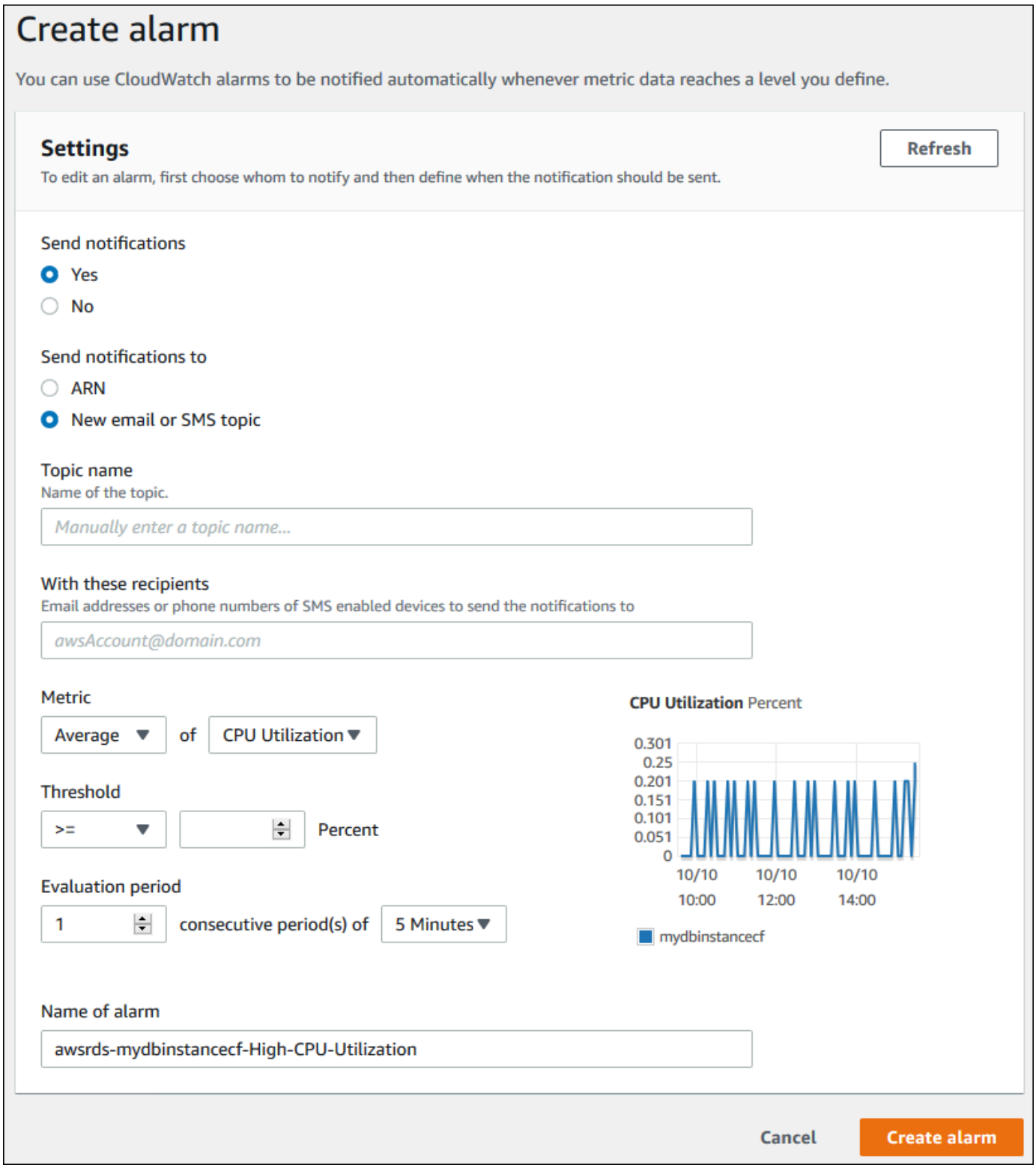
O painel fornece as métricas de performance. Por padrão, as métricas mostram as informações das últimas três horas.

4. Use os botões numerados no canto superior direito da página por meio das métricas adicionais ou ajuste as configurações para ver mais métricas.
5. Escolha uma métrica de performance para ajustar o período para ver os dados além do dia atual. Você pode alterar os valores de Statistic (Estatística), Time Range (Intervalo de tempo) e Period (Período) para ajustar as informações exibidas. Por exemplo, talvez você queira ver os valores de pico de uma métrica para cada dia das últimas duas semanas. Em caso afirmativo, defina Statistic (Estatística) como Maximum (Máximo), Time Range (Intervalo de Tempo) como Last 2 Weeks (Últimas duas semanas) e Period (Período) como Day (Dia).

Você também pode visualizar métricas de performance usando a CLI ou a API. Para obter mais informações, consulte Visualizar métricas no console do Amazon RDS.

Para definir um alarme do CloudWatch

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Databases (Bancos de dados) e uma instância de banco de dados.
3. Escolha Logs & events (Logs e eventos).
4. Na seção CloudWatch alarms (Alarmes do CloudWatch), escolha Create alarm (Criar alarme).



5. Em Send notifications (Enviar notificações), escolha Yes (Sim), e em Send notifications to (Enviar notificações para), escolha New email or SMS topic (Novo tópico de email ou SMS).

6. Em Topic name (Nome do tópico), digite um nome para a notificação e em With these recipients (Com esses destinatários), digite uma lista separada por vírgulas de endereços de e-mail e números de telefone.
7. Em Metric (Métrica), escolha a estatística e a métrica de alarme a serem definidas.
8. Em Threshold (Limite), especifique se a métrica deve ser maior que, menor que ou igual ao limite e especifique o valor do limite.
9. Em Evaluation period (Período de avaliação), selecione o período de avaliação do alarme. Em consecutive period(s) of (períodos consecutivos de), selecione o período durante o qual o limite deve ter sido atingido para acionar o alarme.
10. Em Name of alarm (Nome de alarme), digite um nome para o alarme.
11. Escolha Create Alarm.

O alarme aparece na seção CloudWatch alarms (Alarmes do CloudWatch).

Avaliação de métricas de performance

Uma instância de banco de dados apresenta várias categorias de métricas diferentes, e determinar valores aceitáveis depende de cada métrica.

CPU

- Utilização da CPU – porcentagem da capacidade de processamento computacional utilizada.

Memória

- Memória disponível: a quantidade de RAM disponível na instância de banco de dados, em bytes. A linha vermelha nas métricas da guia Monitoring (Monitoramento) é marcada em 75% para CPU, memória e métricas de armazenamento. Se o consumo de memória da instância cruzar essa linha com frequência, isso indica que é necessário verificar sua workload ou atualizar sua instância.
- Uso de troca: quanto espaço de troca é usado pela instância do banco de dados, em bytes.

Espaço em disco

- Espaço de armazenamento gratuito – quanto espaço de disco não está sendo usado atualmente pela instância de banco de dados, em megabytes.

Operações de entrada/saída

- IOPS de leitura, IOPS de gravação – o número médio de operações de leitura ou gravação de disco por segundo.
- Latência de leitura, Latência de gravação – o tempo médio de uma operação de leitura ou gravação em milissegundos.
- Taxa de transferência de leitura, Taxa de transferência de gravação – o número médio de megabytes lido ou gravado no disco por segundo.
- Profundidade da fila – o número de operações de E/S que aguardam pela gravação ou leitura no disco.

Tráfego de rede

- Taxa de transferência de recepção de rede, Taxa de transferência de transmissão de rede – a taxa de tráfego de rede entre a instância de banco de dados em bytes por segundo.

Conexões de banco de dados

- Conexões de banco de dados – o número de sessões do cliente que estão conectadas à instância do banco de dados.

Para obter descrições individuais mais detalhadas das métricas de performance disponíveis, consulte Monitorar métricas do Amazon RDS com o Amazon CloudWatch.

De um modo geral, os valores aceitáveis para as métricas de performance dependem do aspecto da linha de base e do que o aplicativo está fazendo. Investigue variações consistentes ou tendenciais de sua linha de base. Conselhos sobre tipos específicos de métricas a seguir:

- Alto consumo de CPU ou RAM: valores altos para o consumo de CPU ou RAM podem ser adequados. Por exemplo, eles poderão ficar assim se estiverem de acordo com seus objetivos em relação à aplicação (como throughput ou simultaneidade) e estiverem de acordo com sua expectativa.
- Consumo de espaço em disco: inspecione o consumo de espaço em disco caso o espaço usado seja consistentemente igual ou superior a 85% do espaço total no disco. Veja se é possível excluir dados da instância ou arquivar dados em um sistema diferente para liberar mais espaço.

- Tráfego de rede – em relação ao tráfego de rede, fale com o administrador do sistema para entender qual taxa de transferência é esperada para sua rede de domínio e conexão com a Internet. Inspecione o tráfego de rede caso a taxa de transferência seja consistentemente menor do que a esperada.
- Conexões do banco de dados – considere restringir as conexões do banco de dados caso perceba um alto número de conexões de usuários em conjunto com uma diminuição na performance da instância e no tempo de resposta. O melhor número de conexões de usuários para sua instância de banco de dados vai variar conforme a classe da instância e a complexidade das operações sendo executadas. Para determinar o número de conexões de banco de dados, associe sua instância de banco de dados a um grupo de parâmetros. Nesse grupo, defina o parâmetro User Connections (Conexões de usuário) como diferente de 0 (ilimitado). Você pode usar um parameter group existente ou criar um novo. Para obter mais informações, consulte Grupos de parâmetros para Amazon RDS.
- Métricas de IOPS – os valores esperados para as métricas de IOPS dependem da especificação do disco e da configuração do servidor, por isso, use sua linha de base para saber os valores típicos. Inspecione caso os valores sejam consistentemente diferentes da sua linha de base. Para obter o melhor performance de IOPS, confira se o seu conjunto de trabalho típico se adequa à memória para minimizar as operações de leitura e gravação.

Para problemas com métricas de performance, a primeira etapa para melhorar a performance é ajustar as consultas mais usadas e mais caras. Ajuste-os para ver se isso diminui a pressão sobre os recursos do sistema. Para obter mais informações, consulte Ajuste das consultas.

Se suas consultas forem ajustadas e o problema persistir, considere atualizar a Classes de instâncias de banco de dados do Amazon RDS. Você pode atualizá-la para uma que tenha mais do recurso (CPU, RAM, espaço em disco, largura de banda de rede, capacidade de E/S) que está relacionado ao problema.

Ajuste das consultas

Uma das melhores maneiras de melhorar a performance da instância de banco de dados é ajustar as consultas mais utilizadas e que requerem mais recursos. Aqui, você os ajusta para ficar mais barato executá-los. Para obter informações sobre como aprimorar consultas, use os seguintes recursos:

- MySQL – Consulte Otimizar instruções SELECT na documentação do MySQL. Para obter recursos adicionais de ajuste da consulta, consulte MySQL performance tuning and optimization resources (Recursos de otimização e ajuste de performance do MySQL).

- Oracle: consulte Database SQL Tuning Guide (Guia de ajuste do banco de dados SQL) na documentação do Oracle Database.
- SQL Server – Consulte Analyzing a query (Analisar uma consulta) na documentação da Microsoft. Você também pode usar as visualizações de gerenciamento de dados (DMVs) execution- e index-, relacionadas à E/S descritas na documentação System Dynamic Management Views (Visualizações de gerenciamento dinâmico de sistemas) para solucionar problemas de consulta do SQL Server.

  Um aspecto comum do ajuste de consulta é a criação de índices efetivos. Para obter melhorias do índice potencial para sua instância de banco de dados, consulte Database Engine Tuning Advisor (Orientador de Otimização do Mecanismo de Banco de dados) na documentação da Microsoft. Para obter informações sobre como usar o Orientador de Otimização em RDS for SQL Server, consulte Análise da workload do banco de dados em uma instância de banco de dados do Amazon RDS para SQL Server com o Orientador de Otimização do Mecanismo de Banco de Dados.
- PostgreSQL – Consulte Using EXPLAIN na documentação do PostgreSQL para saber como analisar um plano de consulta. Você pode usar essas informações para modificar uma consulta ou tabelas subjacentes para melhorar a performance da consulta.

  Para obter informações sobre como especificar as junções na consulta visando a melhor performance, consulte Controlling the planner with explicit JOIN clauses (Controlar o planejador com cláusulas JOIN explícitas).
- MariaDB – Consulte Otimizações de consulta na documentação do MariaDB.

Práticas recomendadas para trabalhar com o MySQL

Ambos os tamanhos de tabela e o número de tabelas em um banco de dados MySQL podem afetar a performance.

Tamanho da tabela

Normalmente, as restrições do sistema operacional em tamanhos de arquivo determinam o tamanho máximo efetivo da tabela para bancos de dados MySQL. Assim, os limites geralmente não são determinados por restrições internas do MySQL.

Em uma instância de banco de dados MySQL, evite que as tabelas em seu banco de dados se tornem muito grandes. Embora o limite geral de armazenamento seja de 64 TiB, os limites de armazenamento provisionados restringem o tamanho máximo de um arquivo de tabela do MySQL a

16 TiB. Divida suas tabelas grandes para que os tamanhos de arquivo estejam bem abaixo do limite de 16 TiB. Esta abordagem também pode melhorar a performance e o tempo de recuperação. Para obter mais informações, consulte Limites de tamanho de arquivo do MySQL no Amazon RDS.

Tabelas muito grandes (maiores que 100 GB de tamanho) podem afetar negativamente a performance de leituras e gravações (incluindo instruções DML e especialmente instruções DDL). Os índices em tabelas grandes podem melhorar significativamente a performance de seleção, mas também podem degradar a performance das instruções DML. As instruções DDL, como `ALTER TABLE`, podem ser significativamente mais lentas para as grandes tabelas porque essas operações podem reconstruir completamente uma tabela em alguns casos. Essas instruções DDL podem bloquear as tabelas durante a operação.

A quantidade de memória exigida pelo MySQL para leituras e gravações depende das tabelas envolvidas nas operações. É uma melhor prática ter pelo menos RAM suficiente para manter os índices de tabelas usadas ativamente. Para encontrar as dez maiores tabelas e índices em um banco de dados, use a seguinte consulta:

```
select table_schema, TABLE_NAME, dat, idx from
(SELECT table_schema, TABLE_NAME,
       ( data_length ) / 1024 / 1024 as dat,
       ( index_length ) / 1024 / 1024 as idx
FROM information_schema.TABLES
order by 3 desc ) a
order by 3 desc
limit 10;
```

Número de tabelas

Seu sistema de arquivos subjacente pode ter um limite no número de arquivos que representam tabelas. No entanto, o MySQL não tem limite para o número de tabelas. No entanto, o número total de tabelas no mecanismo de armazenamento MySQL InnoDB pode contribuir para a degradação da performance, independentemente do tamanho dessas tabelas. Para limitar o impacto do sistema operacional, você pode dividir as tabelas em vários bancos de dados na mesma instância de banco de dados MySQL. Isso pode limitar o número de arquivos em um diretório, mas não resolverá o problema geral.

Quando há degradação de performance devido a um grande número de tabelas (mais de 10 mil), ela é causada pelo MySQL trabalhando com arquivos de armazenamento, incluindo a abertura e o fechamento deles. Para resolver esse problema, você pode aumentar o tamanho dos parâmetros

`table_open_cache` e `table_definition_cache`. No entanto, aumentar os valores desses parâmetros pode aumentar significativamente a quantidade de memória que o MySQL usa, e pode até usar toda a memória disponível. Para obter mais informações, consulte How MySQL opens and closes tables na documentação do MySQL.

Além disso, muitas tabelas podem afetar significativamente o tempo de inicialização do MySQL. Tanto um desligamento limpo como uma reinicialização e uma recuperação de falha podem ser afetados, especialmente em versões anteriores ao MySQL 8.0.

Recomendamos ter menos de dez mil tabelas no total em todos os bancos de dados em uma instância de banco de dados. Para um caso de uso com um grande número de tabelas em um banco de dados MySQL, consulte Um milhão de tabelas no MySQL 8.0.

Mecanismo de armazenamento

Os recursos de recuperação a um ponto anterior no tempo restauração de snapshot do Amazon RDS para MySQL exigem um mecanismo de armazenamento de recuperação de falhas. Esses recursos só são compatíveis com o mecanismo de armazenamento InnoDB. Embora o MySQL suporte vários mecanismos de armazenamento com recursos variados, nem todos eles são otimizados para durabilidade de dados e recuperação de falha. Por exemplo, o mecanismo de armazenamento MyISAM não é compatível com a recuperação de falha confiável e pode impedir que uma recuperação a um ponto anterior no tempo ou uma restauração de snapshot funcionem da forma pretendida. Isso pode resultar em dados perdidos ou corrompidos quando o MySQL for reiniciado após uma falha.

O InnoDB é o mecanismo de armazenamento recomendado e compatível com instâncias de banco de dados MySQL no Amazon RDS. As instâncias do InnoDB também podem ser migradas para o Aurora, enquanto as instâncias do MyISAM não podem ser migradas. No entanto, o MyISAM funciona melhor do que o InnoDB se você precisar de uma capacidade de pesquisa intensa de texto completo. Se você ainda optar por usar o MyISAM com o Amazon RDS, seguir as etapas descritas em Backups automáticos com mecanismos de armazenamento MySQL sem suporte pode ser útil em certos cenários para a funcionalidade de restauração de snapshot.

Se você quiser converter tabelas do MyISAM existentes em tabelas do InnoDB, poderá usar o processo descrito em Converting Tables from MyISAM to InnoDB na documentação do MySQL. O MyISAM e a InnoDB têm pontos fortes e fracos diferentes. Portanto, é necessário que você avalie totalmente o impacto que essas alterações terão em seus aplicativos antes de fazê-las.

Além disso, o Federated Storage Engine atualmente não é compatível com o Amazon RDS for MySQL.

Práticas recomendadas para trabalhar com o MariaDB

Tanto os tamanhos de tabelas quanto o número de tabelas em um banco de dados MariaDB podem afetar a performance.

Tamanho da tabela

Normalmente, as restrições do sistema operacional em tamanhos de arquivo determinam o tamanho máximo efetivo da tabela para bancos de dados MariaDB. Assim, os limites geralmente não são determinados por restrições internas do MariaDB.

Em uma instância de banco de dados MariaDB, evite que as tabelas em seu banco de dados se tornem muito grandes. Embora o limite geral de armazenamento seja de 64 TiB, os limites de armazenamento provisionados restringem o tamanho máximo de um arquivo de tabela do MariaDB a 16 TiB. Divida suas tabelas grandes para que os tamanhos de arquivo estejam bem abaixo do limite de 16 TiB. Esta abordagem também pode melhorar a performance e o tempo de recuperação.

Tabelas muito grandes (maiores que 100 GB de tamanho) podem afetar negativamente a performance de leituras e gravações (incluindo instruções DML e especialmente instruções DDL). Os índices em tabelas grandes podem melhorar significativamente a performance de seleção, mas também podem degradar a performance das instruções DML. As instruções DDL, como `ALTER TABLE`, podem ser significativamente mais lentas para as grandes tabelas porque essas operações podem reconstruir completamente uma tabela em alguns casos. Essas instruções DDL podem bloquear as tabelas durante a operação.

A quantidade de memória exigida pelo MariaDB para leituras e gravações depende das tabelas envolvidas nas operações. É uma melhor prática ter pelo menos RAM suficiente para manter os índices de tabelas usadas ativamente. Para encontrar as dez maiores tabelas e índices em um banco de dados, use a seguinte consulta:

```
select table_schema, TABLE_NAME, dat, idx from
(SELECT table_schema, TABLE_NAME,
       ( data_length ) / 1024 / 1024 as dat,
       ( index_length ) / 1024 / 1024 as idx
FROM information_schema.TABLES
```

```
order by 3 desc ) a
order by 3 desc
limit 10;
```

Número de tabelas

Seu sistema de arquivos subjacente pode ter um limite no número de arquivos que representam tabelas. No entanto, o MariaDB não tem limite para o número de tabelas. No entanto, o número total de tabelas no mecanismo de armazenamento MariaDB InnoDB pode contribuir para a degradação da performance, independentemente do tamanho dessas tabelas. Para limitar o impacto no sistema operacional, você pode dividir as tabelas em vários bancos de dados na mesma instância de banco de dados MariaDB. Isso pode limitar o número de arquivos em um diretório, mas não resolverá o problema geral.

Quando há degradação da performance devido a um grande número de tabelas (mais de dez mil), o motivo é porque o MariaDB está trabalhando com arquivos de armazenamento. Esse trabalho inclui a abertura e o fechamento de arquivos de armazenamento do MariaDB. Para resolver esse problema, você pode aumentar o tamanho dos parâmetros `table_open_cache` e `table_definition_cache`. No entanto, aumentar os valores desses parâmetros pode aumentar significativamente a quantidade de memória que o MariaDB utiliza. Ele pode até usar toda a memória disponível. Para obter mais informações, consulte Optimizing table_open_cache na documentação do MariaDB.

Além disso, muitas tabelas podem afetar significativamente o tempo de inicialização do MariaDB. Tanto um desligamento limpo quanto uma reinicialização e uma recuperação de falha podem ser afetados. Recomendamos ter menos de dez mil tabelas no total em todos os bancos de dados em uma instância de banco de dados.

Mecanismo de armazenamento

Os recursos Point-In-Time Restore e Snapshot Restore do Amazon RDS for MariaDB exigem um mecanismo de armazenamento de recuperação de falhas. Embora o MariaDB suporte vários mecanismos de armazenamento com recursos variados, nem todos eles são otimizados para durabilidade de dados e recuperação de falha. Por exemplo, embora o Aria seja uma substituição segura contra falhas para o MyISAM, isso ainda pode impedir que uma Point-In-Time Restore ou uma restauração de snapshot funcione conforme o previsto. Isso pode resultar em dados perdidos ou corrompidos quando o MariaDB for reiniciado após uma falha. O XtraDB é o mecanismo de armazenamento recomendado e compatível com instâncias de banco de dados MariaDB no Amazon

RDS. Se você ainda optar por usar o Aria com o Amazon RDS, seguir as etapas descritas em Backups automáticos com mecanismos de armazenamento MariaDB sem suporte pode ser útil em certos cenários para a funcionalidade de restauração de snapshot.

Se você quiser converter tabelas do MyISAM existentes em tabelas do InnoDB, poderá usar o processo descrito em Converting Tables from MyISAM to InnoDB na documentação do MariaDB. O MyISAM e a InnoDB têm pontos fortes e fracos diferentes. Portanto, é necessário que você avalie totalmente o impacto que essas alterações terão em seus aplicativos antes de fazê-las.

Práticas recomendadas para trabalhar com o Oracle

Para obter informações sobre as práticas recomendadas para trabalhar com o Amazon RDS for Oracle, consulte Práticas recomendadas para execução do banco de dados Oracle no Amazon Web Services.

Um workshop virtual da AWS de 2020 teve uma apresentação sobre a execução de bancos de dados Oracle de produção no Amazon RDS. Um vídeo da apresentação está disponível aqui.

Práticas recomendadas para trabalhar com PostgreSQL

Das duas áreas importantes em que você pode melhorar a performance do RDS para PostgreSQL, uma é ao carregar dados em uma instância de banco de dados. Outra é ao usar o recurso autovacuum do PostgreSQL. As seções a seguir abrangem algumas das práticas que recomendamos para essas áreas.

Para obter informações sobre como Amazon RDS implementar outras tarefas comuns de DBA PostgreSQL, consulte Tarefas comuns de DBA do Amazon RDS para PostgreSQL.

Carregamento de dados em uma instância de banco de dados PostgreSQL

Ao carregar dados em uma instância de banco de dados do Amazon RDS para PostgreSQL, modifique as configurações de instância de banco de dados e os valores do grupo de parâmetros. Configure-os para permitir a importação mais eficiente de dados para sua instância de banco de dados.

Modifique as configurações da instância de banco de dados da seguinte forma:

- Desabilite os backups de instâncias de banco de dados (defina backup_retention como 0)

- Desabilite o Multi-AZ

Modifique o parameter group de banco de dados para incluir as seguintes configurações. Teste também as configurações dos parâmetros para encontrar as que forem mais eficientes para a instância do banco de dados.

- Aumente o valor do parâmetro `maintenance_work_mem`. Para obter mais informações sobre os parâmetros de consumo de recursos do PostgreSQL, consulte a documentação do PostgreSQL.
- Aumente o valor dos parâmetros `max_wal_size` e `checkpoint_timeout` para reduzir o número de gravações no log write-ahead (WAL).
- Desativar o parâmetro `synchronous_commit`.
- Desabilite o parâmetro autovacuum do PostgreSQL.
- Verifique se alguma das tabelas que você está importando não está registrada. Os dados armazenados em tabelas não registradas podem ser perdidos durante um failover. Para obter mais informações, consulte CREATE TABLE UNLOGGED.

Use os comandos `pg_dump -Fc` (compactado) ou `pg_restore -j` (paralelo) com essas configurações.

Após a conclusão da operação de carregamento, retorne a instância de banco de dados e os parâmetros de banco de dados para as configurações normais.

Trabalhar com o recurso autovacuum do PostgreSQL

O recurso autovacuum para bancos de dados PostgreSQL é um recurso o qual recomendamos que você use para manter a integridade de sua instância de banco de dados PostgreSQL. O autovacuum automatiza a execução do comando VACUUM e ANALYZE. O uso do autovacuum é exigido pelo PostgreSQL, não imposto pelo Amazon RDS, e é fundamental para a boa performance. O recurso é habilitado por padrão para todas as novas instâncias de banco de dados do Amazon RDS para PostgreSQL e os parâmetros de configuração relacionados são definidos apropriadamente por padrão.

O administrador de banco de dados precisa conhecer e entender esta operação de manutenção. Para obter a documentação do PostgreSQL sobre autovacuum, consulte The Autovacuum Daemon.

O autovacuum não é uma operação "livre de recursos", mas funciona em segundo plano e respeita as operações do usuário o máximo possível. Quando ativado, o autovacuum verifica as tabelas que

tiveram um grande número de ênuplas atualizadas ou excluídas. Ele também protege contra a perda de dados muito antigos devido ao encapsulamento do ID de transação. Para obter mais informações, consulte Prevenção contra falhas de encapsulamento de IDs de transação.

O autovacuum não deve ser considerado como uma operação de alta sobrecarga que pode ser reduzida para obter melhor performance. Em contrapartida, as tabelas que têm uma alta velocidade de atualizações e exclusões se deteriorarão rapidamente ao longo do tempo se o autovacuum não for executado.

> **Important**
>
> Não executar o autovacuum pode resultar em uma eventual interrupção necessária para executar uma operação de vacuum muito mais intrusiva. Em alguns casos, uma instância de banco de dados do RDS para PostgreSQL pode ficar indisponível devido ao uso de autovacuum. Nesses casos, o banco de dados do PostgreSQL é desligado para se proteger. Nesse ponto, o Amazon RDS deve executar um vacuum completo no modo de usuário único diretamente na instância de banco de dados. Esse vacuum completo pode ocasionar uma interrupção de várias horas. Portanto, é altamente recomendável não desativar o autovacuum, que está ativado por padrão.

Os parâmetros de autovacuum determinam quando funciona e a intensidade do autovacuum. Os parâmetros `autovacuum_vacuum_threshold` e `autovacuum_vacuum_scale_factor` determinam quando o autovacuum é executado. Os parâmetros `autovacuum_max_workers`, `autovacuum_nap_time`, `autovacuum_cost_limit` e `autovacuum_cost_delay` determinam a intensidade do autovacuum. Para obter mais informações sobre o autovacuum, quando ele é executado e quais parâmetros são necessários, consulte Routine Vacuuming na documentação do PostgreSQL.

A consulta a seguir mostra o número de tuplas "mortas" em uma tabela chamada table1:

```
SELECT relname, n_dead_tup, last_vacuum, last_autovacuum FROM
pg_catalog.pg_stat_all_tables
WHERE n_dead_tup > 0 and relname = 'table1';
```

Os resultados da consulta serão os seguintes:

```
relname | n_dead_tup | last_vacuum | last_autovacuum
```

```
---------+------------+-------------+-----------------
 tasks   |   81430522 |             |
(1 row)
```

Amazon RDS for PostgreSQL vídeo de práticas recomendadas

A conferência AWS re:Invent de 2020 teve uma apresentação sobre novos recursos e práticas recomendadas para trabalhar com o PostgreSQL no Amazon RDS. Um vídeo da apresentação está disponível aqui.

Práticas recomendadas para trabalhar com o SQL Server

As práticas recomendadas para uma implantação Multi-AZ com uma instância de banco de dados SQL Server incluem o seguinte:

- Usar eventos de banco de dados do Amazon RDS para monitorar failovers. Por exemplo, você pode ser notificado por mensagem de texto ou por e-mail quando uma instância de banco de dados falhar. Para obter mais informações sobre eventos do Amazon RDS, consulte Trabalhar com a notificação de eventos do Amazon RDS.
- Se seu aplicativo armazenar em cache os valores de DNS, defina o tempo de vida (TTL) para menos de 30 segundos. Definir o TTL dessa forma é uma prática recomendada caso haja um failover. Em um failover, o endereço IP pode mudar e o valor em cache pode não estar mais funcionando.
- Recomendamos que você não habilite os seguintes modos porque eles desligam o registro de log das transações, o que é necessário para o Multi-AZ:
  - Modo de recuperação simples
  - Modo off-line
  - Modo somente leitura
- Teste para determinar quanto tempo leva para ocorrer um failover na sua instância de banco de dados. O tempo de failover pode variar devido ao tipo de banco de dados, a classe da instância e o tipo de armazenamento que você usa. Você também deve testar a capacidade de continuar trabalhando do seu aplicativo caso ocorra um failover.
- Para reduzir o tempo de failover, faça o seguinte:
  - Certifique-se de ter IOPS provisionadas suficientes alocadas para sua workload. A E/S inadequada pode prolongar os tempos de failover. A recuperação do banco de dados requer E/S.

  - Use transações menores. A recuperação do banco de dados depende das transações, portanto, se for possível dividir grandes transações em várias transações menores, seu tempo de failover deverá ser menor.
- Leve em consideração que, durante um failover, haverá latências elevadas. Como parte do processo de failover, o Amazon RDS replica automaticamente seus dados para uma nova instância em espera. Essa replicação significa que novos dados estão sendo confirmados em duas instâncias de banco de dados diferentes. Portanto, pode haver alguma latência até que a instância de banco de dados de espera tenha alcançado a nova instância de banco de dados primária.
- Implante seus aplicativos em todas as zonas de disponibilidade. Se uma zona de disponibilidade ficar inativa, seus aplicativos nas outras zonas de disponibilidade ainda estarão disponíveis.

Ao trabalhar com uma implantação Multi-AZ do SQL Server, lembre-se de que o Amazon RDS cria réplicas para todos os bancos de dados do SQL Server na instância. Se você não quiser que bancos de dados específicos tenham réplicas secundárias, configure uma instância de banco de dados separada que não use o Multi-AZ nesses bancos de dados.

Amazon RDS for SQL Server vídeo de práticas recomendadas

A conferência AWS re:Invent de 2019 teve uma apresentação sobre novos recursos e práticas recomendadas para trabalhar com o SQL Server no Amazon RDS. Um vídeo da apresentação está disponível aqui.

Trabalhar com grupos de parâmetros de banco de dados

Recomendamos que você experimente fazer mudanças de parameter group de banco de dados em uma instância de banco de dados de teste antes de aplicar alterações de parameter group às instâncias de banco de dados de produção. Definir incorretamente os parâmetros do mecanismo de banco de dados em um parameter group de banco de dados pode causar efeitos adversos não intencionais, inclusive diminuição no performance e instabilidade do sistema. Sempre tenha cuidado ao modificar os parâmetros do mecanismo de banco de dados e faça backup de sua instância de banco de dados antes de modificar um parameter group de banco de dados.

Para obter informações sobre o backup da instância de banco de dados, consulte Backup, restauração e exportação de dados.

Práticas recomendadas para automatizar a criação de instâncias de banco de dados

É uma prática recomendada do Amazon RDS criar uma instância de banco de dados com a versão secundária preferida do mecanismo de banco de dados. Você pode usar a AWS CLI, a API do Amazon RDS ou o AWS CloudFormation para automatizar a criação de instâncias de banco de dados. Ao usar esses métodos, você pode especificar apenas a versão principal e o Amazon RDS cria automaticamente a instância com a versão secundária preferida. Por exemplo, se a versão 12.5 do PostgreSQL for a versão secundária preferida e você especificar a 12 com `create-db-instance`, a instância de banco de dados será da versão 12.5.

Para determinar a versão secundária preferida, você pode executar o comando `describe-db-engine-versions` com a opção `--default-only`, conforme mostrado no exemplo a seguir.

```
aws rds describe-db-engine-versions --default-only --engine postgres

{
    "DBEngineVersions": [
        {
            "Engine": "postgres",
            "EngineVersion": "12.5",
            "DBParameterGroupFamily": "postgres12",
            "DBEngineDescription": "PostgreSQL",
            "DBEngineVersionDescription": "PostgreSQL 12.5-R1",
            ...some output truncated...
        }
    ]
}
```

Para obter informações sobre como criar instâncias de banco de dados programaticamente, consulte os seguintes recursos:

- Usar a AWS CLI: create-db-instance
- Usar a API do Amazon RDS – CreateDBInstance
- Usar o AWS CloudFormation: AWS::RDS::DBInstance

Vídeo de novos recursos do Amazon RDS

A conferência AWS re:Invent de 2023 teve uma apresentação sobre novos recursos do Amazon RDS. Um vídeo da apresentação está disponível aqui.

Acesso programático ao Amazon RDS

O Amazon RDS fornece as seguintes ferramentas para gerenciar seus recursos do Amazon RDS de forma programática.

AWS Command Line Interface (AWS CLI)

Você pode criar e gerenciar seus recursos do RDS usando a AWS CLI em um shell da linha de comandos. A AWS CLI fornece acesso direto às APIs para serviços da AWS, como o Amazon RDS. Consulte a sintaxe e exemplos dos comandos do Amazon RDS em rds na Referência de comandos da AWS CLI.

AWS CloudFormation

Com essa ferramenta de infraestrutura como código (IaC) da AWS, é possível criar modelos que descrevem todos os recursos do Amazon RDS que você desejar, e o AWS CloudFormation provisiona e configura esses recursos para você. Para ter mais informações, consulte the section called “Criar recursos com o AWS CloudFormation”.

Kits de desenvolvimento de software (SDKs) da AWS

A AWS fornece SDKs para muitas tecnologias e linguagens de programação conhecidas. Eles facilitam a chamada de serviços da AWS de dentro de suas aplicações nessa linguagem ou tecnologia. Consulte mais informações sobre esses SDKs em Ferramentas para desenvolver e gerenciar aplicações na AWS.

API do Amazon RDS

Essa API é a interface em nível de protocolo para o Amazon RDS. Ao usar essa API, você deve formatar cada solicitação HTTPS corretamente e adicionar uma assinatura digital válida a cada solicitação. Para ter mais informações, consulte Referência da API do Amazon RDS.

Console-to-Code

Com essa ferramenta, é possível gerar código para ações que você executa no console do Amazon RDS e usar esse código em outras ferramentas, como o AWS CloudFormation. Para ter mais informações, consulte the section called “Console-to-Code”.

Usar o Console-to-Code para gerar código para as suas ações no console do Amazon RDS

O console oferece um caminho guiado para criar recursos e testar protótipos. Se quiser criar os mesmos recursos em escala, você precisará de um código de automação. O Console-to-Code é um atributo do Amazon Q Developer que pode ajudar você a começar a usar seu código de automação. O Console-to-Code registra as ações que você executa no console, incluindo os valores padrão e os valores de parâmetros fornecidos. Depois, ele usa IA generativa para sugerir código no idioma e formato de sua preferência para as ações desejadas. Como o fluxo de trabalho do console garante que os valores dos parâmetros que você especifica sejam válidos juntos, o código que você gera usando o Console-to-Code tem valores de parâmetros compatíveis. Você pode usar o código como ponto de partida e, depois, personalizá-lo para deixá-lo pronto para produção no seu caso de uso específico.

Por exemplo, com o Console-to-Code, é possível gravar a criação de uma instância de banco de dados do RDS e optar por gerar código no formato JSON do AWS CloudFormation. Em seguida, você pode copiar esse código e personalizá-lo para uso em seu modelo do AWS CloudFormation.

Atualmente, o Console-to-Code pode gerar infraestrutura como código (IaC) nos seguintes formatos e linguagens:

- Java do CDK
- Python do CDK
- TypeScript do CDK
- JASON do CloudFormation
- YAML do CloudFormation

Para obter mais informações e instruções sobre como usar o Console-to-Code, consulte Automatizar serviços da AWS com o Console-to-Code do Amazon Q Developer no Guia do usuário do Amazon Q Developer.

Configurar uma instância de banco de dados do Amazon RDS

Esta seção mostra como configurar a instância de banco de dados do Amazon RDS. Antes de criar uma instância de banco de dados, decida sobre a classe de instância de banco de dados que executará a instância de banco de dados. Além disso, decida em que local a instância de banco de dados será executada escolhendo uma região da AWS. Depois, crie a instância de banco de dados.

Você pode configurar uma instância de banco de dados com um grupo de opções e um grupo de parâmetros de banco de dados.

- Um grupo de opções pode especificar recursos, chamados de opções, que estão disponíveis para uma determinada instância de banco de dados do Amazon RDS.
- Um grupo de parâmetros de banco de dados atua como um contêiner para valores de configuração de mecanismo que são aplicados a uma ou mais instâncias de bancos de dados.

As opções e parâmetros disponíveis dependem do mecanismo de banco de dados e da versão do mecanismo de banco de dados. Você pode especificar um grupo de opções e um grupo de parâmetros de banco de dados ao criar uma instância de banco de dados. Você também pode modificar uma instância de banco de dados para especificá-los.

Tópicos

- Criar uma instância de banco de dados do Amazon RDS
- Criar recursos do Amazon RDS com o AWS CloudFormation
- Conectar a uma instância de banco de dados do Amazon RDS
- Trabalhar com grupos de opções
- Grupos de parâmetros para Amazon RDS
- Criar um cache do Amazon ElastiCache usando as configurações da instância de banco de dados do Amazon RDS
- Migração automática de bancos de dados do EC2 para o Amazon RDS usando o AWS Database Migration Service
- Tutorial: criar uma instância de banco de dados do MySQL com um grupo de parâmetros personalizado e um novo grupo de opções

Criar uma instância de banco de dados do Amazon RDS

O bloco de construção básico do Amazon RDS é a instância de banco de dados, onde você cria os bancos de dados. Você escolhe as características específicas do mecanismo da instância de banco de dados ao criá-la. Você também escolhe a capacidade de armazenamento, CPU, memória e assim por diante, da instância AWS em que o servidor de banco de dados é executado.

Tópicos

- Pré-requisitos da instância de banco de dados
- Criar uma instância de banco de dados
- Configurações para instâncias de banco de dados

Pré-requisitos da instância de banco de dados

> **Important**
>
> É necessário concluir as tarefas em Configuração do ambiente do Amazon RDS antes de criar uma instância de banco de dados do Amazon RDS.

Veja a seguir os pré-requisitos para a criação de um cluster de banco de dados do .

Tópicos

- Configurar a rede para a instância de banco de dados
- Pré-requisitos adicionais

Configurar a rede para a instância de banco de dados

Só é possível criar uma instância de banco de dados do Amazon RDS em uma nuvem privada virtual (VPC) com base no serviço da Amazon VPC. Além disso, ele deve estar em uma Região da AWS que tenha pelo menos duas zonas de disponibilidade. O grupo de sub-redes do banco de dados escolhido para a instância de banco de dados deve incluir pelo menos duas zonas de disponibilidade. Essa configuração garante que você possa configurar uma implantação multi-AZ ao criar a instância de banco de dados ou migrar facilmente para uma no futuro.

Para configurar a conectividade entre sua nova instância de banco de dados e uma instância do Amazon EC2 na mesma VPC, faça isso durante a criação da instância. Para se conectar à instância de banco de dados usando recursos que não sejam instâncias do EC2 na mesma VPC, configure as conexões de rede manualmente.

Tópicos

- Configurar a conectividade automática de rede com uma instância do EC2
- Configurar a rede manualmente

Configurar a conectividade automática de rede com uma instância do EC2

Ao criar uma instância de banco de dados do RDS, use o AWS Management Console para configurar a conectividade entre uma instância do EC2 e a nova instância de banco de dados. Quando você faz isso, o RDS configura suas definições de VPC e rede automaticamente. A instância de banco de dados é criada na mesma VPC da instância do EC2 para que esta possa acessar a instância de banco de dados.

Confira a seguir os requisitos para conectar uma instância do EC2 à instância de banco de dados:

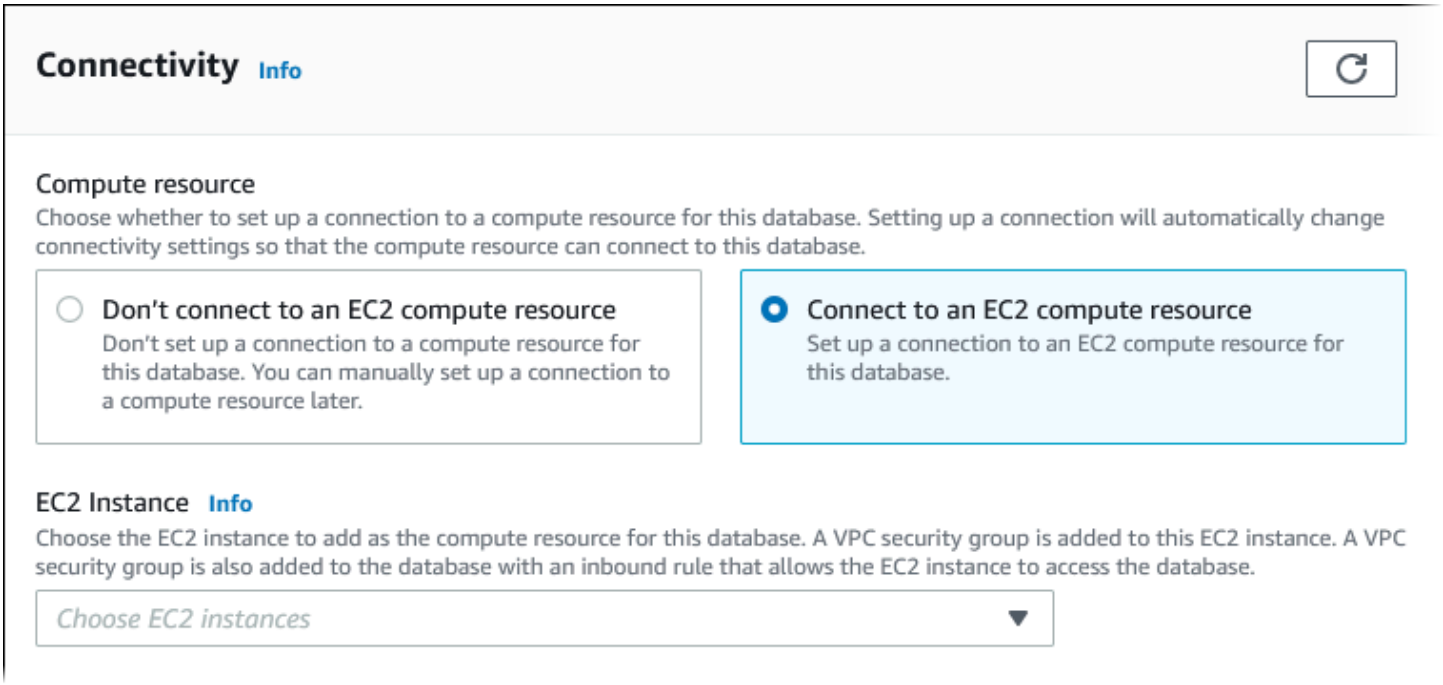
- A instância do EC2 deve existir na Região da AWS antes da criação da instância de banco de dados.

  Se não houver nenhuma instância do EC2 na Região da AWS, o console fornecerá um link para que você crie uma.
- O usuário que está criando a instância de banco de dados deve ter permissões para realizar as seguintes operações:
  - `ec2:AssociateRouteTable`
  - `ec2:AuthorizeSecurityGroupEgress`
  - `ec2:AuthorizeSecurityGroupIngress`
  - `ec2:CreateRouteTable`
  - `ec2:CreateSubnet`
  - `ec2:CreateSecurityGroup`
  - `ec2:DescribeInstances`
  - `ec2:DescribeNetworkInterfaces`
  - `ec2:DescribeRouteTables`

- `ec2:DescribeSecurityGroups`
- `ec2:DescribeSubnets`
- `ec2:ModifyNetworkInterfaceAttribute`
- `ec2:RevokeSecurityGroupEgress`

Usar essa opção cria uma instância de banco de dados privada. A instância de banco de dados usa um grupo de sub-redes de banco de dados somente com sub-redes privadas para restringir o acesso aos recursos da VPC.

Para conectar uma instância do EC2 à instância de banco de dados, escolha Connect to an EC2 compute resource (Conectar-se a um recurso de computação do EC2) na seção Connectivity (Conectividade) da página Create database (Criar banco de dados).



Quando você escolhe Connect to an EC2 compute resource (Conectar-se a um recurso de computação do EC2), o RDS define as opções a seguir automaticamente. Você não pode alterar essas configurações, a menos que opte por não configurar a conectividade com uma instância do EC2 escolhendo Don't connect to an EC2 compute resource (Não conectar a um recurso de computação do EC2).

Opção do console	Configuração automática

Opção do console	Configuração automática
Tipo de rede	O RDS define o tipo de rede como IPv4. No momento, o modo de pilha dupla não é compatível quando você configura uma conexão entre uma instância do EC2 e a instância de banco de dados.
Virtual Private Cloud (VPC)	O RDS define a VPC como aquela associada à instância do EC2.

Opção do console	Configuração automática
DB subnet group (Grupo de subredes do banco de dados)	O RDS requer um grupo de sub-redes de banco de dados com uma sub-rede privada na mesma zona de disponibilidade da instância do EC2. Se existir um grupo de sub-redes de banco de dados que atenda a esse requisito, o RDS usará o grupo de sub-redes de banco de dados existente. Por padrão, essa opção está definida como Automatic setup (Configuração automática). Quando você escolhe Automatic setup (Configuração automátic a) e não há nenhum grupo de sub-redes de banco de dados que atenda a esse requisito, ocorre a ação a seguir. O RDS usa três sub-redes privadas disponíveis em três zonas de disponibi lidade, das quais uma é a mesma da instância do EC2. Se não houver uma sub-rede privada disponível em uma zona de disponibilidade, o RDS criará uma sub-rede privada na zona de disponibilidade. O RDS cria o grupo de sub-redes de banco de dados. Quando houver uma sub-rede privada disponível, o RDS usará a tabela de rotas associada a ela e adicionará todas as sub-redes que criar a essa tabela de rotas. Quando não houver nenhuma sub-rede privada disponível, o RDS criará uma tabela de rotas sem acesso ao gateway da Internet e adicionará as sub-redes que criar à tabela de rotas. O RDS também permite que você use grupos de sub-redes de banco de dados existentes. Selecione Choose existing (Selecion ar existente) se quiser usar um grupo de sub-redes de banco de dados existente de sua escolha.
Acesso público	O RDS escolhe No (Não) para que a instância de banco de dados não fique acessível ao público. Por motivos de segurança, é uma prática recomendada manter o banco de dados privado e garantir que ele não seja acessado pela Internet.

Opção do console	Configuração automática
Grupo de segurança da VPC (firewall)	O RDS cria um grupo de segurança associado à instância de banco de dados. O grupo de segurança é chamado de `rds-ec2-`*n*, em que *n* é um número. Esse grupo de segurança inclui uma regra de entrada com o grupo de segurança da VPC do EC2 (firewall) como origem. Esse grupo de segurança associado à instância de banco de dados permite que a instância do EC2 acesse a respectiva instância. O RDS também cria um grupo de segurança associado à instância do EC2. O grupo de segurança é chamado de `ec2-rds-`*n*, em que *n* é um número. Esse grupo de segurança inclui uma regra de saída no grupo de segurança da VPC da instância de banco de dados como origem. Esse grupo de segurança permite que a instância do EC2 envie tráfego à instância de banco de dados. Para adicionar outro novo grupo de segurança, escolha Create new (Criar novo) e digite o nome do novo grupo de segurança. Para adicionar grupos de segurança existentes, escolha Choose existing (Escolher existente) e selecione os grupos de segurança que deseja adicionar.

Opção do console	Configuração automática
Zona de disponibilidade	Quando você escolher Single DB instance (Instância de banco de dados única) em Availability & durability (Disponibilidade e durabilidade) (implantação single-AZ), o RDS escolherá a zona de disponibilidade da instância do EC2. Quando você escolher Multi-AZ DB instance (Instância de banco de dados multi-AZ) em Availability & durability (Disponibilidade e durabilidade) (implantação de instância de banco de dados multi-AZ), o RDS escolherá a zona de disponibilidade da instância do EC2 para uma instância de banco de dados da implantação. O RDS escolhe aleatoriamente uma zona de disponibilidade diferente para a outra instância de banco de dados. A instância de banco de dados primária ou a réplica de espera é criada na mesma zona de disponibilidade da instância do EC2. Quando você escolhe Multi-AZ DB instance (Instância de banco de dados multi-AZ), existe a possibilidade de custos entre zonas de disponibilidade se a instância de banco de dados e a instância do EC2 estiverem em zonas de disponibilidade diferentes.

Para ter mais informações sobre essas configurações, consulte Configurações para instâncias de banco de dados.

Se você alterar essas configurações depois da criação da instância de banco de dados, as alterações poderão afetar a conexão entre a instância do EC2 e a instância de banco de dados.

Configurar a rede manualmente

Para se conectar à instância de banco de dados usando recursos que não sejam instâncias do EC2 na mesma VPC, configure as conexões de rede manualmente. Se você usar o AWS Management Console para criar a instância de banco de dados, o Amazon RDS poderá criar automaticamente uma VPC para você. Como alternativa, você pode usar a VPC existente ou criar uma VPC para sua instância de banco de dados. Com qualquer abordagem, sua VPC requer pelo menos uma sub-rede de cada em pelo menos duas zonas de disponibilidade para você usá-la com uma instância de banco de dados do RDS.

Por padrão, o Amazon RDS cria automaticamente a instância de banco de dados em uma zona de disponibilidade para você. Para escolher uma zona de disponibilidade específica, altere a configuração Availability & durability (Disponibilidade e durabilidade) para Single DB instance (Instância de banco de dados única). Essa ação exibe uma configuração de Availability Zone (Zona de disponibilidade) que permite escolher entre as zonas de disponibilidade em sua VPC. No entanto, se você escolher uma implantação multi-AZ, o RDS escolherá automaticamente a zona de disponibilidade da instância de banco de dados primária ou gravadora, e a configuração Availability Zone (Zona de disponibilidade) não será exibida.

Em alguns casos, você pode não ter uma VPC padrão ou não ter criado uma VPC. Nesses casos, o Amazon RDS pode criar automaticamente uma VPC para você ao criar uma instância de bancos de dados utilizando o console. Caso contrário, faça o seguinte:

- Crie uma VPC com com no mínimo uma sub-rede em pelo menos duas das zonas de disponibilidade na Região da AWS em que você deseja implantar a instância de banco de dados. Para ter mais informações, consulte Trabalhar com uma instância de banco de dados em uma VPC e Tutorial: Criar uma VPC para usar com uma instância de banco de dados (somente IPv4).
- Especifique um grupo de segurança da VPC que autorize conexões à sua instância de banco de dados. Para ter mais informações, consulte Fornecer acesso à instância de banco de dados na VPC criando um grupo de segurança e Controlar acesso com grupos de segurança.
- Especifique um grupo de sub-rede de banco de dados do RDS que defina pelo menos duas sub-redes na VPC as quais possam ser usadas pela instância de banco de dados. Para ter mais informações, consulte Trabalhar com grupos de sub-redes de banco de dados.

Se você quiser se conectar a um recurso que não esteja na mesma VPC que a instância de banco de dados, veja os cenários apropriados em Cenários para acessar uma instância de banco de dados em uma VPC.

Pré-requisitos adicionais

Antes de criar a instância de banco de dados, considere os seguintes pré-requisitos adicionais:

- Se você estiver se conectando à AWS utilizando credenciais do AWS Identity and Access Management (IAM), sua conta da AWS deverá ter determinadas políticas do IAM. Elas concedem as permissões necessárias para realizar operações do Amazon RDS. Para ter mais informações, consulte Gerenciamento de identidade e acesso no Amazon RDS.

Para utilizar o IAM para acessar o console do RDS, faça login no AWS Management Console com suas credenciais de usuário do IAM. Depois, acesse o console do Amazon RDS em https://console.aws.amazon.com/rds/.

- Para personalizar os parâmetros de configuração de sua instância de banco de dados, especifique um grupo de parâmetros de banco de dados com as configurações de parâmetro necessárias. Para obter informações sobre como criar ou modificar um grupo de parâmetros de banco de dados, consulte Grupos de parâmetros para Amazon RDS.

> **Important**
>
> Se você estiver usando o modelo BYOL para o Amazon RDS para Db2, antes de criar uma instância de banco de dados, primeiro crie um grupo de parâmetros personalizado que contenha o IBM Site ID e o IBM Customer ID. Para ter mais informações, consulte Traga a sua própria licença para Db2.

- Determine o número de porta de TCP/IP a ser especificado para sua instância de banco de dados. Em algumas empresas, os firewalls bloqueiam conexões com as portas padrão para instâncias de banco de dados do RDS. Se o firewall da sua empresa bloquear a porta padrão, escolha outra porta para a instância de banco de dados. As portas padrão para os mecanismos de banco de dados do Amazon RDS são:

RDS para Db2	RDS para MariaDB	RDS para MySQL	RDS para Oracle	RDS para PostgreSQL	RDS para SQL Server
50000	3306	3306	1521	5432	1433

Para o RDS para SQL Server, as seguintes portas são reservadas, e não é possível usá-las ao criar uma instância de banco de dados: `1234`, `1434`, `3260`, `3343`, `3389`, `47001`, e `49152-49156`.

Criar uma instância de banco de dados

Você pode criar e modificar uma instância de banco de dados do Amazon RDS usando o AWS Management Console, a AWS CLI ou a API do RDS.

> **Note**
>
> Em relação ao RDS para Db2, recomendamos configurar os itens necessários para o modelo de licença antes de criar uma instância de banco de dados do RDS para Db2. Para ter mais informações, consulte Opções de licenciamento do Amazon RDS para Db2.

Console

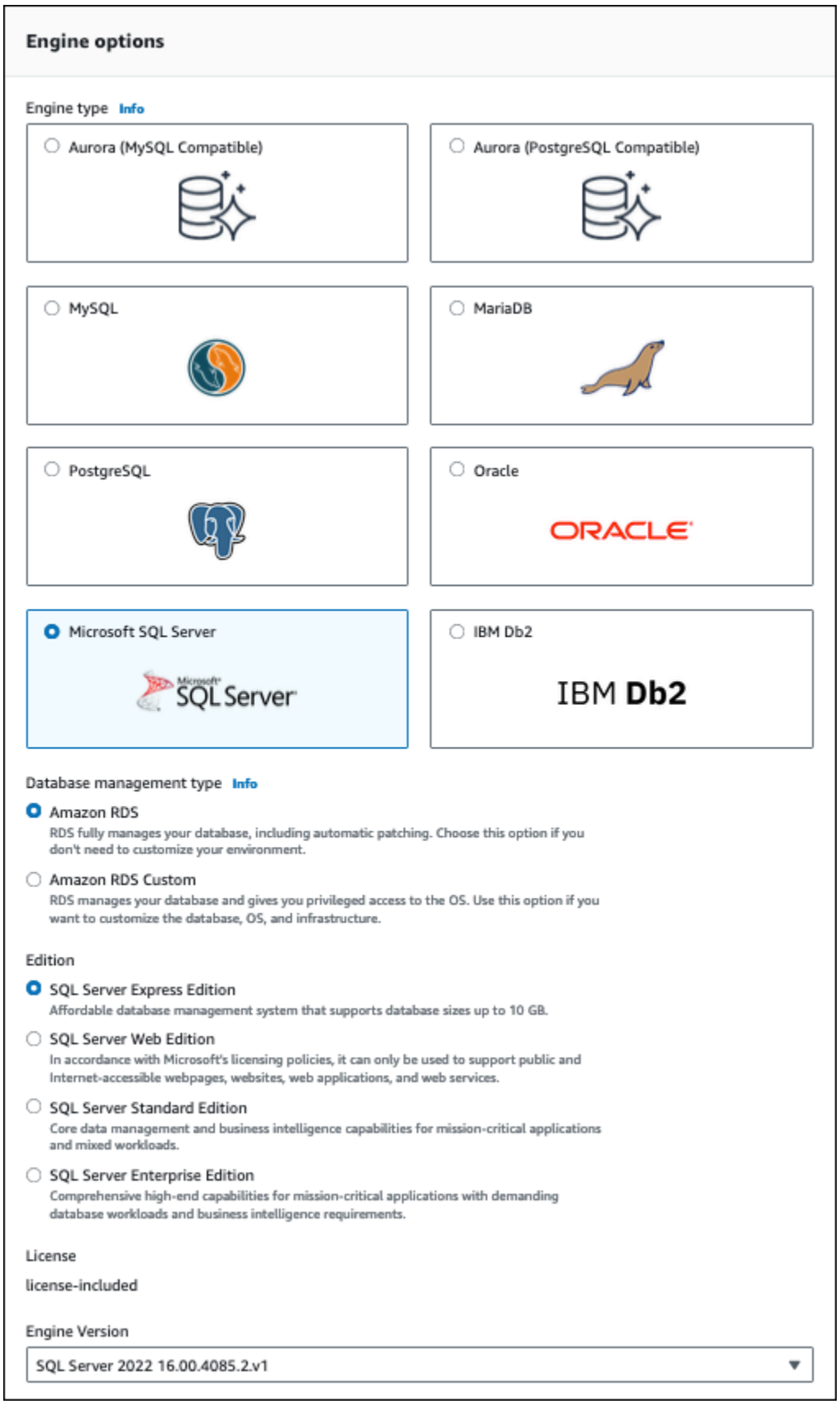
É possível criar uma instância de banco de dados usando o AWS Management Console com a opção Easy Create (Criação fácil) habilitada ou não. Com a Easy create (Criação fácil) habilitada, você especifica apenas o tipo de mecanismo de banco de dados, o tamanho da instância de banco de dados e o identificador da instância de banco de dados. A Easy create (Criação fácil) usa a configuração padrão para outras opções de configuração. Com a Easy create (Criação fácil) desabilitada, você especifica mais opções de configuração ao criar um banco de dados, incluindo as de disponibilidade, segurança, backups e manutenção.

> **Note**
>
> No procedimento a seguir, a opção Standard Create (Criação padrão) está habilitada e Easy Create (Criação fácil) não está habilitada. Este procedimento usa o Microsoft SQL Server como exemplo.
> Para obter exemplos que usam a Easy Create (Criação fácil) para orientar você a criar e conectar a instâncias de banco de dados de exemplo para cada mecanismo, consulte Conceitos básicos do Amazon RDS.

Como criar uma instância de banco de dados

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No canto superior direito do console do Amazon RDS, escolha a região da AWS na qual você deseja criar a instância de banco de dados.
3. No painel de navegação, escolha Databases (Bancos de dados).
4. Selecione Criar banco de dados e escolha Criação padrão.
5. Em Tipo de mecanismo, selecione IBM Db2, MariaDB, Microsoft SQL Server, MySQL, Oracle ou PostgreSQL.

O Microsoft SQL Server é exibido aqui.


Engine options
Engine type Info
Aurora (MySQL Compatible)
Aurora (PostgreSQL Compatible)
MySQL
MariaDB
PostgreSQL
Oracle
ORACLE
Microsoft SQL Server
SQL Server
IBM Db2
IBM Db2
Database management type Info
Amazon RDS
RDS fully manages your database, including automatic patching. Choose this option if you don't need to customize your environment.
Amazon RDS Custom
RDS manages your database and gives you privileged access to the OS. Use this option if you want to customize the database, OS, and infrastructure.
Edition
SQL Server Express Edition
Affordable database management system that supports database sizes up to 10 GB.
SQL Server Web Edition
In accordance with Microsoft's licensing policies, it can only be used to support public and Internet-accessible webpages, websites, web applications, and web services.
SQL Server Standard Edition
Core data management and business intelligence capabilities for mission-critical applications and mixed workloads.
SQL Server Enterprise Edition
Comprehensive high-end capabilities for mission-critical applications with demanding database workloads and business intelligence requirements.
License
license-included
Engine Version
SQL Server 2022 16.00.4085.2.v1

6. Em Tipo de gerenciamento de banco de dados, se você estiver usando Oracle ou SQL Server, escolha Amazon RDS ou Amazon RDS Custom.

   O Amazon RDS é exibido aqui. Para saber mais sobre o RDS Custom, consulte Amazon RDS Custom.

7. Para Edição, se você estiver usando Db2, Oracle ou SQL Server, selecione a edição do mecanismo de banco de dados que deseja usar.

   O MySQL tem apenas uma opção para a edição e MariaDB e PostgreSQL não têm opções.

8. Em Version (Versão), escolha a versão do mecanismo.

9. Em Templates (Modelos), escolha o modelo que corresponde ao seu caso de uso. Se você escolher Production (Produção), as seguintes opções serão pré-selecionadas em uma etapa posterior:

   - Opção de failover multi-AZ
   - Opção de armazenamento Provisioned IOPS (IOPS provisionadas)
   - Opção Enable deletion protection (Habilitar proteção contra exclusão)

   Recomendamos essas características para qualquer ambiente de produção.

   > **Note**
   >
   > As escolhas de modelo variam de acordo com a edição.

10. Na seção Settings (Configurações), abra Credential Settings (Configurações de credencial). Faça o seguinte:

    a. (Opcional) Altere o valor Master username (Nome de usuário primário).

    b. Escolha uma das seguintes opções de gerenciamento de credenciais:

       - Gerenciado no AWS Secrets Manager

         Em Selecionar a chave de criptografia, escolha uma chave do KMS que o Secrets Manager cria ou uma chave que você criou.

> **Note**
>
> Recomendamos o AWS Secrets Manager como a técnica mais segura para gerenciar credenciais. Aplicam-se cobranças adicionais. Para ter mais informações, consulte Gerenciamento de senhas com Amazon RDS e AWS Secrets Manager.

- Autogerenciado

  Para especificar uma senha, desmarque a caixa de seleção Gerar uma senha automaticamente se ela estiver marcada. Insira a mesma senha em Senha primária e Confirmar senha.

11. (Opcional) Configure uma conexão com um recurso de computação para essa instância de banco de dados.

    Você pode configurar a conectividade entre uma instância do Amazon EC2 e a nova instância de banco de dados durante a criação da instância. Para ter mais informações, consulte Configurar a conectividade automática de rede com uma instância do EC2.

12. Na seção Conectividade em Grupo de segurança da VPC (firewall), se você selecionar Criar, um grupo de segurança da VPC será criado com uma regra de entrada que permite que o endereço IP do computador local acesse o banco de dados.

13. Nas seções restantes, especifique suas configurações de instância de banco de dados. Para obter informações sobre cada configuração, consulte Configurações para instâncias de banco de dados.

14. Escolha Create database (Criar banco de dados).

    Se você optar por usar uma senha gerada automaticamente, o botão View credential details (Ver detalhes da credencial) será exibido na página Databases (Bancos de dados).

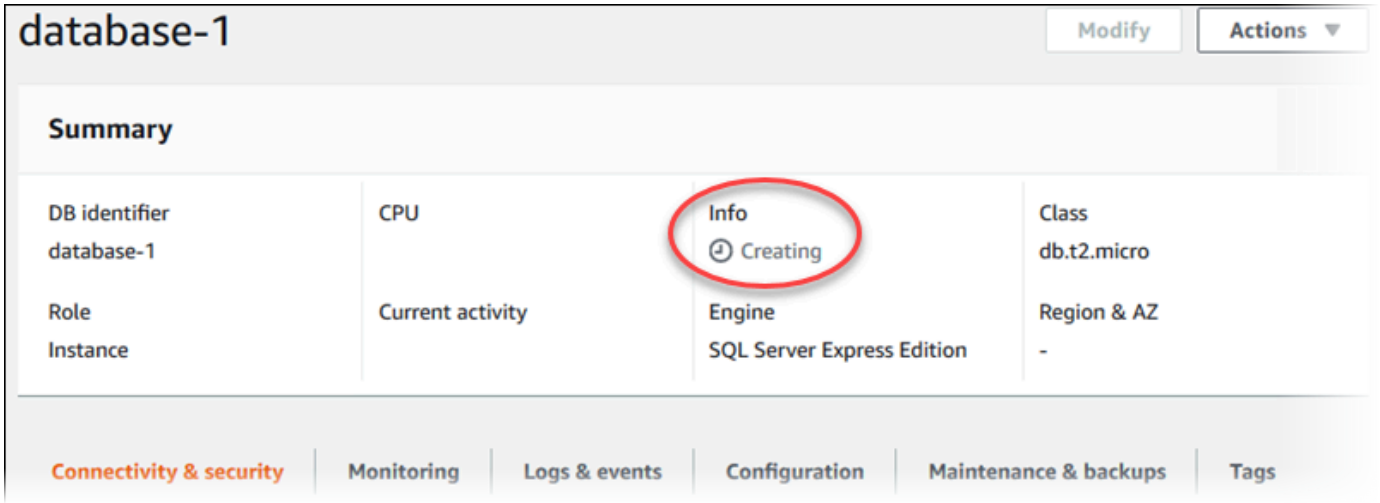
    Para exibir o nome de usuário mestre e a senha da instância de banco de dados, escolha View credential details (Ver detalhes da credencial).

    Para se conectar à instância de banco de dados como o usuário mestre, use o nome de usuário e a senha exibidos.

> **Important**
>
> Você não pode visualizar a senha do usuário principal novamente. Caso você não a registre, talvez seja necessário alterá-la. Se for necessário alterar a senha do usuário mestre depois que a instância de banco de dados estiver disponível, modifique a instância de banco de dados para fazer isso. Para obter mais informações sobre a modificação de uma instância de banco de dados , consulte Modificar uma instância de banco de dados do Amazon RDS.

15. Em Databases (Bancos de dados), escolha o nome da nova instância de banco de dados.

    No console do RDS, os detalhes para a nova instância de banco de dados são exibidos. A instância de banco de dados fica com o status Creating (Criando) até que esteja criada e pronta para uso. Quando o status muda para Available (Disponível), você pode se conectar à instância de banco de dados. Dependendo da classe da instância de banco de dados e do armazenamento alocado, pode levar alguns minutos até que a nova instância fique disponível.



AWS CLI

> **Note**
>
> Se quiser usar a licença do Db2 por meio do AWS Marketplace, é necessário primeiro assinar o AWS Marketplace e cadastrar-se na IBM usando o AWS Management Console.

Para ter mais informações, consulte Assinar ofertas do Db2 Marketplace e cadastrá-las na IBM.

Para criar uma instância de banco de dados utilizando a AWS CLI, chame o comando create-db-instance com os seguintes parâmetros:

- `--db-instance-identifier`
- `--db-instance-class`
- `--vpc-security-group-ids`
- `--db-subnet-group`
- `--engine`
- `--master-username`
- `--master-user-password` ou `--manage-master-user-password`
- `--allocated-storage`
- `--backup-retention-period`

Para obter informações sobre cada configuração, consulte Configurações para instâncias de banco de dados.

Este exemplo usa o Microsoft SQL Server.

Example

Para Linux, macOS ou Unix:

```
aws rds create-db-instance \
    --engine sqlserver-se \
    --db-instance-identifier mymsftsqlserver \
    --allocated-storage 250 \
    --db-instance-class db.t3.large \
    --vpc-security-group-ids mysecuritygroup \
    --db-subnet-group mydbsubnetgroup \
    --master-username masterawsuser \
    --manage-master-user-password \
    --backup-retention-period 3
```

Para Windows:

```
aws rds create-db-instance ^
    --engine sqlserver-se ^
    --db-instance-identifier mydbinstance ^
    --allocated-storage 250 ^
    --db-instance-class db.t3.large ^
    --vpc-security-group-ids mysecuritygroup ^
    --db-subnet-group mydbsubnetgroup ^
    --master-username masterawsuser ^
    --manage-master-user-password ^
    --backup-retention-period 3
```

O comando gerará uma saída semelhante à seguinte:

```
DBINSTANCE  mydbinstance  db.t3.large  sqlserver-se  250  sa  creating  3  ****  n
 10.50.2789
SECGROUP  default  active
PARAMGRP  default.sqlserver-se-14  in-sync
```

API do RDS

> **Note**
>
> Se quiser usar a licença do Db2 por meio do AWS Marketplace, é necessário primeiro assinar o AWS Marketplace e cadastrar-se na IBM usando o AWS Management Console. Para ter mais informações, consulte Assinar ofertas do Db2 Marketplace e cadastrá-las na IBM.

Para criar uma instância de banco de dados utilizando a API do Amazon RDS, chame a operação CreateDBInstance com os seguintes parâmetros.

Para obter informações sobre cada configuração, consulte Configurações para instâncias de banco de dados.

Configurações para instâncias de banco de dados

Na tabela a seguir, veja detalhes sobre as configurações que você escolhe ao criar uma instância de banco de dados. A tabela também exibe os mecanismos de banco de dados para cada configuração compatível.

É possível criar uma instância de banco de dados utilizando o console, o comando create-db-instance da CLI ou a operação da API CreateDBInstance do RDS.

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS	Mecanismos de banco de dados com suporte

Armazenamento alocado	O valor de armazenamento a ser alocado para a sua instância de banco de dados (em gigabytes). Em alguns casos, alocar uma quantidade de armazenamento para a instância de banco de dados maior do que o tamanho do banco de dados pode melhorar a performance de E/S.
	
	Para obter mais informações, consulte Armazenamento de instâncias de banco de dados do Amazon RDS.
		Opção da CLI:
		
		`--allocated-storage`
		
		Parâmetro da API:
		
		`AllocatedStorage`
			Tudo

Configurações de arquitetura	Se você escolher Use multitenant architecture (Usar arquitetura multilocatário), o RDS for Oracle criará um banco de dados de contêiner (CDB). Se você não escolher essa opção, o RDS for Oracle criará um não CDB. Um não CDB usa a arquitetura Oracle tradicional. Um não CDB não pode conter bancos de dados conectáveis (PDBs). O Oracle Database 21c usa somente arquitetura CDB. O Oracle Database 19c pode usar arquitetura CDB ou não CDB. Versões anteriores do Oracle Database 19c usam somente não CDB.	Opção da CLI: Oracle Multilocatário Oracle Multilocatário `--engine oracle-ee` (tradicional) `--engine oracle-se2` (tradicional) Parâmetro da API: `Engine`	Oracle

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS	Mecanismos de banco de dados com suporte
	Para obter mais informações, consulte Visão geral dos CDBs RDS para Oracle.		

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS	Mecanismos de banco de dados com suporte
Configuração de arquitetura	Essas configurações são válidas somente quando você escolhe a arquitetura multilocatária Oracle para configurações de arquitetura. Escolha uma das seguintes opções: • Na configuração multilocatária, a instância de CDB do RDS para Oracle pode conter bancos de dados de um a trinta locatários, dependendo da edição do banco de dados e de quaisquer licenças de opção necessárias. No contexto de um banco de dados Oracle, um banco de dados de inquilinos é um PDB. Não há suporte para PDBs de aplicativos e PDBs de proxy. Sua instância de banco de dados é criada com 1 banco de dados inicial de inquilinos. Escolha valores para nome do banco de dados do inquilino, nome de usuário mestre do banco de dados do inquilino, senha mestre do banco de dados do inquilino e conjunto de caracteres do banco de dados do inquilino. A configuração multilocatária é permanente. Portanto, você não pode	Opção da CLI: Configuração de locação múltipla Configuração de locatário único Parâmetro da API: `MultiTenant`	Oracle

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS	Mecanismos de banco de dados com suporte
	converter a configuração de vários locatários de volta para a configuração de inquilino único. A atualização de lançamento (RU) mínima com suporte para a configuração de vários locatários é 19.0.0.0.0.0.ru-2022.r1. **Note** A configuração do Amazon RDS é chamada de “multi-tenant”, em vez de “multitenant”, porque é um recurso da plataforma RDS, não apenas do mecanismo de banco de dados Oracle. Da mesma forma, o termo “locatário” do RDS refere-se a qualquer locatário em uma configuração do RDS, não apenas aos PDBs da Oracle. Na documentação do RDS, o termo “multilocatário da Oracle” (Oracle multitenant), sem hífen, refere-se exclusivamente à arquitetura CDB de banco de dados Oracle, que é compatível com implantações on-premises e do RDS.		

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS	Mecanismos de banco de dados com suporte
	• Com a configuração Single-tenant, seu RDS for Oracle CDB contém 1 PDB. Essa é a configuração padrão quando você cria um CDB. Você não pode excluir o PDB inicial nem adicionar mais PDBs. Posteriormente, você pode converter a configuração de inquilino único do seu CDB na configuração de vários locatários, mas não pode depois converter novamente para a configuração de inquilino único. Independentemente da configuração escolhida, seu CDB contém um único PDB inicial. Na configuração de vários locatários, você pode criar mais PDBs posteriormente usando as APIs do RDS. Para obter mais informações, consulte Visão geral dos CDBs RDS para Oracle.		

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS	Mecanismos de banco de dados com suporte

Atualização da versão secundária automática	Selecione Habilitar o upgrade automático da versão secundária para permitir que a instância de banco de dados receba automaticamente atualizações da versão do mecanismo de banco de dados secundário preferencial quando elas forem disponibilizadas. Esse é o comportamento padrão. O Amazon RDS executa atualizações automáticas de versões secundárias na janela de manutenção. Se você não selecionar Habilitar o upgrade automático da versão secundária, a instância de banco de dados não será atualizada automaticamente quando novas versões secundárias forem disponibilizadas.
	
	Para obter mais informações, consulte Atualizar automaticamente a versão do mecanismo espelho.
		Opção da CLI:
		
		`--auto-minor-version-upgrade`
		
		`--no-auto-minor-version-upgrade`
		
		Parâmetro da API:
		
		`AutoMinorVersionUpgrade`
			Todos

Availability zone	A zona de disponibilidade da sua instância de banco de dados. Use o valor padrão de No Preference (Sem preferência), a menos que você queira especificar uma zona de disponibilidade.
	
	Para obter mais informações, consulte Regiões, zonas de disponibilidade e Local Zones.
		Opção da CLI:
		
		`--availability-zone`
		
		Parâmetro da API:
		
		`AvailabilityZone`
			Todos

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS	Mecanismos de banco de dados com suporte

AWS KMS key	Disponível apenas quando Encryption (Criptografia) estiver definido como Enable encryption (Habilitar criptografia). Selecione a AWS KMS key a ser usada para criptografar esta instância de banco de dados. Para obter mais informações, consulte Criptografar recursos do Amazon RDS	Opção da CLI:
		`--kms-key-id`
		Parâmetro da API:
		`KmsKeyId`
			Todos

Configuração da AWS License Manager	Insira um nome para uma configuração da licença da AWS License Manager. O nome deve ter 100 caracteres ou menos e incluir apenas a-z, A-Z e 0-9. Para obter mais informações, consulte the section called “Integração com o AWS License Manager”.	Opção da CLI: Para obter mais informações, consulte AWS License Manager CLI. Parâmetro da API: Para obter mais informações, consulte API do AWS License Manager.	Db2

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS	Mecanismos de banco de dados com suporte
Backup replication (Replicação de backup)	Selecione Habilitar a replicação em outra região da AWS para criar backups em uma região adicional para recuperação de desastres. Em seguida, escolha a Destination Region (Região de destino) para os backups adicionais.	Não disponível ao criar uma instância de banco de dados. Para obter informações sobre como ativar backups entre regiões usando a AWS CLI ou a API do RDS, consulte Habilitar backups automatizados entre regiões.	Oracle PostgreSQL SQL Server

Período de retenção de backup	O número de dias que você deseja que os backups automáticos da sua instância de banco de dados fiquem retidos. Para instâncias de banco de dados incomuns, defina esse valor como **1** ou maior.
	
	Para obter mais informações, consulte Introdução aos backups.
		Opção da CLI:
		
		`--backup-retention-period`
		
		Parâmetro da API:
		
		`BackupRetentionPeriod`
			Todos

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS	Mecanismos de banco de dados com suporte

Destino de backup	Escolha Nuvem AWS para armazenar backups automatizados e snapshots manuais na região da AWS principal : Escolha Outposts (on-premises) para armazená-los localmente no seu Outpost.
	
	Essa configuração de opção se aplica somente ao RDS on Outposts. Para obter mais informações, consulteCriar instâncias de banco de dados para o Amazon RDS on AWS Outposts
		Opção da CLI:
		
		`--backup-target`
		
		Parâmetro da API:
		
		`BackupTarget`
			MySQL, PostgreSQL, SQL Server

Janela de backup	O período durante o qual o Amazon RDS faz um backup automático da sua instância de banco de dados. A menos que você tenha um horário específico no qual deseja que o backup do banco de dados seja feito, use o padrão No Preference (Sem preferência).
	
	Para ter mais informações, consulte Introdução aos backups.
		Opção da CLI:
		
		`--preferred-backup-window`
		
		Parâmetro da API:
		
		`PreferredBackupWindow`
			Todos

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS	Mecanismos de banco de dados com suporte

Autoridade certificadora	A autoridade de certificação (CA) para o certificado do servidor usado pela instância de banco de dados.
	
	Para obter mais informações, consulte Usar SSL/TLS para criptografar uma conexão com uma instância ou um cluster de banco de dados.
		Opção da CLI:
		
		`--ca-certificate-identifier`
		
		Parâmetro da API do RDS:
		
		`CACertificateIdentifier`
			Todos

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS	Mecanismos de banco de dados com suporte

Character set (Conjunto de caracteres)	O conjunto de caracteres da sua instância de banco de dados. O valor padrão AL32UTF8 para o conjunto de caracteres de banco de dados Unicode 5.0 UTF-8 Universal. Você não pode alterar o conjunto de caracteres de banco de dados depois de criar a instância de banco de dados.
	
	Em uma configuração de locatário único, um conjunto de caracteres de banco de dados não padrão afeta apenas o PDB, não o CDB. Para obter mais informações, consulte Configuração de locatário único da arquitetura CDB.
	
	O conjunto de caracteres de banco de dados é diferente do conjunto de caracteres nacional, que é chamado de conjunto de caracteres NCHAR. Ao contrário do conjunto de caracteres de banco de dados, o conjunto de caracteres NCHAR especifica a codificação para colunas de tipos de dados NCHAR (NCHAR, NVARCHAR2 e NCLOB) sem afetar os metadados do banco de dados.
	
	Para obter mais informações, consulte Conjuntos de caracteres do RDS for Oracle.
		Opção da CLI:
		
		`--character-set-name`
		
		Parâmetro da API:
		
		`CharacterSetName`
			Oracle

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS	Mecanismos de banco de dados com suporte

Collation (Agrupamento)	Um agrupamento em nível de servidor para a sua instância de banco de dados.
	
	Para obter mais informações, consulte Agrupamento no nível do servidor para o Microsoft SQL Server.
		Opção da CLI:
		
		`--character-set-name`
		
		Parâmetro da API:
		
		`CharacterSetName`
			SQL Server

Copiar tags para snapshots	Essa opção copia qualquer tag da instância de banco de dados para um snapshot de banco de dados quando você cria um snapshot.
	
	Para obter mais informações, consulte Marcar recursos do Amazon RDS.
		Opção da CLI:
		
		`--copy-tags-to-snapshot`
		
		`--no-copy-tags-to-snapshot`
		
		Parâmetro da API do RDS:
		
		`CopyTagsToSnapshot`
			Tudo

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS	Mecanismos de banco de dados com suporte

Autenticação de banco de dados	A opção de autenticação de banco de dados que você deseja usar.
	
	Selecione Password authentication (Autenticação de senha) para autenticar usuários de banco de dados somente com senhas de banco de dados.
	
	Selecione Password and IAM DB authentication (Senha e autenticação de banco de dados do IAM) para autenticar usuários de banco de dados com senhas de banco de dados e credenciais de usuário por meio de perfis e usuários. Para obter mais informações, consulte Autenticação do banco de dados do IAM para MariaDB, MySQL e PostgreSQL. Esta opção é compatível somente com MySQL e PostgreSQL.
	
	Escolha Password and Kerberos authentication (Senha e autenticação Kerberos) para autenticar usuários de banco de dados com senhas de banco de dados e a autenticação Kerberos pelo AWS Managed Microsoft AD criado com o AWS Directory Service. Depois, escolha o diretório ou selecione Create a new Directory (Criar um diretório).
		IAM:
		
		Opção da CLI:
		
		`--enable-iam-database-authentication`
		
		`--no-enable-iam-database-authentication`
		
		Parâmetro da API do RDS:
		
		`EnableIAMDatabaseAuthentication`
		
		Kerberos:
		
		Opção da CLI:
		
		`--domain`
		
		`--domain-iam-role-name`
		
		Parâmetro da API do RDS:
		
		`Domain`
		
		`DomainIAMRoleName`
			Varia por tipo de autenticação

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS	Mecanismos de banco de dados com suporte
	Para obter mais informações, consulte um dos seguintes: • Usar a autenticação do Kerberos para o Amazon RDS para Db2 • Uso da autenticação do Kerberos para o Amazon RDS para MySQL • Configurar a autenticação Kerberos do Amazon RDS para Oracle • Usar a autenticação Kerberos com o Amazon RDS para PostgreSQL		
Tipo de gerenciamento de banco de dados	Escolha Amazon RDS se não precisar personalizar seu ambiente. Escolha Amazon RDS Custom se quiser personalizar o banco de dados, o sistema operacional e a infraestrutura. Para obter mais informações, consulteAmazon RDS Custom	Para a CLI e a API, especifique o tipo de mecanismo de banco de dados.	Oracle SQL Server

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS	Mecanismos de banco de dados com suporte

Porta de banco de dados	A porta pela qual você deseja acessar a instância de banco de dados. A porta padrão é exibida.
	
	Note
	Os firewalls em algumas empresas bloqueiam conexões com as portas padrão do MariaDB, MySQL e PostgreSQL. Se o firewall da sua empresa bloquear a porta padrão, insira outra porta para a instância de banco de dados.
		Opção da CLI:
		
		`--port`
		
		Parâmetro da API do RDS:
		
		`Port`
			Todos

DB engine version (Versão do mecanismo de banco de dados)	A versão do mecanismo de banco de dados que você deseja usar.	Opção da CLI:
		
		`--engine-version`
		
		Parâmetro da API do RDS:
		
		`EngineVersion`
			Tudo

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS	Mecanismos de banco de dados com suporte

Classe de instância de banco de dados	A configuração da sua instância de banco de dados. Por exemplo, uma classe de instância de banco de dados db.t3.small tem 2 GiB de memória, 2 vCPUs, 1 núcleo virtual, uma ECU variável e uma capacidade de E/S moderada.
	
	Se possível, escolha uma classe de instância de banco de dados grande o suficiente para um conjunto de trabalho de consulta típico pode ser sustentado na memória. Quando os conjuntos de trabalho são mantidos na memória o sistema pode evitar a gravação em disco, o que aprimora a performance. Para obter mais informações, consulteClasses de instâncias de banco de dados
	
	No RDS para Oracle, você pode selecionar Include additional memory configurations (Incluir configurações de memória adicionais). Essas configurações são otimizadas para uma alta proporção de memória para vCPU. Por exemplo, db.r5.6xlarge.tpc2.mem4x é uma instância de banco de dados db.r5.8x que tem dois threads por núcleo (tpc2) e quatro vezes a memória de uma
		Opção da CLI:
		
		`--db-instance-class`
		
		Parâmetro da API do RDS:
		
		`DBInstanceClass`
			Tudo

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS	Mecanismos de banco de dados com suporte
	instância de banco de dados db.r5.6xlarge padrão. Para obter mais informações, consulte Classes de instância de banco de dados do RDS para Oracle		

DB instance identifier	O nome da sua instância de banco de dados. Nomeie suas instâncias de banco de dados da mesma forma que nomeia seus servidores no local. O identificador da instância do banco de dados pode conter até 63 caracteres alfanuméricos e deve ser exclusivo para a conta na região da AWS que você escolher.	Opção da CLI:
		`--db-instance-identifier`
		Parâmetro da API do RDS:
		`DBInstanceIdentifier`
			Tudo

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS	Mecanismos de banco de dados com suporte

Grupo de parâmetros de banco de dados	Um parameter group para a sua instância de banco de dados. Você pode escolher um grupo de parâmetros padrão ou criar o seu próprio grupo personalizado de parâmetros.
	
	Se você estiver usando o modelo BYOL para RDS para Db2, antes de criar uma instância de banco de dados, primeiro crie um grupo de parâmetros personalizado que contenha o IBM Site ID e o IBM Customer ID. Para obter mais informações, consulte Traga a sua própria licença para Db2.
	
	Para obter mais informações, consulte Grupos de parâmetros para Amazon RDS.
		Opção da CLI:
		
		`--db-parameter-group-name`
		
		Parâmetro da API do RDS:
		
		`DBParameterGroupName`
			Todos

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS	Mecanismos de banco de dados com suporte

Grupo de sub-redes de banco de dados	O grupo de sub-redes de banco de dados que você deseja usar para o cluster de banco de dados.
	Selecione Choose existing (Selecionar existente) para usar um grupo de sub-redes de banco de dados existente. Depois, escolha o grupo de sub-redes necessário na lista suspensa Existing DB subnet groups (Grupos de sub-redes de banco de dados existentes).
	Escolha Automatic setup (Configuração automática) para permitir que o RDS selecione um grupo de sub-redes de banco de dados compatível. Se não existir nenhum, o RDS criará um grupo de sub-redes para o cluster.
	Para ter mais informações, consulte Trabalhar com grupos de sub-redes de banco de dados.
		Opção da CLI:
		`--db-subnet-group-name`
		Parâmetro da API do RDS:
		`DBSubnetGroupName`
			Todos

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS	Mecanismos de banco de dados com suporte

Volume de registro dedicado	Use um volume de log dedicado (DLV) para armazenar registros de transações do banco de dados em um volume de armazenamento separado do volume que contém as tabelas do banco de dados.
	
	Para obter mais informações, consulte Usando um volume de registro dedicado (DLV).
		Opção da CLI:
		
		`--dedicated-log-volume`
		
		Parâmetro da API do RDS:
		
		`DedicatedLogVolume`
			Tudo

Deletion protection (Proteção contra exclusão)	Selecione Enable deletion protection (Habilitar proteção contra exclusão) para impedir que sua instância de banco de dados seja excluída. Por padrão, se você criar uma instância de banco de dados de produção com o AWS Management Console, a proteção contra exclusão será habilitada.
	
	Para obter mais informações, consulte Excluir uma instância de banco de dados.
		Opção da CLI:
		
		`--deletion-protection`
		
		`--no-deletion-protection`
		
		Parâmetro da API do RDS:
		
		`DeletionProtection`
			Tudo

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS	Mecanismos de banco de dados com suporte

Criptografia	Selecione Enable Encryption (Habilitar criptografia) para habilitar a criptografia em repouso para essa instância de banco de dados.
	
	Para obter mais informações, consulte Criptografar recursos do Amazon RDS.
		Opção da CLI:
		
		`--storage-encrypted`
		
		`--no-storage-encrypted`
		
		Parâmetro da API do RDS:
		
		`StorageEncrypted`
			Tudo

Monitoramento avançado	Selecione Enable enhanced monitoring (Habilitar o monitoramento avançado) para habilitar a coleta de métricas em tempo real do sistema operacional em que a instância de banco de dados é executada.
	
	Para obter mais informações, consulte Monitorar métricas do SO com o monitoramento avançado.
		Opções da CLI:
		
		`--monitoring-interval`
		
		`--monitoring-role-arn`
		
		Parâmetros da API do RDS:
		
		`MonitoringInterval`
		
		`MonitoringRoleArn`
			Todos

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS	Mecanismos de banco de dados com suporte

Tipo de mecanismo	Escolha o mecanismo de banco de dados a ser usado para essa instância de banco de dados.	Opção da CLI:
		`--engine`
		Parâmetro da API do RDS:
		`Engine`
			Todos

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS	Mecanismos de banco de dados com suporte

Nome do banco de dados inicial	O nome do banco de dados na sua instância de banco de dados. Se você não fornecer um nome, o Amazon RDS não criará um banco de dados na instância de banco de dados (exceto Oracle e PostgreSQL). O nome não pode ser uma palavra reservada pelo mecanismo de banco de dados e tem outras restrições dependendo do mecanismo de banco de dados.
	
	Db2:
	
	• Deve conter 1 – 8 caracteres alfanuméricos.
	• Ele deve começar com a-z, A-Z, @, $ ou # e ser seguido por a-z, A-Z, 0-9, _, @, # ou $.
	• Ele não pode conter espaços.
	• Para obter mais informações, consulte Considerações adicionais.
	
	MariaDB e MySQL:
	
	• Deve conter de 1 a 64 caracteres alfanuméricos.
		Opção da CLI:
		
		`--db-name`
		
		Parâmetro da API do RDS:
		
		`DBName`
			Todos, exceto o SQL Server

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS	Mecanismos de banco de dados com suporte
	Oracle: • Deve conter 1 – 8 caracteres alfanuméricos. • Não pode ser `NULL`. O valor padrão é `ORCL`. • Deve começar com uma letra. PostgreSQL: • Deve conter de 1 a 63 caracteres alfanuméricos. • Deve começar com uma letra ou um sublinhado. Os caracteres subsequentes podem ser letras, sublinhado ou dígitos (0 a 9). • O nome do banco de dados inicial é `postgres`.		

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS	Mecanismos de banco de dados com suporte

License (Licença)	Valores válidos para o modelo de licença:
	• bring-your-own-license ou marketplace-license para Db2.
	• general-public-license para MariaDB.
	• license-included para Microsoft SQL Server.
	• general-public-license para MySQL.
	• license-included ou bring-your-own-license para Oracle.
	• postgresql-license para PostgreSQL.
		Opção da CLI:
		`--license-model`
		Parâmetro da API do RDS:
		`LicenseModel`
			Todos

Exportações de log	Os tipos de arquivos de log de bancos de dados a serem publicados no Amazon CloudWatch Logs.
	Para ter mais informações, consulte Publicação de logs de banco de dados no Amazon CloudWatch Logs.
		Opção da CLI:
		`--enable-cloudwatch-logs-exports`
		Parâmetro da API do RDS:
		`EnableCloudwatchLogsExports`
			Tudo

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS	Mecanismos de banco de dados com suporte

Janela de manutenção	A janela de 30 minutos na qual as modificações pendentes da sua instância de banco de dados serão aplicadas. Se o período não for relevante, selecione No Preference (Sem preferência).
	
	Para ter mais informações, consulte A janela de manutenção do Amazon RDS.
		Opção da CLI:
		
		`--preferred-maintenance-window`
		
		Parâmetro da API do RDS:
		
		`PreferredMaintenanceWindow`
			Todos

Gerenciar credenciais principais no AWS Secrets Manager	Selecione Gerenciar credenciais principais no AWS Secrets Manager para gerenciar a senha do usuário principal em um segredo no Secrets Manager.
	
	Opcionalmente, selecione uma chave do KMS a ser usada para proteger o segredo. Escolha entre uma das chaves do KMS da sua conta ou insira a chave de uma conta distinta.
	
	Para ter mais informações, consulte Gerenciamento de senhas com Amazon RDS e AWS Secrets Manager.
		Opção da CLI:
		
		`--manage-master-user-password | --no-manage-master-user-password`
		
		`--master-user-secret-kms-key-id`
		
		Parâmetro da API do RDS:
		
		`ManageMasterUserPassword`
		
		`MasterUserSecretKmsKeyId`
			Tudo

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS	Mecanismos de banco de dados com suporte

Senha mestre	A senha da conta de usuário principal . A senha tem o seguinte número de caracteres ASCII imprimíveis (excluindo `/`, `"`, um espaço e `@`) dependendo do mecanismo de banco de dados:
	• Db2: 8–255
	• Oracle: 8–30
	• MariaDB e MySQL: 8–41
	• SQL Server e PostgreSQL: 8–128
		Opção da CLI:
		`--master-user-password`
		Parâmetro da API do RDS:
		`MasterUserPassword`
			Tudo

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS	Mecanismos de banco de dados com suporte

Nome do usuário principal	O nome que você usa como o nome de usuário principal para fazer login na instância de banco de dados com todos os privilégios do banco de dados. Observe as seguintes restrições:
	• Pode conter 1–16 caracteres alfanuméricos e sublinhados.
	• O primeiro caractere deve ser uma letra.
	• O nome não pode ser uma palavra reservada pelo mecanismo de banco de dados.
	
	Não é possível alterar o nome de usuário principal depois que a instância de banco de dados está criada.
	
	Para o Db2, recomendamos usar o mesmo nome de usuário principal do nome da instância do Db2 autogerenciado.
	
	Para ter mais informações sobre os privilégios concedidos ao usuário primário, consulte Privilégios da conta de usuário mestre.
		Opção da CLI:
		`--master-username`
		
		Parâmetro da API do RDS:
		`MasterUsername`
			Tudo

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS	Mecanismos de banco de dados com suporte

Microsoft SQL Server Windows Authentication	Enable Microsoft SQL Server Windows authentication (Habilitar a autenticação do Microsoft SQL Server Windows) e Browse Directory (Procurar no diretório) para escolher o diretório no qual deseja permitir que usuários autorizados do domínio autentiquem esta instância do SQL Server usando a Autenticação do Windows.	Opções da CLI:
		`--domain`
		`--domain-iam-role-name`
		Parâmetros da API do RDS:
		`Domain`
		`DomainIAMRoleName`
			SQL Server

implantação multi-AZ	Create a standby instance (Criar uma instância em espera) para criar uma réplica secundária passiva da instância de banco de dados em outra zona de disponibilidade para oferece suporte a failover. Recomendamos o multi-AZ para workloads de produção a fim de manter a alta disponibilidade.
	Para desenvolvimento e teste, você pode selecionar Do not create a standby instance (Não criar uma instância em espera).
	Para obter mais informações, consulte Configurar e gerenciar uma implantação multi-AZ para o Amazon RDS.
		Opção da CLI:
		`--multi-az`
		`--no-multi-az`
		Parâmetro da API do RDS:
		`MultiAZ`
			Tudo

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS	Mecanismos de banco de dados com suporte

Conjunto de caracteres nacionais (NCHAR)	O conjunto de caracteres nacionais para sua instância de banco de dados, comumente chamado de conjunto de caracteres NCHAR. Você pode definir o conjunto de caracteres nacionais como AL16UTF16 (padrão) ou UTF-8. Você não pode alterar o conjunto de caracteres nacionais depois de criar a instância de banco de dados.
	
	O conjunto de caracteres nacionais é diferente do conjunto de caracteres de banco de dados. Ao contrário do conjunto de caracteres de banco de dados, o conjunto de caracteres nacionais especifica a codificação apenas para as colunas de tipos de dados NCHAR (NCHAR, NVARCHAR2 e NCLOB) sem afetar os metadados do banco de dados.
	
	Para obter mais informações, consulte Conjuntos de caracteres do RDS for Oracle.
		Opção da CLI:
		
		`--nchar-character-set-name`
		
		Parâmetro da API:
		
		`NcharCharacterSetName`
			Oracle

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS	Mecanismos de banco de dados com suporte

Tipo de rede	Os protocolos de endereçamento IP compatíveis com a instância de banco de dados.
	
	IPv4 (o padrão) para especificar que os recursos podem se comunicar com a instância de banco de dados somente por meio do protocolo de endereçamento Internet Protocol versão 4 (IPv4).
	
	Modo de pilha dupla para especificar que os recursos podem se comunicar com a instância de banco de dados por IPv4, Internet Protocol versão 6 (IPv6) ou ambos. Use o modo de pilha dupla se você tiver algum recurso que precise se comunicar com sua instância de banco de dados pelo protocolo de endereçamento IPv6. Além disso, associe um bloco CIDR IPv6 a todas as sub-redes no grupo de sub-redes de banco de dados especificado.
	
	Para obter mais informações, consulteEndereçamento IP do Amazon RDS
		Opção da CLI:
		
		`--network-type`
		
		Parâmetro da API do RDS:
		
		`NetworkType`
			Todos

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS	Mecanismos de banco de dados com suporte

Option group	Um grupo de opções para a sua instância de banco de dados. Você pode escolher um grupo de opções padrão ou criar o seu próprio grupo personalizado de opções.
	
	Para obter mais informações, consulte Trabalhar com grupos de opções.
		Opção da CLI:
		
		`--option-group-name`
		
		Parâmetro da API do RDS:
		
		`OptionGroupName`
			Tudo

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS	Mecanismos de banco de dados com suporte

Performance Insights	Selecione Enable Performance Insights (Habilitar Performance Insights) para monitorar a carga da instância de banco de dados do para que você possa analisar e solucionar problemas relacionados à performance do banco de dados.
	
	Escolha um período de retenção para determinar a quantidade de histórico de dados do Performance Insights que deve ser mantida. A configuração de retenção no nível gratuito é Default (7 days) (Padrão (7 dias)). Para reter seus dados de performance por mais tempo, especifique entre 1 e 24 meses. Para obter mais informações sobre os períodos de retenção, consulte Preços e retenção de dados para o Performance Insights.
	
	Escolha uma chave do KMS a ser usada para proteger a chave usada para criptografar esse volume de banco de dados. Escolha entre uma das chaves do KMS da sua conta ou insira a chave de uma conta distinta.
	
	Para obter mais informações, consulteMonitorar a carga de banco de
		Opções da CLI:
		
		`--enable-performance-insights`
		
		`--no-enable-performance-insights`
		
		`--performance-insights-retention-period`
		
		`--performance-insights-kms-key-id`
		
		Parâmetros da API do RDS:
		
		`EnablePerformanceInsights`
		
		`PerformanceInsightsRetentionPeriod`
		
		`PerformanceInsightsKMSKeyId`
			Todos, exceto o Db2

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS	Mecanismos de banco de dados com suporte
	dados com o Performance Insights no Amazon RDS		

IOPS provisionadas	As IOPS provisionadas (operações de E/S por segundo) para a instância de banco de dados. Essa configuração estará disponível somente se você escolher uma das seguintes opções para Storage type (Tipo de armazenamento):
	• General purpose SSD (gp3) (SSD de uso geral (gp3))
	• Provisioned IOPS SSD (io1) (SSD de IOPS provisionadas (io1))
	• SSD de IOPS provisionadas (io2)
	
	Para obter mais informações, consulte Armazenamento de instâncias de banco de dados do Amazon RDS.
		Opção da CLI:
		`--iops`
		Parâmetro da API do RDS:
		`Iops`
			Todos

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS	Mecanismos de banco de dados com suporte

Acesso público	Yes (Sim) para fornecer à instância de banco de dados um endereço IP público, o que significa que ela é acessível fora da VPC. Para ser acessível publicamente, a instância de banco de dados também deve estar em uma sub-rede pública na VPC.
	
	No (Não) para tornar a instância de banco de dados acessível apenas a partir da VPC.
	
	Para obter mais informações, consulte Ocultar uma instância de banco de dados em uma VPC da Internet.
	
	Para se conectar a uma instância de banco de dados de fora de sua VPC, a instância de banco de dados deve estar acessível publicamente. Além disso, o acesso deve ser concedido usando as regras de entrada do grupo de segurança da instância de banco de dados. Além disso, outros requisitos devem ser atendidos. Para obter mais informações, consulte Não é possível conectar-se à instância de banco de dados do Amazon RDS.
	
	Se sua instância de banco de dados não estiver acessível ao público, use
		Opção da CLI:
		
		`--publicly-accessible`
		
		`--no-publicly-accessible`
		
		Parâmetro da API do RDS:
		
		`PubliclyAccessible`
			Todos

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS	Mecanismos de banco de dados com suporte
	uma conexão AWS Site-to-Site VPN ou uma conexão do AWS Direct Connect para acessá-la de uma rede privada. Para obter mais informações, consulte Privacidade do tráfego entre redes.		

Suporte estendido do RDS	Selecione Habilitar Suporte estendido do RDS para permitir que as versões principais do mecanismo compatíveis continuem funcionando após a data de término do suporte padrão do RDS.
	
	Quando você cria uma instância de banco de dados, o Amazon RDS usa como padrão o Suporte estendido do RDS. Para evitar a criação de uma instância de banco de dados após a data de fim do suporte padrão do RDS e para evitar cobranças pelo Suporte estendido do RDS, desabilite essa configuração. As instâncias de banco de dados existentes não incorrerão em cobranças até a data de início dos preços do Suporte estendido do RDS.
	
	Para obter mais informações, consulte Suporte estendido do Amazon RDS com Amazon RDS.
		Opção da CLI:
		
		`--engine-lifecycle-support`
		
		Parâmetro da API do RDS:
		
		`EngineLifecycleSupport`
			MySQL
			
			PostgreSQL

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS	Mecanismos de banco de dados com suporte
RDS Proxy	Selecione Create an RDS Proxy (Criar um RDS Proxy) para criar um proxy para sua instância de banco de dados. O Amazon RDS cria automaticamente um perfil do IAM e um segredo do Secrets Manager para o proxy. Para obter mais informações, consulte Amazon RDS Proxy.	Não disponível ao criar uma instância de banco de dados.	MariaDB MySQL PostgreSQL

Storage autoscaling (Autoscaling do armazenamento)	Enable storage autoscaling (Habilitar o autoscaling do armazenamento) para permitir que o Amazon RDS aumente automaticamente o armazenamento quando necessário para evitar que a instância de banco de dados seja executada sem espaço de armazenamento.
	
	Use Maximum storage threshold (Limite máximo de armazenamento) para definir o limite máximo para o Amazon RDS aumentar automaticamente o armazenamento da instância de banco de dados. O padrão é de 1.000 GiB.
	
	Para obter mais informações, consulte Gerenciar a capacidade automaticamente com a escalabilidade automática de armazenamento do Amazon RDS.
		Opção da CLI:
		
		`--max-allocated-storage`
		
		Parâmetro da API do RDS:
		
		`MaxAllocatedStorage`
			Todos

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS	Mecanismos de banco de dados com suporte

Storage throughput (Throughput de armazenamento)	O valor do throughput de armazenamento da instância de banco de dados. Essa configuração estará disponível somente se você escolher General purpose SSD (gp3) (SSD de uso geral (gp3)) para Storage type (Tipo de armazenamento).
	
	Para obter mais informações, consulte Armazenamento gp3 (recomendado).
		Opção da CLI:
		
		`--storage-throughput`
		
		Parâmetro da API do RDS:
		
		`StorageThroughput`
			Tudo

Tipo de armazenamento	O tipo de armazenamento da sua instância de banco de dados.
	
	Se você selecionar General Purpose SSD (gp3) (SSD de uso geral (gp3)), poderá provisionar IOPS provisionadas adicionais e throughput de armazenamento em Advanced settings (Configurações avançadas).
	
	Se você selecionar SSD de IOPS provisionadas (io1) ou SSD de IOPS provisionadas (io2), insira o valor IOPS provisionadas.
	
	Para obter mais informações, consulte Tipos de armazenamento do Amazon RDS.
		Opção da CLI:
		
		`--storage-type`
		
		Parâmetro da API do RDS:
		
		`StorageType`
			Tudo

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS	Mecanismos de banco de dados com suporte

Grupo de sub-redes	Um grupo de sub-redes de banco de dados a ser associado a essa instância de banco de dados.
	
	Para obter mais informações, consulte Trabalhar com grupos de sub-redes de banco de dados
		Opção da CLI:
		
		`--db-subnet-group-name`
		
		Parâmetro da API do RDS:
		
		`DBSubnetGroupName`
			Todos

Nome do banco de dados do inquilino	O nome do seu PDB inicial na configuração multilocatária da arquitetura Oracle. Essa configuração estará disponível somente se você escolher Configuração multilocatária para configuração de arquitetura.
	
	O nome do banco de dados do inquilino deve ser diferente do nome do seu CDB, que é nomeado. `RDSCDB` Não é possível mudar o nome de uma pilha.
		Opção da CLI:
		
		`--db-name`
		
		Parâmetro da API do RDS:
		
		`DBName`
			Oracle

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS	Mecanismos de banco de dados com suporte

Nome de usuário principal do banco de dados do inquilino	O nome que você usa como o nome de usuário primário para fazer login no cluster de banco de dados com todos os privilégios do banco de dados. Essa configuração estará disponível somente se você escolher Configuração multilocatária para configuração de arquitetura.
	
	Observe as seguintes restrições:
	
	• Pode conter 1–16 caracteres alfanuméricos e sublinhados.
	• O primeiro caractere deve ser uma letra.
	• O nome não pode ser uma palavra reservada pelo mecanismo de banco de dados.
	
	Você não pode fazer o seguinte:
	
	• Altere o nome de usuário principal do inquilino depois de criar o banco de dados do inquilino.
	• Faça login com o nome de usuário do tenant master no CDB.
		Opção da CLI:
		
		`--master-username`
		
		Parâmetro da API do RDS:
		
		`MasterUsername`
			Oracle

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS	Mecanismos de banco de dados com suporte

Senha mestre do banco de dados do inquilino	A senha da conta de usuário principal do banco de dados de locatários (PDB). Essa configuração estará disponível somente se você escolher Configuração multilocatária para configuração de arquitetura.
	
	A senha tem de 8 a 30 caracteres ASCII imprimíveis, excluindo,, um espaço / e. " @
		Opção da CLI:
		
		`--master-password`
		
		Parâmetro da API do RDS:
		
		`MasterPassword`
			Oracle

Conjunto de caracteres do banco de dados do inquilino	O conjunto de caracteres do banco de dados inicial do inquilino. Essa configuração estará disponível somente se você escolher Configuração multilocatária para configuração de arquitetura. Somente instâncias de CDB do RDS para Oracle são suportadas.
	
	O valor padrão AL32UTF8 para o conjunto de caracteres de banco de dados Unicode 5.0 UTF-8 Universal . Você pode escolher um conjunto de caracteres do banco de dados do inquilino que seja diferente do conjunto de caracteres do CDB.
	
	Para obter mais informações, consulte Conjuntos de caracteres do RDS for Oracle.
		Opção da CLI:
		
		`--character-set-name`
		
		Parâmetro da API do RDS:
		
		`CharacterSetName`
			Oracle

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS	Mecanismos de banco de dados com suporte

Conjunto de caracteres nacionais do banco de dados do inquilino	O conjunto de caracteres nacionais para sua instância de banco de dados, comumente chamado de conjunto de caracteres NCHAR. Essa configuração estará disponível somente se você escolher Configuração multilocatária para configuração de arquitetura. Somente instâncias de CDB do RDS para Oracle são suportadas.
	
	Você pode definir o conjunto de caracteres nacionais como AL16UTF16 (padrão) ou UTF-8. Não é possível alterar o conjunto de caracteres nacionais depois de criar o banco de dados.
	
	O conjunto de caracteres nacionais é diferente do conjunto de caracteres de banco de dados. O conjunto de caracteres nacionais especifica a codificação somente para colunas que usam o tipo de NCHAR dados (NCHAR,NVARCHAR2 , eNCLOB) e não afeta os metadados do banco de dados.
	
	Para obter mais informações, consulte Conjuntos de caracteres do RDS for Oracle.
		Opção da CLI:
		
		`--nchar-character-set-name`
		
		Parâmetro da API:
		
		`NcharCharacterSetName`
			Oracle

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS	Mecanismos de banco de dados com suporte

Time zone (Fuso horário)	O fuso horário da sua instância de banco de dados. Se você não escolher um fuso horário, a instância de banco de dados usará o fuso horário padrão. Não é possível alterar o fuso horário depois que a instância de banco de dados é criada.
	
	Para obter mais informações, consulte Fuso horário local para instâncias de banco de dados do Amazon RDS para Db2 e Fuso horário local para instâncias de banco de dados do Microsoft SQL Server.
		Opção da CLI:
		
		`--timezone`
		
		Parâmetro da API do RDS:
		
		`Timezone`
			Db2
			
			SQL Server
			
			RDS Custom para SQL Server:

Virtual Private Cloud (VPC)	Uma VPC baseada no serviço da Amazon VPC para associar a essa instância de banco de dados. Para obter mais informações, consulte Amazon VPC e Amazon RDS.	Para a CLI e a API, especifique os IDs do grupo de segurança da VPC.	Todos

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS	Mecanismos de banco de dados com suporte

Grupo de segurança da VPC (firewall)	O grupo de segurança a ser associado à instância de banco de dados.
	
	Para obter mais informações, consulteVisão geral dos grupos de segurança de VPC
		Opção da CLI:
		
		`--vpc-security-group-ids`
		
		Parâmetro da API do RDS:
		
		`VpcSecurityGroupIds`
			Todos

Criar recursos do Amazon RDS com o AWS CloudFormation

O Amazon RDS é integrado ao AWS CloudFormation, um serviço que ajuda você a modelar e configurar os recursos da AWS para que você possa passar menos tempo criando e gerenciando os recursos e a infraestrutura. Você cria um modelo que descreve todos os recursos AWS desejados (como instâncias de banco de dados e grupos de parâmetros de banco de dados), e o AWS CloudFormation provisiona e configura esses recursos para você.

Ao usar o AWS CloudFormation, você poderá reutilizar o modelo para configurar os recursos do RDS de forma repetida e consistente. Descreva seus recursos uma vez e, depois, provisione os mesmos recursos repetidamente em várias contas e regiões da AWS.

RDS e modelos do AWS CloudFormation

Modelos do AWS CloudFormation são arquivos de texto formatados em JSON ou YAML. Esses modelos descrevem os atributos que você deseja provisionar nas suas pilhas AWS CloudFormation. Se não estiver familiarizado com JSON ou YAML, você pode usar AWS CloudFormation Designer para ajudá-lo a começar a usar os modelos AWS CloudFormation. Para obter mais informações, consulte O que é o AWS CloudFormation Designer? no Guia do usuário do AWS CloudFormation.

O RDS é compatível com a criação de recursos no AWS CloudFormation. Para obter mais informações, incluindo exemplos de modelos JSON e YAML para esses recursos, consulte Referência de tipo de recurso do RDS no Guia do usuário do AWS CloudFormation.

Saiba mais sobre o AWS CloudFormation

Para mais sobre o AWS CloudFormation, consulte os seguintes atributos:

- AWS CloudFormation
- Guia do usuário do AWS CloudFormation
- Referência de API AWS CloudFormation
- Guia do Usuário da Interface de Linha de ComandoAWS CloudFormation

Conectar a uma instância de banco de dados do Amazon RDS

Antes de se conectar a uma instância de banco de dados, você deve criar a instância de banco de dados. Para obter mais informações, consulte Criar uma instância de banco de dados do Amazon RDS. Depois que o Amazon RDS provisionar sua instância de banco de dados, use qualquer aplicação ou utilitário cliente padrão para o mecanismo de banco de dados se conectar à instância de banco de dados. Na string de conexão, especifique o endereço DNS do endpoint de instância de banco de dados como o parâmetro de host. Além disso, especifique o número da porta do endpoint da instância de banco de dados como o parâmetro da porta.

Para ter mais detalhes sobre como encontrar informações de conexão para uma instância de banco de dados do Amazon RDS ou cenários para acessar uma instância de banco de dados em uma VPC, consulte os tópicos a seguir.

- Localizando as informações de conexão para uma instância de Amazon RDS banco de dados
- Cenários para acessar uma instância de banco de dados em uma VPC

Localizando as informações de conexão para uma instância de Amazon RDS banco de dados

As informações de conexão de uma instância de banco de dados incluem seu endpoint, porta e um usuário de banco de dados válido, como o usuário mestre. Por exemplo, para uma instância de banco de dados MySQL, suponha que o valor do endpoint seja `mydb.123456789012.us-east-1.rds.amazonaws.com`. Nesse caso, o valor da porta é `3306`, e o usuário do banco de dados é `admin`. Com essas informações, você especifica os seguintes valores em uma string de conexão:

- Para o host, nome de host ou nome DNS, especifique `mydb.123456789012.us-east-1.rds.amazonaws.com`.
- Para a porta, especifique `3306`.
- Para usuário, especifique `admin`.

O endpoint é exclusivo para cada instância de banco de dados e os valores da porta e do usuário podem variar. A seguinte lista mostra a porta mais comum para cada mecanismo de banco de dados:

- Db2: 50000

- MariaDB – 3306
- Microsoft SQL Server – 1433
- MySQL – 3306
- Oracle – 1521
- PostgreSQL – 5432

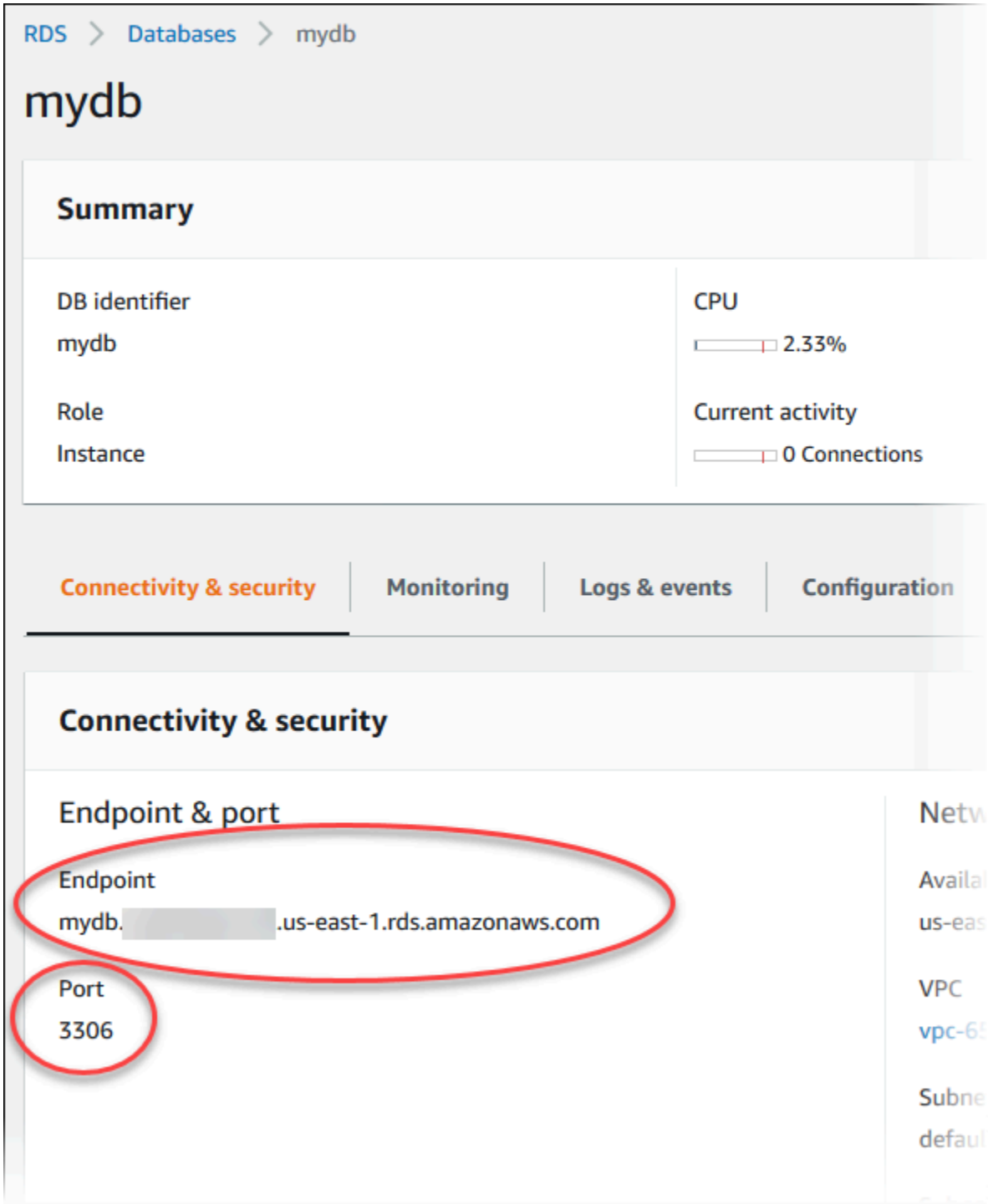
Para se conectar a uma instância de banco de dados, use qualquer cliente para um mecanismo de banco de dados. Por exemplo, você pode usar o utilitário mysql para se conectar a uma instância de banco de dados MariaDB ou MySQL. Você pode usar o Microsoft SQL Server Management Studio para se conectar a uma instância de banco de dados SQL Server. Você pode usar o Oracle SQL Developer para se conectar a uma instância de banco de dados Oracle. Da mesma forma, você pode usar uma instância local do utilitário de linha de comando psql para se conectar a uma instância de banco de dados do PostgreSQL.

Para encontrar as informações de conexão de uma instância de banco de dados, use o AWS Management Console Você também pode usar o comando describe-db-instances da AWS Command Line Interface (AWS CLI) ou a operação DescribeDBInstances da API do RDS.

Console

Para encontrar as informações de conexão de uma instância de banco de dados no AWS Management Console

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Bancos de dados para exibir uma lista de suas instâncias de banco de dados.
3. Escolha o nome da instância de banco de dados para mostrar os detalhes.
4. Na guia Connectivity & security (Conectividade e segurança), copie o endpoint. Além disso, anote o número da porta. Você precisará do endpoint e do número da porta para conectar-se à instância de banco de dados.



5. Se você precisar encontrar o nome de usuário mestre, escolha a guia Configuração e visualize o valor do nome de usuário mestre .

AWS CLI

Para encontrar as informações de conexão de uma instância de banco de dados usando o AWS CLI, chame o comando describe-db-instances . Na chamada, consulte o ID da instância de banco de dados, o endpoint, a porta e o nome de usuário mestre.

Para Linux, macOS ou Unix:

```
aws rds describe-db-instances \
  --query "*[].[DBInstanceIdentifier,Endpoint.Address,Endpoint.Port,MasterUsername]"
```

Para Windows:

```
aws rds describe-db-instances ^
  --query "*[].[DBInstanceIdentifier,Endpoint.Address,Endpoint.Port,MasterUsername]"
```

Sua saída deve ser similar à seguinte.

```
[
    [
        "mydb",
        "mydb.123456789012.us-east-1.rds.amazonaws.com",
        3306,
        "admin"
    ],
    [
        "myoracledb",
        "myoracledb.123456789012.us-east-1.rds.amazonaws.com",
        1521,
        "dbadmin"
    ],
    [
        "mypostgresqldb",
        "mypostgresqldb.123456789012.us-east-1.rds.amazonaws.com",
        5432,
        "postgresadmin"
    ]
]
```

API do RDS

Para localizar as informações de conexão de uma instância de banco de dados usando a Amazon RDS API, chame a operação DescribeDBInstances . Na saída, encontre os valores para o endereço do endpoint, a porta do endpoint e o nome do usuário mestre.

Cenários para acessar uma instância de banco de dados em uma VPC

Usando o Amazon Virtual Private Cloud (Amazon VPC) você pode executar recursos da AWS, como instâncias de bancos de dados do Amazon RDS, em uma nuvem privada virtual (VPC). Ao usar uma Amazon VPC, você tem controle sobre o ambiente de rede virtual. É possível escolher seu próprio intervalo de endereços IP, criar sub-redes e configurar o roteamento e listas de controle de acesso.

Um grupo de segurança da VPC controla o acesso a uma instância de banco de dados dentro de uma VPC. Cada regra de grupo de segurança de VPC permite que uma origem específica acesse uma instância de banco de dados em uma VPC que esteja associada a esse grupo de segurança de VPC. A origem pode ser uma gama de endereços (por exemplo, 203.0.113.0/24) ou outro grupo de segurança da VPC. Ao especificar um grupo de segurança de VPC como origem, você permite o tráfego recebido de todas as instâncias (geralmente servidores de aplicativos) que usam o grupo de segurança de VPC de origem.

Antes de tentar se conectar à sua instância de banco de dados, configure a VPC para o caso de uso. Estes são alguns cenários comuns para acessar uma instância de banco de dados em uma VPC:

- Uma instância de banco de dados em uma VPC acessada por uma instância Amazon EC2 na mesma VPC – Um uso comum de uma instância de banco de dados em uma VPC é compartilhar dados com um servidor de aplicações que estiver sendo executada em uma instância do EC2 na mesma VPC. A instância do EC2 pode executar um servidor Web com uma aplicação que interaja com a instância de banco de dados.
- Uma instância de banco de dados em uma VPC acessada por uma instância do EC2 em uma VPC diferente: em alguns casos, sua instância de banco de dados está em uma VPC diferente da instância do EC2 que você está usando para acessá-la. Nesse caso, você pode usar o emparelhamento de VPC para acessar a instância de banco de dados.
- Uma instância de banco de dados em uma VPC acessada por uma aplicação cliente pela Internet: para acessar uma instância de banco de dados em uma VPC de uma aplicação cliente pela Internet, configure uma VPC com uma única sub-rede pública. Não há nenhum gateway da Internet para permitir comunicação pela Internet.

Para se conectar a uma instância de banco de dados de fora de sua VPC, a instância de banco de dados deve estar acessível publicamente. Além disso, o acesso deve ser concedido usando as regras de entrada do grupo de segurança da instância de banco de dados, e os outros requisitos devem ser atendidos. Para ter mais informações, consulte Não é possível conectar-se à instância de banco de dados do Amazon RDS.

- Uma instância de banco de dados em uma VPC acessada por uma rede privada: se sua instância de banco de dados não estiver acessível ao público, você poderá usar uma das seguintes opções para acessá-la de uma rede privada:
  - Uma conexão do AWS Site-to-Site VPN.
  - Uma conexão do AWS Direct Connect.
  - Uma conexão do AWS Client VPN.

Para ter mais informações, consulte Cenários para acessar uma instância de banco de dados em uma VPC.

Conectar-se a instâncias de banco de dados com os drivers da AWS

O pacote de drivers da AWS foram projetados para comportar tempos mais rápidos de transição e de failover, além de autenticação com o AWS Secrets Manager, o AWS Identity and Access Management (IAM) e identidades federadas. Os drivers da AWS dependem do monitoramento do status da instância de banco de dados e do conhecimento da topologia da instância para determinar a nova instância primária. Essa abordagem reduz os tempos de transição e de failover para segundos de um dígito, em comparação com dezenas de segundos para drivers de código aberto.

A tabela a seguir lista os recursos aceitos para cada um dos drivers. Como novos recursos do serviço são introduzidos, o objetivo do pacote de drivers da AWS é ter suporte integrado para esses recursos do serviço.

Atributo	Driver JDBC da AWS	Driver Python da AWS	Driver ODBC da AWS para MySQL
Suporte a failover	Sim	Sim	Sim
Monitoramento aprimorado de failover	Sim	Sim	Sim

Atributo	Driver JDBC da AWS	Driver Python da AWS	Driver ODBC da AWS para MySQL
Divisão de leituras/ gravações	Sim	Sim	Não
Conexão de metadados do driver	Sim	N/D	N/D
Telemetria	Sim	Sim	Não
Secrets Manager	Sim	Sim	Sim
Autenticação do IAM	Sim	Sim	Sim
Identidades federadas (AD FS)	Sim	Sim	Não
Identidade federada (Okta)	Sim	Não	Não
Clusters de banco de dados multi-AZ	Sim	Sim	Não

Para ter mais informações sobre os drivers da AWS, consulte o driver de linguagem correspondente para a instância de banco de dados do RDS para MariaDB, RDS para MySQL ou RDS para PostgreSQL.

> **Note**
>
> Os únicos recursos compatíveis com o RDS para MariaDB são a autenticação com o AWS Secrets Manager, o AWS Identity and Access Management (IAM) e a identidade federada.

Conectar-se a uma instância de banco de dados que está executando um mecanismo de banco de dados específico

Para saber como se conectar a uma instância de banco de dados que esteja executando um mecanismo de banco de dados específico, siga as instruções do mecanismo de banco de dados:

- RDS para Db2
- RDS para MariaDB
- RDS para SQL Server
- RDS para MySQL
- RDS para Oracle
- RDS para PostgreSQL

Gerenciar conexões com o RDS Proxy

Também é possível usar o Amazon RDS Proxy para gerenciar conexões com instâncias de banco de dados do RDS para MariaDB, do RDS para Microsoft SQL Server, do RDS para MySQL e do RDS para PostgreSQL. O RDS Proxy permite que as aplicações agrupem e compartilhem conexões de banco de dados para melhorar a escalabilidade. Para ter mais informações, consulte Amazon RDS Proxy.

Opções de autenticação do banco de dados

Amazon RDS suporta as seguintes maneiras de autenticar usuários do banco de dados:

- Com a autenticação com senha, –sua instância de banco de dados executa toda a administração de contas do usuário. Você cria usuários e especifica senhas com instruções SQL. As instruções SQL que você pode usar dependem do mecanismo de banco de dados.
- Autenticação de banco de dados do AWS Identity and Access Management (IAM): você não precisa usar uma senha ao se conectar a uma instância de banco de dados. Em vez disso, você usa um token de autenticação.
- Autenticação Kerberos – Você usa autenticação externa de usuários de banco de dados usando Kerberos e Microsoft Active Directory. O Kerberos é um protocolo de autenticação de rede que usa tíquetes e criptografia de chave simétrica para eliminar a necessidade de transmitir senhas pela rede. O Kerberos foi integrado ao Active Directory e foi projetado para autenticar usuários em recursos de rede, como bancos de dados.

IAM autenticação de banco de dados e autenticação Kerberos estão disponíveis apenas para mecanismos de banco de dados específicos e versões.

Para obter mais informações, consulte Autenticação do banco de dados com Amazon RDS.

Conexões criptografadas

É possível usar o Security Socket Layer (SSL) ou o Transport Layer Security (TLS) na aplicação para criptografar uma conexão com a uma instância de banco de dados. Cada mecanismo de banco de dados tem seu próprio processo de implementação do SSL/TLS. Para ter mais informações, consulte Usar SSL/TLS para criptografar uma conexão com uma instância ou um cluster de banco de dados.

Trabalhar com grupos de opções

Alguns mecanismos de banco de dados oferecem recursos adicionais que facilitam o gerenciamento de dados e de bancos de dados e oferecem segurança adicional ao seu banco de dados. O Amazon RDS usa grupos de opções para habilitar e configurar esses recursos. Um grupo de opções pode especificar recursos, chamados de opções, que estão disponíveis para uma determinada instância de banco de dados do Amazon RDS. Opções podem ter configurações que especificam como a opção funciona. Quando você associa uma instância de banco de dados a um grupo de opções, as opções e as configurações de opções especificadas são habilitadas para essa instância de banco de dados.

O Amazon RDS oferece suporte para os seguintes mecanismos de banco de dados:

Mecanismo do banco de dados	Documentação relevante
`Db2`	Opções para instâncias de banco de dados do RDS para Db2
`MariaDB`	Opções para o mecanismo de banco de dados do MariaDB
`Microsoft SQL Server`	Opções para o mecanismo de banco de dados do Microsoft SQL Server
`MySQL`	Opções para instâncias de bancos de dados MySQL
`Oracle`	Adição de opções a instâncias de banco de dados Oracle
`PostgreSQL`	O PostgreSQL não usa opções e grupos de opções. O PostgreSQL usa extensões e módulos para fornecer recursos adicionais. Para obter mais informações, consulte Versões de extensões do PostgreSQL compatíveis.

Visão geral de grupos de opções

O Amazon RDS fornece um grupo de opções padrão vazio para cada nova instância de banco de dados. Não é possível modificar ou excluir esse grupo de opções padrão, mas qualquer novo grupo de opções que você criar derivará suas configurações do grupo de opções padrão. Para aplicar uma opção a uma instância de banco de dados, você deve fazer o seguinte:

1. Crie um novo grupo de opções, ou copie ou modifique um existente.
2. Adicionar uma ou mais opções ao grupo de opções.
3. Associe o grupo de opções à instância de banco de dados.

   Para associar um grupo de opções a uma instância de banco de dados, modifique a instância de banco de dados. Para obter mais informações, consulte Modificar uma instância de banco de dados do Amazon RDS.

Tanto as instâncias de banco de dados quanto os snapshot de banco de dadoss podem ser associados a um grupo de opções. Em alguns casos, você pode restaurar a partir de um snapshot de banco de dados ou pode realizar uma restauração pontual time para uma instância de banco de dados. Nesses casos, o grupo de opções associado ao snapshot de banco de dados ou à instância de banco de dados está, por padrão, associado à instância de banco de dados restaurada. Você pode associar um grupo de opções diferente a uma instância de banco de dados restaurada. No entanto, o novo grupo de opções deve conter quaisquer opções persistentes ou permanentes que foram incluídas no grupo de opções original. As opções persistentes e permanentes estão descritas a seguir.

As opções exigem memória adicional para serem executadas em uma instância de banco de dados. Portanto, pode ser necessário iniciar uma instância maior para usá-las, dependendo do uso atual da sua instância de banco de dados. Por exemplo, o Oracle Enterprise Manager Database Control usa cerca de 300 MB de RAM. Se você habilitar essa opção para uma instância de banco de dados pequena, poderá encontrar problemas de performance devido a erros de memória insuficiente.

Opções persistentes e permanentes

Dois tipos de opções, persistentes e permanentes, exigem consideração especial quando você as adiciona a um grupo de opções.

Ações persistentes não podem ser removidas de um grupo de opções enquanto instâncias de banco de dados estão associadas a esse grupo. Um exemplo de uma opção persistente é a opção TDE para o recurso TDE (Transparent Data Encryption) do Microsoft SQL Server. Você deve desassociar todas as instâncias de bancos de dados do grupo de opções para que uma opção persistente possa ser removida do grupo de opções. Em alguns casos, você pode restaurar ou realizar uma restauração pontual a partir de um snapshot de banco de dados. Nesses casos, se o grupo de opções associado a esse snapshot de banco de dados contiver uma opção persistente, você só poderá associar a instância de banco de dados restaurada a esse grupo de opções.

Opções permanentes, como a opção TDE para o TDE do Oracle Advanced Security, nunca podem ser removidas de um grupo de opções. Você pode alterar o grupo de opções de uma instância de banco de dados que esteja usando a opção permanente. No entanto, o grupo de opções associado à instância de banco de dados deve incluir a mesma opção permanente. Em alguns casos, você pode restaurar ou realizar uma restauração pontual a partir de um snapshot de banco de dados. Nesses casos, se o grupo de opções associado a esse snapshot de banco de dados contiver uma opção permanente, você só poderá associar a instância de banco de dados restaurada a um grupo de opções com essa opção permanente.

Para instâncias de banco de dados Oracle, você pode copiar snapshots de banco de dados compartilhados que tenham as opções `Timezone` ou `OLS` (ou ambas). Para fazer isso, especifique um grupo de opções de destino que inclua essas opções ao copiar o snapshot de banco de dados. A opção OLS é permanente e persistente apenas para instâncias de banco de dados Oracle executando o Oracle versão 12.2 ou superior. Para obter mais informações sobre essas opções, consulte Fuso horário da Oracle e Oracle Label Security.

Considerações de VPC

O grupo de opções associado à instância de banco de dados é vinculado à VPC da instância de banco de dados. Isso significa que você não poderá usar o grupo de opções atribuído a uma instância de banco de dados se tentar restaurar essa instância em uma VPC diferente. Se você restaurar uma instância de banco de dados para uma VPC diferente, poderá realizar um dos seguintes procedimentos:

- Atribuir o grupo de opções padrão à instância de banco de dados.
- Atribuir um grupo de opções vinculado a essa VPC.
- Criar um novo grupo de opções e atribui-lo à instância de banco de dados.

Com opções persistentes ou permanentes, como o Oracle TDE, você deve criar um grupo de opções. Esse grupo de opções deve incluir a opção persistente ou permanente ao restaurar uma instância de banco de dados em uma VPC diferente.

Configurações de opções controlam o comportamento de uma opção. Por exemplo, a opção Oracle Advanced Security `NATIVE_NETWORK_ENCRYPTION` tem uma configuração que você pode usar para especificar o algoritmo de criptografia do tráfego de rede para e a partir da instância de banco de dados. Algumas configurações de opções são otimizadas para uso com o Amazon RDS e não podem ser alteradas.

Opções mutuamente exclusivas

Algumas opções são mutuamente exclusivas. Você pode usar uma ou outro, mas não ambas ao mesmo tempo. As seguintes opções são mutuamente exclusivas:

- Oracle Enterprise Manager Database Express e Oracle Management Agent para Enterprise Manager Cloud Control.
- Oracle Native Network Encryption e Oracle Secure Sockets Layer.

Criar um grupo de opções

Você pode criar um grupo de opções que obtenha suas configurações do grupo de opções padrão. Então, você adiciona uma ou mais opções ao novo grupo de opções. Ou, se você já possui um grupo de opções, pode copiar esse grupo com todas as suas opções em um novo grupo de opções. Para ter mais informações, consulte Copiar um grupo de opções.

Depois de criar um novo grupo de opções, ele não possui opções. Para saber como adicionar opções ao grupo de opções, consulte Adicionar uma opção a um grupo de opções. Depois de adicionar as opções desejadas, você poderá então associar o grupo de opções a uma instância de banco de dados. Dessa forma, as opções ficam disponíveis na instância de banco de dados. Para obter informações sobre como associar um grupo de opções a uma instância de banco de dados, consulte a documentação do mecanismo em Trabalhar com grupos de opções.

Console

Uma maneira de criar um grupo de opções é usando o AWS Management Console.

Para criar um novo grupo de opções usando o console

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Option groups (Grupos de opções).
3. Escolha Create group (Criar grupo).
4. Na janela Create option group (Criar grupo de opções), faça o seguinte:
    a. Em Name (Nome), digite um nome para o grupo de opções que seja exclusivo na sua conta da AWS. O nome pode conter somente letras, números e hifens.

b. Em Description (Descrição), digite uma breve descrição do grupo de opções. A descrição é usada para fins de exibição.

c. Em Engine (Mecanismo), escolha o mecanismo de banco de dados desejado.

d. Em Major engine version (Versão principal do mecanismo), escolha a versão principal do mecanismo de banco de dados desejado.

5. Escolha Create (Criar) para continuar. Para cancelar a operação, escolha Cancel (Cancelar).

AWS CLI

Para criar um grupo de opções, use o comando AWS CLI da `create-option-group` com os seguintes parâmetros obrigatórios.

- `--option-group-name`
- `--engine-name`
- `--major-engine-version`
- `--option-group-description`

Example

O exemplo a seguir cria um grupo de opções denominado `testoptiongroup`, que está associado ao mecanismo de banco de dados Oracle Enterprise Edition. A descrição está entre aspas.

Para Linux, macOS ou Unix:

```
aws rds create-option-group \
    --option-group-name testoptiongroup \
    --engine-name oracle-ee \
    --major-engine-version 19 \
    --option-group-description "Test option group for Oracle Database 19c EE"
```

Para Windows:

```
aws rds create-option-group ^
```

```
    --option-group-name testoptiongroup ^
    --engine-name oracle-ee ^-
    --major-engine-version 19 ^
    --option-group-description "Test option group for Oracle Database 19c EE"
```

API do RDS

Para criar um grupo de opções, chame a operação `CreateOptionGroup` da API do Amazon RDS. Inclua os seguintes parâmetros:

- `OptionGroupName`
- `EngineName`
- `MajorEngineVersion`
- `OptionGroupDescription`

Copiar um grupo de opções

Você pode usar a AWS CLI ou a API do Amazon RDS para copiar um grupo de opções. Copiar um grupo de opções pode ser conveniente. Um exemplo é quando há um grupo de opções e você deseja incluir a maioria de seus parâmetros e valores personalizados em um novo grupo de opções. Você também pode fazer uma cópia de um grupo de opções que costuma usar em um ambiente de produção e depois modificar essa cópia para testar outras configurações de opções.

> **Note**
>
> No momento, você não pode copiar um grupo de opções para uma região diferente da AWS.

AWS CLI

Para copiar um grupo de opções, use o comando copy-option-group da AWS CLI. Inclua as seguintes opções necessárias:

- `--source-option-group-identifier`
- `--target-option-group-identifier`
- `--target-option-group-description`

Example

O exemplo a seguir cria um grupo de opções denominado `new-option-group`, que é uma cópia local do grupo de opções `my-option-group`.

Para Linux, macOS ou Unix:

```
aws rds copy-option-group \
    --source-option-group-identifier my-option-group \
    --target-option-group-identifier new-option-group \
    --target-option-group-description "My new option group"
```

Para Windows:

```
aws rds copy-option-group ^
    --source-option-group-identifier my-option-group ^
    --target-option-group-identifier new-option-group ^
    --target-option-group-description "My new option group"
```

API do RDS

Para copiar um grupo de opções, chame a operação CopyOptionGroup da API do Amazon RDS. Inclua os seguintes parâmetros necessários.

- `SourceOptionGroupIdentifier`
- `TargetOptionGroupIdentifier`
- `TargetOptionGroupDescription`

Adicionar uma opção a um grupo de opções

Você pode adicionar uma opção a um grupo de opções existente. Depois de ter adicionado as opções desejadas, você poderá então associar o grupo de opções com uma instância de banco de dados para que as opções se tornem disponíveis na instância de banco de dados. Para obter informações sobre como associar um grupo de opções a uma instância de banco de dados, consulte a documentação do mecanismo de banco de dados específico, listada em Trabalhar com grupos de opções.

As alterações do grupo de opções devem ser aplicadas imediatamente em dois casos:

- Quando você adiciona uma opção que adiciona ou atualiza um valor de porta, como a opção `OEM`.
- Quando você adiciona ou remove um grupo de opções com uma opção que inclui um valor de porta.

Nesses casos, escolha a opção Apply Immediately (Aplicar imediatamente) no console. Se preferir, inclua a opção `--apply-immediately` quando usar a AWS CLI ou definir o parâmetro `ApplyImmediately` como `true` ao usar a API do Amazon RDS. As opções que não incluem valores de porta podem ser aplicadas imediatamente ou podem ser aplicadas durante a próxima janela de manutenção da instância de banco de dados.

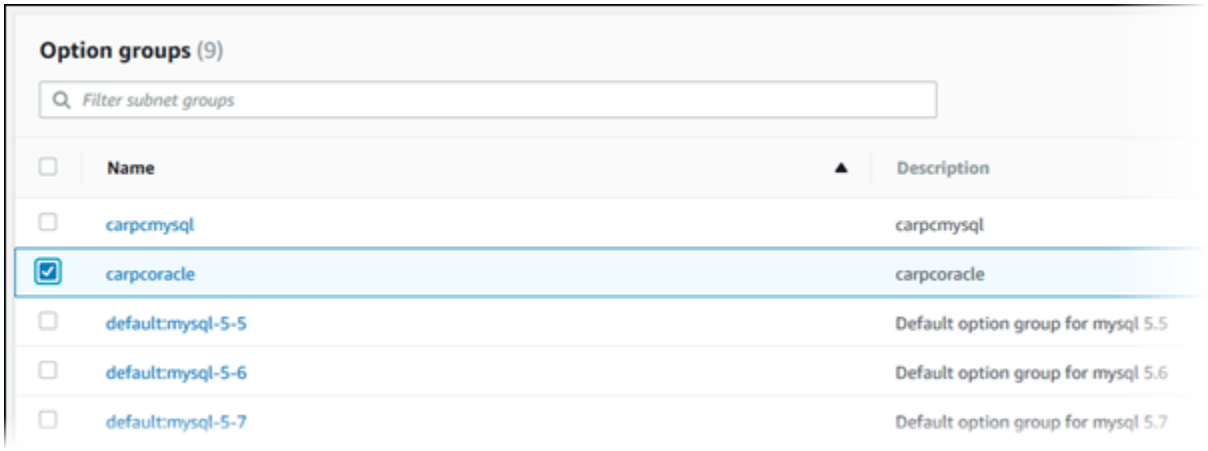
> **Note**
>
> Se você especificar um grupo de segurança como um valor para uma opção em um grupo de opções, gerencie o grupo de segurança modificando o grupo de opções. Não é possível alterar ou remover esse grupo de segurança modificando uma instância de banco de dados. Além disso, o grupo de segurança não aparece nos detalhes da instância de banco de dados no AWS Management Console ou na saída do comando AWS CLI da `describe-db-instances`.

Console

Você pode usar o AWS Management Console para adicionar uma opção a um grupo de opções.

Para adicionar uma opção a um grupo de opções usando o console

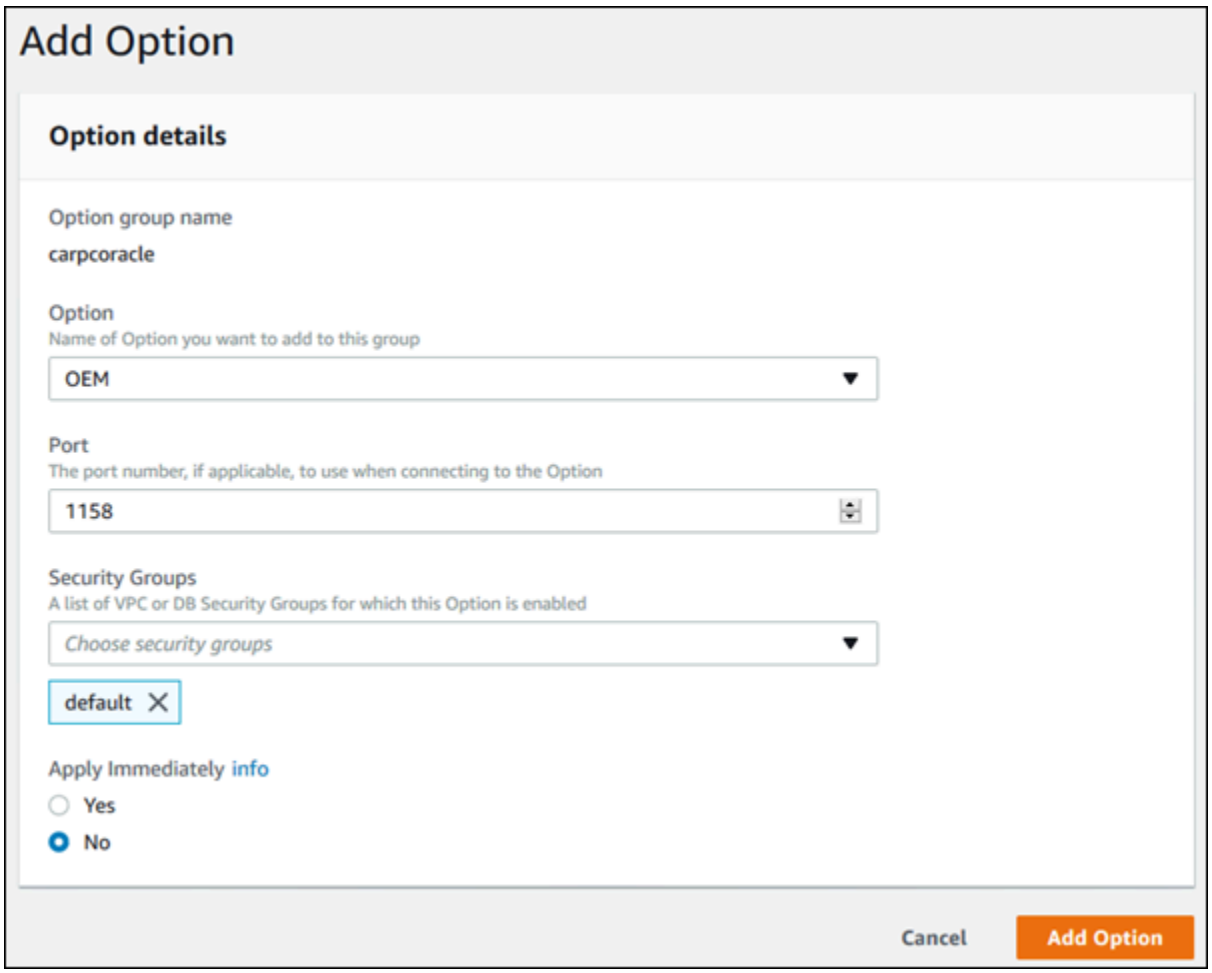
1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Option groups (Grupos de opções).
3. Selecione o grupo de opções que você deseja modificar e escolha Add Option (Adicionar opção).



4. Na janela Add option (Adicionar opção), faça o seguinte:

   a. Escolha a opção que você deseja adicionar. Talvez seja necessário fornecer valores adicionais, dependendo da opção que você selecionar. Por exemplo, ao escolher a opção `OEM`, também é necessário inserir um valor de porta e especificar um grupo de segurança.

   b. Para habilitar a opção em todas as instâncias de banco de dados associadas assim que você a adicionar, em Apply Immediately (Aplicar imediatamente), escolha Yes (Sim). Se você escolher No (Não) (o padrão), a opção será habilitada para cada instância de banco de dados associada durante sua próxima janela de manutenção.



5. Quando estiver satisfeito com as configurações, escolha Add Option (Adicionar opção).

AWS CLI

Para adicionar uma opção a um grupo de opções, execute o comando add-option-to-option-group da AWS CLI com a opção que você deseja adicionar. Para habilitar a nova opção imediatamente em todas as instâncias de bancos de dados associadas, inclua o parâmetro `--apply-immediately`. Por padrão, a opção é habilitada para cada instância de banco de dados associada durante sua próxima janela de manutenção. Inclua o seguinte parâmetro necessário:

- `--option-group-name`

Example

O exemplo a seguir adiciona a opção `Timezone`, com a configuração `America/Los_Angeles`, a um grupo de opções denominado `testoptiongroup` e a habilita imediatamente.

Para Linux, macOS ou Unix:

```
aws rds add-option-to-option-group \
    --option-group-name testoptiongroup \
    --options "OptionName=Timezone,OptionSettings=[{Name=TIME_ZONE,Value=America/
Los_Angeles}]" \
    --apply-immediately
```

Para Windows:

```
aws rds add-option-to-option-group ^
    --option-group-name testoptiongroup ^
    --options "OptionName=Timezone,OptionSettings=[{Name=TIME_ZONE,Value=America/
Los_Angeles}]" ^
    --apply-immediately
```

A saída do comando é semelhante à seguinte:

```
...{
       "OptionName": "Timezone",
       "OptionDescription": "Change time zone",
       "Persistent": true,
       "Permanent": false,
       "OptionSettings": [
           {
               "Name": "TIME_ZONE",
               "Value": "America/Los_Angeles",
               "DefaultValue": "UTC",
               "Description": "Specifies the timezone the user wants to change the
 system time to",
               "ApplyType": "DYNAMIC",
               "DataType": "STRING",
               "AllowedValues": "Africa/Cairo,...",
               "IsModifiable": true,
               "IsCollection": false
           }
       ],
       "DBSecurityGroupMemberships": [],
```

```
        "VpcSecurityGroupMemberships": []
    }...
```

Example

O exemplo a seguir adiciona a opção Oracle OEM a um grupo de opções. Ele também especifica uma porta personalizada e um par de grupos de segurança de VPC do Amazon EC2 a ser usado para essa porta.

Para Linux, macOS ou Unix:

```
aws rds add-option-to-option-group \
    --option-group-name testoptiongroup \
    --options OptionName=OEM,Port=5500,VpcSecurityGroupMemberships="sg-test1,sg-test2"
 \
    --apply-immediately
```

Para Windows:

```
aws rds add-option-to-option-group ^
    --option-group-name testoptiongroup ^
    --options OptionName=OEM,Port=5500,VpcSecurityGroupMemberships="sg-test1,sg-test2"
 ^
    --apply-immediately
```

A saída do comando é semelhante à seguinte:

```
OPTIONGROUP  False  oracle-ee  19 arn:aws:rds:us-east-1:1234567890:og:testoptiongroup
 Test Option Group  testoptiongroup vpc-test
OPTIONS Oracle 12c EM Express   OEM     False   False   5500
VPCSECURITYGROUPMEMBERSHIPS     active  sg-test1
VPCSECURITYGROUPMEMBERSHIPS     active  sg-test2
```

Example

O exemplo a seguir adiciona a opção Oracle `NATIVE_NETWORK_ENCRYPTION` a um grupo de opções e especifica as configurações de opções. Se nenhuma configuração de opção for especificada, os valores padrão serão usados.

Para Linux, macOS ou Unix:

```
aws rds add-option-to-option-group \
```

```
    --option-group-name testoptiongroup \
    --options '[{"OptionSettings":
[{"Name":"SQLNET.ENCRYPTION_SERVER","Value":"REQUIRED"},
{"Name":"SQLNET.ENCRYPTION_TYPES_SERVER","Value":"AES256,AES192,DES"}],"OptionName":"NATIVE_NET
 \
    --apply-immediately
```

Para Windows:

```
aws rds add-option-to-option-group ^
    --option-group-name testoptiongroup ^
    --options "OptionSettings"=[{"Name"="SQLNET.ENCRYPTION_SERVER","Value"="REQUIRED"},
{"Name"="SQLNET.ENCRYPTION_TYPES_SERVER","Value"="AES256\,AES192\,DES"}],"OptionName"="NATIVE_N
 ^
    --apply-immediately
```

A saída do comando é semelhante à seguinte:

```
...{
    "OptionName": "NATIVE_NETWORK_ENCRYPTION",
    "OptionDescription": "Native Network Encryption",
    "Persistent": false,
    "Permanent": false,
    "OptionSettings": [
        {
            "Name": "SQLNET.ENCRYPTION_TYPES_SERVER",
             "Value": "AES256,AES192,DES",
            "DefaultValue":
 "RC4_256,AES256,AES192,3DES168,RC4_128,AES128,3DES112,RC4_56,DES,RC4_40,DES40",
            "Description": "Specifies list of encryption algorithms in order of
 intended use",
            "ApplyType": "STATIC",
            "DataType": "STRING",
            "AllowedValues":
 "RC4_256,AES256,AES192,3DES168,RC4_128,AES128,3DES112,RC4_56,DES,RC4_40,DES40",
            "IsModifiable": true,
            "IsCollection": true
        },
        {
            "Name": "SQLNET.ENCRYPTION_SERVER",
            "Value": "REQUIRED",
            "DefaultValue": "REQUESTED",
            "Description": "Specifies the desired encryption behavior",
```

```
            "ApplyType": "STATIC",
            "DataType": "STRING",
            "AllowedValues": "ACCEPTED,REJECTED,REQUESTED,REQUIRED",
            "IsModifiable": true,
            "IsCollection": false
        },...
```

API do RDS

Para adicionar uma opção a um grupo de opções usando a API do Amazon RDS, chame a operação ModifyOptionGroup com a opção que você deseja adicionar. Para habilitar a nova opção imediatamente em todas as instâncias de bancos de dados associadas, inclua o parâmetro `ApplyImmediately` e defina-o como `true`. Por padrão, a opção é habilitada para cada instância de banco de dados associada durante sua próxima janela de manutenção. Inclua o seguinte parâmetro necessário:

- `OptionGroupName`

Listar as opções e as configurações de opções para um grupo de opções

Você pode listar todas as opções e configurações de opções para um grupo de opções.

Console

Você pode usar o AWS Management Console para listar todas as opções e configurações de opções para um grupo de opções.

Para listar as opções e as configurações de opções para um grupo de opções

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Option groups (Grupos de opções).
3. Escolha o nome do grupo de opções para exibir seus detalhes. As opções e as configurações de opções no grupo de opções estão listadas.

AWS CLI

Para listar as opções e as configurações de opções para um grupo de opções, use o comando AWS CLI da `describe-option-groups`. Especifique o nome do grupo de opções cujas opções e

configurações você deseja visualizar. Se você não especificar um nome de grupo de opções, todos os grupos de opções serão descritos.

Example

O exemplo a seguir lista as opções e as configurações de opções para todos os grupos de opções.

```
aws rds describe-option-groups
```

Example

O exemplo a seguir lista as opções e as configurações de opções para um grupo de opções denominado `testoptiongroup`.

```
aws rds describe-option-groups --option-group-name testoptiongroup
```

API do RDS

Para listar as opções e as configurações de opções para um grupo de opções, use a operação `DescribeOptionGroups` da API do Amazon RDS. Especifique o nome do grupo de opções cujas opções e configurações você deseja visualizar. Se você não especificar um nome de grupo de opções, todos os grupos de opções serão descritos.

Modificar uma configuração de opção

Depois de adicionar uma opção que tenha configurações de opções modificáveis, você pode modificar as configurações a qualquer momento. Se você alterar opções ou configurações de opções em um grupo de opções, essas alterações serão aplicadas a todas as instâncias de bancos de dados associadas a esse grupo de opções. Para obter mais informações sobre quais configurações estão disponíveis para as várias opções, consulte a documentação do mecanismo específico em Trabalhar com grupos de opções.

As alterações do grupo de opções devem ser aplicadas imediatamente em dois casos:

- Quando você adiciona uma opção que adiciona ou atualiza um valor de porta, como a opção `OEM`.
- Quando você adiciona ou remove um grupo de opções com uma opção que inclui um valor de porta.

Nesses casos, escolha a opção Apply Immediately (Aplicar imediatamente) no console. Outra alternativa é incluir a opção `--apply-immediately` quando usar a AWS CLI ou definir o parâmetro

`ApplyImmediately` como `true` ao usar a API do RDS. As opções que não incluem valores de porta podem ser aplicadas imediatamente ou podem ser aplicadas durante a próxima janela de manutenção da instância de banco de dados.

> **Note**
>
> Se você especificar um grupo de segurança como um valor para uma opção em um grupo de opções, gerencie o grupo de segurança modificando o grupo de opções. Não é possível alterar ou remover esse grupo de segurança modificando uma instância de banco de dados. Além disso, o grupo de segurança não aparece nos detalhes da instância de banco de dados no AWS Management Console ou na saída do comando AWS CLI da `describe-db-instances`.

Console

Você pode usar o AWS Management Console para modificar uma configuração de opção.

Para modificar uma configuração de opção usando o console

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Option groups (Grupos de opções).
3. Selecione o grupo de opções cuja opção você deseja modificar e escolha Modify option (Modificar opção).
4. Na janela Modify option (Modificar opção), em Installed Options (Opções instaladas), escolha a opção cuja configuração você deseja modificar. Faça as alterações desejadas.
5. Para habilitar a opção assim que você a adicionar, em Apply Immediately (Aplicar imediatamente), escolha Yes (Sim). Se você escolher No (Não) (o padrão), a opção será habilitada para cada instância de banco de dados associada durante sua próxima janela de manutenção.
6. Quando estiver satisfeito com as configurações, escolha Modify Option (Modificar opção).

AWS CLI

Para modificar uma configuração de opção, use o comando AWS CLI da `add-option-to-option-group` com o grupo de opções e a opção que você deseja modificar. Por padrão, a opção

é habilitada para cada instância de banco de dados associada durante sua próxima janela de manutenção. Para aplicar a alteração imediatamente a todas as instâncias de bancos de dados associadas, inclua o parâmetro `--apply-immediately`. Para modificar uma configuração de opção, use o argumento `--settings`.

Example

O exemplo a seguir modifica a porta usada pelo Oracle Enterprise Manager Database Control (OEM) em um grupo de opções denominado `testoptiongroup` e aplica imediatamente a alteração.

Para Linux, macOS ou Unix:

```
aws rds add-option-to-option-group \
    --option-group-name testoptiongroup \
    --options OptionName=OEM,Port=5432,DBSecurityGroupMemberships=default \
    --apply-immediately
```

Para Windows:

```
aws rds add-option-to-option-group ^
    --option-group-name testoptiongroup ^
    --options OptionName=OEM,Port=5432,DBSecurityGroupMemberships=default ^
    --apply-immediately
```

A saída do comando é semelhante à seguinte:

```
OPTIONGROUP   False  oracle-ee  19  arn:aws:rds:us-east-1:1234567890:og:testoptiongroup
   Test Option Group    testoptiongroup
OPTIONS Oracle 12c EM Express   OEM     False   False   5432
DBSECURITYGROUPMEMBERSHIPS   default  authorized
```

Example

O seguinte exemplo modifica a opção Oracle `NATIVE_NETWORK_ENCRYPTION` e altera as configurações de opções.

Para Linux, macOS ou Unix:

```
aws rds add-option-to-option-group \
```

```
    --option-group-name testoptiongroup \
    --options '[{"OptionSettings":
[{"Name":"SQLNET.ENCRYPTION_SERVER","Value":"REQUIRED"},
{"Name":"SQLNET.ENCRYPTION_TYPES_SERVER","Value":"AES256,AES192,DES,RC4_256"}],"OptionName":"NA
 \
    --apply-immediately
```

Para Windows:

```
aws rds add-option-to-option-group ^
    --option-group-name testoptiongroup ^
    --options "OptionSettings"=[{"Name"="SQLNET.ENCRYPTION_SERVER","Value"="REQUIRED"},
{"Name"="SQLNET.ENCRYPTION_TYPES_SERVER","Value"="AES256\,AES192\,DES
\,RC4_256"}],"OptionName"="NATIVE_NETWORK_ENCRYPTION" ^
    --apply-immediately
```

A saída do comando é semelhante à seguinte:

```
OPTIONGROUP   False  oracle-ee  19  arn:aws:rds:us-east-1:1234567890:og:testoptiongroup
   Test Option Group    testoptiongroup
OPTIONS Oracle Advanced Security - Native Network Encryption
 NATIVE_NETWORK_ENCRYPTION       False   False
OPTIONSETTINGS
 RC4_256,AES256,AES192,3DES168,RC4_128,AES128,3DES112,RC4_56,DES,RC4_40,DES40 STATIC
 STRING
   RC4_256,AES256,AES192,3DES168,RC4_128,AES128,3DES112,RC4_56,DES,RC4_40,DES40
 Specifies list of encryption algorithms in order of intended use
   True     True    SQLNET.ENCRYPTION_TYPES_SERVER  AES256,AES192,DES,RC4_256
OPTIONSETTINGS  ACCEPTED,REJECTED,REQUESTED,REQUIRED    STATIC  STRING  REQUESTED
 Specifies the desired encryption behavior  False   True  SQLNET.ENCRYPTION_SERVER
 REQUIRED
OPTIONSETTINGS  SHA1,MD5   STATIC  STRING  SHA1,MD5    Specifies list of
 checksumming algorithms in order of intended use      True    True
 SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER  SHA1,MD5
OPTIONSETTINGS  ACCEPTED,REJECTED,REQUESTED,REQUIRED  STATIC  STRING
 REQUESTED     Specifies the desired data integrity behavior   False   True
 SQLNET.CRYPTO_CHECKSUM_SERVER  REQUESTED
```

API do RDS

Para modificar uma configuração de opção, use o comando `ModifyOptionGroup` da API do Amazon RDS com o grupo de opções e a opção que você deseja modificar. Por padrão, a opção

é habilitada para cada instância de banco de dados associada durante sua próxima janela de manutenção. Para aplicar a alteração imediatamente a todas as instâncias de bancos de dados associadas, inclua o parâmetro `ApplyImmediately` e defina-o como `true`.

Remover uma opção de um grupo de opções

Algumas opções podem ser removidas de um grupo de opções e outras não. Uma opção persistente não pode ser removida de um grupo de opções até que todas as instâncias de bancos de dados associadas a esse grupo de opções sejam desassociadas. Uma opção permanente nunca pode ser removida de um grupo de opções. Para obter mais informações sobre quais opções podem ser removidas, consulte a documentação do seu mecanismo específico, listada em Trabalhar com grupos de opções.

Se você remover todas as opções de um grupo de opções, o Amazon RDS não excluirá o grupo de opções. As instâncias de banco de dados associadas ao grupo de opções vazio continuam associadas a ele. Essas instâncias simplesmente não terão opções ativas. Alternativamente, para remover todas as opções de uma instância de banco de dados, você pode associar a instância de banco de dados ao grupo de opções padrão (vazio).

Console

Você pode usar o AWS Management Console para remover uma opção de um grupo de opções.

Para remover uma opção de um grupo de opções usando o console

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Option groups (Grupos de opções).
3. Selecione o grupo de opções cuja opção você deseja remover e escolha Delete option (Excluir opção).
4. Na janela Delete option (Excluir opção), faça o seguinte:
   - Marque a caixa de seleção para a opção que você deseja excluir.
   - Para que a exclusão entre em vigor assim que você a fizer, em Apply immediately (Aplicar imediatamente), escolha Yes (Sim). Se você escolher No (Não) (o padrão), a opção será excluída de cada instância de banco de dados associada durante a próxima janela de manutenção.



5. Quando estiver satisfeito com as configurações, escolha Yes, Delete (Sim, excluir).

AWS CLI

Para remover uma opção de um grupo de opções, use o comando da AWS CLI `remove-option-from-option-group` com a opção que você deseja excluir. Por padrão, a opção será removida de cada instância de banco de dados associada durante sua próxima janela de manutenção. Para aplicar a alteração imediatamente, inclua o parâmetro `--apply-immediately`.

Example

O exemplo a seguir remove a opção Oracle Enterprise Manager Database Control (OEM) de um grupo de opções chamado `testoptiongroup` e aplica imediatamente a alteração.

Para Linux, macOS ou Unix:

```
aws rds remove-option-from-option-group \
    --option-group-name testoptiongroup \
    --options OEM \
    --apply-immediately
```

Para Windows:

```
aws rds remove-option-from-option-group ^
    --option-group-name testoptiongroup ^
    --options OEM ^
    --apply-immediately
```

A saída do comando é semelhante à seguinte:

```
OPTIONGROUP    testoptiongroup oracle-ee   19    Test option group
```

API do RDS

Para remover uma opção de um grupo de opções, use a ação `ModifyOptionGroup` da API do Amazon RDS. Por padrão, a opção será removida de cada instância de banco de dados associada durante sua próxima janela de manutenção. Para aplicar a alteração imediatamente, inclua o parâmetro `ApplyImmediately` e configure-o como `true`.

Inclua os seguintes parâmetros:

- `OptionGroupName`
- `OptionsToRemove.OptionName`

Excluir um grupo de opções

Só será possível excluir um grupo de opções se ele atender aos seguintes critérios:

- Ele não está associado a nenhum recurso do Amazon RDS. Um grupo de opções pode ser associado a uma instância de banco de dados, a um snapshot de banco de dados manual ou a um snapshot de banco de dados automatizado.
- Ele não é um grupo de opções padrão.

Para identificar os grupos de opções usados pelas instâncias de banco de dados e snapshots de banco de dados, é possível usar os seguintes comandos da CLI:

```
aws rds describe-db-instances \
    --query 'DBInstances[*].
[DBInstanceIdentifier,OptionGroupMemberships[].OptionGroupName]'
```

```
aws rds describe-db-snapshots | jq -r '.DBSnapshots[] | "\(.DBInstanceIdentifier),
\(.OptionGroupName)"' | sort | uniq
```

Se você tentar excluir um grupo de opções associado a um recurso do RDS, será retornado um erro semelhante ao seguinte.

```
An error occurred (InvalidOptionGroupStateFault) when calling the DeleteOptionGroup
 operation: The option group 'optionGroupName' cannot be deleted because it is in use.
```

Para encontrar os recursos do Amazon RDS associados a um grupo de opções

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Option groups (Grupos de opções).
3. Escolha o nome do grupo de opções para mostrar seus detalhes.
4. Consulte a seção Instâncias e snapshots associados para conhecer os recursos do Amazon RDS associados.

Se uma instância de banco de dados estiver associada ao grupo de opções, modifique-a para usar um grupo de opções diferente. Para ter mais informações, consulte Modificar uma instância de banco de dados do Amazon RDS.

Se um snapshot de banco de dados manual estiver associado ao grupo de opções, modifique esse snapshot para usar um grupo de opções diferente. Faça isso usando o comando `modify-db-snapshot` da AWS CLI.

> **Note**
>
> Não é possível modificar o grupo de opções de um snapshot de banco de dados automatizado.

Console

Uma maneira de excluir um grupo de opções é usando o AWS Management Console.

Para excluir um grupo de opções usando o console

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Option groups (Grupos de opções).
3. Escolha o grupo de opções.
4. Selecione Delete group (Excluir grupo).
5. Na página de confirmação, escolha Delete (Excluir) para concluir a exclusão do grupo de opções ou escolha Cancel (Cancelar) para cancelar a exclusão.

AWS CLI

Para excluir um grupo de opções, use o comando da AWS CLI `delete-option-group` com o seguinte parâmetro necessário.

- `--option-group-name`

Example

O exemplo a seguir exclui um grupo de opções denominado `testoptiongroup`.

Para Linux, macOS ou Unix:

```
aws rds delete-option-group \
    --option-group-name testoptiongroup
```

Para Windows:

```
aws rds delete-option-group ^
    --option-group-name testoptiongroup
```

API do RDS

Para excluir um grupo de opções, chame a operação `DeleteOptionGroup` da API do Amazon RDS. Inclua o seguinte parâmetro:

- `OptionGroupName`

Grupos de parâmetros para Amazon RDS

Database parameters (Parâmetros do banco de dados) especifica como o banco de dados é configurado. Por exemplo, os parâmetros do banco de dados podem especificar a quantidade de recursos, como memória, a serem alocados para um banco de dados.

Você gerencia a configuração do banco de dados associando suas instâncias e seus clusters de banco de dados multi-AZ com grupos de parâmetros. O Amazon RDS define grupos de parâmetros com configurações padrão. Você também pode definir seus próprios grupos de parâmetros com configurações personalizadas.

> **Note**
>
> Alguns mecanismos de banco de dados oferecem recursos adicionais que você pode adicionar ao seu banco de dados como opções em um grupo de opções. Para obter informações sobre grupos de opções, consulte Trabalhar com grupos de opções.

Tópicos

- Visão geral dos grupos de parâmetros
- Grupos de parâmetros de banco de dados para instâncias de banco de dados do Amazon RDS
- Trabalhar com grupos de parâmetros de clusters de banco de dados multi-AZ
- Comparação de grupos de parâmetros de banco de dados
- Especificação de parâmetros de banco de dados

Visão geral dos grupos de parâmetros

Um grupo de parâmetros de banco de dados atua como um contêiner para valores de configuração de mecanismo que são aplicados a uma ou mais instâncias de bancos de dados.

Grupos de parâmetros de cluster de banco de dados se aplicam somente a clusters de banco de dados multi-AZ. As configurações do grupo de parâmetros de um cluster de banco de dados multi-AZ se aplicam a todas as instâncias de banco de dados contidas no cluster. O grupo de parâmetros de banco de dados padrão do mecanismo e da versão do mecanismo de banco de dados é usado para cada instância de banco de dados contida no cluster de banco de dados.

Tópicos

- Grupos de parâmetros padrão e personalizados
- Parâmetros estáticos e dinâmicos de instância de banco de dados
- Parâmetros estáticos e dinâmicos de cluster de banco de dados
- Parâmetros de conjunto de caracteres
- Parâmetros e valores de parâmetros compatíveis

Grupos de parâmetros padrão e personalizados

Se você criar uma instância de banco de dados sem especificar um grupo de parâmetros de banco de dados, essa instância usará o grupo de parâmetros de banco de dados padrão. Da mesma forma, se você criar um cluster de banco de dados multi-AZ sem especificar um grupo de parâmetros de cluster de banco de dados, esse cluster usará um grupo de parâmetros de cluster de banco de dados padrão. Cada grupo de parâmetros de banco de dados padrão contém padrões de mecanismo de banco de dados e do sistema Amazon RDS com base no mecanismo, na classe de computação e no armazenamento alocado da instância.

Não é possível modificar as configurações de parâmetros de um grupo de parâmetros padrão. Em vez disso, você pode fazer o seguinte:

1. Crie um novo grupo de parâmetros.
2. Altere as configurações dos parâmetros desejados. Nem todos os parâmetros de mecanismo de banco de dados em um grupo de parâmetros podem ser modificados.
3. Modifique a instância de banco de dados ou o cluster de banco de dados para associar o novo grupo de parâmetros.

   Ao associar um novo grupo de parâmetros de banco de dados a uma instância de banco de dados, a associação acontece imediatamente. Para ter mais informações sobre como modificar uma instância de banco de dados, consulte Modificar uma instância de banco de dados do Amazon RDS. Para obter informações sobre como modificar clusters de banco de dados multi-AZ, consulte Modificar um cluster de banco de dados multi-AZ para o Amazon RDS.

   > **Note**
   >
   > Se você modificou sua instância de banco de dados para usar um grupo de parâmetros personalizado, e iniciar a instância de banco de dados, o RDS reinicializará automaticamente a instância de banco de dados como parte do processo de inicialização.

> Para instâncias multi-AZ do RDS para SQL Server com a opção “Sempre ativa” ou “Espelhamento” habilitada, um failover é esperado quando a instância é reiniciada após o processo de inicialização.

O RDS aplica os parâmetros estáticos e dinâmicos modificados a um grupo de parâmetros recém-associado somente depois que a instância de banco de dados é reinicializada. No entanto, se você modificar parâmetros dinâmicos no grupo de parâmetros de banco de dados depois de associá-lo à instância de banco de dados, essas alterações serão aplicadas imediatamente sem uma reinicialização. Para obter mais informações sobre como alterar o grupo de parâmetros de banco de dados, consulte Modificar uma instância de banco de dados do Amazon RDS.

Se você atualizar parâmetros dentro de um grupo de parâmetros de banco de dados, as alterações se aplicarão a todas as instâncias de banco de dados associadas a esse grupo de parâmetros. Da mesma forma, se você atualizar os parâmetros dentro de um grupo de parâmetros de cluster de banco de dados multi-AZ, as alterações se aplicarão a todos os clusters do Aurora que estiverem associados a esse grupo.

Se você não quiser criar um grupo de parâmetros do zero, poderá copiar um grupo de parâmetros existente com o comando copy-db-parameter-group ou o comando copy-db-cluster-parameter-group da AWS CLI. Você pode perceber que copiar um grupo de parâmetros é útil em alguns casos. Por exemplo, talvez você queira incluir a maioria dos valores e parâmetros personalizados de um grupo de parâmetros de banco de dados existente em um novo grupo de parâmetros de banco de dados.

Parâmetros estáticos e dinâmicos de instância de banco de dados

Os parâmetros de instância de banco de dados são estáticos ou dinâmicos. As diferenças são as seguintes:

- Quando você altera um parâmetro estático e salva o grupo de parâmetros de banco de dados, a alteração entra em vigor depois que você reinicializa as instâncias de banco de dados manualmente. Para parâmetros estáticos, o console sempre usa `pending-reboot` para o `ApplyMethod`.
- Quando você altera um parâmetro dinâmico, por padrão, a alteração de parâmetro entra em vigor imediatamente, sem a necessidade de reinicialização. Quando você usa o AWS Management Console para alterar os valores dos parâmetros da instância de banco de dados, ele sempre usa `immediate` para `ApplyMethod` para parâmetros dinâmicos. Para adiar a alteração do parâmetro até a finalização da reinicialização de uma instância de banco de dados associada, use a AWS

CLI ou a API do RDS. Defina o `ApplyMethod` como `pending-reboot` para a alteração do parâmetro.

> **Note**
>
> O uso do `pending-reboot` com parâmetros dinâmicos no AWS CLI ou no API do RDS em instâncias de banco de dados do RDS for SQL Server gera um erro. Use `apply-immediately` no RDS for SQL Server.

Para obter mais informações sobre como usar a AWS CLI para alterar um valor de parâmetro, consulte modify-db-parameter-group. Para obter mais informações sobre como usar a API do RDS para alterar um valor de parâmetro, consulte ModifyDBParameterGroup.

Se uma instância de banco de dados não estiver usando as últimas alterações de seu grupo de parâmetros de banco de dados associado, o console mostrará o status pending-reboot para o grupo de parâmetros de banco de dados. Esse status não ocasionará uma reinicialização automática durante a próxima janela de manutenção. Para aplicar as alterações de parâmetro mais recentes a essa instância de banco de dados, reinicialize-a manualmente.

Parâmetros estáticos e dinâmicos de cluster de banco de dados

Os parâmetros de cluster de banco de dados são estáticos ou dinâmicos. As diferenças são as seguintes:

- Quando você altera um parâmetro estático e salva o grupo de parâmetros do cluster de banco de dados, a alteração do parâmetro entra em vigor depois que você reinicializa manualmente os clusters de banco de dados associados. Para parâmetros estáticos, o console sempre usa `pending-reboot` para o `ApplyMethod`.
- Quando você altera um parâmetro dinâmico, por padrão, a alteração de parâmetro entra em vigor imediatamente, sem a necessidade de reinicialização. Quando você usa o AWS Management Console para alterar os valores dos parâmetros do cluster de banco de dados, ele sempre usa `immediate` para `ApplyMethod` para parâmetros dinâmicos. Para adiar a alteração do parâmetro até a reinicialização de um cluster de banco de dados, use a AWS CLI ou a API do RDS. Defina o `ApplyMethod` como `pending-reboot` para a alteração do parâmetro.

Para obter mais informações sobre como usar a AWS CLI para alterar um valor de parâmetro, consulte modify-db-cluster-parameter-group. Para obter mais informações sobre como usar a API do RDS para alterar um valor de parâmetro, consulte ModifyDBClusterParameterGroup.

Parâmetros de conjunto de caracteres

Antes de criar uma instância de banco de dados ou um cluster de banco de dados multi-AZ, defina todos os parâmetros relacionados ao conjunto de caracteres ou ao agrupamento do seu banco de dados no grupo de parâmetros. Também faça isso antes de criar um banco de dados nele. Dessa forma, você garante que o banco de dados padrão e os novos bancos de dados usem o conjunto de caracteres e os valores de agrupamento que você especificar. Se você alterar o conjunto de caracteres ou os parâmetros de agrupamento, as alterações de parâmetros não serão aplicadas aos bancos de dados existentes.

Em mecanismos de bancos de dados, você pode alterar o conjunto de caracteres ou os valores de agrupamento para um banco de dados existente usando o comando `ALTER DATABASE`. Por exemplo:

```
ALTER DATABASE database_name CHARACTER SET character_set_name COLLATE collation;
```

Para obter mais informações sobre como alterar o conjunto de caracteres ou valores de agrupamento de um banco de dados, consulte a documentação referente ao seu mecanismo de banco de dados.

Parâmetros e valores de parâmetros compatíveis

Para determinar os parâmetros compatíveis com seu mecanismo de banco de dados, visualize os parâmetros no grupo de parâmetros de banco de dados e no grupo de parâmetros do cluster de banco de dados utilizados pela instância ou pelo cluster de banco de dados. Para obter mais informações, consulte Visualizar valores de parâmetros para um grupo de parâmetros de banco de dados no Amazon RDS e Visualizar valores de parâmetros de um grupo de parâmetros do cluster de banco de dados.

Em muitos casos, é possível especificar valores de parâmetros inteiros e boolianos usando expressões, fórmulas e funções. As funções podem incluir uma expressão matemática de log. No entanto, nem todos os parâmetros são compatíveis com expressões, fórmulas e funções para valores de parâmetros. Para obter mais informações, consulte Especificação de parâmetros de banco de dados.

Definir incorretamente os parâmetros em um grupo de parâmetros pode causar efeitos adversos não intencionais, inclusive diminuição da performance e instabilidade no sistema. Sempre tenha cuidado ao modificar parâmetros de bancos de dados e faça backup dos dados antes de modificar um grupo de parâmetros. Faça testes com alterações de configuração de grupos de parâmetros em uma instância ou um cluster de banco de dados de teste antes de aplicar essas alterações de grupos de parâmetros a uma instância ou um cluster de banco de dados de produção.

Grupos de parâmetros de banco de dados para instâncias de banco de dados do Amazon RDS

As instâncias de banco de dados usam grupos de parâmetros de banco de dados. As seções a seguir descrevem a configuração e o gerenciamento de grupos de parâmetros de instância de banco de dados.

Tópicos

- Criar um grupo de parâmetros de banco de dados no Amazon RDS
- Associar um grupo de parâmetros de banco de dados a uma instância de banco de dados no Amazon RDS
- Modificar parâmetros em um grupo de parâmetros de banco de dados no Amazon RDS
- Redefinir parâmetros em um grupo de parâmetros de banco de dados como seus valores padrão no Amazon RDS
- Copiar um grupo de parâmetros de banco de dados no Amazon RDS
- Listar grupos de parâmetros de banco de dados no Amazon RDS
- Visualizar valores de parâmetros para um grupo de parâmetros de banco de dados no Amazon RDS
- Excluir um grupo de parâmetros de banco de dados no Amazon RDS

Criar um grupo de parâmetros de banco de dados no Amazon RDS

Você pode criar um novo grupo de parâmetros de banco de dados usando o AWS Management Console, a AWS CLI ou a API do RDS.

As seguintes limitações se aplicam ao nome do grupo de parâmetros de banco de dados:

- O nome deve ter de 1 a 255 letras, números ou hifens.

Os nomes de grupos de parâmetros padrão podem incluir um ponto, como `default.mysql8.0`. No entanto, nomes de grupos de parâmetros personalizados não podem incluir um ponto.

- O primeiro caractere deve ser uma letra.
- O nome não pode terminar com hífen nem conter dois hífens consecutivos.

Console

Para criar um grupo de parâmetros de banco de dados

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Parameter groups (Grupos de parâmetros).
3. Escolha Create parameter group (Criar grupo de parâmetros).
4. Em Nome do grupo de parâmetros, digite o nome do novo grupo de parâmetros de banco de dados.
5. Em Descrição, digite uma descrição para o novo grupo de parâmetros de banco de dados.
6. Em Tipo de mecanismo, escolha o mecanismo de banco de dados.
7. Em Família de grupos de parâmetros, selecione uma família de grupos de parâmetros de banco de dados.
8. Em Tipo, se aplicável, selecione Grupo de parâmetros de banco de dados.
9. Escolha Criar.

AWS CLI

Para criar um grupo de parâmetros de banco de dados, use o comando AWS CLI `create-db-parameter-group`da . O exemplo a seguir cria um grupo de parâmetros de banco de dados chamado mydbparametergroup para o MySQL versão 8.0 com a descrição "My new parameter group (Meu novo grupo de parâmetros)."

Inclua os seguintes parâmetros necessários:

- `--db-parameter-group-name`
- `--db-parameter-group-family`
- `--description`

Para listar todas as famílias de grupos de parâmetros disponíveis, use o comando a seguir:

```
aws rds describe-db-engine-versions --query "DBEngineVersions[].DBParameterGroupFamily"
```

> **Note**
>
> A saída contém duplicatas.

Example

Para Linux, macOS ou Unix:

```
aws rds create-db-parameter-group \
    --db-parameter-group-name mydbparametergroup \
    --db-parameter-group-family MySQL8.0 \
    --description "My new parameter group"
```

Para Windows:

```
aws rds create-db-parameter-group ^
    --db-parameter-group-name mydbparametergroup ^
    --db-parameter-group-family MySQL8.0 ^
    --description "My new parameter group"
```

O comando gerará uma saída semelhante à seguinte:

```
DBPARAMETERGROUP  mydbparametergroup  mysql8.0  My new parameter group
```

API do RDS

Para criar um grupo de parâmetros de banco de dados, use a operação da API `CreateDBParameterGroup` do RDS.

Inclua os seguintes parâmetros necessários:

- `DBParameterGroupName`
- `DBParameterGroupFamily`

- `Description`

Associar um grupo de parâmetros de banco de dados a uma instância de banco de dados no Amazon RDS

Você pode criar seus próprios grupos de parâmetros de banco de dados com configurações personalizadas. Você pode associar um grupo de parâmetros de banco de dados a uma instância de banco de dados usando a AWS Management Console, a AWS CLI ou a API do RDS. Você pode fazer isso ao criar ou modificar uma instância de banco de dados.

Para obter mais informações sobre como criar um grupo de parâmetros de banco de dados, consulte Criar um grupo de parâmetros de banco de dados no Amazon RDS. Para obter informações sobre como criar uma instância de banco de dados, consulte Criar uma instância de banco de dados do Amazon RDS. Para ter mais informações sobre como modificar uma instância de banco de dados , consulte Modificar uma instância de banco de dados do Amazon RDS.

> **Note**
>
> Ao associar um novo grupo de parâmetros de banco de dados a uma instância de banco de dados, os parâmetros estáticos e dinâmicos modificados serão aplicados somente após a reinicialização da instância de banco de dados. No entanto, se você modificar parâmetros dinâmicos no grupo de parâmetros de banco de dados depois de associá-lo à instância de banco de dados, essas alterações serão aplicadas imediatamente sem uma reinicialização.

Console

Para associar um grupo de parâmetros de banco de dados a uma instância de banco de dados

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Databases (Bancos de dados) e a instância de banco de dados que você deseja modificar.
3. Selecione Modify. A página Modify DB instance (Modificar instância de banco de dados) será exibida.
4. Altere a configuração do grupo de parâmetros de banco de dados.
5. Escolha Continue (Continuar) e verifique o resumo de modificações.

6. (Opcional) Escolha Apply immediately (Aplicar imediatamente) para aplicar as alterações imediatamente. Escolher essa opção pode causar uma interrupção em alguns casos. Para obter mais informações, consulte Usar a configuração de programação de modificações.
7. Na página de confirmação, revise suas alterações. Se estiverem corretas, escolha Modify DB Instance (Modificar instância de banco de dados) para salvar suas alterações.

   Ou escolha Back (Voltar) para editar as alterações ou Cancel (Cancelar) para cancelar as alterações.

AWS CLI

Para associar um grupo de parâmetros de banco de dados a uma instância de banco de dados, use o comando da AWS CLI `modify-db-instance` com as seguintes opções:

- `--db-instance-identifier`
- `--db-parameter-group-name`

O exemplo a seguir associa o grupo de parâmetros de banco de dados `mydbpg` à instância de banco de dados `database-1`. As alterações são aplicadas imediatamente usando `--apply-immediately`. Use `--no-apply-immediately` para aplicar alterações durante a próxima janela de manutenção. Para obter mais informações, consulte Usar a configuração de programação de modificações.

Example

Para Linux, macOS ou Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier database-1 \
    --db-parameter-group-name mydbpg \
    --apply-immediately
```

Para Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier database-1 ^
    --db-parameter-group-name mydbpg ^
    --apply-immediately
```

API do RDS

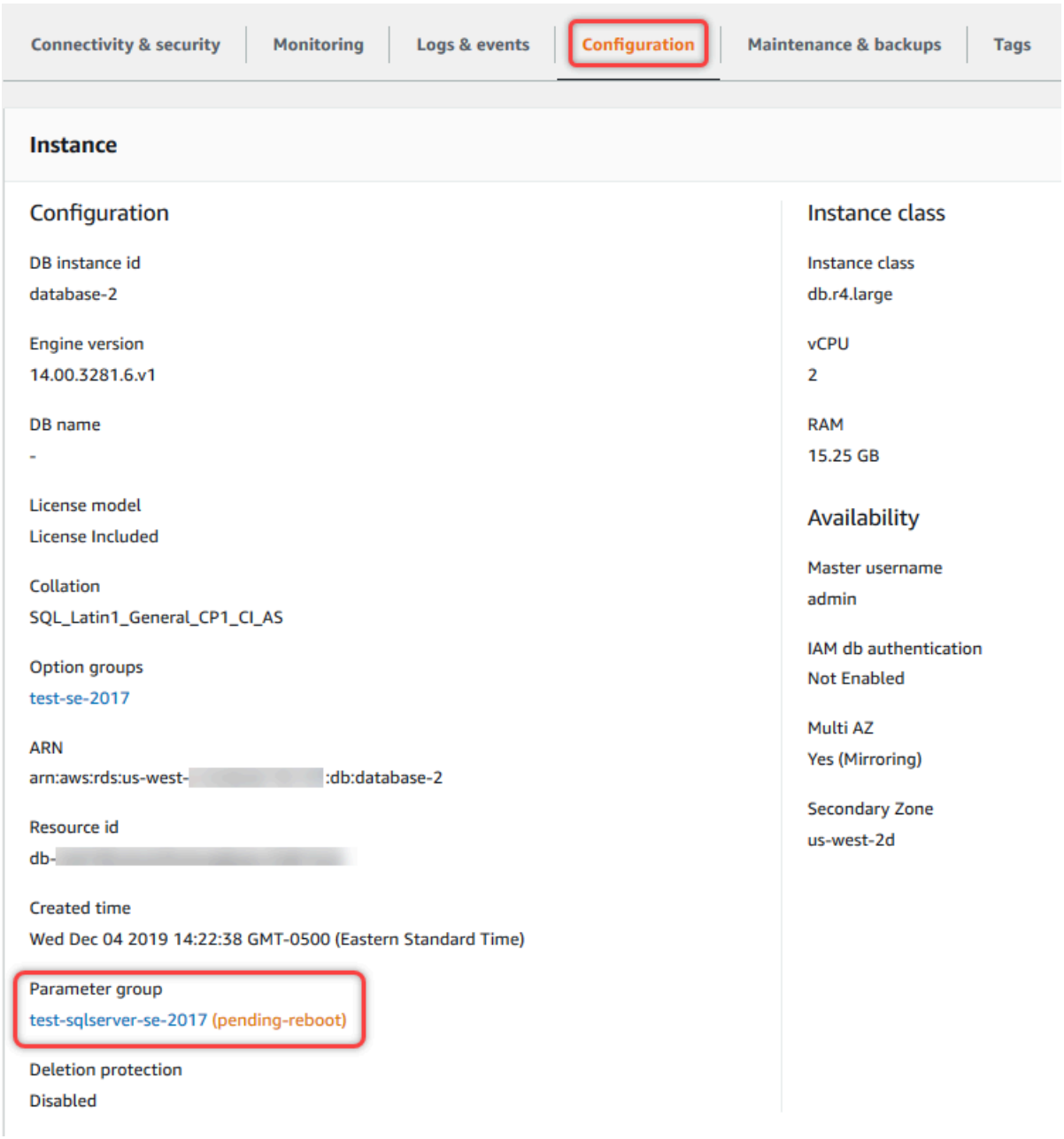
Para associar um grupo de parâmetros de banco de dados a uma instância de banco de dados, use a operação da API `ModifyDBInstance` do RDS com os seguintes parâmetros:

- `DBInstanceName`
- `DBParameterGroupName`

Modificar parâmetros em um grupo de parâmetros de banco de dados no Amazon RDS

Você pode modificar valores de parâmetros em um grupo de parâmetros de banco de dados criado pelo cliente, mas não pode alterar os valores dos parâmetros em um grupo de parâmetros de banco de dados padrão. As alterações em parâmetros em um grupo de parâmetros de banco de dados criado pelo cliente são aplicadas a todas as instâncias de bancos de dados que estão associadas a esse grupo de parâmetros.

As alterações em alguns parâmetros são aplicadas à instância de banco de dados imediatamente sem uma reinicialização. As alterações feitas em outros parâmetros só serão aplicadas depois que a instância de banco de dados no cluster de banco de dados for reinicializada. O console do RDS mostra o status do grupo de parâmetros de banco de dados associado a uma instância de banco de dados na guia Configuration (Configuração). Por exemplo, suponha que a instância de banco de dados não esteja usando as últimas alterações do grupo de parâmetros de banco de dados associado. Se sim, o console do RDS mostrará o grupo de parâmetros de banco de dados com o status pending-reboot. Para aplicar as alterações de parâmetro mais recentes a essa instância de banco de dados, reinicialize-a manualmente. Recomendamos que você sempre reinicie a instância de banco de dados depois de modificar os parâmetros. Se você não reiniciar a instância de banco de dados, uma operação de failover poderá levar mais tempo do que o normal.



Console

Como modificar os parâmetros em um grupo de parâmetros de banco de dados

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Parameter groups (Grupos de parâmetros).
3. Na lista, selecione o grupo de parâmetros que você deseja modificar.
4. Em Parameter group actions (Ações do grupo de parâmetros), escolha Edit (Editar).

5. Altere os valores dos parâmetros que você deseja modificar. Você pode percorrer os parâmetros usando as teclas de seta no canto superior direito da caixa de diálogo.

   Não altere valores em um grupo de parâmetros padrão.

6. Escolha Save changes (Salvar alterações).

AWS CLI

Para copiar um grupo de parâmetros de banco de dados, use o comando AWS CLI`modify-db-parameter-group` com as seguintes opções obrigatórias:

- `--db-parameter-group-name`
- `--parameters`

O exemplo a seguir modifica os valores `max_connections` e `max_allowed_packet` no grupo de parâmetros de banco de dados chamado mydbparametergroup.

Example

Para Linux, macOS ou Unix:

```
aws rds modify-db-parameter-group \
    --db-parameter-group-name mydbparametergroup \
    --parameters
 "ParameterName=max_connections,ParameterValue=250,ApplyMethod=immediate" \

 "ParameterName=max_allowed_packet,ParameterValue=1024,ApplyMethod=immediate"
```

Para Windows:

```
aws rds modify-db-parameter-group ^
    --db-parameter-group-name mydbparametergroup ^
    --parameters
 "ParameterName=max_connections,ParameterValue=250,ApplyMethod=immediate" ^

 "ParameterName=max_allowed_packet,ParameterValue=1024,ApplyMethod=immediate"
```

O comando produz uma saída como a seguinte:

```
DBPARAMETERGROUP  mydbparametergroup
```

API do RDS

Para copiar um grupo de parâmetros de banco de dados, use a operação da API `ModifyDBParameterGroup` do RDS com os seguintes parâmetros obrigatórios:

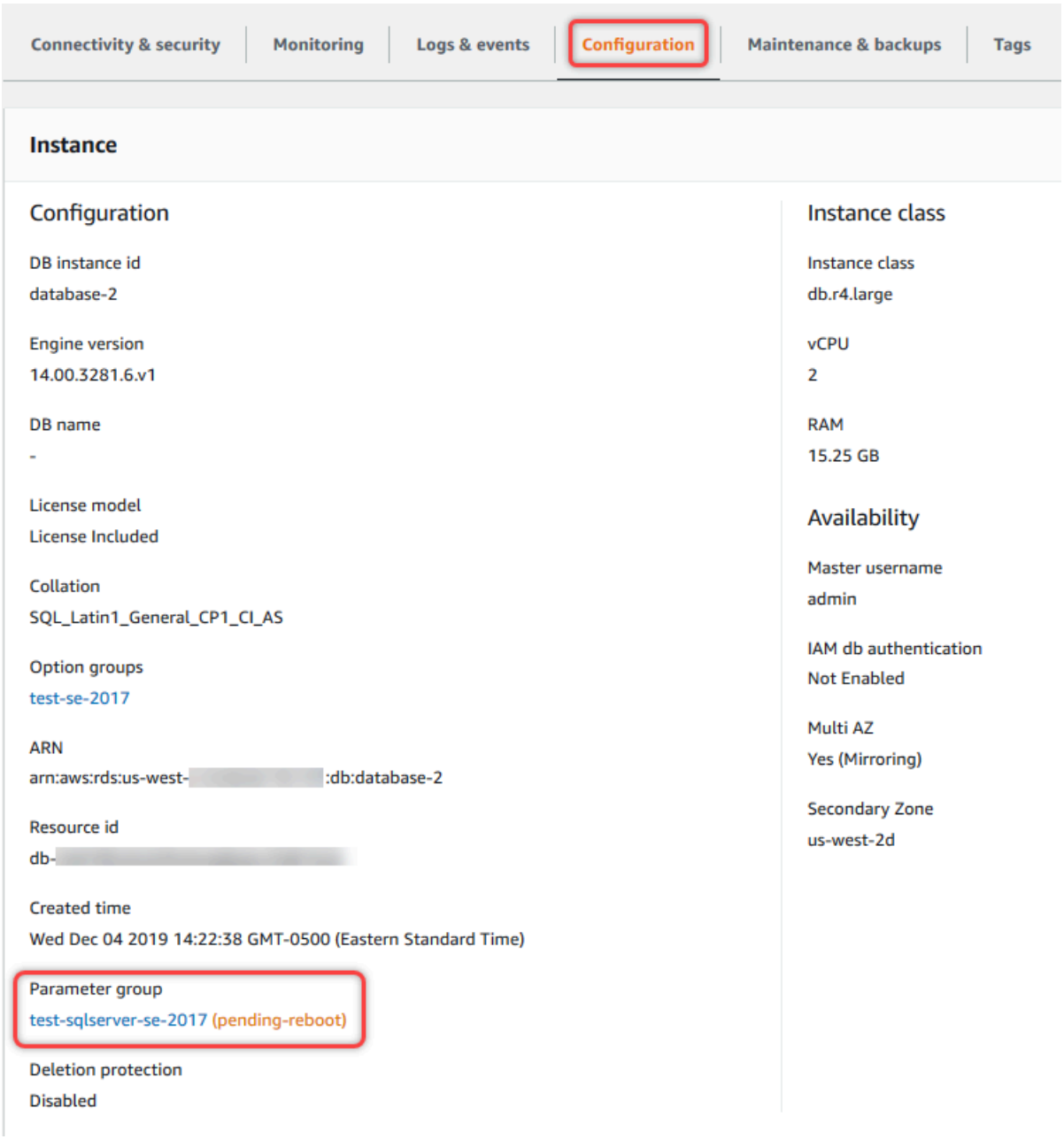
- `DBParameterGroupName`
- `Parameters`

Redefinir parâmetros em um grupo de parâmetros de banco de dados como seus valores padrão no Amazon RDS

Você pode redefinir valores de parâmetro em um grupo de parâmetros de banco de dados criado pelo cliente para seus valores padrão. As alterações em parâmetros em um grupo de parâmetros de banco de dados criado pelo cliente são aplicadas a todas as instâncias de bancos de dados que estão associadas a esse grupo de parâmetros.

Ao usar o console, você pode redefinir parâmetros específicos como seus valores padrão. No entanto, não é possível redefinir facilmente todos os parâmetros no grupo de parâmetros de banco de dados de uma só vez. Ao usar A AWS CLI ou a API do RDS, você pode redefinir parâmetros específicos como seus valores padrão. Você também pode redefinir todos os parâmetros no grupo de parâmetros de banco de dados de uma só vez.

As alterações em alguns parâmetros são aplicadas à instância de banco de dados imediatamente sem uma reinicialização. As alterações feitas em outros parâmetros só serão aplicadas depois que a instância de banco de dados no cluster de banco de dados for reinicializada. O console do RDS mostra o status do grupo de parâmetros de banco de dados associado a uma instância de banco de dados na guia Configuration (Configuração). Por exemplo, suponha que a instância de banco de dados não esteja usando as últimas alterações do grupo de parâmetros de banco de dados associado. Se sim, o console do RDS mostrará o grupo de parâmetros de banco de dados com o status pending-reboot. Para aplicar as alterações de parâmetro mais recentes a essa instância de banco de dados, reinicialize-a manualmente.



> **Note**
>
> Em um grupo de parâmetros de banco de dados padrão, os parâmetros são sempre definidos para seus valores padrão.

Console

Para redefinir parâmetros em um grupo de parâmetros de banco de dados para seus valores padrão

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, selecione Parameter groups.
3. Na lista, escolha o grupo de parâmetros.
4. Em Parameter group actions (Ações do grupo de parâmetros), escolha Edit (Editar).
5. Escolha os parâmetros que você deseja redefinir para os valores padrão. Você pode percorrer os parâmetros usando as teclas de seta no canto superior direito da caixa de diálogo.

   Não redefina os valores em um grupo de parâmetros padrão.
6. Escolha Reset (Redefinir) e, em seguida, confirme escolhendo Reset parameters (Redefinir parâmetros).

AWS CLI

Para redefinir alguns ou todos os parâmetros em um grupo de parâmetros de banco de dados, use o comando AWS CLI da `reset-db-parameter-group` com a seguinte opção necessária: `--db-parameter-group-name`.

Para redefinir todos os parâmetros no grupo de parâmetros de banco de dados, especifique a opção `--reset-all-parameters`. Para redefinir parâmetros específicos, especifique a opção `--parameters`.

O exemplo a seguir redefine todos os parâmetros no grupo de parâmetros de banco de dados chamado mydbparametergroup para seus valores padrão.

Example

Para Linux, macOS ou Unix:

```
aws rds reset-db-parameter-group \
    --db-parameter-group-name mydbparametergroup \
    --reset-all-parameters
```

Para Windows:

```
aws rds reset-db-parameter-group ^
    --db-parameter-group-name mydbparametergroup ^
    --reset-all-parameters
```

O exemplo a seguir redefine as pções `max_connections` e `max_allowed_packet` para seus valores padrão no grupo de parâmetros de banco de dados chamado mydbparametergroup.

Example

Para Linux, macOS ou Unix:

```
aws rds reset-db-parameter-group \
    --db-parameter-group-name mydbparametergroup \
    --parameters "ParameterName=max_connections,ApplyMethod=immediate" \
                 "ParameterName=max_allowed_packet,ApplyMethod=immediate"
```

Para Windows:

```
aws rds reset-db-parameter-group ^
    --db-parameter-group-name mydbparametergroup ^
    --parameters "ParameterName=max_connections,ApplyMethod=immediate" ^
                 "ParameterName=max_allowed_packet,ApplyMethod=immediate"
```

O comando produz uma saída como a seguinte:

```
DBParameterGroupName  mydbparametergroup
```

API do RDS

Para redefinir parâmetros em um grupo de parâmetros de banco de dados para seus valores padrão, use o comando `ResetDBParameterGroup` da API do RDS com o seguinte parâmetro necessário: `DBParameterGroupName`.

Para redefinir todos os parâmetros no grupo de parâmetro de banco de dados, defina o parâmetro `ResetAllParameters` como `true`. Para redefinir parâmetros específicos, especifique o parâmetro `Parameters`.

Copiar um grupo de parâmetros de banco de dados no Amazon RDS

Você pode copiar os grupos de parâmetros de banco de dados personalizados que criou. Copiar um grupo de parâmetros pode ser uma solução conveniente. Um exemplo é quando você criou

um grupo de parâmetros de banco de dados e quer incluir a maioria de seus parâmetros e valores personalizados em um novo grupo de parâmetros de banco de dados. Você pode copiar um grupo de parâmetros de banco de dados usando o AWS Management Console. Também é possível usar o comando copy-db-parameter-group da AWS CLI ou a operação da API CopyDBParameterGroup do RDS.

Depois de copiar um grupo de parâmetros de banco de dados, você deve esperar pelo menos 5 minutos antes de criar sua primeira instância de banco de dados que usa esse grupo de parâmetros de banco de dados como o padrão. Isso permite que o Amazon RDS conclua completamente a ação de cópia antes que o grupo de parâmetros seja usado. Isso é especialmente importante para parâmetros que são críticos ao criar o banco de dados padrão para uma instância de banco de dados. Um exemplo é o conjunto de caracteres para o banco de dados padrão definido pelo parâmetro `character_set_database`. Use opção Parameter Groups (Grupos de parâmetros) do console do Amazon RDS ou o comando describe-db-parameters para verificar se o seu grupo de parâmetros de banco de dados foi criado.

> **Note**
>
> Não é possível copiar um grupo de parâmetros padrão. No entanto, é possível criar um grupo de parâmetros com base em um grupo de parâmetros padrão.
> Não é possível copiar um grupo de parâmetros de banco de dados para uma Conta da AWS ou Região da AWS diferente.

Console

Para copiar um grupo de parâmetros de banco de dados

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, selecione Parameter groups.
3. Na lista, escolha o grupo de parâmetros personalizado que você deseja copiar.
4. Em Parameter group actions (Ações do grupo de parâmetros), escolha Copy (Copiar).
5. Em New DB parameter group identifier (Novo identificador do grupo de parâmetros do banco de dados), digite um nome para o novo grupo de parâmetros.
6. Em Description (Descrição), digite uma descrição para o novo grupo de parâmetros.
7. Escolha Copiar.

AWS CLI

Para copiar um grupo de parâmetros de banco de dados, use o comando da AWS CLI `copy-db-parameter-group` com as seguintes opções obrigatórias:

- `--source-db-parameter-group-identifier`
- `--target-db-parameter-group-identifier`
- `--target-db-parameter-group-description`

O exemplo a seguir cria um novo grupo de parâmetros de banco de dados chamado `mygroup2` que é uma cópia do grupo de parâmetros de banco de dados `mygroup1`.

Example

Para Linux, macOS ou Unix:

```
aws rds copy-db-parameter-group \
    --source-db-parameter-group-identifier mygroup1 \
    --target-db-parameter-group-identifier mygroup2 \
    --target-db-parameter-group-description "DB parameter group 2"
```

Para Windows:

```
aws rds copy-db-parameter-group ^
    --source-db-parameter-group-identifier mygroup1 ^
    --target-db-parameter-group-identifier mygroup2 ^
    --target-db-parameter-group-description "DB parameter group 2"
```

API do RDS

Para copiar um grupo de parâmetros de banco de dados, use a operação da API do RDS `CopyDBParameterGroup` com os seguintes parâmetros obrigatórios:

- `SourceDBParameterGroupIdentifier`
- `TargetDBParameterGroupIdentifier`
- `TargetDBParameterGroupDescription`

Listar grupos de parâmetros de banco de dados no Amazon RDS

Você pode listar os grupos de parâmetros de banco de dados que criou para sua conta da AWS.

> **Note**
>
> Grupos de parâmetros padrão são criados automaticamente a partir de um template de parâmetro padrão quando você cria uma instância de banco de dados para um mecanismo e uma versão de banco de dados específicos. Esses grupos de parâmetros padrão contêm configurações de parâmetros preferenciais e não podem ser modificados. Quando você cria um grupo de parâmetros personalizado, pode modificar as configurações desses parâmetros.

Console

Para listar todos os grupos de parâmetros de banco de dados para uma conta da AWS

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, selecione Parameter groups.

   Os grupos de parâmetros de banco de dados aparecem em uma lista.

AWS CLI

Para listar todos os grupos de parâmetros de banco de dados de uma conta da AWS, use o comando AWS CLI da `describe-db-parameter-groups`.

Example

O exemplo a seguir lista todos os grupos de parâmetros de banco de dados para uma conta da AWS.

```
aws rds describe-db-parameter-groups
```

O comando retorna uma resposta como a seguinte:

```
DBPARAMETERGROUP  default.mysql8.0     mysql8.0  Default parameter group for MySQL8.0
```

```
DBPARAMETERGROUP  mydbparametergroup   mysql8.0  My new parameter group
```

O exemplo a seguir descreve o grupo de parâmetros mydbparamgroup1.

Para Linux, macOS ou Unix:

```
aws rds describe-db-parameter-groups \
    --db-parameter-group-name mydbparamgroup1
```

Para Windows:

```
aws rds describe-db-parameter-groups ^
    --db-parameter-group-name mydbparamgroup1
```

O comando retorna uma resposta como a seguinte:

```
DBPARAMETERGROUP  mydbparametergroup1  mysql8.0  My new parameter group
```

API do RDS

Para listar todos os grupos de parâmetros de banco de dados para uma conta da AWS, use a operação da API do RDS `DescribeDBParameterGroups`.

Visualizar valores de parâmetros para um grupo de parâmetros de banco de dados no Amazon RDS

Você pode obter uma lista de todos os parâmetros em um grupo de parâmetros de banco de dados e seus valores.

Console

Para visualizar os valores de parâmetros para um grupo de parâmetros de banco de dados

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, selecione Parameter groups.

   Os grupos de parâmetros de banco de dados aparecem em uma lista.
3. Escolha o nome do grupo de parâmetros para ver sua lista de parâmetros.

AWS CLI

Para visualizar os valores de parâmetros para um grupo de parâmetros de banco de dados, use o comando AWS CLI da `describe-db-parameters` com o seguinte parâmetro obrigatório.

- `--db-parameter-group-name`

Example

O exemplo a seguir lista os parâmetros e os valores de parâmetros para um grupo de parâmetros de banco de dados chamado mydbparametergroup.

```
aws rds describe-db-parameters --db-parameter-group-name mydbparametergroup
```

O comando retorna uma resposta como a seguinte:

```
DBPARAMETER  Parameter Name            Parameter Value  Source           Data Type
 Apply Type  Is Modifiable
DBPARAMETER  allow-suspicious-udfs                      engine-default   boolean
 static      false
DBPARAMETER  auto_increment_increment                   engine-default   integer
 dynamic     true
DBPARAMETER  auto_increment_offset                      engine-default   integer
 dynamic     true
DBPARAMETER  binlog_cache_size         32768            system           integer
 dynamic     true
DBPARAMETER  socket                    /tmp/mysql.sock  system           string
 static      false
```

API do RDS

Para visualizar os valores de parâmetros para um grupo de parâmetros de banco de dados, use o comando `DescribeDBParameters` da API do RDS com o seguinte parâmetro necessário.

- `DBParameterGroupName`

Excluir um grupo de parâmetros de banco de dados no Amazon RDS

É possível excluir um grupo de parâmetros de banco de dados usando o AWS Management Console, a AWS CLI ou a API do RDS. Um grupo de parâmetros será elegível para exclusão somente se não estiver associado a uma instância de banco de dados.

Console

Como excluir um grupo de parâmetros de banco de dados

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, selecione Parameter groups.

   Os grupos de parâmetros de banco de dados aparecem em uma lista.
3. Escolha o nome dos grupos de parâmetros a serem excluídos.
4. Escolha Ações e então Excluir.
5. Revise os nomes dos grupos de parâmetros e escolha Excluir.

AWS CLI

Para excluir um grupo de parâmetros de banco de dados, use o comando `delete-db-parameter-group` da AWS CLI com o parâmetro exigido a seguir.

- `--db-parameter-group-name`

Example

O exemplo a seguir exclui um grupo de parâmetros de banco de dados chamado mydbparametergroup.

```
aws rds delete-db-parameter-group --db-parameter-group-name mydbparametergroup
```

API do RDS

Para excluir um grupo de parâmetros de banco de dados, use o comando `DeleteDBParameterGroup` da API do RDS com o parâmetro exigido a seguir.

- `DBParameterGroupName`

Trabalhar com grupos de parâmetros de clusters de banco de dados multi-AZ

Clusters de banco de dados multi-AZ usam grupos de parâmetros de cluster de banco de dados As seções a seguir descrevem a configuração e o gerenciamento de grupos de parâmetros de cluster de banco de dados.

Tópicos

- Criar um grupo de parâmetros do cluster de banco de dados
- Modificar parâmetros em um grupo de parâmetros do cluster de banco de dados
- Redefinir parâmetros em um grupo de parâmetros do cluster de banco de dados
- Copiar um grupo de parâmetros do cluster de banco de dados
- Listar grupos de parâmetros do cluster de banco de dados
- Visualizar valores de parâmetros de um grupo de parâmetros do cluster de banco de dados
- Excluir um grupo de parâmetros do cluster de banco de dados

Criar um grupo de parâmetros do cluster de banco de dados

Você pode criar um novo grupo de parâmetros de cluster de banco de dados usando o AWS Management Console, a AWS CLI ou a API do RDS.

Depois de criar um grupo de parâmetros de cluster de banco de dados, você deve esperar pelo menos 5 minutos antes de criar seu primeiro cluster que usa esse grupo de parâmetros de cluster de banco de dados. Isso permite que o Amazon RDS conclua totalmente a criação do grupo de parâmetros antes que seja usado pelo novo cluster de banco de dados. É possível usar a página Parameter Groups (Grupos de parâmetros) do console do Amazon RDS ou o comando describe-db-cluster-parameters para verificar se o seu grupo de parâmetros de cluster de banco de dados foi criado.

As seguintes limitações se aplicam ao nome do grupo de parâmetros do cluster de banco de dados:

- O nome deve ter de 1 a 255 letras, números ou hifens.

  Os nomes de grupos de parâmetros padrão podem incluir um ponto, como `default.mysql5.7`. No entanto, nomes de grupos de parâmetros personalizados não podem incluir um ponto.
- O primeiro caractere deve ser uma letra.

- O nome não pode terminar com hífen nem conter dois hífens consecutivos.

Console

Para criar um grupo de parâmetros de cluster de banco de dados

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Parameter groups (Grupos de parâmetros).
3. Escolha Create parameter group (Criar grupo de parâmetros).
4. Em Nome do grupo de parâmetros, insira o nome do novo grupo de parâmetros do cluster de banco de dados.
5. Em Descrição, insira uma descrição para o novo grupo de parâmetros do cluster de banco de dados.
6. Em Tipo de mecanismo, escolha o mecanismo de banco de dados.
7. Em Família de grupos de parâmetros, selecione uma família de grupos de parâmetros de banco de dados.
8. Escolha Criar.

AWS CLI

Para criar um grupo de parâmetros de cluster de banco de dados, use o comando da AWS CLI `create-db-cluster-parameter-group`.

O exemplo a seguir cria um grupo de parâmetros de cluster de banco de dados chamado mydbclusterparametergroup para o RDS para MySQL versão 8.0 com a descrição "My new parameter group” (Meu novo grupo de parâmetros).

Inclua os seguintes parâmetros necessários:

- `--db-cluster-parameter-group-name`
- `--db-parameter-group-family`
- `--description`

Para listar todas as famílias de grupos de parâmetros disponíveis, use o comando a seguir:

```
aws rds describe-db-engine-versions --query "DBEngineVersions[].DBParameterGroupFamily"
```

> **Note**
>
> A saída contém duplicatas.

Example

Para Linux, macOS ou Unix:

```
aws rds create-db-cluster-parameter-group \
    --db-cluster-parameter-group-name mydbclusterparametergroup \
    --db-parameter-group-family mysql8.0 \
    --description "My new cluster parameter group"
```

Para Windows:

```
aws rds create-db-cluster-parameter-group ^
    --db-cluster-parameter-group-name mydbclusterparametergroup ^
    --db-parameter-group-family mysql8.0 ^
    --description "My new cluster parameter group"
```

O comando gerará uma saída semelhante à seguinte:

```
{
    "DBClusterParameterGroup": {
        "DBClusterParameterGroupName": "mydbclusterparametergroup",
        "DBParameterGroupFamily": "mysql8.0",
        "Description": "My new cluster parameter group",
        "DBClusterParameterGroupArn": "arn:aws:rds:us-east-1:123456789012:cluster-
pg:mydbclusterparametergroup2"
    }
}
```

API do RDS

Para criar um grupo de parâmetros de cluster de banco de dados, use a ação da API do RDS `CreateDBClusterParameterGroup`.

Inclua os seguintes parâmetros necessários:

- `DBClusterParameterGroupName`
- `DBParameterGroupFamily`
- `Description`

Modificar parâmetros em um grupo de parâmetros do cluster de banco de dados

É possível modificar valores de parâmetros em um grupo de parâmetros do cluster de banco de dados criado pelo cliente. Não é possível alterar os valores de parâmetros em um grupo de parâmetros do cluster de banco de dados padrão. As alterações dos parâmetros em um grupo de parâmetros de cluster de banco de dados criado pelo cliente são aplicadas a todos os clusters de bancos de dados que estão associados a esse grupo de parâmetros de cluster de banco de dados.

Console

Para modificar um grupo de parâmetros de cluster de banco de dados

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, selecione Parameter groups.
3. Na lista, escolha o grupo de parâmetros que você deseja modificar.
4. Em Parameter group actions (Ações do grupo de parâmetros), escolha Edit (Editar).
5. Altere os valores dos parâmetros que você deseja modificar. Você pode percorrer os parâmetros usando as teclas de seta no canto superior direito da caixa de diálogo.

   Não altere valores em um grupo de parâmetros padrão.
6. Selecione Save changes.
7. Reinicialize a cluster para aplicar as alterações a ela.

   Se você não o cluster, uma operação de failover poderá levar mais tempo do que o normal.

AWS CLI

Para modificar um grupo de parâmetros de cluster de banco de dados, use o comando da AWS CLI `modify-db-cluster-parameter-group` com os seguintes parâmetros obrigatórios:

- `--db-cluster-parameter-group-name`
- `--parameters`

O exemplo a seguir modifica os valores `server_audit_logging` e `server_audit_logs_upload` no grupo de parâmetros de cluster de banco de dados chamado mydbclusterparametergroup.

Example

Para Linux, macOS ou Unix:

```
aws rds modify-db-cluster-parameter-group \
    --db-cluster-parameter-group-name mydbclusterparametergroup \
    --parameters
 "ParameterName=server_audit_logging,ParameterValue=1,ApplyMethod=immediate" \

 "ParameterName=server_audit_logs_upload,ParameterValue=1,ApplyMethod=immediate"
```

Para Windows:

```
aws rds modify-db-cluster-parameter-group ^
    --db-cluster-parameter-group-name mydbclusterparametergroup ^
    --parameters
 "ParameterName=server_audit_logging,ParameterValue=1,ApplyMethod=immediate" ^

 "ParameterName=server_audit_logs_upload,ParameterValue=1,ApplyMethod=immediate"
```

O comando produz uma saída como a seguinte:

```
DBCLUSTERPARAMETERGROUP  mydbclusterparametergroup
```

API do RDS

Para modificar um grupo de parâmetros de cluster de banco de dados, use o comando da API do RDS [ModifyDBClusterParameterGroup] com os seguintes parâmetros obrigatórios:

- `DBClusterParameterGroupName`
- `Parameters`

Redefinir parâmetros em um grupo de parâmetros do cluster de banco de dados

Você pode redefinir parâmetros para seus valores padrão em um grupo de parâmetro de cluster de banco de dados criado pelo cliente. As alterações dos parâmetros em um grupo de parâmetros de cluster de banco de dados criado pelo cliente são aplicadas a todos os clusters de bancos de dados que estão associados a esse grupo de parâmetros de cluster de banco de dados.

> **Note**
>
> Em um grupo de parâmetro de cluster de banco de dados padrão, os parâmetros são sempre definidos como seus valores padrão.

Console

Para redefinir parâmetros em um grupo de parâmetros de cluster de banco de dados para seus valores padrão

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, selecione Parameter groups.
3. Na lista, escolha o grupo de parâmetros.
4. Em Parameter group actions (Ações do grupo de parâmetros), escolha Edit (Editar).
5. Escolha os parâmetros que você deseja redefinir para os valores padrão. Você pode percorrer os parâmetros usando as teclas de seta no canto superior direito da caixa de diálogo.

   Não redefina os valores em um grupo de parâmetros padrão.
6. Escolha Reset (Redefinir) e, em seguida, confirme escolhendo Reset parameters (Redefinir parâmetros).
7. Reinicialize a cluster do banco de dados.

AWS CLI

Para redefinir parâmetros em um grupo de parâmetros de cluster de banco de dados para seus valores padrão, use o comando da AWS CLI `reset-db-cluster-parameter-group` com a seguinte opção obrigatória: `--db-cluster-parameter-group-name`.

Para redefinir todos os parâmetros no grupo de parâmetro do cluster de banco de dados, especifique a opção `--reset-all-parameters`. Para redefinir parâmetros específicos, especifique a opção `--parameters`.

O exemplo a seguir redefine todos os parâmetros no grupo de parâmetros de banco de dados chamado mydbparametergroup para seus valores padrão.

Example

Para Linux, macOS ou Unix:

```
aws rds reset-db-cluster-parameter-group \
    --db-cluster-parameter-group-name mydbparametergroup \
    --reset-all-parameters
```

Para Windows:

```
aws rds reset-db-cluster-parameter-group ^
    --db-cluster-parameter-group-name mydbparametergroup ^
    --reset-all-parameters
```

O exemplo a seguir redefine os valores padrão `server_audit_logging` e `server_audit_logs_upload` no grupo de parâmetro de cluster de banco de dados chamado mydbclusterparametergroup.

Example

Para Linux, macOS ou Unix:

```
aws rds reset-db-cluster-parameter-group \
    --db-cluster-parameter-group-name mydbclusterparametergroup \
    --parameters "ParameterName=server_audit_logging,ApplyMethod=immediate" \
                 "ParameterName=server_audit_logs_upload,ApplyMethod=immediate"
```

Para Windows:

```
aws rds reset-db-cluster-parameter-group ^
    --db-cluster-parameter-group-name mydbclusterparametergroup ^
    --parameters
 "ParameterName=server_audit_logging,ParameterValue=1,ApplyMethod=immediate" ^
```

```
"ParameterName=server_audit_logs_upload,ParameterValue=1,ApplyMethod=immediate"
```

O comando produz uma saída como a seguinte:

```
DBClusterParameterGroupName  mydbclusterparametergroup
```

API do RDS

Para redefinir parâmetros em um grupo de parâmetros de cluster de banco de dados para seus valores padrão, use o comando da API do RDS de `ResetDBClusterParameterGroup` com o seguinte parâmetro necessário: `DBClusterParameterGroupName`.

Para redefinir todos os parâmetros no grupo de parâmetros do cluster de banco de dados, defina o parâmetro `ResetAllParameters` para `true`. Para redefinir parâmetros específicos, especifique o parâmetro `Parameters`.

Copiar um grupo de parâmetros do cluster de banco de dados

Você pode copiar os grupos de parâmetros de cluster de banco de dados personalizados que criou. Copiar um grupo de parâmetros é uma solução conveniente quando você já criou um grupo de parâmetros de cluster de banco de dados e deseja incluir a maioria dos parâmetros e valores personalizados desse grupo em um novo grupo de parâmetros de cluster de banco de dados. É possível copiar um grupo de parâmetros de cluster de banco de dados usando o comando AWS CLI copy-db-cluster-parameter-group ou a operação da API do RDS CopyDBClusterParameterGroup.

Depois de copiar um grupo de parâmetros de cluster de banco de dados, você deve esperar pelo menos 5 minutos antes de criar seu primeiro cluster que usa esse grupo de parâmetros de cluster de banco de dados. Isso permite que o Amazon RDS conclua totalmente a cópia do grupo de parâmetros antes que seja usado pelo novo cluster de banco de dados. É possível usar a página Parameter Groups (Grupos de parâmetros) do console do Amazon RDS ou o comando describe-db-cluster-parameters para verificar se o seu grupo de parâmetros de cluster de banco de dados foi criado.

> **Note**
>
> Não é possível copiar um grupo de parâmetros padrão. No entanto, é possível criar um grupo de parâmetros com base em um grupo de parâmetros padrão.

Não é possível copiar um grupo de parâmetros de cluster de banco de dados para uma Conta da AWS ou Região da AWS diferente.

Console

Para copiar um grupo de parâmetros de cluster de banco de dados

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, selecione Parameter groups.
3. Na lista, escolha o grupo de parâmetros personalizado que você deseja copiar.
4. Em Parameter group actions (Ações do grupo de parâmetros), escolha Copy (Copiar).
5. Em New DB parameter group identifier (Novo identificador do grupo de parâmetros do banco de dados), digite um nome para o novo grupo de parâmetros.
6. Em Description (Descrição), digite uma descrição para o novo grupo de parâmetros.
7. Escolha Copiar.

AWS CLI

Para copiar um grupo de parâmetros de cluster de banco de dados, use o comando da AWS CLI `copy-db-cluster-parameter-group` com os seguintes parâmetros obrigatórios:

- `--source-db-cluster-parameter-group-identifier`
- `--target-db-cluster-parameter-group-identifier`
- `--target-db-cluster-parameter-group-description`

O exemplo a seguir cria um novo grupo de parâmetros de cluster de banco de dados chamado `mygroup2` que é uma cópia do grupo de parâmetros de cluster de banco de dados `mygroup1`.

Example

Para Linux, macOS ou Unix:

```
aws rds copy-db-cluster-parameter-group \
    --source-db-cluster-parameter-group-identifier mygroup1 \
    --target-db-cluster-parameter-group-identifier mygroup2 \
```

```
    --target-db-cluster-parameter-group-description "DB parameter group 2"
```

Para Windows:

```
aws rds copy-db-cluster-parameter-group ^
    --source-db-cluster-parameter-group-identifier mygroup1 ^
    --target-db-cluster-parameter-group-identifier mygroup2 ^
    --target-db-cluster-parameter-group-description "DB parameter group 2"
```

API do RDS

Para copiar um grupo de parâmetros de cluster de banco de dados, use a operação da API do RDS `CopyDBClusterParameterGroup` com os seguintes parâmetros obrigatórios:

- `SourceDBClusterParameterGroupIdentifier`
- `TargetDBClusterParameterGroupIdentifier`
- `TargetDBClusterParameterGroupDescription`

Listar grupos de parâmetros do cluster de banco de dados

Você pode listar os grupos de parâmetros do cluster de banco de dados que criou para sua conta da AWS.

> **Note**
>
> Grupos de parâmetros padrão são criados automaticamente a partir de um modelo de parâmetro padrão quando você cria um cluster de banco de dados para um mecanismo e uma versão de banco de dados específicos. Esses grupos de parâmetros padrão contêm configurações de parâmetros preferenciais e não podem ser modificados. Quando você cria um grupo de parâmetros personalizado, pode modificar as configurações desses parâmetros.

Console

Para listar todos os grupos de parâmetros do cluster de banco de dados para uma conta da AWS

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, selecione Parameter groups.

Os parameter groups de cluster de banco de dados são exibidos na lista com DB cluster parameter group (Parameter group de cluster de banco de dados) para o Type (Tipo).

AWS CLI

Para listar todos os grupos de parâmetros de cluster de banco de dados de uma conta da AWS, use o comando da AWS CLI `describe-db-cluster-parameter-groups`.

Example

O exemplo a seguir lista todos os grupos de parâmetros de cluster de banco de dados de uma conta da AWS.

```
aws rds describe-db-cluster-parameter-groups
```

O exemplo a seguir descreve o grupo de parâmetros mydbclusterparametergroup.

Para Linux, macOS ou Unix:

```
aws rds describe-db-cluster-parameter-groups \
    --db-cluster-parameter-group-name mydbclusterparametergroup
```

Para Windows:

```
aws rds describe-db-cluster-parameter-groups ^
    --db-cluster-parameter-group-name mydbclusterparametergroup
```

O comando retorna uma resposta como a seguinte:

```
{
    "DBClusterParameterGroups": [
        {
            "DBClusterParameterGroupName": "mydbclusterparametergroup2",
            "DBParameterGroupFamily": "mysql8.0",
            "Description": "My new cluster parameter group",
            "DBClusterParameterGroupArn": "arn:aws:rds:us-east-1:123456789012:cluster-
pg:mydbclusterparametergroup"
        }
    ]
}
```

API do RDS

Para listar todos os grupos de parâmetros de cluster de banco de dados de uma conta da AWS, use a ação da API do RDS `DescribeDBClusterParameterGroups`.

Visualizar valores de parâmetros de um grupo de parâmetros do cluster de banco de dados

Você pode obter uma lista de todos os parâmetros em um grupo de parâmetros de cluster de banco de dados e seus valores.

Console

Para visualizar os valores de parâmetros para um grupo de parâmetros de cluster de banco de dados

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, selecione Parameter groups.

   Os grupos de parâmetros de cluster de banco de dados são exibidos na lista com DB cluster parameter group (Grupo de parâmetros de cluster de banco de dados) para o Type (Tipo).
3. Escolha o nome do grupo de parâmetros de cluster de banco de dados para ver sua lista de parâmetros.

AWS CLI

Para visualizar os valores de parâmetros para um grupo de parâmetros de cluster de banco de dados, use o comando da AWS CLI `describe-db-cluster-parameters` com o parâmetro obrigatório a seguir.

- `--db-cluster-parameter-group-name`

Example

O exemplo a seguir lista os parâmetros e os valores de parâmetros para um grupo de parâmetros de cluster de banco de dados chamado mydbclusterparametergroup, no formato JSON.

O comando retorna uma resposta como a seguinte:

```
aws rds describe-db-cluster-parameters --db-cluster-parameter-group-
name mydbclusterparametergroup
```

```
{
    "Parameters": [
        {
            "ParameterName": "activate_all_roles_on_login",
            "ParameterValue": "0",
            "Description": "Automatically set all granted roles as active after the
 user has authenticated successfully.",
            "Source": "engine-default",
            "ApplyType": "dynamic",
            "DataType": "boolean",
            "AllowedValues": "0,1",
            "IsModifiable": true,
            "ApplyMethod": "pending-reboot",
            "SupportedEngineModes": [
                "provisioned"
            ]
        },
        {
            "ParameterName": "allow-suspicious-udfs",
            "Description": "Controls whether user-defined functions that have only an
 xxx symbol for the main function can be loaded",
            "Source": "engine-default",
            "ApplyType": "static",
            "DataType": "boolean",
            "AllowedValues": "0,1",
            "IsModifiable": false,
            "ApplyMethod": "pending-reboot",
            "SupportedEngineModes": [
                "provisioned"
            ]
        },
...
```

API do RDS

Para visualizar os valores de parâmetros para um grupo de parâmetros de cluster de banco de dados, use o comando `DescribeDBClusterParameters` da API do RDS com o seguinte parâmetro necessário.

- `DBClusterParameterGroupName`

Em alguns casos, os valores permitidos para um parâmetro não são mostrados. São sempre parâmetros em que a origem é o padrão do mecanismo de banco de dados.

Para visualizar os valores desses parâmetros, você pode executar as seguintes instruções SQL:

- MySQL:

```
-- Show the value of a particular parameter
mysql$ SHOW VARIABLES LIKE '%parameter_name%';

-- Show the values of all parameters
mysql$ SHOW VARIABLES;
```

- PostgreSQL:

```
-- Show the value of a particular parameter
postgresql=> SHOW parameter_name;

-- Show the values of all parameters
postgresql=> SHOW ALL;
```

Excluir um grupo de parâmetros do cluster de banco de dados

É possível excluir um grupo de parâmetros de cluster de banco de dados usando o AWS Management Console, a AWS CLI ou a API do RDS. Um grupo de parâmetros do cluster de banco de dados será elegível para exclusão somente se não estiver associado a um cluster de banco de dados.

Console

Para excluir um grupo de parâmetros

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Parameter groups (Grupos de parâmetros).

   Os grupos de parâmetros aparecem em uma lista.

3. Escolha o nome dos grupos de parâmetros do cluster de banco de dados a serem excluídos.
4. Escolha Ações e então Excluir.
5. Revise os nomes dos grupos de parâmetros e escolha Excluir.

AWS CLI

Para excluir um grupo de parâmetros de cluster de banco de dados, use o comando `delete-db-cluster-parameter-group` da AWS CLI com os parâmetros obrigatórios a seguir.

- `--db-parameter-group-name`

Example

O exemplo a seguir exclui um grupo de parâmetros do cluster de banco de dados chamado mydbparametergroup.

```
aws rds delete-db-cluster-parameter-group --db-parameter-group-name mydbparametergroup
```

API do RDS

Para excluir um grupo de parâmetros de cluster de banco de dados, use o comando `DeleteDBClusterParameterGroup` da API do RDS com os parâmetros obrigatórios a seguir.

- `DBParameterGroupName`

Comparação de grupos de parâmetros de banco de dados

É possível usar o AWS Management Console para visualizar as diferenças entre dois grupos de parâmetros de banco de dados.

Os grupos de parâmetros especificados devem ser grupos de parâmetros de banco de dados ou devem ser grupos de parâmetros de cluster de banco de dados. Isso se aplica mesmo se o mecanismo de banco de dados e a versão forem os mesmos. Por exemplo, não é possível comparar um grupo de parâmetros do banco de dados do `aurora-mysql8.0` (Aurora MySQL versão 3) e um grupo de parâmetros do cluster de banco de dados do `aurora-mysql8.0`.

Você pode comparar grupos de parâmetros de banco de dados do Aurora MySQL e do RDS para MySQL, mesmo para versões diferentes, mas não pode comparar grupos de parâmetros de banco de dados do Aurora PostgreSQL e do RDS para PostgreSQL.

Para comparar dois grupos de parâmetros de banco de dados

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, selecione Parameter groups.
3. Na lista, escolha os dois grupos de parâmetros que você deseja comparar.

   > **Note**
   >
   > Para comparar um grupo de parâmetros padrão com um grupo de parâmetros personalizado, primeiro escolha o grupo de parâmetros padrão na guia Padrão e, depois, selecione o grupo de parâmetros personalizado na guia Personalizado.

4. Em Ações, selecione Comparar.

Especificação de parâmetros de banco de dados

Os tipos de parâmetros de banco de dados incluem os seguintes:

- Inteiro
- Booliano
- String
- Longo
- Double
- Timestamp
- Objeto de outros tipos de dados definidos
- Matriz de valores do tipo integer, boolean, string, long, double, timestamp ou object

Você também pode especificar parâmetros inteiros e booleanos usando expressões, fórmulas e funções.

Para o mecanismo do Oracle, você pode usar a variável de fórmula `DBInstanceClassHugePagesDefault` para especificar um parâmetro de banco de dados booleano. Consulte Variáveis de fórmulas de parâmetros de banco de dados.

Para o mecanismo do PostgreSQL, você pode usar uma expressão para especificar um parâmetro de banco de dados booleano. Consulte Expressões de parâmetro de banco de dados booleanas.

Sumário

- Fórmulas de parâmetros de banco de dados
  - Variáveis de fórmulas de parâmetros de banco de dados
  - Operadores de fórmulas de parâmetros de banco de dados
- Funções de parâmetros de banco de dados
- Expressões de parâmetro de banco de dados booleanas
- Expressões de log de parâmetros de banco de dados
- Exemplos de valores de parâmetros de banco de dados

Fórmulas de parâmetros de banco de dados

Uma fórmula de parâmetro de banco de dados é uma expressão resolvida como um valor inteiro ou booliano. A expressão é delimitada por chaves: {}. você pode usar uma fórmula para um valor de parâmetro de banco de dados ou como um argumento para uma função de parâmetro de banco de dados.

Sintaxe

```
{FormulaVariable}
{FormulaVariable*Integer}
{FormulaVariable*Integer/Integer}
{FormulaVariable/Integer}
```

Variáveis de fórmulas de parâmetros de banco de dados

Cada variável de fórmula retorna um inteiro ou um valor booliano. Os nomes das variáveis diferenciam maiúsculas de minúsculas.

AllocatedStorage

Retorna um inteiro que representa o tamanho, em bytes, do volume de dados.

DBInstanceClassHugePagesDefault

Retorna um valor Booliano. Atualmente, é compatível apenas com os mecanismos do Oracle.

Para obter mais informações, consulte Ativar o HugePages para uma instância do RDS para Oracle.

DBInstanceClassMemory

Retorna um número inteiro para o número de bytes de memória disponíveis para o processo do banco de dados. Esse número é calculado internamente começando com a quantidade total de memória para a classe de instância de banco de dados. A partir disso, o cálculo subtrai a memória reservada para o sistema operacional e os processos do RDS que gerenciam a instância. Portanto, o número é sempre um pouco menor do que as figuras de memória mostradas nas tabelas de classe de instância em Classes de instâncias de banco de dados . O valor exato depende de uma combinação de fatores. São eles classe de instância, mecanismo de banco de dados e se ele se aplica a uma instância do RDS ou a uma instância que faça parte de um cluster do Aurora.

DBInstanceVCPU

Retorna um número inteiro representando a quantidade de unidades de processamento centrais virtuais (vCPUs) usadas pelo Amazon RDS para gerenciar a instância.

EndPointPort

Retorna um inteiro representando a porta usada ao se conectar à instância de banco de dados.

Réplica TrueIF

Retorna `1` se a instância de banco de dados é uma réplica de leitura e `0` se não é. Esse é o valor padrão do `read_only` parâmetro no .

Operadores de fórmulas de parâmetros de banco de dados

Fórmulas de parâmetros de banco de dados que oferecem suporte a dois operadores: divisão e multiplicação.

Operador de divisão: /

Divide o dividendo pelo divisor, retornando um quociente inteiro. Casas decimais no quociente são truncadas, não arredondadas.

Sintaxe

```
dividend / divisor
```

Os argumentos de dividendo e divisor devem ser expressões de inteiros.

Operador de multiplicação: *

Multiplica as expressões, retornando seu produto. As casas decimais nas expressões são truncadas, e não arredondadas.

Sintaxe

```
expression * expression
```

Ambas as expressões devem ser números inteiros.

Funções de parâmetros de banco de dados

Você especifica os argumentos das funções de parâmetro de banco de dados como inteiros ou fórmulas. Cada função deve ter pelo menos um argumento. Especifique vários argumentos como uma lista separada por vírgulas. A lista não pode ter membros vazios, como argument1,,argument3. Nomes de função não diferenciam maiúsculas de minúsculas.

IF

Retorna um argumento.

Atualmente, é compatível apenas com mecanismos do Oracle e o único primeiro argumento compatível é `{DBInstanceClassHugePagesDefault}`. Para obter mais informações, consulte Ativar o HugePages para uma instância do RDS para Oracle.

Sintaxe

```
IF(argument1, argument2, argument3)
```

Retorna o segundo argumento se o primeiro argumento é avaliado como verdadeiro. Retorna o terceiro argumento, caso contrário.

GREATEST

Retorna o maior valor de uma lista de números inteiros ou fórmulas de parâmetros.

Sintaxe

```
GREATEST(argument1, argument2,...argumentn)
```

Retorna um número inteiro.

LEAST

Retorna o menor valor de uma lista de números inteiros ou fórmulas de parâmetros.

Sintaxe

```
LEAST(argument1, argument2,...argumentn)
```

Retorna um número inteiro.

SUM

Adiciona os valores dos números inteiros ou fórmulas de parâmetros especificados.

Sintaxe

```
SUM(argument1, argument2,...argumentn)
```

Retorna um número inteiro.

Expressões de parâmetro de banco de dados booleanas

Uma expressão de parâmetro de banco de dados booleana é resolvida como um valor booleano de 1 ou 0. A expressão é delimitada por aspas.

> **Note**
>
> Expressões de parâmetro de banco de dados booleanas são compatíveis apenas com o mecanismo do PostgreSQL.

Sintaxe

```
"expression operator expression"
```

Ambas as expressões devem ser resolvidas como inteiros. Uma expressão pode ser a seguinte:

- Uma constante de inteiro
- Fórmula de parâmetros de banco de dados
- Função de parâmetros de banco de dados
- Variável de parâmetro de banco de dados

As expressões de parâmetro de banco de dados booleanas suportam os seguintes operadores de desigualdade:

O operador maior que: **>**

Sintaxe

```
"expression > expression"
```

O operador menor que: **<**

Sintaxe

```
"expression < expression"
```

Os operadores maior que ou igual a: **>=**, **=>**

Sintaxe

```
"expression >= expression"
"expression => expression"
```

Os operadores menor que ou igual a: **<=**, **=<**

Sintaxe

```
"expression <= expression"
"expression =< expression"
```

Example usando uma expressão de parâmetro de banco de dados booleana

O seguinte exemplo de expressão de parâmetro de banco de dados booliana compara o resultado de uma fórmula de parâmetro com um valor inteiro. Ele faz isso para modificar o parâmetro de banco de dados booliano `wal_compression` para uma instância de banco de dados do PostgreSQL. A expressão do parâmetro compara o número de vCPUs com o valor 2. Se o número de vCPUs for maior que 2, então o parâmetro de banco de dados `wal_compression` é definido como true (verdadeiro).

```
aws rds modify-db-parameter-group --db-parameter-group-name group-name \
--parameters "ParameterName=wal_compression,ParameterValue=\"{DBInstanceVCPU} > 2\" "
```

Expressões de log de parâmetros de banco de dados

Você pode definir um valor de parâmetro de banco de dados inteiro como uma expressão de log. A expressão é delimitada por chaves: {}. Por exemplo:

```
{log(DBInstanceClassMemory/8187281418)*1000}
```

A função `log` representa a base de log 2. Esse exemplo também usa a variável de fórmula `DBInstanceClassMemory`. Consulte Variáveis de fórmulas de parâmetros de banco de dados.

> **Note**
>
> Atualmente, você não pode especificar o parâmetro `innodb_log_file_size` do MySQL com qualquer valor diferente de um inteiro.

Exemplos de valores de parâmetros de banco de dados

Esses exemplos mostram o uso de fórmulas, funções e expressões para os valores de parâmetros de banco de dados.

> **Warning**
>
> Definir parâmetros indevidamente em um grupo de parâmetros de banco de dados pode ter efeitos adversos não intencionais. Estes podem incluir performance degradada e instabilidade do sistema. Sempre tenha cuidado ao modificar os parâmetros do banco de dados e faça backup dos seus dados antes de modificar seu grupo de parâmetros de banco

de dados. Faça testes com alterações de grupos de parâmetros em instâncias de bancos de dados de teste, criadas usando restaurações pontuais, antes de aplicar essas alterações às suas instâncias de banco de dados de produção.

Example uso da função de parâmetros de banco de dados GREATEST (Maior)

Você pode especificar a função `GREATEST` em um parâmetro de processos do Oracle. Use-a para definir a quantidade de processos do usuário para o maior entre 80 ou `DBInstanceClassMemory` dividido por 9.868.951.

```
GREATEST({DBInstanceClassMemory/9868951},80)
```

Example uso da função de parâmetro de banco de dados LEAST (Menor)

Você pode especificar a função `LEAST` em um valor do parâmetro MySQL `max_binlog_cache_size`. Use-a para definir o tamanho máximo de cache que uma transação pode usar em uma instância MySQL para o menor entre 1 MB ou `DBInstanceClass`/256.

```
LEAST({DBInstanceClassMemory/256},10485760)
```

Criar um cache do Amazon ElastiCache usando as configurações da instância de banco de dados do Amazon RDS

O ElastiCache é um serviço de armazenamento em cache em memória totalmente gerenciado que fornece latências de leitura e gravação em microssegundos que são compatíveis com casos de uso flexíveis e em tempo real. O ElastiCache pode ajudar você a acelerar a performance de aplicações e bancos de dados. Você pode usar o ElastiCache como armazenamento de dados primário para casos de uso que não exigem durabilidade de dados, como tabelas de classificação de jogos, streaming e análise de dados. O ElastiCache ajuda a remover a complexidade associada à implantação e ao gerenciamento de um ambiente de cache distribuído. Para ter mais informações, consulte Common ElastiCache Use Cases and How ElastiCache Can Help para Memcached e Common ElastiCache Use Cases and How ElastiCache Can Help para Redis OSS. É possível usar o console do Amazon RDS para criar o cache do ElastiCache.

Você pode operar o Amazon ElastiCache em dois formatos. Você pode começar com um cache sem servidor ou optar por criar seu próprio cluster de cache. Se você optar por desenvolver seu próprio cluster de cache, saiba que o ElastiCache funciona com os mecanismos do Redis OSS e do Memcached. Se você não tiver certeza de qual mecanismo deseja usar, consulte Comparing Memcached and Redis OSS. Para ter mais informações sobre o Amazon ElastiCache, consulte o Guia do usuário do Amazon ElastiCache.

Tópicos

- Visão geral da criação do cache do ElastiCache com as configurações da instância de banco de dados do RDS
- Criar um cache do ElastiCache com as configurações de uma instância de banco de dados do RDS

Visão geral da criação do cache do ElastiCache com as configurações da instância de banco de dados do RDS

É possível criar um cache do ElastiCache pelo Amazon RDS usando as mesmas configurações de uma instância de banco de dados do RDS recém-criada ou existente.

Alguns casos de uso para associar um cache do ElastiCache à instância de banco de dados:

- Você pode economizar custos e melhorar a performance usando o ElastiCache com o RDS em vez de executar somente no RDS.

Por exemplo, você pode economizar até 55% em custos e alcançar performance de leitura até 80 vezes mais rápida usando o ElastiCache com o RDS para MySQL, em comparação com o RDS para MySQL sozinho.

- É possível usar o cache do ElastiCache como um datastore primário para aplicações que não exigem durabilidade de dados. As aplicações que usam Redis OSS ou o Memcached podem usar o ElastiCache sem praticamente nenhuma modificação.

Ao criar um cache do ElastiCache pelo RDS, esse cache herda as seguintes configurações da instância de banco de dados do RDS associada:

- Configurações de conectividade do ElastiCache
- Configurações de segurança do ElastiCache

Também é possível definir as configurações de cache de acordo com seus requisitos.

Configurar o ElastiCache nas aplicações

As aplicações devem ser configuradas para usar o cache do ElastiCache. Também é possível otimizar e melhorar o desempenho do cache configurando as aplicações para usar estratégias de armazenamento em cache, dependendo dos requisitos.

- Para acessar o cache do ElastiCache e começar, consulte Getting started with ElastiCache (Redis OSS) e Getting started with ElastiCache (Memcached).
- Para ter mais informações sobre estratégias de armazenamento em cache, consulte Caching strategies and best practices para Memcached e Caching strategies and best practices para Redis OSS.
- Para ter mais informações sobre alta disponibilidade nos clusters do ElastiCache (Redis OSS), consulte High availability using replication groups.
- É possível incorrer em custos associados a armazenamento de backup, transferência de dados dentro ou entre regiões ou uso do AWS Outposts. Para obter detalhes de preço, consulte Preço do Amazon ElastiCache.

Criar um cache do ElastiCache com as configurações de uma instância de banco de dados do RDS

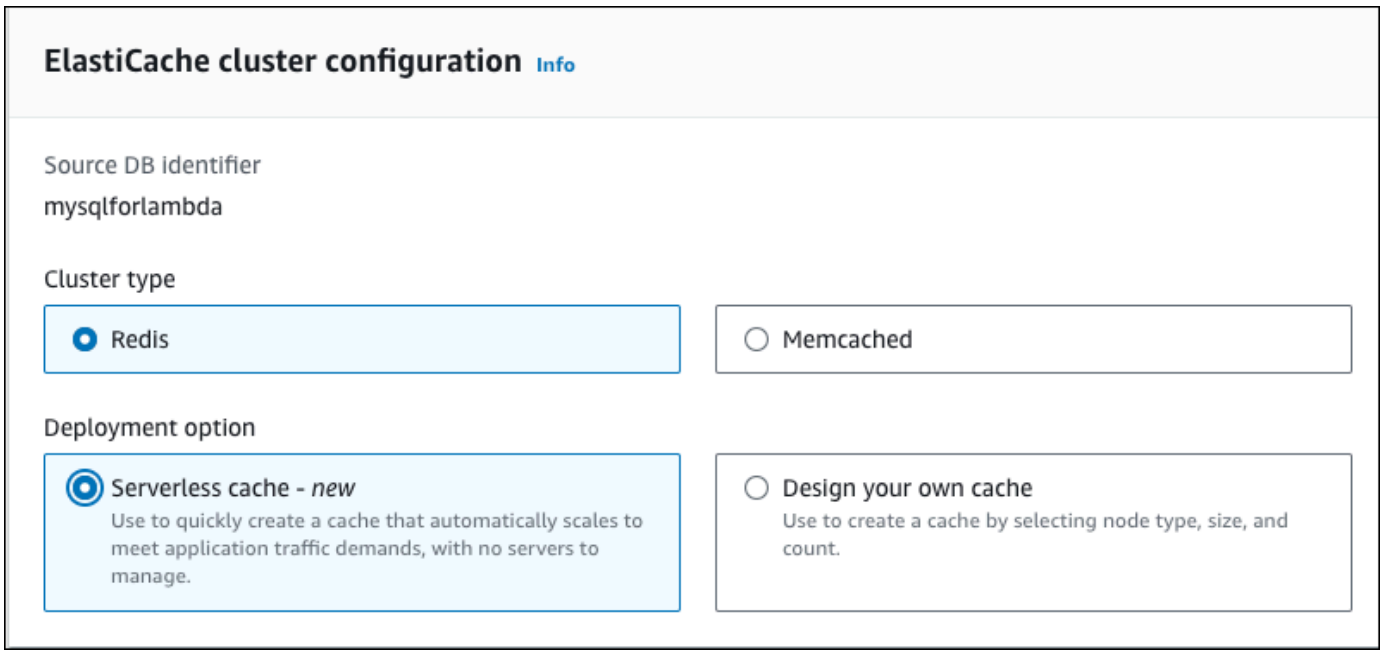
É possível criar um cache do ElastiCache para instância de banco de dados do RDS com configurações herdadas da instância de banco de dados.

Criar um cache do ElastiCache com as configurações de uma instância de banco de dados

1. Para criar uma instância de banco de dados, siga as instruções em Criar uma instância de banco de dados do Amazon RDS.
2. Depois de criar uma instância de banco de dados do RDS, o console exibe a janela Complementos sugeridos. Selecione Criar um cluster do ElastiCache do RDS usando suas configurações de banco de dados.

   Para um banco de dados existente, na página Bancos de dados, selecione a instância de banco de dados. No menu suspenso Ações, selecione Criar cluster do ElastiCache para criar um cache do ElastiCache no RDS que tenha as mesmas configurações da instância de banco de dados do RDS existente.

   Na seção de Configuração do ElastiCache, o Identificador de banco de dados de origem exibe de qual instância de banco de dados o cache do ElastiCache herda as configurações.
3. Escolha se você deseja ou não criar um cluster do Redis OSS ou do Memcached. Para ter mais informações, consulte Comparing Memcached and Redis OSS.



4. Depois disso, escolha entre criar um Cache sem servidor ou Crie seu próprio cache. Consulte mais informações em Choosing between deployment options.

   Se você escolher Cache sem servidor:

   a. Em Configurações de cache, insira valores para Nome e Descrição.

   b. Em Visualizar configurações padrão, deixe as configurações padrão para estabelecer a conexão entre o cache e a instância de banco de dados.

   c. Também é possível editar as configurações padrão escolhendo Personalizar as configurações padrão. Selecione as Configurações de conectividade do ElastiCache, as Configurações de segurança do ElastiCache e os Limites de uso máximos.

5. Se você escolher Crie seu próprio cache:

   a. Se você selecionou Cluster do Redis OSS, escolha se deseja manter o modo de cluster Habilitado ou Desabilitado. Para ter mais informações, consulte Replication: Redis OSS (Cluster Mode Disabled) vs. Redis OSS (Cluster Mode Enabled).

   b. Insira valores para Nome, Descrição e Versão do mecanismo.

      Para Versão de mecanismo, o valor padrão recomendado é a versão mais recente do mecanismo. Você também pode escolher uma Versão do mecanismo para o cache do ElastiCache mais adequada para as seus requisitos.

   c. Escolha o tipo de nó na opção Tipo de nó. Para ter mais informações, consulte Gerenciar nós.

      Se você optar por criar um cluster do Redis OSS com o Modo de cluster definido como Habilitado, insira o número de fragmentos (partições/grupos de nós) na opção Número de fragmentos.

      Insira o número de réplicas de cada fragmento em Número de réplicas.

      > **Note**
      >
      > O tipo de nó selecionado, o número de fragmentos e o número de réplicas afetam o desempenho do cache e os custos de recursos. Essas configurações devem corresponder às necessidades de seu banco de dados. Para ter informações de preços, consulte Definição de preço do Amazon ElastiCache.

d. Selecione as Configurações de conectividade do ElastiCache e as Configurações de segurança do ElastiCache. É possível manter as configurações padrão ou personalizar essas configurações de acordo com seus requisitos.

6. Verifique as configurações padrão e herdadas do cache do ElastiCache. Algumas configurações não podem ser alteradas após a criação.

> **Note**
>
> O RDS pode ajustar a janela de backup do cache do ElastiCache para atender ao requisito mínimo de janela de 60 minutos. A janela de backup de seu banco de dados de origem permanece a mesma.

7. Quando estiver tudo pronto, selecione Criar cache do ElastiCache.

O console exibe um banner de confirmação para a criação do cache do ElastiCache. Siga o link no banner para o console do ElastiCache para ver os detalhes do cache. O console do ElastiCache exibe o cache do ElastiCache recém-criado.

Migração automática de bancos de dados do EC2 para o Amazon RDS usando o AWS Database Migration Service

Você pode usar o console do RDS para migrar um banco de dados do EC2 para o RDS. O RDS usa o AWS Database Migration Service (AWS DMS) para migrar seu banco de dados do EC2 de origem. O AWS DMS permite que você migre bancos de dados relacionais para a Nuvem AWS. Para obter mais informações sobre o AWS Database Migration Service, consulte O que é AWS Database Migration Service? no Guia do Usuário doAWS Database Migration Service.

Para começar a migração, você deve criar um ou uma instância de banco de dados equivalente do RDS para o qual migrar os dados. Depois de criar o banco de dados de destino, você pode importar seu banco de dados do EC2 para ele. Para bancos de dados de origem menores que 1 TiB, essa ação de migração reduz o tempo e os recursos necessários para migrar seus dados para o RDS.

Visão geral

O console do RDS permite migrar bancos de dados do EC2 para bancos de dados equivalentes do RDS. Você deve criar um banco de dados do RDS para habilitar a migração pelo console.

Você pode migrar bancos de dados do EC2 para os seguintes mecanismos de banco de dados:

- MySQL
- MariaDB
- PostgreSQL

O processo de migração inclui as seguintes etapas:

- Crie um banco de dados equivalente no RDS. Para que os bancos de dados sejam equivalentes, eles devem ter o mesmo mecanismo de banco de dados e versões de mecanismo compatíveis. Eles também devem estar na mesma VPC. Para obter instruções sobre como criar um banco de dados, consulte Criar uma instância de banco de dados do Amazon RDS.
- Escolha o tipo de replicação para seu banco de dados:
  - Migração de carga máxima: o RDS copia o banco de dados de origem completo para o banco de dados de destino, criando novas tabelas no destino quando necessário.

> **Note**
>
> Essa opção causa uma interrupção no banco de dados do RDS.

- Migração de carga máxima e captura de dados de alteração (CDC): semelhante à migração de carga máxima, com essa opção, o RDS copia todo o banco de dados de origem para o banco de dados de destino. No entanto, após a migração de carga máxima, o RDS aplica todas as alterações capturadas na origem ao banco de dados de destino. A captura de dados de alteração coleta alterações nos logs de banco de dados utilizando a API nativa do mecanismo do banco de dados.

> **Note**
>
> Essa opção causa uma interrupção no banco de dados do RDS.

- Captura de dados de alteração (CDC): use essa opção para manter seu banco de dados de destino disponível durante a migração. O RDS migra as alterações em andamento no banco de dados de origem para o banco de dados de destino.

- O RDS cria os recursos de rede necessários para facilitar a migração. Depois que o RDS cria os recursos necessários, ele notifica você sobre os recursos criados e permite que inicie a transferência de dados.

  O tempo necessário para concluir a migração depende do tipo de replicação e do tamanho do banco de dados de origem.

Pré-requisitos

MySQL e MariaDB

Antes de começar a trabalhar com um banco de dados MySQL ou MariaDB como origem, confirme se você tem os pré-requisitos a seguir. Esses pré-requisitos se aplicam às gerenciadas pela AWS.

Você deve ter uma conta do AWS DMS com o perfil Replication Admin. O perfil precisa dos seguintes privilégios:

- REPLICATION CLIENT: este privilégio é necessário apenas para tarefas de CDC. Em outras palavras, as tarefas somente com carga completa não requerem este privilégio.

- REPLICATION SLAVE: este privilégio é necessário apenas para tarefas de CDC. Em outras palavras, as tarefas somente com carga completa não requerem este privilégio.

O usuário do AWS DMS deve ter privilégios SELECT para as tabelas de origem designadas para replicação.

Conceda os seguintes privilégios se você usar avaliações de pré-migração específicas do MySQL.

```
grant select on mysql.user to <dms_user>;
grant select on mysql.db to <dms_user>;
grant select on mysql.tables_priv to <dms_user>;
grant select on mysql.role_edges to <dms_user>  #only for MySQL version 8.0.11 and
 higher
```

PostgreSQL

Antes de migrar dados de um banco de dados de origem PostgreSQL gerenciado pela AWS, faça o seguinte:

- É recomendável utilizar uma conta de usuário da AWS com as permissões mínimas necessárias para a instância de banco de dados PostgreSQL como a conta de usuário do endpoint de origem do PostgreSQL do AWS DMS. A utilização de uma conta mestre não é recomendada. A conta deve ter o perfil `rds_superuser` e o perfil `rds_replication`. O perfil `rds_replication` concede permissões para gerenciar slots lógicos e transmitir dados utilizando slots lógicos.

> **Note**
>
> Algumas transações do AWS DMS ficam ociosas por algum tempo antes que o mecanismo do DMS as utilize novamente. Com a utilização do parâmetro `idle_in_transaction_session_timeout` no PostgreSQL versões 9.6 e superior é possível fazer com que as transações ociosas atinjam o tempo limite e falhem.

Limitações

As seguintes limitações se aplicam ao processo de migração automática:

- O status do banco de dados de destino deve ser Disponível para iniciar a migração do banco de dados de origem.
- Ao migrar de um banco de dados de origem MySQL, sua conta do RDS deve ter a função de Administrador de Replicação. Você também deve ter os privilégios adequados aplicados para essa função.
- A instância do EC2 e o banco de dados de destino devem estar na mesma VPC.
- Você não pode migrar o banco de dados do EC2 para os seguintes bancos de dados de destino ao usar a ação Migrar dados do banco de dados do EC2:
  - Banco de dados que é membro de um cluster
  - Bancos de dados Oracle, SQL Server e Db2
  - Bancos de dados com versão do MySQL anterior a 5.7
  - Bancos de dados com versão do PostgreSQL anterior a 10.4
  - Bancos de dados com versão do MariaDB anterior a 10.2

Criação de recursos do IAM para migrações homogêneas

O RDS usa AWS DMS para migrar seus dados. Para acessar os bancos de dados e migrar os dados, o AWS DMS cria um ambiente com tecnologia sem servidor para migrações de dados homogêneas. Nesse ambiente, o AWS DMS exige acesso ao emparelhamento de VPC, a tabelas de rotas, a grupos de segurança e a outros recursos da AWS. Além disso, o AWS DMS armazena os logs, as métricas e o progresso de cada migração de dados no Amazon CloudWatch. Para criar um projeto de migração de dados, o AWS DMS precisa acessar esses serviços.

Além disso, o AWS DMS requer acesso aos segredos que representam um conjunto de credenciais de usuário para autenticar a conexão de banco de dados para as conexões de origem e de destino.

> **Note**
>
> Ao usar a ação Migrar dados da instância do EC2, você pode usar o console do RDS para gerar esses recursos do IAM. Ignore esta etapa se você estiver usando os recursos do IAM gerados pelo console.

Para esse processo, é necessário ter estes recursos do IAM:

Tópicos

- Criação de uma política do IAM para migrações de dados homogêneas
- Criação de um perfil do IAM para migrações de dados homogêneas
- Criação de uma política de acesso secreto e um perfil
- Criação de um perfil do IAM para AWS DMS para gerenciar a Amazon VPC

Criação de uma política do IAM para migrações de dados homogêneas

Nesta etapa, você cria uma política do IAM que fornece acesso aos recursos do Amazon EC2 e do CloudWatch ao AWS DMS. Crie um perfil do IAM e anexe uma política.

Como criar uma política do IAM para migrações de dados

1. Faça login no AWS Management Console e abra o console do IAM, em https://console.aws.amazon.com/iam/.
2. No painel de navegação, selecione Políticas.
3. Selecione Criar política.
4. Na página Criar política, escolha a guia JSON.
5. Cole a política o JSON a seguir no editor.

   JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeRouteTables",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeVpcPeeringConnections",
                "ec2:DescribeVpcs",
                "ec2:DescribePrefixLists",
                "logs:DescribeLogGroups"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
```

```
            "Action": [
                "servicequotas:GetServiceQuota"
            ],
            "Resource": "arn:aws:servicequotas:*:*:vpc/L-0EA8095F"
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogGroup",
                "logs:DescribeLogStreams"
            ],
            "Resource": "arn:aws:logs:*:*:log-group:dms-data-migration-*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogStream",
                "logs:PutLogEvents"
            ],
            "Resource": "arn:aws:logs:*:*:log-group:dms-data-migration-*:log-
stream:dms-data-migration-*"
        },
        {
            "Effect": "Allow",
            "Action": "cloudwatch:PutMetricData",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateRoute",
                "ec2:DeleteRoute"
            ],
            "Resource": "arn:aws:ec2:*:*:route-table/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateTags"
            ],
            "Resource": [
                "arn:aws:ec2:*:*:security-group/*",
                "arn:aws:ec2:*:*:security-group-rule/*",
                "arn:aws:ec2:*:*:route-table/*",
```

```
                "arn:aws:ec2:*:*:vpc-peering-connection/*",
                "arn:aws:ec2:*:*:vpc/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:AuthorizeSecurityGroupEgress",
                "ec2:AuthorizeSecurityGroupIngress"
            ],
            "Resource": "arn:aws:ec2:*:*:security-group-rule/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:AuthorizeSecurityGroupEgress",
                "ec2:AuthorizeSecurityGroupIngress",
                "ec2:RevokeSecurityGroupEgress",
                "ec2:RevokeSecurityGroupIngress"
            ],
            "Resource": "arn:aws:ec2:*:*:security-group/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:AcceptVpcPeeringConnection",
                "ec2:ModifyVpcPeeringConnectionOptions"
            ],
            "Resource": "arn:aws:ec2:*:*:vpc-peering-connection/*"
        },
        {
            "Effect": "Allow",
            "Action": "ec2:AcceptVpcPeeringConnection",
            "Resource": "arn:aws:ec2:*:*:vpc/*"
        }
    ]
}
```

6. Selecione Avançar: tags e Avançar: revisar.
7. Insira **`HomogeneousDataMigrationsPolicy`** para Nome* e escolha Criar política.

Criação de um perfil do IAM para migrações de dados homogêneas

Nesta etapa, você cria um perfil do IAM que fornece acesso ao AWS Secrets Manager, ao Amazon EC2 e ao CloudWatch.

Como criar um perfil do IAM para migrações de dados

1. Faça login no AWS Management Console e abra o console do IAM, em https://console.aws.amazon.com/iam/.
2. No painel de navegação, selecione Perfis.
3. Escolha Criar Perfil.
4. Na página Selecionar entidade confiável, em Tipo de entidade confiável, escolha Serviço da AWS. Em Casos de uso para outros serviços da AWS, escolha DMS.
5. Marque a caixa de seleção DMS e escolha Próximo.
6. Na página Adicionar permissões, escolha HomogeneousDataMigrationsPolicy criada anteriormente. Escolha Próximo.
7. Na página Nomear, revisar e criar, insira **`HomogeneousDataMigrationsRole`** em Nome do perfil e escolha Criar função.
8. Na página Perfis, insira **`HomogeneousDataMigrationsRole`** em Nome do perfil. Escolha HomogeneousDataMigrationsRole.
9. Na página HomogeneousDataMigrationsRole, escolha a guia Relações de confiança. Selecione Edit trust policy (Editar política de confiança).
10. Na página Editar política de confiança, cole o seguinte JSON no editor, substituindo o texto existente.

    JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "dms-data-migrations.amazonaws.com",
                    "dms.your_region.amazonaws.com"
```

```
                ]
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

No exemplo anterior, substitua *your_region* pelo nome da sua Região da AWS.

A política anterior baseada em recursos fornece às entidades principais do AWS DMS as permissões para executar tarefas de acordo com a política HomogeneousDataMigrationsPolicy gerenciada pelo cliente.

11. Escolha Atualizar política.

Criação de uma política de acesso secreto e um perfil

Siga os procedimentos abaixo para criar sua política de acesso secreto e seu perfil que permitem que o DMS acesse as credenciais do usuário para os bancos de dados de origem e destino.

Como criar a política de acesso secreto e o perfil, que permite que o Amazon RDS acesse o AWS Secrets Manager para acessar o segredo apropriado

1. Faça login no AWS Management Console e abra o console do AWS Identity and Access Management (IAM) em https://console.aws.amazon.com/iam/.
2. Escolha Políticas e Criar política.
3. Escolha JSON e insira a política a seguir para ativar o acesso e a descriptografia do segredo.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "secretsmanager:GetSecretValue",
            "Resource": secret_arn,
        },
        {
             "Effect": "Allow",
             "Action": [
                        "kms:Decrypt",
```

```
                    "kms:DescribeKey"
                ],
            "Resource": kms_key_arn,
        }
    ]
}
```

Aqui, *secret_arn* é o ARN do segredo, que é possível obter de `SecretsManagerSecretId` conforme apropriado, e *kms_key_arn* é o ARN da chave do AWS KMS que você está utilizando para criptografar o segredo, como no exemplo a seguir.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "secretsmanager:GetSecretValue",
            "Resource": "arn:aws:secretsmanager:us-
east-2:123456789012:secret:MySQLTestSecret-qeHamH"
        },
        {
             "Effect": "Allow",
             "Action": [
                        "kms:Decrypt",
                        "kms:DescribeKey"
                    ],
             "Resource": "arn:aws:kms:us-
east-2:123456789012:key/761138dc-0542-4e58-947f-4a3a8458d0fd"
        }
    ]
}
```

> **Note**
>
> Se você utilizar a chave de criptografia padrão criada pelo AWS Secrets Manager, não precisará especificar as permissões do AWS KMS para *kms_key_arn*.

Se quiser que a política forneça acesso aos dois segredos, basta especificar um objeto de recurso JSON adicional para o outro *secret_arn*.

4. Revise e crie a política com um nome amigável e uma descrição opcional.
5. Escolha Funções e Criar função.
6. Escolha Serviço da AWS como o tipo de entidade confiável.
7. Escolha DMS na lista de serviços como o serviço confiável e escolha Próximo: Permissões.
8. Pesquise e anexe a política que criada na etapa 4 e continue para adicionar quaisquer tags e revisar o perfil. Nesse ponto, edite as relações de confiança do perfil para utilizar a entidade principal do serviço Amazon RDS regional como a entidade confiável. Essa entidade principal tem o seguinte formato.

```
dms.region-name.amazonaws.com
```

Aqui, *region-name* é nome da sua região, como `us-east-1`. Desse modo, segue uma entidade principal do serviço Amazon RDS regional para essa região.

```
dms.us-east-1.amazonaws.com
dms-data-migrations.amazonaws.com
```

Criação de um perfil do IAM para AWS DMS para gerenciar a Amazon VPC

Você deve criar um perfil do IAM para AWS DMS para gerenciar as configurações de VPC para seus recursos. Esse perfil deve estar disponível para uma migração bem-sucedida.

Criação de **`dms-vpc-role`** para migração de banco de dados

<result>

Isso cria o perfil para o DMS para gerenciar as configurações de VPC para a migração.
</result>

1. Faça login no AWS Management Console e abra o console do IAM, em https://console.aws.amazon.com/iam/.
2. No painel de navegação do console, escolha Roles (Perfis) e, em seguida, clique em Create role (Criar perfil).
3. Selecione a opção AWS service (Serviço da AWS) para Selecionar entidade confiável.

Em Caso de uso, selecione DMS.

4. Na etapa Adicionar permissões, selecione `AmazonDMSVPCManagementRole` e escolha Próximo.
5. Na página Nomear, revisar e criar, defina o Nome do perfil como `dms-vpc-role` e escolha Criar perfil.

Configurar a migração de dados para o banco de dados do EC2

Para começar a migrar dados do banco de dados de origem do EC2, é necessário criar um banco de dados do RDS equivalente. Para obter instruções sobre como criar um banco de dados, consulte Criar uma instância de banco de dados do Amazon RDS.

Depois de criar o banco de dados de destino, use as etapas a seguir para configurar a migração de dados:

Configurar o projeto de migração de dados

1. Selecione o banco de dados de destino na página Bancos de dados no console do RDS.
2. Escolha o menu suspenso Ações e selecione a opção Migrar dados do banco de dados do EC2. Para ver os bancos de dados de destino compatíveis, consulte Limitações.
3. Na seção Selecionar banco de dados de origem do EC2:

    1. Verifique o Tipo de mecanismo e certifique-se de que seja o mesmo do banco de dados de origem.

        Além disso, verifique se as versões do mecanismo são compatíveis.
    2. Em Instância do EC2, escolha a instância do EC2 onde o banco de dados de origem reside.
    3. Em Porta, insira a porta na qual o banco de dados de origem permite tráfego.
    4. Em Segredo, escolha Criar e usar um novo segredo se você não tiver um segredo existente. Insira o Nome de usuário e a Senha do banco de dados de origem. Escolha também a chave do KMS com a qual criptografar seu segredo.

        Para você já tiver um segredo, selecione Usar um segredo existente e escolha um segredo no menu suspenso.

5. Em Perfil do IAM para segredo, se você tiver um perfil do IAM existente, selecione Usar um perfil do IAM existente e escolha um perfil do IAM no menu suspenso que possa acessar o ID do segredo da etapa anterior.

   Se você não tem um perfil do IAM, escolha Criar e usar novo perfil do IAM. Insira o novo nome para o perfil no campo Nome do perfil do IAM. Você pode ver as permissões associadas a essa função no link abaixo.

4. Na seção Visualizar banco de dados de destino do RDS:

   1. Confirme as configurações do banco de dados de destino na parte superior da seção.
   2. Em Segredo, escolha Criar e usar um novo segredo se você não tiver um segredo existente que contenha as credenciais do banco de dados de destino.

      Para você já tiver um segredo, selecione o segredo no menu suspenso.
   3. Em Perfil do IAM para segredo, selecione um perfil do IAM que possa acessar o segredo da etapa anterior. Você também pode criar um perfil do IAM se já não tiver um.

      Se o menu suspenso não preencher os perfis do IAM, especifique o ARN do perfil do IAM no formato `arn:aws:iam:`*`account_id`*`:role/`*`roleName`*.

5. Na seção Configurar migração de dados:

   1. Selecione o tipo de migração de dados escolhendo entre Carga máxima, Captura de dados de alteração e carga máxima (CDC) ou Captura de dados de alteração (CDC). Para obter mais informações sobre essas opções, consulte Visão geral.

      Não é possível modificar o tipo de migração após o início da migração.
   2. Em Perfil do IAM para migração de dados, se você já tiver um perfil do IAM, selecione Usar um perfil do IAM existente e escolha um perfil do IAM no menu suspenso que conceda ao DMS as permissões para criar os recursos necessários para a migração. Se você não tem um perfil do IAM, escolha Criar e usar novo perfil do IAM.

6. Confirme se a guia Exibir configurações de migração mostra as configurações necessárias para que a migração de dados seja configurada com sucesso.
7. Selecione Migrar para concluir a configuração da migração.

Depois de concluir essas etapas, você pode ver os recursos que estão sendo configurados para a migração de dados escolhendo Visualizar detalhes no banner de progresso no console. Depois que

os recursos necessários estiverem configurados, a migração será iniciada automaticamente. Se você criar

Para migrar vários bancos de dados para o banco de dados de destino, inicie esse processo novamente com detalhes sobre o novo banco de dados do EC2.

Gerenciar migrações de dados

Depois de usar a ação Migrar dados do banco de dados do EC2 do console do RDS, o RDS inicia a migração automaticamente.

Se você tiver usado o console do AWS DMS para criar os recursos de migração, poderá iniciar o processo de migração.

Início da migração de dados

Siga estas etapas para iniciar a migração de dados:

Início de uma migração de dados

1. Escolha o banco de dados de destino na página Bancos de dados no console do RDS.
2. Na página de detalhes do banco de dados, escolha a guia Migrações de dados.
3. Na guia Migrações de dados, a opção Migrações de dados associadas lista as migrações de dados disponíveis.

   As migrações configuradas usando o console do RDS são iniciadas automaticamente quando os recursos necessários são configurados.

   As migrações configuradas usando o console do DMS são definidas como Prontas.

   Para começar essas migrações, selecione o menu suspenso Ações e selecione Iniciar.
4. Isso inicia a migração de dados para seu banco de dados do EC2.

Interrupção da migração de dados

Para migrações de dados cujo tipo de replicação é de carga completa, interromper a migração faz com que o processo pare e não possa ser retomado. Depois de interrompida, a migração deve ser reiniciada.

Para migrações com o tipo de replicação definido para captura de dados de alteração (CDC) ou carga total e CDC, você pode interromper o processo de replicação contínua e retomar o processo posteriormente.

Interrupção de uma migração de dados

1. Escolha o banco de dados de destino na página Bancos de dados no console do RDS.
2. Na página de detalhes do banco de dados, escolha a guia Migrações de dados.
3. Na guia Migrações de dados, a opção Migrações de dados associadas lista as migrações de dados em andamento.

   Para interromper uma migração, selecione uma migração de dados e selecione Parar no menu suspenso Ações.
4. Isso para a migração de dados para seu banco de dados do EC2.

Retomada da migração de dados

Para migrações de dados cujo tipo de replicação é carga total e captura de dados de alteração (CDC) ou migração de captura de dados de alteração (CDC), você pode retomar o processo CDC a partir do último ponto de interrupção.

Retomada de uma migração de dados

1. Escolha o banco de dados de destino na página Bancos de dados no console do RDS.
2. Na página de detalhes do banco de dados, escolha a guia Migrações de dados.
3. Na guia Migrações de dados, a opção Migrações de dados associadas lista as migrações de dados interrompidas.

   Para retomar uma migração, selecione uma migração de dados e selecione Retomar processamento no menu suspenso Ações.
4. Isso retoma a migração de dados para seu banco de dados do EC2.

Exclusão da migração de dados

Para excluir uma migração de dados associada, use as instruções a seguir

Exclusão de uma migração de dados

1. Escolha o banco de dados de destino na página Bancos de dados no console do RDS.
2. Na página de detalhes do banco de dados, escolha a guia Migrações de dados.
3. Para excluir uma migração, selecione uma migração de dados e selecione Excluir no menu suspenso Ações.
4. Isso exclui a migração de dados.

A exclusão de uma migração de dados que estava em andamento não afeta nenhum dado que já tenha sido carregado no banco de dados de destino.

Reinicialização da migração de dados

Para reiniciar uma migração de dados associada a partir de um ponto inicial do CDC, use as instruções a seguir

Reinicialização de uma migração de dados

1. Escolha o banco de dados de destino na página Bancos de dados no console do RDS.
2. Na página de detalhes do banco de dados, escolha a guia Migrações de dados.
3. Para reiniciar uma migração, selecione uma migração de dados e escolha Reiniciar no menu suspenso Ações.
4. Isso reinicia a migração de dados a partir de um ponto inicial do CDC.

A reinicialização de uma migração de dados que estava em andamento não afeta nenhum dado que já tenha sido carregado no banco de dados de destino.

Monitoramento das migrações de dados

Depois de iniciar migrações de dados, é possível monitorar seu status e progresso. As migrações de dados de grandes conjuntos de dados demoram horas para serem concluídas. Para manter a confiabilidade, a disponibilidade e o alto desempenho da migração de dados, monitore o progresso regularmente.

Como verificar o status e o progresso da migração de dados

1. Escolha o banco de dados de destino na página Bancos de dados no console do RDS.

2. Na página de detalhes do banco de dados, escolha a guia Migrações de dados.
3. A seção Migrações de dados associadas lista suas migrações de dados. Verifique a coluna Status.
4. Para migrações de dados em execução, a coluna Processo da migração exibe a porcentagem de dados migrados.
5. Para monitorar o processo no CloudWatch, use o link na coluna CloudWatch.

Status das migrações

Para cada migração de dados executada, o console do RDS exibe o Status. A lista a seguir inclui os status:

- `Ready`: a migração de dados está pronta para ser iniciada.
- `Starting`: o RDS está criando o ambiente com tecnologia sem servidor para a migração de dados.
- `Load running`: o RDS está executando a migração de carga máxima.
- `Load complete, replication ongoing`: o RDS concluiu a carga máxima e agora replica as alterações em andamento. Esse status só se aplica a migrações de carga máxima e do tipo CDC.
- `Replication ongoing`: o RDS está replicando as alterações em andamento. Esse status só se aplica a migrações do tipo CDC.
- `Stopping`: o RDS está interrompendo as migrações de dados. Esse status se aplica quando você opta por interromper a migração de dados no menu Ações.
- `Stopped`: o RDS interrompeu as migrações de dados.
- `Failed`: falha na migração de dados. Para obter mais informações, consulte os arquivos de log.
- `Restarting`: a migração de dados reiniciou uma replicação de dados em andamento a partir de um ponto inicial do CDC.

Tutorial: criar uma instância de banco de dados do MySQL com um grupo de parâmetros personalizado e um novo grupo de opções

Neste tutorial, é necessário criar uma instância de banco de dados do MySQL com um grupo de parâmetros personalizado e um novo grupo de opções. Para ter mais informações sobre grupos de parâmetros personalizados e grupos de opções, consulte Grupos de parâmetros para Amazon RDS e Trabalhar com grupos de opções.

> **Important**
>
> Não há cobrança para criar uma conta da AWS. No entanto, ao concluir este tutorial, é possível gerar custos para os recursos da AWS que você usa. Se esses recursos não forem mais necessários após a conclusão do tutorial, você poderá excluí-los.

Para criar uma instância de banco de dados com configurações e definições personalizadas, é possível usar grupos de parâmetros personalizados e novos grupos de opções. Os grupos de parâmetros personalizados e os novos grupos de opções são particularmente úteis se você trabalha com vários bancos de dados e quer definir as configurações de modo uniforme para eles.

Ao concluir essas etapas, você aprende a:

- Usar o Amazon RDS para criar uma instância de banco de dados do MySQL com um grupo de parâmetros personalizado e um novo grupo de opções.
- Como usar parâmetros específicos e opções específicas para instâncias de banco de dados do MySQL.

Para concluir este tutorial, realize as seguintes tarefas:

1. Crie um grupo de parâmetros personalizado com os parâmetros `default_password_lifetime` e `disconnect_on_expired_password` do MySQL.
2. Crie um grupo de opções com o recurso opcional `MariaDB Audit Plugin` do MySQL. Para conhecer as etapas para criar um grupo de opções, consulte Trabalhar com grupos de opções.
3. Crie uma instância de banco de dados do MySQL com o grupo de parâmetros personalizado e o grupo de opções que você criou.

Tópicos

- Pré-requisitos
- Criar um grupo de parâmetros do Amazon RDS
- Modificar os valores dos parâmetros em seu grupo de parâmetros personalizado
- Criar um grupo de opções do Amazon RDS
- Adicionar uma opção ao seu novo grupo de opções
- Criar uma instância de banco de dados do MySQL com um grupo de parâmetros personalizado e um novo grupo de opções

Pré-requisitos

Este tutorial exige que você tenha uma Conta da AWS e um usuário com acesso administrativo. Se ainda não os configurou, conclua as etapas apresentadas nas seguintes seções:

- Inscrever-se para uma Conta da AWS
- Criar um usuário com acesso administrativo

Criar um grupo de parâmetros do Amazon RDS

Neste tutorial, você aprende a criar um grupo de parâmetros personalizado com default_password_lifetime e disconnect_on_expired_password para uma instância de banco de dados do MySQL no console. O parâmetro `default_password_lifetime` controla o número de dias antes da expiração automática da senha do cliente. O parâmetro `disconnect_on_expired_password` controla se a instância de banco de dados do MySQL desconecta o cliente quando a senha expira. Para ter mais informações sobre outros parâmetros disponíveis para instâncias de banco de dados do MySQL, consulte Parâmetros do MySQL.

Para criar um grupo de parâmetros

1. Abra o console do Amazon RDS e selecione Grupos de parâmetros.
2. Em Grupos de parâmetros personalizados, selecione Criar grupo de parâmetros.
3. Defina os detalhes do grupo de parâmetros.
    1. Insira um nome para o grupo de parâmetros.
    2. Insira uma descrição do grupo de parâmetros.

3. Em Tipo de mecanismo, escolha MySQL Community.
4. Em Família de grupos de parâmetros, selecione MySQL 8.0.
4. Escolha Criar.

O novo grupo de parâmetros aparece na página Grupos de parâmetros no console do Amazon RDS. As etapas a seguir mostram como modificar valores de parâmetro para personalizar um grupo de parâmetros.

Modificar os valores dos parâmetros em seu grupo de parâmetros personalizado

Use as etapas a seguir para modificar os valores dos parâmetros no grupo de parâmetros que você criou em Criar um grupo de parâmetros do Amazon RDS.

Como modificar os valores dos parâmetros em seu grupo de parâmetros

1. Abra o console do Amazon RDS e selecione Grupos de parâmetros.
2. Em Grupos de parâmetros personalizados, escolha o nome do grupo de parâmetros que você criou.
3. Escolha Editar.
4. Na caixa de pesquisa Filtrar parâmetros, pesquise o parâmetro personalizado `default_password_lifetime`.
5. Marque a caixa de seleção ao lado do parâmetro e insira um valor para o número de dias a ser definido para esse parâmetro de ciclo de vida da senha.
6. Escolha Salvar alterações.
7. Repita as mesmas etapas para o parâmetro `disconnect_on_expired_password`. Ao escolher esse parâmetro, será solicitado que você selecione um valor de 0 ou 1 no menu suspenso. Selecione 1 para desconectar quando a senha expirar.

O grupo de parâmetros personalizado agora está disponível para ser associado ao Amazon RDS para instâncias de banco de dados do MySQL 8.0. Em seguida, crie um grupo de opções para sua instância de banco de dados.

Criar um grupo de opções do Amazon RDS

Crie um grupo de opções com o MariaDB Audit Plugin opcional. Esse plug-in registra a atividade do servidor para segurança e conformidade. Para ter mais informações sobre outras opções personalizadas disponíveis para instâncias de banco de dados do MySQL, consulte Opções para instâncias de bancos de dados MySQL.

Como criar um grupo de opções

1. Abra o console do Amazon RDS e selecione Grupos de opções.
2. Em Grupos de opções, escolha Criar grupo.
3. Defina os detalhes do grupo de opções.
   - Insira um nome para o grupo de opções.
   - Insira uma descrição do grupo de opções.
   - Em Mecanismo, selecione mysql.
   - Em Versão do mecanismo principal, selecione 8.0.
4. Escolha Criar.

O novo grupo de opções aparece na página Grupos de opções no console do Amazon RDS. As etapas a seguir mostram como adicionar opções ao grupo de opções.

Adicionar uma opção ao seu novo grupo de opções

Use as etapas a seguir para adicionar uma opção ao grupo de opções que você criou em Criar um grupo de opções do Amazon RDS.

Como adicionar uma opção ao seu grupo de opções

1. Abra o console do Amazon RDS e selecione Grupos de opções.
2. Em Grupos de opções, selecione o nome do grupo de opções que você criou.
3. Em Opções, escolha Adicionar opção.
4. Defina os detalhes do grupo de opções.
   - Em Nome da opção, selecione a opção MariaDB Audit Plugin, MARIADB_AUDIT_PLUGIN.
   - Em Configurações de opção, mantenha todas as opções padrão selecionadas.

- Em Aplicar imediatamente, escolha Sim.

5. Selecione Criar opção.

O novo grupo de opões agora deve estar disponível para todas as instâncias de banco de dados associadas. Depois, crie uma instância de banco de dados do MySQL com o grupo de parâmetros personalizado e o novo grupo de opções.

Criar uma instância de banco de dados do MySQL com um grupo de parâmetros personalizado e um novo grupo de opções

Por fim, crie uma instância de banco de dados do MySQL com o grupo de parâmetros personalizado e o grupo de opções que você criou nas etapas anteriores. As etapas a seguir mostram como criar a instância de banco de dados do MySQL com um grupo de parâmetros personalizado e um novo grupo de opções.

Como criar uma instância de banco de dados com um grupo de parâmetros personalizado e um novo grupo de opções

1. Abra o console do Amazon RDS e selecione Bancos de dados.
2. Selecione Criar banco de dados.
3. Em Escolher um método de criação de banco de dados, escolha Criação padrão.
4. Em Opções do mecanismo, escolha MySQL.
5. Em Disponibilidade e durabilidade, selecione Instância de banco de dados única. Essa etapa é necessária para comportar um grupo de parâmetros personalizado ou um novo grupo de opções.
6. Selecione Configuração adicional.
   - Em Nome do banco de dados inicial, selecione um nome para a instância de banco de dados.
   - No menu suspenso Grupo de parâmetros do banco de dados, selecione o nome do grupo de parâmetros personalizado que você criou anteriormente.
   - No menu suspenso Grupo de opções, selecione o nome do novo grupo de opções que você criou anteriormente.
7. Neste tutorial, é possível deixar as configurações padrão para qualquer outra configuração de banco de dados ou modificá-las conforme necessário.
8. Selecione Criar banco de dados.

O RDS cria uma instância de banco de dados do MySQL com um grupo de parâmetros personalizado e um novo grupo de opções. Para ver mais informações sobre esse banco de dados, consulte a página Bancos de dados do console do Amazon RDS.

Neste tutorial, você vai configurar uma instância de banco de dados do MySQL com configurações personalizadas usando um grupo de parâmetros personalizado e um novo grupo de opções. Essa instância de banco dados recém-criada do MySQL gerencia o ciclo de vida da senha do usuário usando o parâmetro `default_password_lifetime`. Essa instância de banco de dados também desconecta usuários que se conectam com uma senha vencida usando o parâmetro `disconnect_on_expired_password`. A opção `MariaDB Audit Plugin` também é usada para acompanhar a atividade do servidor. Para otimizar o banco de dados, é possível aplicar configurações adicionais no grupo de parâmetros personalizado e adicionar opções.

Assim que criar sua instância de banco de dados personalizada, você deve excluir seus recursos para evitar custos indesejados. Para excluir uma instância de banco de dados, siga as instruções em Excluir uma instância de banco de dados.

Gerenciar uma instância de banco de dados do Amazon RDS

Veja a seguir instruções para gerenciar e manter a instância de banco de dados do Amazon RDS.

Tópicos

- Parar uma instância de banco de dados do Amazon RDS temporariamente
- Iniciar uma instância de banco de dados do Amazon RDS que foi anteriormente interrompida
- Reinicializar uma instância de banco de dados
- Conectar automaticamente uma instância do EC2 e uma instância de banco de dados
- Conectar automaticamente uma função do Lambda e uma instância de banco de dados
- Modificar uma instância de banco de dados do Amazon RDS
- Manutenção de uma instância de banco de dados
- Atualizar a versão de mecanismo de uma instância de banco de dados
- Renomear uma instância de banco de dados
- Trabalhar com réplicas de leitura de instância de banco de dados
- Marcar recursos do Amazon RDS
- Nomes de recurso da Amazon (ARNs) no Amazon RDS
- Trabalhar com armazenamento para instâncias de banco de dados do Amazon RDS
- Excluir uma instância de banco de dados
- Tutorial: Gerenciar um ambiente de instância de banco de dados do MySQL do desenvolvimento à produção

Parar uma instância de banco de dados do Amazon RDS temporariamente

É possível interromper a instância de banco de dados de modo intermitente para testes temporários ou para uma atividade de desenvolvimento diária, por no máximo sete dias consecutivos. O caso de uso mais comum é a otimização de custos.

O tempo para interromper o cluster de banco de dados varia de acordo com determinados fatores, como classes de instância de banco de dados, estado da rede, tipo de mecanismo de banco de dados e estado do banco de dados. O processo pode demorar alguns minutos. O serviço deve executar as seguintes ações:

- Encerrar os processos do mecanismo de banco de dados.
- Encerrar os processos da plataforma do RDS.
- Separar os volumes de armazenamento do EBS associados à sua instância de banco de dados.
- Encerrar a instância subjacente do Amazon EC2.

> **Warning**
>
> Iniciar uma instância de banco de dados exige a recuperação da instância e pode levar de minutos a horas. Portanto, se a disponibilidade da instância for uma preocupação, tenha cuidado ao interromper temporariamente uma instância de produção. Para ter mais informações, consulte Iniciar uma instância de banco de dados do Amazon RDS que foi anteriormente interrompida.

Para interromper e iniciar a instância de banco de dados na mesma operação, reinicie a instância de banco de dados. Para ter mais informações, consulte Reinicializar uma instância de banco de dados .

Tópicos

- Casos de uso para interromper uma instância de banco de dados
- Mecanismos de banco de dados compatíveis, classes de instância e regiões
- Interromper uma instância de banco de dados em uma implantação multi-AZ
- Como funciona a interrupção de uma instância de banco de dados

- Limitações decorrentes da interrupção de uma instância de banco de dados
- Considerações de grupos de parâmetros e grupos de opções
- Considerações sobre endereços IP públicos
- Interromper uma instância de banco de dados temporariamente: etapas básicas

Casos de uso para interromper uma instância de banco de dados

Interromper e iniciar uma instância de banco de dados é mais rápido do que criar um snapshot de banco de dados, excluir a instância de banco de dados e, depois, restaurar o snapshot quando você quiser acessar a instância. Entre os casos de uso comuns para interromper uma instância estão os seguintes:

- Otimização de custos: para bancos de dados que não são de produção, você pode interromper a instância de banco de dados do Amazon RDS temporariamente para economizar dinheiro. Enquanto a instância estiver interrompida, você não precisará pagar por nenhuma hora da instância de banco de dados.

  > **Important**
  >
  > Durante a interrupção de sua instância de banco de dados, você será cobrado pelo armazenamento provisionado (incluindo IOPS provisionadas). Você também é cobrado pelo armazenamento de backup, incluindo snapshots manuais e backups automatizados durante a janela de retenção especificada. No entanto, você não será cobrado por horas de instância de banco de dados. Para obter mais informações, consulte Perguntas frequentes sobre o faturamento.

- Desenvolvimento diário: se mantiver uma instância de banco de dados para fins de desenvolvimento, você poderá iniciá-la quando for necessário e, depois, desligá-la quando ela não for necessária.
- Teste: talvez você precise de uma instância de banco de dados temporária para testar procedimentos de backup e recuperação, migrações, atualizações de aplicações ou atividades relacionadas. Nesses casos de uso, você poderá interromper a instância de banco de dados quando ela não for necessária.
- Treinamento: se você estiver conduzindo um treinamento no RDS, talvez seja necessário iniciar instâncias de banco de dados durante a sessão de treinamento e encerrá-las depois.

Mecanismos de banco de dados compatíveis, classes de instância e regiões

Você pode interromper e iniciar instâncias de banco de dados do Amazon RDS que estejam executando os seguintes mecanismos de banco de dados:

- Db2
- MariaDB
- Microsoft SQL Server, incluindo RDS Custom para SQL Server.
- MySQL
- Oracle
- PostgreSQL

Há suporte para o recurso de interromper e iniciar uma instância de banco de dados em todas as classes de instância de banco de dados e em todas as regiões da AWS.

Interromper uma instância de banco de dados em uma implantação multi-AZ

Você pode interromper e iniciar uma instância de banco de dados em uma implantação multi-AZ. Observe as seguintes limitações:

- Você só poderá criar uma implantação multi-AZ se o mecanismo de banco de dados for compatível com ela. Consulte mais informações sobre o suporte do mecanismo em Regiões e mecanismos de banco de dados compatíveis com clusters de banco de dados multi-AZ no Amazon RDS.
- O RDS para SQL Server não é compatível com a interrupção de uma instância de banco de dados em uma implantação multi-AZ. Para ter mais informações, consulte Limitações, observações e recomendações de implantação multi-AZ do Microsoft SQL Server.
- Pode levar muito tempo para interromper uma instância de banco de dados. Se você tiver pelo menos um backup após um failover anterior, poderá acelerar a operação de interrupção executando uma reinicialização com operação de failover. Para ter mais informações, consulte Reinicializar uma instância de banco de dados .

Como funciona a interrupção de uma instância de banco de dados

A operação de interrupção ocorre nos seguintes estágios:

1. A instância de banco de dados inicia o processo normal de desligamento.

   O status da instância de banco de dados muda para `stopping`.
2. A instância para de ser executada e fica nesse estado até sete dias consecutivos.

   O status da instância de banco de dados muda para `stopped`.

Características de uma instância de banco de dados interrompida

Quando em um estado interrompido, a instância de banco de dados tem as seguintes características:

- A instância de banco de dados interrompida retém o seguinte:
  - ID da instância
  - Endpoint do servidor de nomes de domínio (DNS)
  - Grupo de parâmetros
  - Grupo de segurança
  - Option group
  - Logs de transações do Amazon S3 (necessários para uma restauração para um ponto no tempo)

  Quando você reinicia uma instância de banco de dados, ela tem a mesma configuração de quando você a interrompeu.
- Todos os volumes de armazenamento permanecem anexados à instância de banco de dados e os dados são mantidos. O RDS exclui qualquer dado armazenado na RAM da instância de banco de dados.

  Durante a interrupção de sua instância de banco de dados, você será cobrado pelo armazenamento provisionado (incluindo IOPS provisionadas). Você também é cobrado pelo armazenamento de backup, incluindo snapshots manuais e backups automatizados durante a janela de retenção especificada.
- O RDS remove ações pendentes, incluindo atualizações de manutenção programadas, exceto as ações pendentes para o grupo de opções ou o grupo de parâmetros da instância de banco de dados.

> **Note**
>
> Ocasionalmente, uma instância de banco de dados do RDS for PostgreSQL não é desligada apropriadamente. Se isso acontecer, você verá que a instância passa por um processo de recuperação ao reiniciá-la mais tarde. Esse é o comportamento esperado do mecanismo de banco de dados, destinado a proteger a integridade do banco de dados. Algumas estatísticas e contadores baseados em memória não retêm o histórico e são reinicializados após a reinicialização, para capturar o workload operacional avançando.

Reinício automático de uma instância de banco de dados interrompida

Se você não iniciar a instância de banco de dados manualmente depois que ela ficar sete dias consecutivos interrompida, o RDS a iniciará automaticamente para você. Dessa forma, a instância não perde nenhuma atualização de manutenção necessária. Para saber como interromper e iniciar a instância de acordo com uma programação, consulte Como posso usar o Step Functions para interromper uma instância do Amazon RDS por mais de sete dias?.

Limitações decorrentes da interrupção de uma instância de banco de dados

Veja a seguir algumas limitações da operação de interrupção:

- Não é possível interromper uma instância de banco de dados que tem uma réplica de leitura ou que é uma réplica de leitura.
- Você não pode modificar uma instância de banco de dados parada.
- Você não pode excluir um grupo de opções que esteja associado a uma instância de banco de dados parada.
- Você não pode excluir um parameter group de banco de dados que esteja associado a uma instância de banco de dados parada.
- Em uma implantação multi-AZ, observe as seguintes limitações:
  - Não é possível interromper uma instância de banco de dados do RDS para SQL Server.
  - As zonas de disponibilidade primária e secundária podem ser alternadas após o início da instância de banco de dados.

Outras limitações se aplicam ao RDS Custom para SQL Server: Para ter mais informações, consulte Iniciar e interromper uma instância de banco de dados do RDS Custom para SQL Server.

Considerações de grupos de parâmetros e grupos de opções

Você não pode remover opções persistentes (incluindo opções permanentes) de um grupo de opções se houver instâncias de banco de dados associadas com aquele grupo de opções. Essa funcionalidade também é verdadeira para qualquer instância de banco de dados com um estado de `stoppingstopped` ou `starting`.

Você não pode alterar o grupo de opções nem o grupo de parâmetros de banco de dados associado a uma instância de banco de dados parada. No entanto, a alteração não ocorrerá até a próxima vez que você iniciar a instância de banco de dados. Se você optar por aplicar imediatamente alterações, a alteração ocorrerá quando iniciar a instância de banco de dados. Caso contrário, as alterações ocorrerão durante a próxima janela de manutenção depois que você iniciar a instância de banco de dados.

Considerações sobre endereços IP públicos

Quando você interrompe uma instância de banco de dados, ela retém o endpoint de DNS. Se você interromper uma instância de banco de dados que tenha um endereço IP público, o Amazon RDS liberará seu endereço IP público. Quando a instância de banco de dados for reiniciada, ela terá um endereço IP público diferente.

> **Note**
>
> Você deve sempre se conectar a uma instância de banco de dados usando o endpoint de DNS, não o endereço IP.

Interromper uma instância de banco de dados temporariamente: etapas básicas

Você pode interromper um banco de dados usando o AWS Management Console, a AWS CLI ou a API do RDS.

Console

Para encerrar uma instância de banco de dados

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.

2. No painel de navegação, escolha Databases (Bancos de dados) e a instância de banco de dados que você deseja interromper.
3. Em Actions (Ações), escolha Stop temporarily (Parar temporariamente).
4. Na janela Stop DB instance temporarily (Interromper temporariamente a instância de banco de dados), selecione a confirmação de que a instância de banco de dados será reiniciada automaticamente após 7 dias.
5. (Opcional) Selecione Save the DB instance in a snapshot (Salvar a instância de banco de dados em um snapshot) e insira o nome do snapshot em Snapshot name (Nome do snapshot). Escolha esta opção se você quiser criar um snapshot da instância de banco de dados antes de interrompê-la.
6. Escolha Stop temporarily (Parar temporariamente) para interromper a instância de banco de dados, ou escolha Cancel (Cancelar) para cancelar a operação.

AWS CLI

Para interromper uma instância de banco de dados usando a AWS CLI, chame o comando stop-db-instance com a seguinte opção:

- `--db-instance-identifier` – o nome da instância de banco de dados.

Example

```
aws rds stop-db-instance --db-instance-identifier mydbinstance
```

API do RDS

Para interromper uma instância de banco de dados usando a API do Amazon RDS, chame a operação StopDBInstance com o seguinte parâmetro:

- `DBInstanceIdentifier` – o nome da instância de banco de dados.

Iniciar uma instância de banco de dados do Amazon RDS que foi anteriormente interrompida

Este tópico explica como iniciar uma instância de banco de dados do Amazon RDS que foi interrompida anteriormente, descrevendo as etapas necessárias e as principais considerações para retomar o uso do banco de dados.

Você pode interromper sua instância de banco de dados do Amazon RDS temporariamente para economizar dinheiro. Depois de interromper sua instância de banco de dados, você poderá reiniciá-la para usá-la novamente. Para ter mais informações sobre como interromper uma instância de banco de dados, consulte Parar uma instância de banco de dados do Amazon RDS temporariamente.

Quando você inicia uma instância de banco de dados interrompida, ela retém informações como as seguintes:

- ID da instância
- Endpoint do servidor de nomes de domínio (DNS)
- DB parameter group (grupo de parâmetros de banco de dados)
- Grupo de segurança de VPC
- Grupo de opções do banco de dados

O Amazon RDS fatura em incrementos de um segundo para instâncias de banco de dados e armazenamento anexado. Há uma cobrança mínima de dez minutos quando uma instância é iniciada.

Para iniciar a instância, o serviço Amazon RDS executa ações como as seguintes:

- Provisionar a instância do Amazon EC2 subjacente
- Iniciar os processos do RDS
- Iniciar os processos do mecanismo do banco de dados
- Anexar os volumes de armazenamento do EBS
- Habilitar o Insights de Performance, caso tenha sido habilitado anteriormente
- Recuperar a instância de banco de dados (a recuperação ocorre mesmo após um desligamento normal)

O tempo para reiniciar o cluster de banco de dados varia de acordo com determinados fatores, como tamanho do banco de dados, classes de instância de banco de dados, estado da rede, tipo de mecanismo de banco de dados e estado do banco de dados quando o cluster foi encerrado. O processo de inicialização pode demorar de minutos a horas. Ao criar o plano de disponibilidade, recomendamos pensar na variabilidade do tempo de inicialização.

Console

Para iniciar uma instância de banco de dados

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Databases (Bancos de dados) e a instância de banco de dados que você deseja iniciar.
3. Em Actions (Ações), escolha Start (Iniciar).

AWS CLI

Para iniciar uma instância de banco de dados usando a AWS CLI, chame o comando start-db-instance com a seguinte opção:

- `--db-instance-identifier` – o nome da instância de banco de dados.

Example

```
aws rds start-db-instance --db-instance-identifier mydbinstance
```

API do RDS

Para iniciar uma instância de banco de dados usando a API do Amazon RDS, chame a operação StartDBInstance com o seguinte parâmetro:

- `DBInstanceIdentifier` – o nome da instância de banco de dados.

Reinicializar uma instância de banco de dados

Você pode interromper e iniciar o serviço de banco de dados na instância de banco de dados do RDS em uma única operação, chamada reinicialização. A reinicialização pode ser necessária para aplicar alterações na configuração, resolver problemas menores ou problemas de rede sem precisar reiniciar ou migrar completamente o banco de dados.

> **Note**
>
> Este tópico se aplica somente à reinicialização de uma instância de banco de dados single-AZ ou multi-AZ. Consulte instruções sobre a reinicialização de clusters de banco de dados multi-AZ em the section called “Reinicializar um cluster de banco de dados multi-AZ”.

Tópicos

- Casos de uso para reinicializar uma instância de banco de dados
- Como a reinicialização de uma instância de banco de dados funciona
- Como a reinicialização de uma instância de banco de dados em uma implantação multi-AZ funciona
- Considerações sobre quando reinicializar uma instância de banco de dados
- Pré-requisitos para reinicializar uma instância de banco de dados
- Reinicializar uma instância de banco de dados: etapas básicas

Casos de uso para reinicializar uma instância de banco de dados

Normalmente, você reinicializa a instância de banco de dados por motivos de manutenção para que as alterações entrem em vigor. Estes casos de uso são comuns:

- Associar um novo grupo de parâmetros de banco de dados: ao associar um novo grupo de parâmetros de banco de dados a uma instância de banco de dados, o RDS aplica os parâmetros estáticos e dinâmicos modificados somente após a reinicialização da instância de banco de dados. No entanto, se você modificar parâmetros dinâmicos no grupo de parâmetros de banco de dados depois de associá-lo à instância de banco de dados, essas alterações serão aplicadas imediatamente sem uma reinicialização. Para ter mais informações, consulte Grupos de parâmetros para Amazon RDS.

- Aplicar uma alteração em um parâmetro estático em um grupo de parâmetros de banco de dados existente: quando você altera um parâmetro estático e salva o grupo de parâmetros de banco de dados, o status das instâncias de banco de dados associadas a esse grupo de parâmetros no console muda para reinicialização pendente. A alteração do parâmetro entrará em vigor somente depois que as instâncias de banco de dados associadas forem reinicializadas. Quando você altera um parâmetro dinâmico em um grupo de parâmetros existente, a alteração entra em vigor imediatamente por padrão, sem exigir uma reinicialização.

  > **Note**
  >
  > O status Reinicialização pendente não resultará em uma reinicialização automática durante a próxima janela de manutenção. Para aplicar as alterações de parâmetro mais recentes à instância de banco de dados, reinicialize-a manualmente. Para ter mais informações sobre parameter groups, consulte Grupos de parâmetros para Amazon RDS.

- Solução de problemas: você pode encontrar problemas de desempenho ou outros problemas operacionais que exijam uma reinicialização. Por exemplo, talvez a instância de banco de dados não esteja respondendo.

Como a reinicialização de uma instância de banco de dados funciona

Quando o Amazon RDS reinicia a instância de banco de dados, ele executa as seguintes tarefas sequenciais:

1. Interrompe o serviço de banco de dados na instância de banco de dados
2. Inicia o serviço de banco de dados na instância de banco de dados

O processo de reinicialização causa uma breve interrupção. Durante essa interrupção, o status da instância de banco de dados é reinicializando. Uma interrupção ocorre tanto para uma implantação single-AZ quanto para uma implantação de instância de banco de dados multi-AZ, mesmo quando você faz a reinicialização com um failover.

Como a reinicialização de uma instância de banco de dados em uma implantação multi-AZ funciona

Se a instância de banco de dados do Amazon RDS estiver em uma implantação multi-AZ, você poderá reinicializar com um failover. Essa operação é útil para simular uma falha de uma instância de banco de dados ou restaurar operações para a zona de disponibilidade original após um failover.

Durante a reinicialização com failover, o Amazon RDS faz o seguinte

- Interrompe o banco de dados abruptamente. A instância de banco de dados e suas sessões de cliente podem não ter tempo para desligar com clareza.

  > **Warning**
  >
  > Para evitar a possibilidade de perda de dados, recomendamos interromper as transações em sua instância de banco de dados antes de reinicializar com um failover.

- Executa a recuperação de falhas do banco de dados, se necessário.
- Muda automaticamente para uma réplica em espera em outra zona de disponibilidade. A alteração da zona de disponibilidade pode não ser refletida no AWS Management Console e em chamadas para a AWS CLI e a API do RDS por vários minutos.
- Atualiza o registro DNS da instância de banco de dados para apontar para a instância de banco de dados em espera. Como resultado, você precisará limpar e restabelecer todas as conexões existentes com a sua instância de banco de dados. Para ter mais informações, consulte Configurar e gerenciar uma implantação multi-AZ para o Amazon RDS.
- Cria um evento do Amazon RDS após a reinicialização.

No RDS para Microsoft SQL Server, o failover reinicializa somente a instância de banco de dados principal. Após o failover, a instância de banco de dados primária torna-se a nova instância de banco de dados secundária. Os parâmetros podem não ser atualizados para instâncias Multi-AZ. Para a reinicialização sem failover, as instâncias de banco de dados primárias e secundárias são reinicializadas e os parâmetros são atualizados após a reinicialização. Se a instância de banco de dados não responder, recomendamos reinicializar sem failover.

Considerações sobre quando reinicializar uma instância de banco de dados

Antes de reinicializar a instância, considere o seguinte:

- Para uma instância de banco de dados com réplicas de leitura, é possível reinicializar a instância de banco de dados de origem e suas réplicas de leitura independentemente. Após a conclusão de uma reinicialização, a replicação é retomada automaticamente.
- O tempo de reinicialização depende do processo de recuperação de falhas, da atividade do banco de dados no momento da reinicialização e do comportamento do mecanismo de banco de dados específico. Para melhorar o tempo de reinicialização, recomendamos reduzir as atividades do banco de dados o máximo possível durante a reinicialização. Essa técnica reduz as atividades de reversão para transações em trânsito.

Pré-requisitos para reinicializar uma instância de banco de dados

Você deve atender aos seguintes pré-requisitos:

- Sua instância de banco de dados deve estar no estado `available`. O banco de dados pode ficar indisponível devido a vários motivos, como um backup em andamento, uma modificação solicitada anteriormente ou uma operação de janela de manutenção.
- Se você forçar um failover para uma zona de disponibilidade diferente, a instância de banco de dados deverá estar configurada para multi-AZ.
- Se você forçar um failover para uma zona de disponibilidade diferente, recomendamos primeiro interromper as transações na instância de banco de dados para evitar uma possível perda de dados.

Reinicializar uma instância de banco de dados: etapas básicas

Você pode reinicializar a instância de banco de dados usando o AWS Management Console, a AWS CLI ou a API do RDS.

Console

Para reiniciar uma instância de banco de dados

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Databases (Bancos de dados) e a instância de banco de dados que você deseja reiniciar.
3. Em Actions (Ações), escolha Reboot (Reiniciar).

A página Reinicializar instância de banco de dados é exibida.

4. (Opcional) Escolha Reboot with failover? (Reiniciar com failover?) para forçar um failover de uma AZ para outra.
5. Escolha Reboot (Reinicializar) para reinicializar sua instância de banco de dados.

   Ou escolha Cancel (Cancelar).

AWS CLI

Para reinicializar uma instância de banco de dados usando a AWS CLI, chame o comando `reboot-db-instance`.

Example Reinicialização simples

Para Linux, macOS ou Unix:

```
aws rds reboot-db-instance \
    --db-instance-identifier mydbinstance
```

Para Windows:

```
aws rds reboot-db-instance ^
    --db-instance-identifier mydbinstance
```

Example Reinicializar com failover

Para forçar um failover de uma zona de disponibilidade para outra em um cluster de banco de dados multi-AZ, use o parâmetro `--force-failover`.

Para Linux, macOS ou Unix:

```
aws rds reboot-db-instance \
    --db-instance-identifier mydbinstance \
    --force-failover
```

Para Windows:

```
aws rds reboot-db-instance ^
```

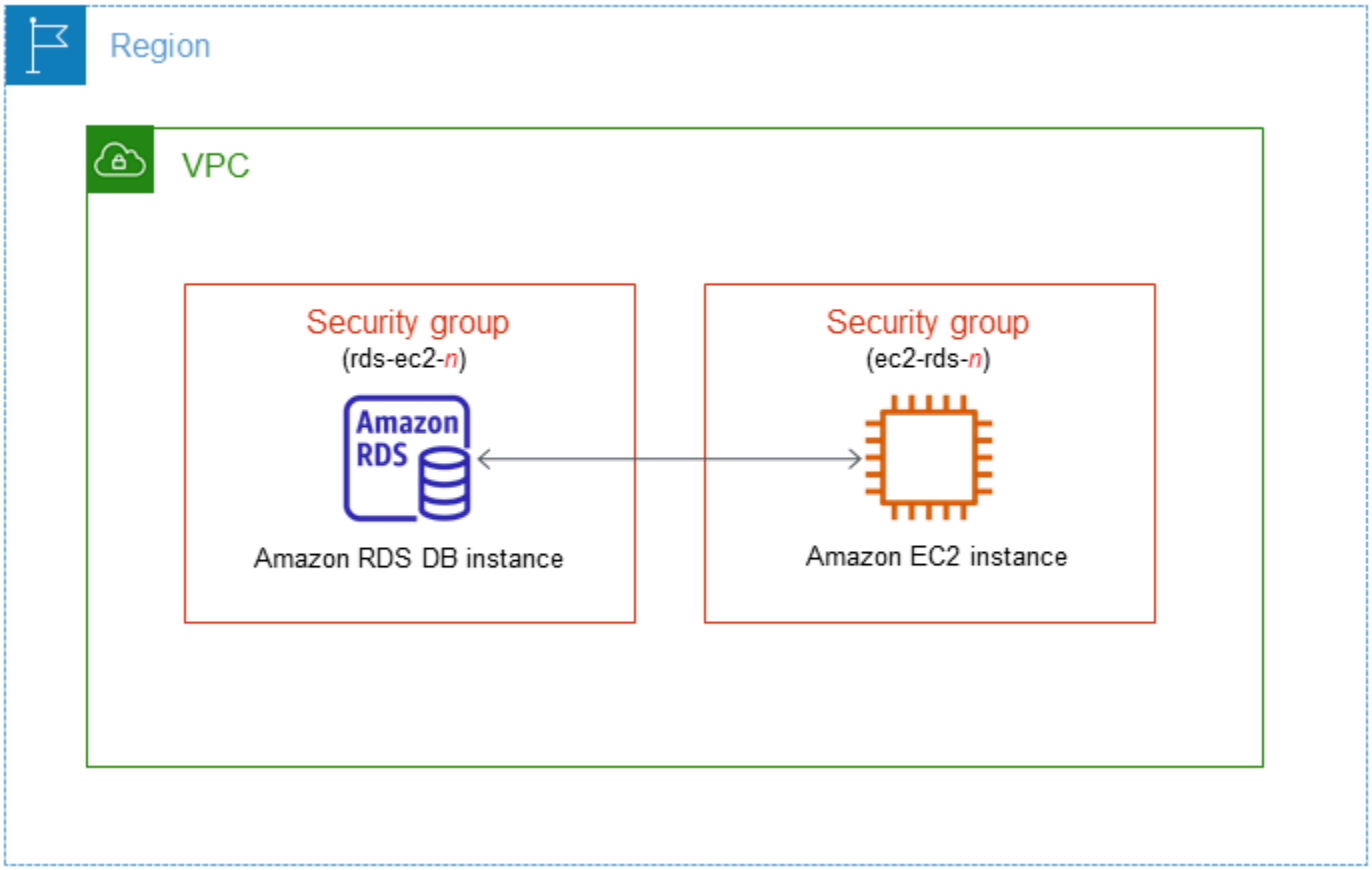
```
    --db-instance-identifier mydbinstance ^
    --force-failover
```

API do RDS

Para reinicializar uma instância de banco de dados usando a API do Amazon RDS, chame a operação `RebootDBInstance`.

Conectar automaticamente uma instância do EC2 e uma instância de banco de dados

Você pode usar o console do Amazon RDS para simplificar a configuração de uma conexão entre uma instância do Amazon Elastic Compute Cloud (Amazon EC2) e uma instância de banco de dados. Geralmente, sua instância de banco de dados está em uma sub-rede privada, e sua instância do EC2 está em uma sub-rede pública em uma VPC. É possível utilizar um cliente SQL na instância do EC2 para se conectar à instância de banco de dados. A instância do EC2 também pode executar aplicações ou servidores que acessam sua instância de banco de dados privado. Para receber instruções sobre como configurar uma conexão entre uma instância do EC2 e um cluster de banco de dados multi-AZ, consulte the section called “Conectar uma instância do EC2 e um cluster de banco de dados multi-AZ”.



Se você quiser se conectar a uma instância do EC2 que não esteja na mesma VPC da instância de banco de dados, veja os cenários em Cenários para acessar uma instância de banco de dados em uma VPC.

Tópicos

- Visão geral da conectividade automática com uma instância do EC2
- Conectar automaticamente uma instância do EC2 e um banco de dados RDS
- Visualizar recursos computacionais conectados
- Conectar a uma instância de banco de dados que está executando um mecanismo de banco de dados específico

Visão geral da conectividade automática com uma instância do EC2

Quando você configura uma conexão entre uma instância do EC2 e um banco de dados do RDS, o RDS automaticamente configura o grupo de segurança da VPC para a instância do EC2 e para o banco de dados do RDS.

Confira a seguir os requisitos para conectar uma instância do EC2 ao banco de dados do RDS:

- A instância do EC2 deve existir na mesma VPC do banco de dados do RDS.

  Se não houver nenhuma instância do EC2 na mesma VPC, o console fornecerá um link para que você crie uma.
- O usuário que configura a conectividade deve ter permissões para realizar as seguintes operações do Amazon EC2:
  - `ec2:AuthorizeSecurityGroupEgress`
  - `ec2:AuthorizeSecurityGroupIngress`
  - `ec2:CreateSecurityGroup`
  - `ec2:DescribeInstances`
  - `ec2:DescribeNetworkInterfaces`
  - `ec2:DescribeSecurityGroups`
  - `ec2:ModifyNetworkInterfaceAttribute`
  - `ec2:RevokeSecurityGroupEgress`

Se a instância de banco de dados e a instância do EC2 estiverem em zonas de disponibilidade diferentes, sua conta poderá incorrer em custos entre as zonas.

Quando você configura uma conexão com uma instância do EC2, o Amazon RDS atua de acordo com a configuração atual dos grupos de segurança associados ao banco de dados do RDS e à instância do EC2, conforme descrito na tabela a seguir.

Configuração atual do grupo de segurança do RDS	Configuração atual do grupo de segurança do EC2	Ação do RDS
Há um ou mais grupos de segurança associados ao banco de dados do RDS com um nome que corresponde ao padrão `rds-ec2-`*n* (em que *n* é um número). Um grupo de segurança que corresponde ao padrão não foi modificado. Esse grupo de segurança tem apenas uma regra de saída com o grupo de segurança da VPC da instância do EC2 como origem.	Há um ou mais grupos de segurança associados à instância do EC2 com um nome que corresponde ao padrão `ec2-rds-`*n* (em que *n* é um número). Um grupo de segurança que corresponde ao padrão não foi modificado. Esse grupo de segurança tem apenas uma regra de saída com o grupo de segurança da VPC do banco de dados do RDS como origem.	O RDS não realiza nenhuma ação. Uma conexão já foi configura da automaticamente entre a instância do EC2 e o banco de dados do RDS. Como já existe uma conexão entre a instância do EC2 e o banco de dados do RDS, os grupos de segurança não são modificad os.
Qualquer uma das seguintes condições se aplica: • Não há um grupo de segurança associado ao banco de dados do RDS com um nome que corresponde ao padrão `rds-ec2-`*n*. • Há um ou mais grupos de segurança associado s ao banco de dados do RDS com um nome que corresponde ao padrão `rds-ec2-`*n*. No entanto, o Amazon RDS não pode usar nenhum desses grupos de segurança para a conexão com a instância do EC2. O Amazon RDS	Qualquer uma das seguintes condições se aplica: • Não há um grupo de segurança associado à instância do EC2 com um nome que corresponde ao padrão `ec2-rds-`*n*. • Há um ou mais grupos de segurança associados à instância do EC2 com um nome que corresponde ao padrão `ec2-rds-`*n*. No entanto, o Amazon RDS não pode usar nenhum desses grupos de segurança para a conexão com o banco de dados do RDS. O Amazon RDS não pode usar um grupo de	RDS action: create new security groups

Configuração atual do grupo de segurança do RDS	Configuração atual do grupo de segurança do EC2	Ação do RDS
não pode usar um grupo de segurança que não tenha uma regra de entrada no grupo de segurança da VPC da instância do EC2 como origem. O Amazon RDS também não pode usar um grupo de segurança que tenha sido modificado. São exemplos de modificação a adição de uma regra ou a alteração da porta de uma regra existente.	segurança que não tenha uma regra de saída com o grupo de segurança da VPC do banco de dados do RDS como origem. O Amazon RDS também não pode usar um grupo de segurança que tenha sido modificado.	
Há um ou mais grupos de segurança associados ao banco de dados do RDS com um nome que correspon de ao padrão `rds-ec2-`*n*. Um grupo de segurança que corresponde ao padrão não foi modificado. Esse grupo de segurança tem apenas uma regra de saída com o grupo de segurança da VPC da instância do EC2 como origem.	Há um ou mais grupos de segurança associados à instância do EC2 com um nome que corresponde ao padrão `ec2-rds-`*n*. No entanto, o Amazon RDS não pode usar nenhum desses grupos de segurança para a conexão com o banco de dados do RDS. O Amazon RDS não pode usar um grupo de segurança que não tenha uma regra de saída com o grupo de segurança da VPC do banco de dados do RDS como origem. O Amazon RDS também não pode usar um grupo de segurança que tenha sido modificado.	RDS action: create new security groups

Configuração atual do grupo de segurança do RDS	Configuração atual do grupo de segurança do EC2	Ação do RDS
Há um ou mais grupos de segurança associados ao banco de dados do RDS com um nome que corresponde ao padrão `rds-ec2-`*n*. Um grupo de segurança que corresponde ao padrão não foi modificado. Esse grupo de segurança tem apenas uma regra de saída com o grupo de segurança da VPC da instância do EC2 como origem.	Existe um grupo de segurança do EC2 válido para a conexão, mas ele não está associado à instância do EC2. Esse grupo de segurança tem um nome que corresponde ao padrão `ec2-rds-`*n*. Não foi modificado. Ele tem apenas uma regra de saída com o grupo de segurança da VPC do banco de dados do RDS como origem.	RDS action: associate EC2 security group

Configuração atual do grupo de segurança do RDS	Configuração atual do grupo de segurança do EC2	Ação do RDS
Qualquer uma das seguintes condições se aplica: • Não há um grupo de segurança associado ao banco de dados do RDS com um nome que corresponde ao padrão `rds-ec2-`*n*. • Há um ou mais grupos de segurança associados ao banco de dados do RDS com um nome que corresponde ao padrão `rds-ec2-`*n*. No entanto, o Amazon RDS não pode usar nenhum desses grupos de segurança para a conexão com a instância do EC2. O Amazon RDS não pode usar um grupo de segurança que não tenha uma regra de entrada no grupo de segurança da VPC da instância do EC2 como origem. O Amazon RDS também não pode usar um grupo de segurança que tenha sido modificado.	Há um ou mais grupos de segurança associados à instância do EC2 com um nome que corresponde ao padrão `ec2-rds-`*n*. Um grupo de segurança que corresponde ao padrão não foi modificado. Esse grupo de segurança tem apenas uma regra de saída com o grupo de segurança da VPC do banco de dados do RDS como origem.	RDS action: create new security groups

Ação do RDS: criar grupos de segurança

O Amazon RDS realiza as seguintes ações:

- Cria um grupo de segurança que corresponde ao padrão `rds-ec2-`*n*. Esse grupo de segurança tem uma regra de entrada com o grupo de segurança da VPC da instância do EC2 como origem. Esse grupo de segurança está associado ao banco de dados do RDS e permite que a instância do EC2 acesse o banco de dados do RDS.
- Cria um grupo de segurança que corresponde ao padrão `ec2-rds-`*n*. Esse grupo de segurança tem uma regra de saída com o grupo de segurança da VPC do banco de dados do RDS como destino. Esse grupo de segurança está associado à instância do EC2 e permite que ela envie tráfego ao banco de dados do RDS.

Ação do RDS: associar o grupo de segurança do EC2

O Amazon RDS associa o grupo de segurança do EC2 existente e válido à instância do EC2. Esse grupo de segurança permite que a instância do EC2 envie tráfego ao banco de dados do RDS.

Conectar automaticamente uma instância do EC2 e um banco de dados RDS

Antes de configurar uma conexão entre uma instância do EC2 e um banco de dados do RDS, atenda aos requisitos descritos em Visão geral da conectividade automática com uma instância do EC2.

Se você alterar esses grupos de segurança depois de configurar a conectividade, as alterações poderão afetar a conexão entre a instância do EC2 e o banco de dados do RDS.

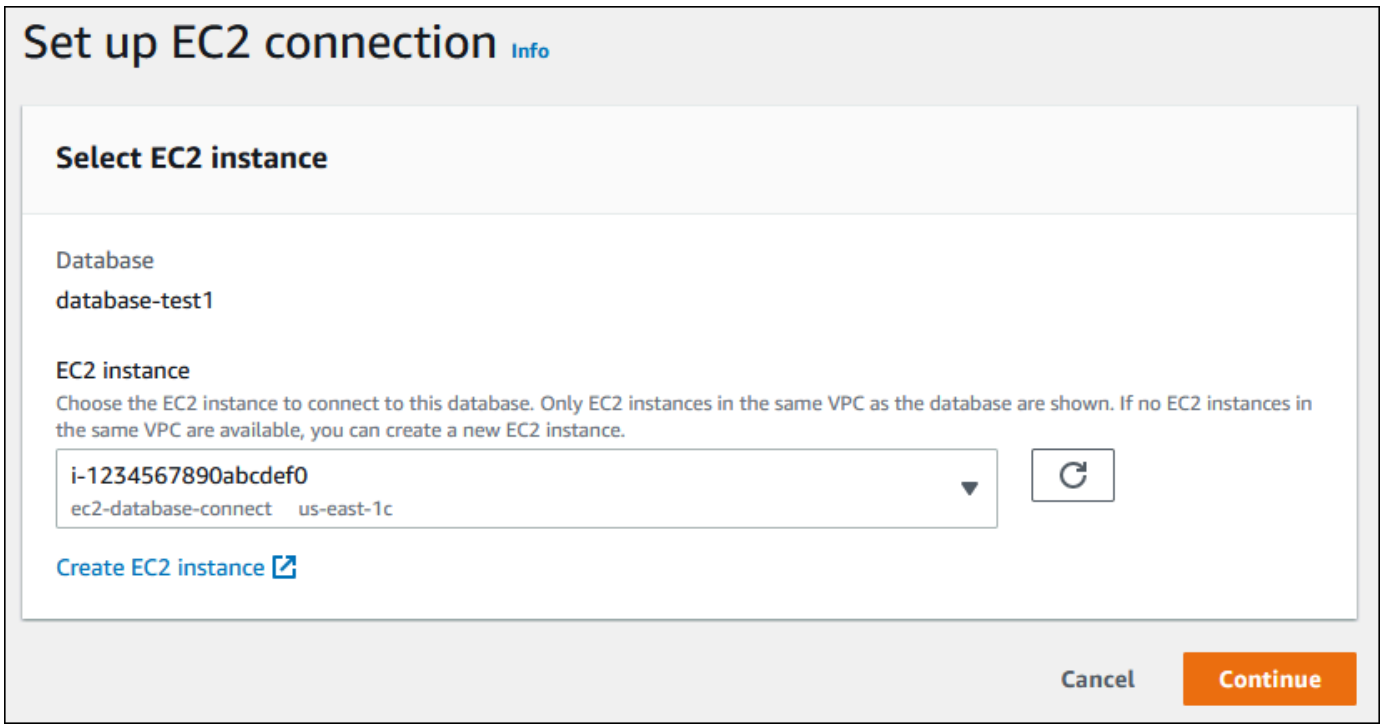
> **Note**
>
> Você só pode configurar automaticamente uma conexão entre uma instância do EC2 e um banco de dados do RDS utilizando o AWS Management Console. Você não pode configurar uma conexão automaticamente com a AWS CLI nem a API do RDS.

Como conectar automaticamente uma instância do EC2 e um banco de dados do RDS

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, selecione Databases (Bancos de dados) e depois selecione banco de dados do RDS.
3. Em Ações, selecione Configurar conexão do EC2.

A página Set up EC2 connection (Configurar conexão do EC2) é exibida.

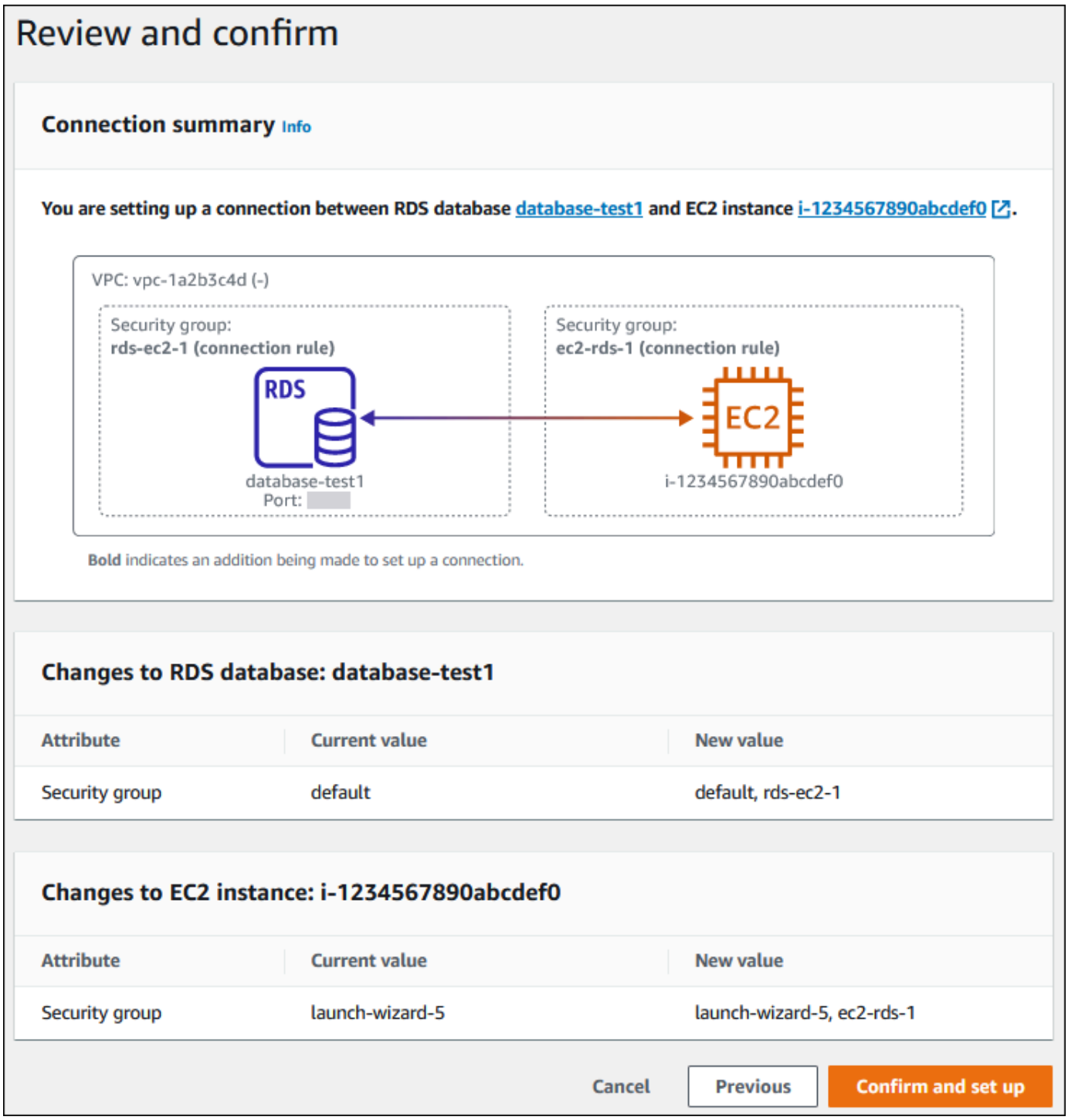
4. Na página Set up EC2 connection (Configurar conexão do EC2), selecione a instância do EC2.



Se não houver nenhuma instância do EC2 na mesma VPC, selecione Create EC2 instance (Criar instância do EC2) para criar uma. Nesse caso, a nova instância do EC2 deve estar na mesma VPC do banco de dados do RDS.

5. Escolha Continuar.

A página Review and confirm (Revisar e confirmar) é exibida.



6. Na página Review and confirm (Revisar e confirmar), analise as alterações que o RDS fará para configurar a conectividade com a instância do EC2.

   Se as alterações estiverem corretas, selecione Confirmar e configurar.

   Se as alterações não estiverem corretas, selecione Previous (Anterior) ou Cancel (Cancelar).

Visualizar recursos computacionais conectados

Você pode usar o AWS Management Console para visualizar os recursos computacionais conectados a um banco de dados do RDS. Os recursos mostrados incluem conexões de recursos computacionais que foram configuradas automaticamente. Você pode configurar a conectividade com recursos computacionais automaticamente das seguintes maneiras:

- Você pode selecionar o recurso computacional ao criar o banco de dados.

  Para ter mais informações, consulte Criar uma instância de banco de dados do Amazon RDS e Criar um cluster de banco de dados multi-AZ para o Amazon RDS.
- Você pode configurar a conectividade entre um banco de dados existente e um recurso computacional.

  Para ter mais informações, consulte Conectar automaticamente uma instância do EC2 e um banco de dados RDS.

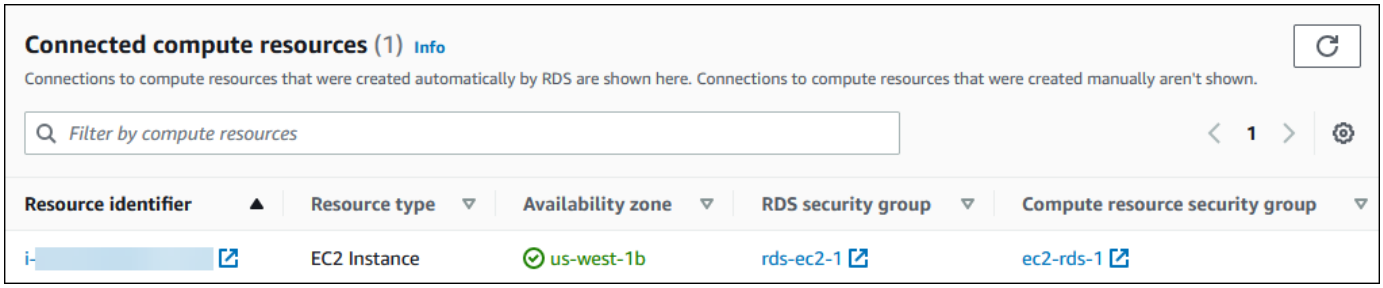
Os recursos computacionais listados não incluem aqueles que foram conectados manualmente ao banco de dados. Por exemplo, você pode permitir que um recurso computacional acesse um banco de dados manualmente adicionando uma regra ao grupo de segurança da VPC associado ao banco de dados.

Para que um recurso computacional seja listado, as seguintes condições devem ser atendidas:

- O nome do grupo de segurança associado ao recurso de computação corresponde ao padrão `ec2-rds-`*n* (em que *n* é um número).
- O grupo de segurança associado ao recurso de computação tem uma regra de saída com o intervalo de portas definido como a porta usada pelo banco de dados RDS.
- O grupo de segurança associado ao recurso computacional tem uma regra de saída com o intervalo de portas definido como um grupo de segurança associado ao banco de dados do RDS.
- O nome do grupo de segurança associado ao banco de dados RDS corresponde ao padrão `rds-ec2-`*n* (em que *n* é um número).
- O grupo de segurança associado ao banco de dados RDS tem uma regra de entrada com o intervalo de portas definido como a porta usada pelo banco de dados RDS.
- O grupo de segurança associado ao banco de dados RDS tem uma regra de entrada com a fonte definida como um grupo de segurança associado ao recurso computacional.

Como visualizar os recursos de computação conectados a um banco de dados do RDS

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, selecione Databases (Bancos de dados) e depois selecione o nome do banco de dados RDS.
3. Na guia Connectivity & security (Conectividade e segurança), veja os recursos computacionais em Connected compute resources (Recursos computacionais conectados).



Conectar a uma instância de banco de dados que está executando um mecanismo de banco de dados específico

Para obter informações sobre a conexão a uma instância de banco de dados que esteja executando um mecanismo de banco de dados específico, siga as instruções do mecanismo de banco de dados:

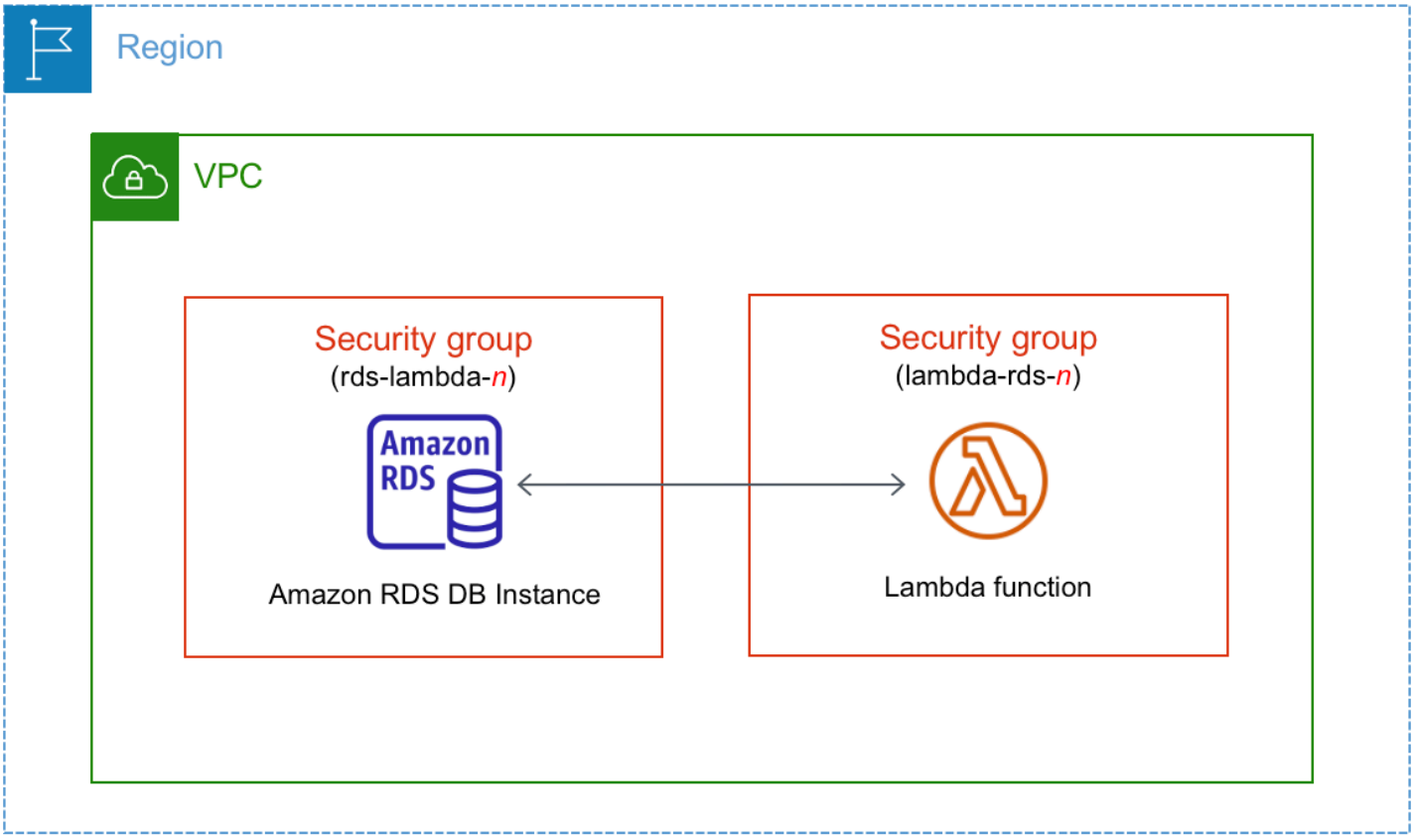
- Conectar-se à instância de banco de dados do MariaDB
- Conectar-se à instância de banco de dados do Microsoft SQL Server
- Conectar-se à instância de banco de dados do MySQL
- Conectar-se a uma instância de banco de dados Oracle
- Conectar a uma instância de banco de dados executando o mecanismo de banco de dados do PostgreSQL

Conectar automaticamente uma função do Lambda e uma instância de banco de dados

Você pode usar o console do Amazon RDS para simplificar a configuração de uma conexão entre uma função do Lambda e uma instância de banco de dados. Muitas vezes, a instância de banco de dados está em uma sub-rede privada dentro de uma VPC. A função do Lambda pode ser usada por aplicações para acessar a instância de banco de dados privada.

Para receber instruções sobre como configurar uma conexão entre uma função do Lambda e um cluster de banco de dados multi-AZ, consulte the section called “Conectar uma função do Lambda e um cluster de banco de dados multi-AZ”.

A imagem a seguir mostra uma conexão direta entre a instância de banco de dados e a função do Lambda.

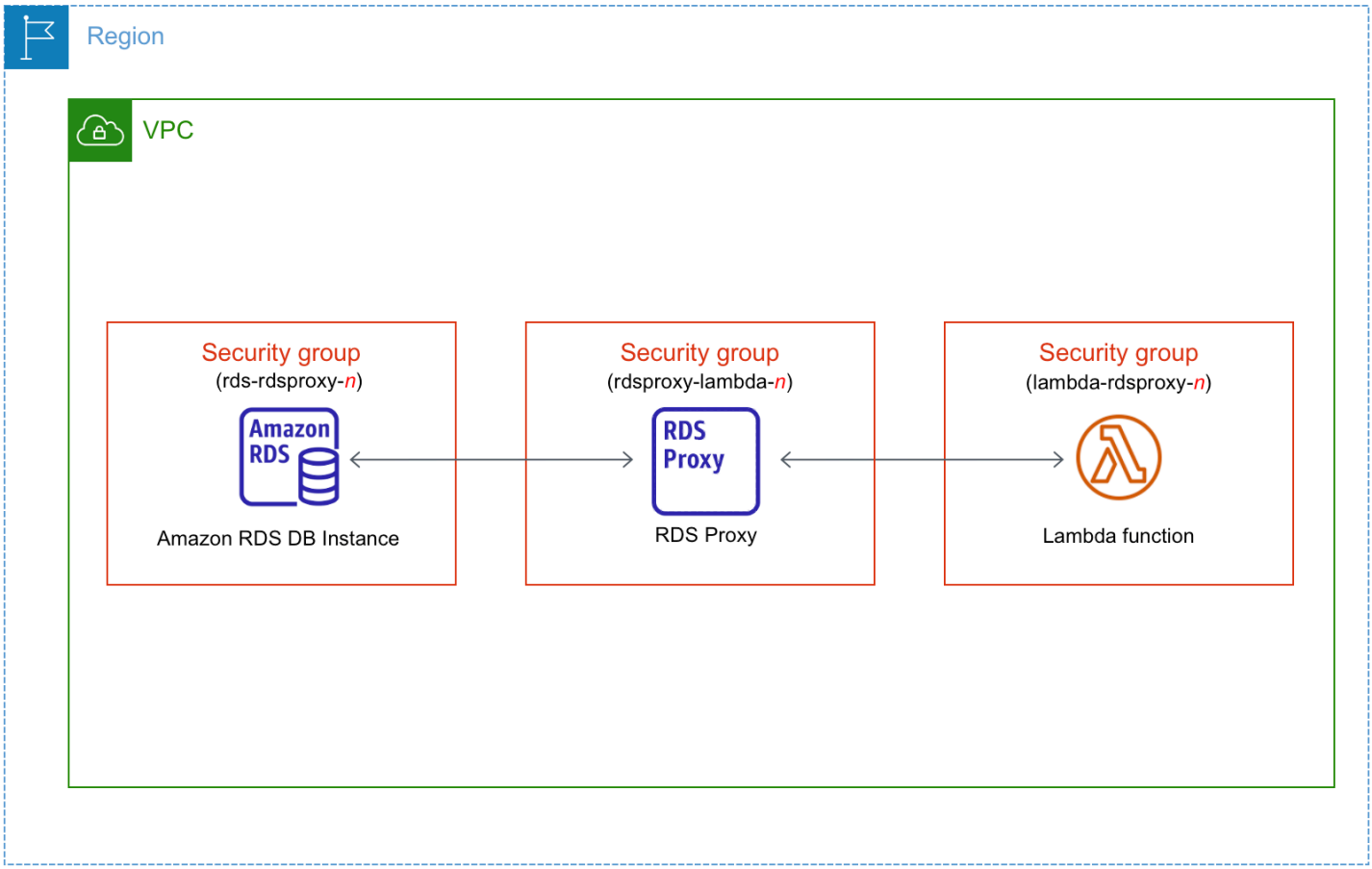


Você pode configurar a conexão entre a função do Lambda e a instância de banco de dados por meio do RDS Proxy para melhorar a performance e a resiliência do banco de dados. Em geral, as funções do Lambda fazem conexões curtas frequentes com o banco de dados que se beneficiam do grupo de conexões oferecido pelo RDS Proxy. É possível aproveitar qualquer autenticação

do AWS Identity and Access Management (IAM) que você já tenha para funções do Lambda, em vez de gerenciar credenciais de banco de dados no código de aplicação do Lambda. Para ter mais informações, consulte Amazon RDS Proxy.

Quando você usa o console para se conectar a um proxy existente, o Amazon RDS atualiza o grupo de segurança do proxy para permitir conexões da instância de banco de dados e a função do Lambda.

Você também pode criar um proxy na mesma página do console. Ao criar um proxy no console, para acessar a instância de banco de dados, você deve inserir suas credenciais do banco de dados ou selecionar um segredo do AWS Secrets Manager.



Tip

Para conectar rapidamente uma função do Lambda a uma instância de banco de dados, você também pode usar o assistente guiado no console. Para abrir o assistente, faça o seguinte:

1. Abra a página Funções do console do Lambda.

2. Selecione a função à qual você deseja conectar um banco de dados.
3. Na guia Configuração, selecione bancos de dados do RDS.
4. Escolha Conectar ao banco de dados do RDS.

Depois de conectar a função a um banco de dados, você pode criar um proxy escolhendo Adicionar proxy.

Tópicos

- Visão geral da conectividade automática com uma função do Lambda
- Conectar automaticamente uma função do Lambda e um banco de dados do RDS
- Visualizar recursos computacionais conectados

Visão geral da conectividade automática com uma função do Lambda

Confira abaixo os requisitos para conectar uma função do Lambda a uma instância de banco de dados do RDS:

- A função do Lambda deve existir na mesma VPC que a instância de banco de dados.
- O usuário que configura a conectividade deve ter permissões para realizar as seguintes operações do Amazon RDS, do Amazon EC2, do Lambda, do Secrets Manager e do IAM:
  - Amazon RDS
    - `rds:CreateDBProxies`
    - `rds:DescribeDBInstances`
    - `rds:DescribeDBProxies`
    - `rds:ModifyDBInstance`
    - `rds:ModifyDBProxy`
    - `rds:RegisterProxyTargets`
  - Amazon EC2
    - `ec2:AuthorizeSecurityGroupEgress`
    - `ec2:AuthorizeSecurityGroupIngress`
    - `ec2:CreateSecurityGroup`

    - `ec2:DeleteSecurityGroup`
    - `ec2:DescribeSecurityGroups`
    - `ec2:RevokeSecurityGroupEgress`
    - `ec2:RevokeSecurityGroupIngress`
- Lambda
    - `lambda:CreateFunctions`
    - `lambda:ListFunctions`
    - `lambda:UpdateFunctionConfiguration`
- Secrets Manager
    - `secretsmanager:CreateSecret`
    - `secretsmanager:DescribeSecret`
- IAM
    - `iam:AttachPolicy`
    - `iam:CreateRole`
    - `iam:CreatePolicy`
- AWS KMS
    - `kms:describeKey`

> **Note**
>
> Se a instância de banco de dados e a função do Lambda estiverem em zonas de disponibilidade diferentes, sua conta poderá incorrer em custos entre as zonas.

Quando você configura uma conexão entre uma função do Lambda e um banco de dados do RDS, o Amazon RDS configura o grupo de segurança da VPC para sua instância e para a instância banco de dados. Se você usa o RDS Proxy, o Amazon RDS também configura o grupo de segurança da VPC para o proxy. O Amazon RDS atua de acordo com a configuração atual dos grupos de segurança associados à instância de banco de dados, à função do Lambda e ao proxy, conforme descrito na tabela a seguir.

Configuração atual do grupo de segurança do RDS	Configuração do grupo de segurança do Lambda atual	Configuração do grupo de segurança de proxy atual	Ação do RDS
Há um ou mais grupos de segurança associados à instância de banco de dados com um nome correspondente ao padrão `rds-lambda-` *n* ou se um proxy já estiver conectado à instância de banco de dados, o RDS conferirá se o `TargetHealth` de um proxy associado está `AVAILABLE`. Um grupo de segurança que corresponde ao padrão não foi modificado. Esse grupo de segurança tem apenas uma regra de entrada com o grupo de segurança da VPC da função do Lambda ou do proxy como origem.	Há um ou mais grupos de segurança associados à função do Lambda com um nome correspondente ao padrão `lambda-rds-` *n* ou `lambda-rdsproxy-` *n* (em que *n* é um número). Um grupo de segurança que corresponde ao padrão não foi modificado. Esse grupo de segurança tem apenas uma regra de saída com o grupo de segurança da VPC da instância de banco de dados ou do proxy como destino.	Há um ou mais grupos de segurança associados ao proxy com um nome correspondente ao padrão `rdsproxy-lambda-` *n* (em que *n* é um número). Um grupo de segurança que corresponde ao padrão não foi modificado. Esse grupo de segurança tem regras de entrada e saída com os grupos de segurança da VPC da função do Lambda e da instância de banco de dados.	O Amazon RDS não realiza nenhuma ação. Uma conexão já foi configurada automaticamente entre a função do Lambda, o proxy (opcional) e a instância de banco de dados. Como já existe uma conexão entre a função, o proxy e o banco de dados, os grupos de segurança não são modificados.
Qualquer uma das seguintes condições se aplica:	Qualquer uma das seguintes condições se aplica:	Qualquer uma das seguintes condições se aplica:	RDS action: create new security groups

Configuração atual do grupo de segurança do RDS	Configuração do grupo de segurança do Lambda atual	Configuração do grupo de segurança de proxy atual	Ação do RDS
• Não há nenhum grupo de segurança associado à instância de banco de dados com um nome correspondente ao padrão `rds-lambda-` *n* ou se o `TargetHealth` de um proxy associado está `AVAILABLE`. • Há um ou mais grupos de segurança associados à instância de banco de dados com um nome correspondente ao padrão `rds-lambda-` *n* ou se o `TargetHealth` de um proxy associado estiver `AVAILABLE`. No entanto, nenhum desses grupos de segurança pode ser usado para a	• Não há um grupo de segurança associado à função do Lambda com um nome correspondente ao padrão `lambda-rds-` *n* ou `lambda-rdsproxy-` *n*. • Há um ou mais grupos de segurança associados à função do Lambda com um nome correspondente ao padrão `lambda-rds-` *n* ou `lambda-rdsproxy-` *n*. No entanto, o Amazon RDS não pode usar nenhum desses grupos de segurança para a conexão com a instância de banco de dados. O Amazon RDS não pode usar um grupo de segurança que não tenha uma regra de	• Não há um grupo de segurança associado ao proxy com um nome correspondente ao padrão `rdsproxy-lambda-` *n*. • Há um ou mais grupos de segurança associados ao proxy com um nome correspondente ao `rdsproxy-lambda-` *n*. No entanto, o Amazon RDS não pode usar nenhum desses grupos de segurança para a conexão com a instância de banco de dados ou a função do Lambda. O Amazon RDS não pode usar um grupo de segurança que não tenha regras de entrada e saída com o grupo de segurança	

Configuração atual do grupo de segurança do RDS	Configuração do grupo de segurança do Lambda atual	Configuração do grupo de segurança de proxy atual	Ação do RDS
conexão com a função do Lambda. O Amazon RDS não pode usar um grupo de segurança que não tenha uma regra de entrada no grupo de segurança da VPC da função do Lambda ou do proxy como origem. O Amazon RDS também não pode usar um grupo de segurança que tenha sido modificado. São exemplos de modificação a adição de uma regra ou a alteração da porta de uma regra existente.	saída com o grupo de segurança da VPC da instância de banco de dados ou do proxy como origem. O Amazon RDS também não pode usar um grupo de segurança que tenha sido modificado.	da VPC da instância de banco de dados ou da função do Lambda. O Amazon RDS também não pode usar um grupo de segurança que tenha sido modificado.	

Configuração atual do grupo de segurança do RDS	Configuração do grupo de segurança do Lambda atual	Configuração do grupo de segurança de proxy atual	Ação do RDS
Há um ou mais grupos de segurança associados à instância de banco de dados com um nome correspondente ao padrão `rds-lambda-` *n* ou se o `TargetHealth` de um proxy associado estiver `AVAILABLE`. Um grupo de segurança que corresponde ao padrão não foi modificado. Esse grupo de segurança tem apenas uma regra de entrada com o grupo de segurança da VPC da função do Lambda ou do proxy como origem.	Há um ou mais grupos de segurança associados à função do Lambda com um nome correspondente ao padrão `lambda-rds-` *n* ou `lambda-rdsproxy-` *n*. No entanto, o Amazon RDS não pode usar nenhum desses grupos de segurança para a conexão com a instância de banco de dados. O Amazon RDS não pode usar um grupo de segurança que não tenha uma regra de saída com o grupo de segurança da VPC da instância de banco de dados ou do proxy como origem. O Amazon RDS também não pode usar um grupo de segurança que tenha sido modificado.	Há um ou mais grupos de segurança associados ao proxy com um nome correspondente ao padrão `rdsproxy-lambda-` *n*. No entanto, o Amazon RDS não pode usar nenhum desses grupos de segurança para a conexão com a instância de banco de dados ou a função do Lambda. O Amazon RDS não pode usar um grupo de segurança que não tenha regras de entrada e saída com o grupo de segurança da VPC da instância de banco de dados ou da função do Lambda. O Amazon RDS também não pode usar um grupo de segurança que tenha sido modificado.	RDS action: create new security groups

Configuração atual do grupo de segurança do RDS	Configuração do grupo de segurança do Lambda atual	Configuração do grupo de segurança de proxy atual	Ação do RDS
Há um ou mais grupos de segurança associados à instância de banco de dados com um nome correspondente ao padrão `rds-lambda-` *n* ou se o `TargetHealth` de um proxy associado estiver `AVAILABLE`. Um grupo de segurança que corresponde ao padrão não foi modificado. Esse grupo de segurança tem apenas uma regra de entrada com o grupo de segurança da VPC da função do Lambda ou do proxy como origem.	Existe um grupo de segurança do Lambda válido para a conexão, mas ele não está associado à função do Lambda. Esse grupo de segurança tem um nome correspondente ao padrão `lambda-rds-` *n* ou `lambda-rdsproxy-` *n*. Não foi modificado. Ele tem apenas uma regra de saída com o grupo de segurança da VPC da instância de banco de dados ou do proxy como destino.	Existe um grupo de segurança do proxy válido para a conexão, mas ele não está associado ao proxy. Esse grupo de segurança tem um nome que corresponde ao padrão `rdsproxy-lambda-` *n*. Não foi modificado. Ele tem regras de entrada e saída com os grupos de segurança da VPC da instância de banco de dados e da função do Lambda.	RDS action: associate Lambda security group

Configuração atual do grupo de segurança do RDS	Configuração do grupo de segurança do Lambda atual	Configuração do grupo de segurança de proxy atual	Ação do RDS
Qualquer uma das seguintes condições se aplica: • Não há nenhum grupo de segurança associado à instância de banco de dados com um nome correspondente ao padrão `rds-lambda-` *n* ou se o `TargetHealth` de um proxy associado está `AVAILABLE`. • Há um ou mais grupos de segurança associados à instância de banco de dados com um nome correspondente ao padrão `rds-lambda-` *n* ou se o `TargetHealth` de um proxy associado estiver `AVAILABLE`. No entanto, o	Há um ou mais grupos de segurança associados à função do Lambda com um nome correspondente ao padrão `lambda-rds-` *n* ou `lambda-rdsproxy-` *n*. Um grupo de segurança que corresponde ao padrão não foi modificado. Esse grupo de segurança tem apenas uma regra de saída com o grupo de segurança da VPC da instância de banco de dados ou do proxy como destino.	Há um ou mais grupos de segurança associados ao proxy com um nome correspondente ao padrão `rdsproxy-lambda-` *n*. Um grupo de segurança que corresponde ao padrão não foi modificado. Esse grupo de segurança tem regras de entrada e saída com os grupos de segurança da VPC da instância de banco de dados e da função do Lambda.	RDS action: create new security groups

Configuração atual do grupo de segurança do RDS	Configuração do grupo de segurança do Lambda atual	Configuração do grupo de segurança de proxy atual	Ação do RDS
Amazon RDS não pode usar nenhum desses grupos de segurança para a conexão com a função do Lambda ou o proxy. O Amazon RDS não pode usar um grupo de segurança que não tenha uma regra de entrada no grupo de segurança da VPC da função do Lambda ou do proxy como origem. O Amazon RDS também não pode usar um grupo de segurança que tenha sido modificado.			

Configuração atual do grupo de segurança do RDS	Configuração do grupo de segurança do Lambda atual	Configuração do grupo de segurança de proxy atual	Ação do RDS
Qualquer uma das seguintes condições se aplica: • Não há nenhum grupo de segurança associado à instância de banco de dados com um nome correspondente ao padrão `rds-lambda-` *n* ou se o `TargetHealth` de um proxy associado está `AVAILABLE`. • Há um ou mais grupos de segurança associados à instância de banco de dados com um nome correspondente ao padrão `rds-lambda-` *n* ou se o `TargetHealth` de um proxy associado estiver `AVAILABLE`. No entanto, o	Qualquer uma das seguintes condições se aplica: • Não há um grupo de segurança associado à função do Lambda com um nome correspondente ao padrão `lambda-rds-` *n* ou `lambda-rdsproxy-` *n*. • Há um ou mais grupos de segurança associados à função do Lambda com um nome correspondente ao padrão `lambda-rds-` *n* ou `lambda-rdsproxy-` *n*. No entanto, o Amazon RDS não pode usar nenhum desses grupos de segurança para a conexão com a instância de banco de dados.	Qualquer uma das seguintes condições se aplica: • Não há um grupo de segurança associado ao proxy com um nome correspondente ao padrão `rdsproxy-lambda-` *n*. • Há um ou mais grupos de segurança associados ao proxy com um nome correspondente ao `rdsproxy-lambda-` *n*. No entanto, o Amazon RDS não pode usar nenhum desses grupos de segurança para a conexão com a instância de banco de dados ou a função do Lambda. O Amazon RDS não pode usar um grupo	RDS action: create new security groups

Configuração atual do grupo de segurança do RDS	Configuração do grupo de segurança do Lambda atual	Configuração do grupo de segurança de proxy atual	Ação do RDS
Amazon RDS não pode usar nenhum desses grupos de segurança para a conexão com a função do Lambda ou o proxy. O Amazon RDS não pode usar um grupo de segurança que não tenha uma regra de entrada no grupo de segurança da VPC da função do Lambda ou do proxy como origem. O Amazon RDS também não pode usar um grupo de segurança que tenha sido modificado.	O Amazon RDS não pode usar um grupo de segurança que não tenha uma regra de saída com o grupo de segurança da VPC da instância de banco de dados ou do proxy como origem. O Amazon RDS também não pode usar um grupo de segurança que tenha sido modificado.	de segurança que não tenha regras de entrada e saída com o grupo de segurança da VPC da instância de banco de dados ou da função do Lambda. O Amazon RDS também não pode usar um grupo de segurança que tenha sido modificado.	

Ação do RDS: criar grupos de segurança

O Amazon RDS realiza as seguintes ações:

- Cria um grupo de segurança que corresponde ao padrão `rds-lambda-`*n* ou `rds-rdsproxy-`*n* (se você optar por usar o RDS Proxy). Esse grupo de segurança tem uma regra de entrada com o grupo de segurança da VPC da função do Lambda ou do proxy como origem. Esse grupo de segurança está associado à instância de banco de dados e permite que a função ou o proxy acesse a instância de banco de dados.

- Cria um grupo de segurança que corresponde ao padrão `lambda-rds-`*n* ou ao `lambda-rdsproxy-`*n*. Esse grupo de segurança tem uma regra de saída com o grupo de segurança da VPC da instância de banco de dados ou do proxy como destino. Esse grupo de segurança está associado à função do Lambda e permite que a função envie tráfego para a instância de banco de dados ou envie tráfego por meio de um proxy.
- Cria um grupo de segurança que corresponde ao padrão `rdsproxy-lambda-`*n*. Esse grupo de segurança tem regras de entrada e saída com os grupos de segurança da VPC da instância de banco de dados e da função do Lambda.

Ação do RDS: associar o grupo de segurança do Lambda

O Amazon RDS associa o grupo de segurança do Lambda válido e existente à função do Lambda. Esse grupo de segurança permite que a função envie tráfego para a instância de banco de dados ou envie tráfego por meio de um proxy.

Conectar automaticamente uma função do Lambda e um banco de dados do RDS

Você pode usar o console do Amazon RDS para conectar automaticamente uma função do Lambda à instância de banco de dados. Isso simplifica o processo de configuração de uma conexão entre esses recursos.

Você também pode usar o RDS Proxy para incluir um proxy em sua conexão. As funções do Lambda fazem conexões curtas frequentes com o banco de dados que se beneficiam do grupo de conexões oferecido pelo RDS Proxy. Também é possível usar qualquer autenticação do IAM que você já tenha para funções do Lambda, em vez de gerenciar credenciais de banco de dados no código da aplicação do Lambda.

Você pode conectar uma instância de banco de dados existente a funções do Lambda novas e existentes usando a página Configurar conexão do Lambda. O processo de configuração define automaticamente os grupos de segurança necessários para você.

Antes de configurar uma conexão entre uma função do Lambda e uma instância de banco de dados, garanta que:

- Sua função do Lambda e a instância de banco de dados estejam na mesma VPC.

- Você tenha as permissões corretas para sua conta do usuário. Para receber mais informações sobre os requisitos, consulte Visão geral da conectividade automática com uma função do Lambda.

Se você alterar os grupos de segurança depois de configurar a conectividade, as alterações poderão afetar a conexão entre a função do Lambda e a instância de banco de dados.

> **Note**
>
> Você possa configurar automaticamente uma conexão entre uma instância de banco de dados e uma função do Lambda somente no AWS Management Console. Para conectar uma função do Lambda, a instância de banco de dados deve estar no estado Disponível.

Como conectar automaticamente uma função do Lambda e uma instância de banco de dados

<result>

Depois de confirmar a configuração, o Amazon RDS inicia o processo de conexão da função do Lambda, do RDS Proxy (se você usou um proxy) e da instância de banco de dados. O console mostra a caixa de diálogo Detalhes da conexão, que lista as alterações do grupo de segurança que permitem conexões entre seus recursos.
</result>

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, selecione Bancos de dados e, depois, a instância de banco de dados que você deseja conectar a uma função do Lambda.
3. Em Ações, selecione Configurar conexão do Lambda.
4. Na página Configurar conexão do Lambda, em Selecionar função do Lambda, faça o seguinte:
   - Se você tiver uma função do Lambda existente na mesma VPC que a instância de banco de dados, selecione Selecionar função existente e, depois, escolha a função.
   - Se você não tiver uma função do Lambda na mesma VPC, selecione Criar função do Lambda e, depois, insira um Nome da função. O runtime padrão é definido como Nodejs.18. Você pode modificar as configurações de sua nova função do Lambda no console do Lambda depois de concluir a configuração da conexão.
5. (Opcional) Em RDS Proxy, selecione Conectar usando o RDS Proxy e, depois, faça o seguinte:

- Se você tiver um proxy que deseja usar, escolha Selecionar um proxy existente e, depois, escolha o proxy.
- Se você não tiver um proxy e quiser que o Amazon RDS crie um automaticamente para você, selecione Criar um proxy. Então, para Credenciais de banco de dados, faça o seguinte:

  a. Selecione Nome de usuário e senha do banco de dados e, depois, insira o Nome do usuário e a Senha para a instância de banco de dados.

  b. Selecione Segredo do Secrets Manager. Então, em Selecionar segredo, escolha um segredo do AWS Secrets Manager. Se você não tiver um segredo do Secrets Manager, selecione Criar um segredo do Secrets Manager para criar um segredo. Depois de criar o segredo, em Selecionar segredo, escolha o novo segredo.

  Depois de criar o proxy, escolha Selecionar proxy existente e, depois, escolha o proxy. Observe que pode levar algum tempo até que o proxy esteja disponível para conexão.

6. (Opcional) Expanda Resumo da conexão e verifique as atualizações destacadas para seus recursos.
7. Escolha Configurar.

Visualizar recursos computacionais conectados

Você pode usar o AWS Management Console para visualizar as funções do Lambda que estão conectadas à instância de banco de dados. Os recursos mostrados incluem conexões de recursos de computação que o Amazon RDS configurou automaticamente.

Os recursos de computação listados não incluem aqueles que são conectados manualmente à instância de banco de dados. Por exemplo, você pode permitir que um recurso de computação acesse a instância de banco de dados manualmente adicionando uma regra ao grupo de segurança da VPC associado ao banco de dados.

Para que o console liste uma função do Lambda, as seguintes condições devem ser aplicadas:

- O nome do grupo de segurança associado ao recurso de computação corresponde ao padrão `lambda-rds-`*n* ou `lambda-rdsproxy-`*n* (em que *n* é um número).
- O grupo de segurança associado ao recurso de computação tem uma regra de saída com o intervalo de portas definido como a porta da instância de banco de dados ou de um proxy

associado. O destino da regra de saída deve ser definido como um grupo de segurança associado à instância de banco de dados ou a um proxy associado.

- Se a configuração incluir um proxy, o nome do grupo de segurança anexado ao proxy associado ao banco de dados corresponderá ao padrão `rdsproxy-lambda-`*n* (em que *n* é um número).
- O grupo de segurança associado à função tem uma regra de saída com a porta definida como a porta que a instância de banco de dados ou o proxy associado usa. O destino deve ser definido como um grupo de segurança associado à instância de banco de dados ou ao proxy associado.

Como visualizar os recursos de computação conectados automaticamente a uma instância de banco de dados

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, selecione Bancos de dados e escolha a instância de banco de dados.
3. Na guia Conectividade e segurança, veja os recursos de computação em Recursos de computação conectados.

Modificar uma instância de banco de dados do Amazon RDS

Você pode alterar as configurações de uma instância de banco de dados para realizar tarefas, como adicionar mais armazenamento ou alterar a classe da instância de banco de dados. Neste tópico, é possível descobrir como modificar uma instância de banco de dados do Amazon RDS e saber mais sobre as configurações para instâncias de banco de dados.

Convém testar todas as alterações na instância de teste antes de modificar a instância de produção. Isso ajuda você a entender completamente o impacto de cada alteração. Testar é importante especialmente ao atualizar as versões do banco de dados.

A maioria das modificações em uma instância de banco de dados podem ser aplicadas imediatamente ou adiadas até a próxima janela de manutenção. Algumas modificações, como alterações em grupos de parâmetros, exigem que você reinicie manualmente sua instância de banco de dados para que a alteração entre em vigor.

> **Important**
>
> Algumas modificações resultam em inatividade, pois o Amazon RDS deve reiniciar sua instância de banco de dados para que a alteração entre em vigor. Analise o impacto sobre o seu banco de dados e os seus aplicativos antes de modificar as configurações da instância do banco de dados.

Console

Como modificar uma instância de banco de dados

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Databases (Bancos de dados) e a instância de banco de dados que você deseja modificar.
3. Selecione Modify. A página Modify DB instance (Modificar instância de banco de dados) será exibida.
4. Altere qualquer uma das configurações desejadas. Para obter informações sobre cada configuração, consulte Configurações para instâncias de banco de dados.
5. Quando todas as alterações estiverem conforme o desejado, escolha Continue (Continuar) e verifique o resumo das modificações.

6. (Opcional) Escolha Apply immediately (Aplicar imediatamente) para aplicar as alterações imediatamente. Escolher essa opção pode causar uma inatividade em alguns casos. Para obter mais informações, consulte Usar a configuração de programação de modificações.
7. Na página de confirmação, revise suas alterações. Se estiverem corretas, escolha Modify DB Instance (Modificar instância de banco de dados) para salvar suas alterações.

   Ou selecione Back (Voltar) para editar as alterações ou Cancel (Cancelar) para cancelar as alterações.

AWS CLI

Para modificar uma instância de banco de dados usando a AWS CLI, chame o comando modify-db-instance. Especifique o identificador da instância de banco de dados e os valores para as configurações que deseja modificar. Para obter mais informações sobre cada opção, consulte Configurações para instâncias de banco de dados.

Example

O código a seguir modifica `mydbinstance` configurando o período de retenção de backup como 1 semana (7 dias). O código habilita a proteção contra exclusão usando `--deletion-protection`. Para desativar a proteção contra exclusão, use `--no-deletion-protection`. As alterações serão aplicadas durante a janela de manutenção usando `--no-apply-immediately`. Use `--apply-immediately` para aplicar as alterações imediatamente. Para ter mais informações, consulte Usar a configuração de programação de modificações.

Para Linux, macOS ou Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --backup-retention-period 7 \
    --deletion-protection \
    --no-apply-immediately
```

Para Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --backup-retention-period 7 ^
    --deletion-protection ^
```

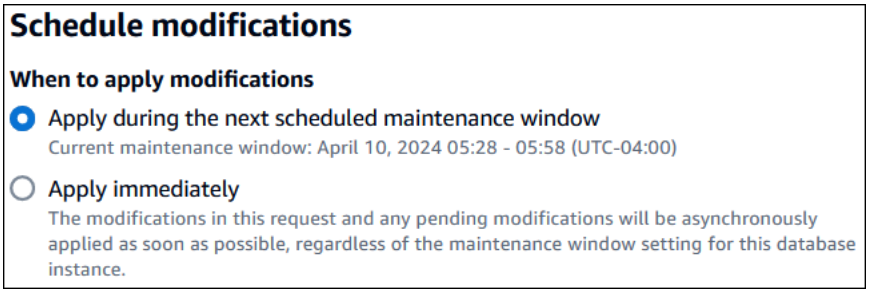
```
    --no-apply-immediately
```

API do RDS

Para modificar uma instância de banco de dados usando a API do Amazon RDS, chame a operação ModifyDBInstance. Especifique o DB instance identifier e os parâmetros para as configurações que você deseja modificar. Para obter informações sobre cada parâmetro, consulte Configurações para instâncias de banco de dados.

Usar a configuração de programação de modificações

Ao modificar sua instância de banco de dados, você decide quando deseja que as modificações ocorram.



Para aplicar as alterações imediatamente e não na próxima janela de manutenção, selecione a opção Aplicar imediatamente no AWS Management Console. Você também pode usar o parâmetro `--apply-immediately` ao chamar a AWS CLI ou definir o parâmetro `ApplyImmediately` como `true` ao usar a API do Amazon RDS.

Se você não optar por aplicar as alterações imediatamente, o RDS as colocará na fila de modificações pendentes. Durante a próxima janela de manutenção, o RDS aplica todas as alterações pendentes na fila. Se você optar por aplicar as alterações imediatamente, as novas alterações e todas as alterações na fila de modificações pendentes serão aplicadas.

Para ver as modificações pendentes para a próxima janela de manutenção, use o comando describe-db-instances da AWS CLI e verifique o campo `PendingModifiedValues`.

> **Important**
>
> Se alguma das modificações pendentes exigir que a instância de banco de dados está temporariamente indisponível (downtime) (inatividade), a escolha da opção para aplicar imediatamente poderá causar um tempo de inatividade inesperado.

Quando você opta por aplicar uma alteração imediatamente, todas as modificações pendentes também são aplicadas de imediato, em vez de durante a próxima janela de manutenção.
Se não quiser que uma alteração pendente seja aplicada na próxima janela de manutenção, é possível modificar a instância de banco de dados para reverter a alteração. Você pode fazer isso usando a AWS CLI e especificando a opção `--apply-immediately`.

As alterações em algumas configurações do banco de dados são aplicadas imediatamente, mesmo que você opte por adiá-las. Para ver como as diferentes configurações do banco de dados interagem com a configuração Aplicar imediatamente, consulte Configurações para instâncias de banco de dados.

Configurações para instâncias de banco de dados

Na tabela a seguir, é possível encontrar detalhes sobre quais configurações podem e não podem ser modificadas. Você também pode descobrir quando as alterações podem ser aplicadas e se elas causam tempo de inatividade para sua instância de banco de dados. Ao usar recursos do Amazon RDS, como multi-AZ, você poderá minimizar o tempo de inatividade se modificar posteriormente a instância de banco de dados. Para obter mais informações, consulte Configurar e gerenciar uma implantação multi-AZ para o Amazon RDS.

É possível modificar uma instância de banco de dados usando o console, o comando da CLI `modify-db-instance` ou a operação da API do RDS `ModifyDBInstance`.

Configuração e descrição do console	Opção da CLI e parâmetro da API do RDS	Quando a alteração ocorre	Observações sobre tempo de inatividade	Mecanismos de banco de dados com suporte
Armazenamento alocado O armazenamento, em gibibytes, que você deseja alocar para sua instância de banco de	Opção da CLI: `--allocated-storage`	Se você optar por aplicar a alteração imediatamente,	Não ocorre uma inatividade durante esta alteração	Todos os mecanismos de

Configuração e descrição do console	Opção da CLI e parâmetro da API do RDS	Quando a alteração ocorre	Observações sobre tempo de inatividade	Mecanismos de banco de dados com suporte
dados. Só é possível aumentar o armazenamento alocado. Não é possível reduzir o armazenamento alocado. Não é possível modificar o armazenamento de algumas instâncias de banco de dados mais antigas e das instâncias de banco de dados restauradas de snapshots de banco de dados mais antigos. A opção Allocated storage (Armazenamento alocado) estará desabilitada no console se a sua instância de banco de dados não for qualificada. É possível verificar se mais armazenamento pode ser alocado usando o comando da CLI describe-valid-db-instance-modifications. Esse comando retorna as opções de armazenamento válidas para sua instância de banco de dados. Não será possível modificar o armazenamento alocado se o status da instância de banco de dados for storage-optimizati	Parâmetro da API do RDS: `AllocatedStorage`	ela ocorrerá imediatamente. Se você não optar por aplicar a alteração imediatamente, ela ocorrerá durante a próxima janela de manutenção.	. A performance pode ser degradada durante a alteração.	banco de dados

Configuração e descrição do console	Opção da CLI e parâmetro da API do RDS	Quando a alteração ocorre	Observações sobre tempo de inatividade	Mecanismos de banco de dados com suporte
on. Também não será possível modificar o armazenamento alocado para uma instância de banco de dados se ela tiver sido modificada nas últimas seis horas. O armazenamento máximo permitido depende do mecanismo do banco de dados e do tipo de armazenamento. Para obter mais informações, consulte Armazenamento de instâncias de banco de dados do Amazon RDS.				

Configuração e descrição do console	Opção da CLI e parâmetro da API do RDS	Quando a alteração ocorre	Observações sobre tempo de inatividade	Mecanismos de banco de dados com suporte
Configuração de arquitetura Uma configuração que permite que vários bancos de dados locatários residam em sua instância de banco de dados. Atualmente, somente o RDS para bancos de dados de contêineres (CDBs) Oracle oferece suporte a essa configuração. Se seu CDB estiver na configuração de locatário único, você poderá modificá-lo para usar a configuração de multilocatário. Nessa configuração, é possível usar as APIs do RDS para criar bancos de dados de um a trinta locatários, dependendo da edição do banco de dados e de quaisquer licenças de opção necessárias. Não há suporte para PDBs de aplicativos e PDBs de proxy. A configuração de multilocatário é permanente, o que significa que você não poderá posteriormente converter seu CDB de volta para a configuração de locatário único.	Opção da CLI: `--multi-tenant` (configuração de multilocatário da arquitetura CDB) `--no-multi-tenant` (configuração de locatário único da arquitetura CDB) Parâmetro da API: `MultiTenant`	A alteração ocorre imediatamente.	Não ocorre uma inatividade durante esta alteração.	Oracle

Configuração e descrição do console	Opção da CLI e parâmetro da API do RDS	Quando a alteração ocorre	Observações sobre tempo de inatividade	Mecanismos de banco de dados com suporte
Note A configuração do Amazon RDS é chamada de “multi-tenant”, em vez de “multitenant”, porque é um recurso da plataforma RDS, não apenas do mecanismo de banco de dados Oracle. Da mesma forma, o termo “locatário” do RDS refere-se a qualquer locatário em uma configuração do RDS, não apenas aos PDBs da Oracle. Na documentação do RDS, o termo “multilocatário da Oracle” (Oracle multitenant), sem hífen, refere-se exclusivamente à arquitetura CDB de banco de dados Oracle, que é compatível com implantações on-premises e do RDS.				

Configuração e descrição do console	Opção da CLI e parâmetro da API do RDS	Quando a alteração ocorre	Observações sobre tempo de inatividade	Mecanismos de banco de dados com suporte
Para obter mais informações, consulte Visão geral dos CDBs RDS para Oracle.				

Configuração e descrição do console	Opção da CLI e parâmetro da API do RDS	Quando a alteração ocorre	Observações sobre tempo de inatividade	Mecanismos de banco de dados com suporte
Configurações de arquitetura A arquitetura do banco de dados Oracle: CDB ou não CDB. Se você escolher Arquitetura multilocatário Oracle, o RDS para Oracle vai converter seu não CDB em um CDB que usa configuração de locatário único. Essa configuração só é compatível se seu banco de dados é um não CDB que executa o Oracle Database 19c com a RU de abril de 2021 ou posterior. Depois da conversão, o CDB conterá um banco de dados inicialmente conectável (PDB). A mudança na arquitetura é permanente, o que significa que você não pode converter seu CDB novamente em um que não seja CDB. **Note** Para converter um CDB na configuração de inquilino único na	Opção da CLI: Oracle Multilocatário `--engine oracle-se2-cdb` (Oracle multilocatário) Parâmetro da API: `Engine`	Se você optar por aplicar a alteração imediatamente, ela ocorrerá imediatamente. Se você não optar por aplicar a alteração imediatamente, ela ocorrerá durante a próxima janela de manutenção.	Ocorre uma inatividade durante esta alteração.	Oracle

Configuração e descrição do console	Opção da CLI e parâmetro da API do RDS	Quando a alteração ocorre	Observações sobre tempo de inatividade	Mecanismos de banco de dados com suporte
configuração de vários locatários, modifique sua instância de CDB novamente e escolha a configuração de vários multilocatários para sua configuração de arquitetura. Para obter mais informações, consulte Configuração de locatário único da arquitetura CDB.				

Configuração e descrição do console	Opção da CLI e parâmetro da API do RDS	Quando a alteração ocorre	Observações sobre tempo de inatividade	Mecanismos de banco de dados com suporte

Atualização da versão secundária automática
 	
Selecione Habilitar o upgrade automático da versão secundária para permitir que a instância de banco de dados receba automaticamente atualizações da versão do mecanismo de banco de dados secundário preferencial quando elas forem disponibilizadas. Esse é o comportamento padrão. O Amazon RDS executa atualizações automáticas de versões secundárias na janela de manutenção. Se você não selecionar Habilitar o upgrade automático da versão secundária, a instância de banco de dados não será atualizada automaticamente quando novas versões secundárias forem disponibilizadas.
 	
Para obter mais informações, consulte Atualizar automaticamente a versão do mecanismo espelho.
	Opção da CLI:
	
	`--auto-minor-version-upgrade | --no-auto-minor-version-upgrade`
	
	Parâmetro da API do RDS:
	
	`AutoMinorVersionUpgrade`
		A alteração ocorre imediatamente. Essa configuração ignora a configuração para aplicar imediatamente.	Não ocorre uma inatividade durante esta alteração.	Todos os mecanismos de banco de dados

Configuração e descrição do console	Opção da CLI e parâmetro da API do RDS	Quando a alteração ocorre	Observações sobre tempo de inatividade	Mecanismos de banco de dados com suporte
Backup replication (Replicação de backup) Escolha Enable replication to another Region da AWS (Habilitar replicação para outra região) para criar backups em uma região da adicional para recuperação de desastres. Em seguida, escolha a Destination Region (Região de destino) para os backups adicionais.	Não disponível ao modificar uma instância de banco de dados. Para obter informações sobre como ativar backups entre regiões usando a AWS CLI ou a API do RDS, consulte Habilitar backups automatizados entre regiões.	A alteração é aplicada de forma assíncrona logo que possível.	Não ocorre uma inatividade durante esta alteração.	Todos os mecanismos de banco de dados

Configuração e descrição do console	Opção da CLI e parâmetro da API do RDS	Quando a alteração ocorre	Observações sobre tempo de inatividade	Mecanismos de banco de dados com suporte

Período de retenção de backup
 	
Por quantos dias os backups automáticos serão mantidos. Para desabilitar backups automáticos, defina o período de retenção de backup como 0.
 	
Para obter mais informações, consulte Introdução aos backups.
 	
Note
Se você usar o AWS Backup para gerenciar os backups, essa opção não será aplicável. Para obter informações sobre o AWS Backup, consulte o Guia do desenvolvedor doAWS Backup.
	Opção da CLI:
	
	`--backup-retention-period`
	
	Parâmetro da API do RDS:
	
	`BackupRetentionPeriod`
		Se você optar por aplicar a alteração imediatamente, ela ocorrerá imediatamente.
		
		Se você não optar por aplicar imediatamente e alterar a configuração de um valor diferente de zero para outro valor diferente de zero, a alteração será aplicada de forma assíncrona o mais rápido possível. Caso contrário, a alteração ocorrerá durante a próxima janela
			Uma inatividade ocorrerá se você alterar de 0 para um valor diferente de zero ou de um valor diferente de zero para 0.
			
			Isso se aplica a instâncias de banco de dados single-AZ e multi-AZ.
				Todos os mecanismos de banco de dados

Configuração e descrição do console	Opção da CLI e parâmetro da API do RDS	Quando a alteração ocorre	Observações sobre tempo de inatividade	Mecanismos de banco de dados com suporte
		de manutenção.		

Janela de backup
 	
O intervalo de tempo durante o qual ocorrem backups automatizados dos seus bancos de dados. A janela de backup é uma hora de início no Tempo Coordenado Universal (UTC) e uma duração em horas.
 	
Para obter mais informações, consulte Introdução aos backups.
 	
Note
Se você usar o AWS Backup para gerenciar os backups, essa opção não será exibida. Para obter mais informações sobre o AWS Backup, consulte o Guia do desenvolvedor do AWS Backup.
	Opção da CLI:
	
	`--preferred-backup-window`
	
	Parâmetro da API do RDS:
	
	`PreferredBackupWindow`
		A alteração é aplicada de forma assíncrona logo que possível.	Não ocorre uma inatividade durante esta alteração.	Todos os mecanismos de banco de dados

Configuração e descrição do console	Opção da CLI e parâmetro da API do RDS	Quando a alteração ocorre	Observações sobre tempo de inatividade	Mecanismos de banco de dados com suporte

Autoridade certificadora
 	
A autoridade de certificação (CA) para o certificado do servidor usado pela instância de banco de dados.
 	
Para obter mais informações, consulte Usar SSL/TLS para criptografar uma conexão com uma instância ou um cluster de banco de dados.
	Opção da CLI:
	
	`--ca-certificate-identifier`
	
	Parâmetro da API do RDS:
	
	`CACertificateIdentifier`
		Se você optar por aplicar a alteração imediatamente, ela ocorrerá imediatamente.
		
		Se você não optar por aplicar a alteração imediatamente, ela ocorrerá durante a próxima janela de manutenção.
			Tempo de inatividade ocorrerá somente se o mecanismo de banco de dados não for compatível com alternância sem reinicialização. Você pode usar o comando da AWS CLI describe-db-engine-versions para determinar se o mecanismo de banco de dados é compatível com alternância sem reinicialização.	Todos os mecanismos de banco de dados

Configuração e descrição do console	Opção da CLI e parâmetro da API do RDS	Quando a alteração ocorre	Observações sobre tempo de inatividade	Mecanismos de banco de dados com suporte

Copiar tags para snapshots
 	
Se tiver tags de instância de banco de dados, habilite essa opção para copiá-las quando você criar um snapshot de banco de dados.
 	
Para obter mais informações, consulte Marcar recursos do Amazon RDS.
	Opção da CLI:
	
	`--copy-tags-to-snapshot` ou `--no-copy-tags-to-snapshot`
	
	Parâmetro da API do RDS:
	
	`CopyTagsToSnapshot`
		A alteração ocorre imediatamente. Essa configuração ignora a configuração para aplicar imediatamente.	Não ocorre uma inatividade durante esta alteração.	Todos os mecanismos de banco de dados

Configuração e descrição do console	Opção da CLI e parâmetro da API do RDS	Quando a alteração ocorre	Observações sobre tempo de inatividade	Mecanismos de banco de dados com suporte

Porta de banco de dados
 	
A porta que você deseja usar para acessar a instância de banco de dados.
 	
O valor da porta não deve corresponder a nenhum dos valores de porta especificados para as opções no grupo de opções associado à instância de banco de dados.
 	
Para obter mais informações, consulte Conectar a uma instância de banco de dados do Amazon RDS.
	Opção da CLI:
	
	`--db-port-number`
	
	Parâmetro da API do RDS:
	
	`DBPortNumber`
		A alteração ocorre imediatamente. Essa configuração ignora a configuração para aplicar imediatamente.	A instância do banco de dados é reinicializada imediatamente.	Todos os mecanismos de banco de dados

Configuração e descrição do console	Opção da CLI e parâmetro da API do RDS	Quando a alteração ocorre	Observações sobre tempo de inatividade	Mecanismos de banco de dados com suporte

DB engine version (Versão do mecanismo de banco de dados)
 	
A versão do mecanismo de banco de dados que você deseja usar. Antes de atualizar sua instância de bancos de dados de produção, recomendamos que você teste o processo de atualização em uma instância de banco de dados de teste. Isso ajuda a verificar sua duração e validar suas aplicações.
 	
Para obter mais informações, consulte Atualizar a versão de mecanismo de uma instância de banco de dados.
	Opção da CLI:
	
	`--engine-version`
	
	Parâmetro da API do RDS:
	
	`EngineVersion`
		Se você optar por aplicar a alteração imediatamente, ela ocorrerá imediatamente.
		
		Se você não optar por aplicar a alteração imediatamente, ela ocorrerá durante a próxima janela de manutenção.
			Ocorre uma inatividade durante esta alteração.	Todos os mecanismos de banco de dados

Configuração e descrição do console	Opção da CLI e parâmetro da API do RDS	Quando a alteração ocorre	Observações sobre tempo de inatividade	Mecanismos de banco de dados com suporte

Classe de instância de banco de dados
 	
A classe da instância de banco de dados que você quer usar.
 	
Para obter mais informações, consulte Classes de instâncias de banco de dados .
	Opção da CLI:
	
	`--db-instance-class`
	
	Parâmetro da API do RDS:
	
	`DBInstanceClass`
		Se você optar por aplicar a alteração imediatamente, ela ocorrerá imediatamente.
		
		Se você não optar por aplicar a alteração imediatamente, ela ocorrerá durante a próxima janela de manutenção.
			Ocorre uma inatividade durante esta alteração.	Todos os mecanismos de banco de dados

Configuração e descrição do console	Opção da CLI e parâmetro da API do RDS	Quando a alteração ocorre	Observações sobre tempo de inatividade	Mecanismos de banco de dados com suporte

DB instance identifier
 	
O novo identificador da instância de banco de dados. Esse valor é armazenado como uma string em minúsculas.
 	
Para obter mais informações sobre os efeitos de renomear uma instância de banco de dados, consulte Renomear uma instância de banco de dados.
	Opção da CLI:
	
	`--new-db-instance-identifier`
	
	Parâmetro da API do RDS:
	
	`NewDBInstanceIdentifier`
		Se você optar por aplicar a alteração imediatamente, ela ocorrerá imediatamente.
		
		Se você não optar por aplicar a alteração imediatamente, ela ocorrerá durante a próxima janela de manutenção.
			O tempo de inatividade ocorre durante essa alteração, a menos que a versão do mecanismo de banco de dados seja compatível com o carregamento dinâmico de SSL. Para determinar se a versão precisa ser reiniciada, execute o seguinte comando AWS CLI:
			
			`aws rds describe-db-engine-versions \ --default-only \`
				Todos os mecanismos de banco de dados

Configuração e descrição do console	Opção da CLI e parâmetro da API do RDS	Quando a alteração ocorre	Observações sobre tempo de inatividade	Mecanismos de banco de dados com suporte
			`--engine your-db-engine \ --query 'DBEngineVersions[*].SupportsCertificateRotationWithoutRestart'`	

Configuração e descrição do console	Opção da CLI e parâmetro da API do RDS	Quando a alteração ocorre	Observações sobre tempo de inatividade	Mecanismos de banco de dados com suporte

Grupo de parâmetros de banco de dados
 	
O grupo de parâmetros de banco de dados que você deseja associar à instância de banco de dados.
 	
Para obter mais informações, consulte Grupos de parâmetros para Amazon RDS.
	Opção da CLI:
	
	`--db-parameter-group-name`
	
	Parâmetro da API do RDS:
	
	`DBParameterGroupName`
		A associação do novo grupo de parâmetros de banco de dados à instância de banco de dados ocorre imediatamente.	Não há tempo de inatividade quando você associa um novo grupo de parâmetros de banco de dados à instância de banco de dados.
			
			A associação de um grupo de parâmetros de banco de dados é diferente da aplicação de alterações de parâmetros dentro de um grupo de parâmetros. O RDS aplica configurações de parâmetros estáticos e dinâmicos
				Todos os mecanismos de banco de dados

Configuração e descrição do console	Opção da CLI e parâmetro da API do RDS	Quando a alteração ocorre	Observações sobre tempo de inatividade	Mecanismos de banco de dados com suporte
			modificadas no grupo recém-associado somente depois que você reinicializa manualmente a instância de banco de dados. No entanto, se você modificar parâmetros dinâmicos no grupo de parâmetros de banco de dados depois de associá-lo à instância de banco de dados, essas configurações de parâmetros serão aplicadas imediatamente sem exigir uma reinicialização.	

Configuração e descrição do console	Opção da CLI e parâmetro da API do RDS	Quando a alteração ocorre	Observações sobre tempo de inatividade	Mecanismos de banco de dados com suporte
			Para obter mais informações, consulte Grupos de parâmetros para Amazon RDS e Reinicializar uma instância de banco de dados .	

Volume de log dedicado
 	
Use um volume de log dedicado (DLV) para armazenar registros de transações do banco de dados em um volume de armazenamento separado do volume que contém as tabelas do banco de dados.
 	
Para obter mais informações, consulte Usando um volume de registro dedicado (DLV).
	Opção da CLI:
	
	`-dedicated-log-volume`
	
	Parâmetro da API do RDS:
	
	`DedicatedLogVolume`
		A alteração só será aplicada depois que a instância de banco de dados for reinicializada.	O tempo de inatividade ocorre enquanto a instância de banco de dados é reinicializada.	MariaDB, MySQL, PostgreSQL

Configuração e descrição do console	Opção da CLI e parâmetro da API do RDS	Quando a alteração ocorre	Observações sobre tempo de inatividade	Mecanismos de banco de dados com suporte

Deletion protection (Proteção contra exclusão)
 	
Selecione Enable deletion protection (Habilitar proteção contra exclusão) para impedir que sua instância de banco de dados seja excluída.
 	
Para obter mais informações, consulte Excluir uma instância de banco de dados.
	Opção da CLI:
	
	`--deletion-protection|--no-deletion-protection`
	
	Parâmetro da API do RDS:
	
	`DeletionProtection`
		A alteração ocorre imediatamente. Essa configuração ignora a configuração para aplicar imediatamente.	Não ocorre uma inatividade durante esta alteração.	Todos os mecanismos de banco de dados

Configuração e descrição do console	Opção da CLI e parâmetro da API do RDS	Quando a alteração ocorre	Observações sobre tempo de inatividade	Mecanismos de banco de dados com suporte

Monitoramento avançado
 	
Selecione Enable enhanced monitoring (Habilitar o monitoramento avançado) para habilitar a coleta de métricas em tempo real do sistema operacional em que a instância de banco de dados é executada.
 	
Para obter mais informações, consulte Monitorar métricas do SO com o monitoramento avançado.
	Opção da CLI:
	
	`--monitoring-interval` e `--monitoring-role-arn`
	
	Parâmetro da API do RDS:
	
	`MonitoringInterval` e `MonitoringRoleArn`
		A alteração ocorre imediatamente. Essa configuração ignora a configuração para aplicar imediatamente.	Não ocorre uma inatividade durante esta alteração.	Todos os mecanismos de banco de dados

Configuração e descrição do console	Opção da CLI e parâmetro da API do RDS	Quando a alteração ocorre	Observações sobre tempo de inatividade	Mecanismos de banco de dados com suporte

IAM DB authentication
 	
Enable IAM DB authentication (Habilitar a autenticação de banco de dados do IAM) para autenticar usuários do banco de dados por meio de usuários e perfis.
 	
Para obter mais informações, consulte Autenticação do banco de dados do IAM para MariaDB, MySQL e PostgreSQL.
	Opção da CLI:
	
	`--enable-iam-database-authentication |--no-enable-iam-database-authentication`
	
	Parâmetro da API do RDS:
	
	`EnableIAMDatabaseAuthentication`
		Se você optar por aplicar a alteração imediatamente, ela ocorrerá imediatamente.
		
		Se você não optar por aplicar a alteração imediatamente, ela ocorrerá durante a próxima janela de manutenção.
			Não ocorre uma inatividade durante esta alteração.	Somente MariaDB, MySQL e PostgreSQL

Configuração e descrição do console	Opção da CLI e parâmetro da API do RDS	Quando a alteração ocorre	Observações sobre tempo de inatividade	Mecanismos de banco de dados com suporte

Autenticação de Kerberos
 	
Escolha o Active Directory para o qual migrar a instância de banco de dados. O diretório deve existir antes que essa operação seja realizada. Se um diretório já estiver selecionado, especifique None (Nenhum) para remover a instância de banco de dados de seu diretório atual.
 	
Para obter mais informações, consulte Autenticação de Kerberos.
	Opção da CLI:
	
	`--domain` e `--domain-iam-role-name`
	
	Parâmetro da API do RDS:
	
	`Domain` e `DomainIAMRoleName`
		Se você optar por aplicar a alteração imediatamente, ela ocorrerá imediatamente.
		
		Se você não optar por aplicar a alteração imediatamente, ela ocorrerá durante a próxima janela de manutenção.
			Um breve tempo de inatividade ocorre durante essa alteração.	Somente Microsoft SQL Server, MySQL, Oracle e PostgreSQL

Configuração e descrição do console	Opção da CLI e parâmetro da API do RDS	Quando a alteração ocorre	Observações sobre tempo de inatividade	Mecanismos de banco de dados com suporte

Modelo de licença
 	
Selecione traga a sua própria licença para usar sua licença do Db2 e da Oracle.
 	
Escolha license-included (licença inclusa) para usar o contrato de licença geral do Microsoft SQL Server ou do Oracle.
 	
Para ter mais informações, consulte Opções de licenciamento do Amazon RDS para Db2, Licenciar o Microsoft SQL Server no Amazon RDS e Opções de licenciamento do RDS para Oracle.
	Opção da CLI:
	
	`--license-model`
	
	Parâmetro da API do RDS:
	
	`LicenseModel`
		Se você optar por aplicar a alteração imediatamente, ela ocorrerá imediatamente.
		
		Se você não optar por aplicar a alteração imediatamente, ela ocorrerá durante a próxima janela de manutenção.
			Ocorre uma inatividade durante esta alteração.	Somente Microsoft SQL Server e Oracle

Configuração e descrição do console	Opção da CLI e parâmetro da API do RDS	Quando a alteração ocorre	Observações sobre tempo de inatividade	Mecanismos de banco de dados com suporte

Exportações de log
 	
Os tipos de arquivos de log de bancos de dados a serem publicados no Amazon CloudWatch Logs.
 	
Para ter mais informações, consulte Publicação de logs de banco de dados no Amazon CloudWatch Logs.
	Opção da CLI:
	
	`--cloudwatch-logs-export-configuration`
	
	Parâmetro da API do RDS:
	
	`CloudwatchLogsExportConfiguration`
		A alteração ocorre imediatamente. Essa configuração ignora a configuração para aplicar imediatamente.	Não ocorre uma inatividade durante esta alteração.	Todos os mecanismos de banco de dados

Configuração e descrição do console	Opção da CLI e parâmetro da API do RDS	Quando a alteração ocorre	Observações sobre tempo de inatividade	Mecanismos de banco de dados com suporte

Janela de manutenção
 	
O intervalo de tempo durante o qual a manutenção do sistema ocorre. A manutenção do sistema inclui upgrades, se aplicáveis. A janela de manutenção é uma hora de início no Tempo Coordenado Universal (UTC) e uma duração em horas.
 	
Se você definir a janela como a hora atual, deverá haver pelo menos 30 minutos entre a hora atual e o final da janela. Esse tempo ajuda a garantir que todas as alterações pendentes sejam aplicadas.
 	
Para obter mais informações, consulte A janela de manutenção do Amazon RDS.
	Opção da CLI:
	
	`--preferred-maintenance-window`
	
	Parâmetro da API do RDS:
	
	`PreferredMaintenanceWindow`
		A alteração ocorre imediatamente. Essa configuração ignora a configuração para aplicar imediatamente.	Se houver uma ou mais ações pendentes que provoquem uma inatividade, e a janela de manutenção for alterada para incluir a hora atual, essas ações pendentes serão aplicadas imediatamente e ocorrerá uma inatividade.	Todos os mecanismos de banco de dados

Configuração e descrição do console	Opção da CLI e parâmetro da API do RDS	Quando a alteração ocorre	Observações sobre tempo de inatividade	Mecanismos de banco de dados com suporte

Gerenciar credenciais principais no AWS Secrets Manager
 	
Selecione Gerenciar credenciais principais no AWS Secrets Manager para gerenciar a senha do usuário principal em um segredo no Secrets Manager.
 	
Opcionalmente, selecione uma chave do KMS a ser usada para proteger o segredo. Escolha entre uma das chaves do KMS da sua conta ou insira a chave de uma conta distinta.
 	
Se o RDS já estiver gerenciando a senha do usuário principal para a instância de banco de dados, você poderá alternar a senha do usuário principal selecionando Rotate secret immediately (Alternar segredo imediatamente).
 	
Para obter mais informações, consulte Gerenciamento de senhas com Amazon RDS e AWS Secrets Manager.
	Opção da CLI:
	
	`--manage-master-user-password | --no-manage-master-user-password`
	
	`--master-user-secret-kms-key-id`
	
	`--rotate-master-user-password | --no-rotate-master-user-password`
	
	Parâmetro da API do RDS:
		Se você estiver ativando ou desativando o gerenciamento automático de senhas de usuário principal, a alteração ocorrerá imediatamente. Essa configuração ignora a configuração para aplicar imediatamente.
		
		Se você estiver alternando a senha do usuário principal, deverá especificar que a alteração seja aplicada imediatamente.
			Não ocorre uma inatividade durante esta alteração.	Todos os mecanismos de banco de dados

Configuração e descrição do console	Opção da CLI e parâmetro da API do RDS	Quando a alteração ocorre	Observações sobre tempo de inatividade	Mecanismos de banco de dados com suporte

	`ManageMasterUserPassword`
	
	`MasterUserSecretKmsKeyId`
	
	`RotateMasterUserPassword`			

Configuração e descrição do console	Opção da CLI e parâmetro da API do RDS	Quando a alteração ocorre	Observações sobre tempo de inatividade	Mecanismos de banco de dados com suporte

Implantação multi-AZ
 	
Yes (Sim) para implantar sua instância de banco de dados em várias zonas de disponibilidade. Caso contrário, No (Não).
 	
Para obter mais informações, consulte Configurar e gerenciar uma implantação multi-AZ para o Amazon RDS.
	Opção da CLI:
	
	`--multi-az|--no-multi-az`
	
	Parâmetro da API do RDS:
	
	`MultiAZ`
		Se você optar por aplicar a alteração imediatamente, ela ocorrerá imediatamente.
		
		Se você não optar por aplicar a alteração imediatamente, ela ocorrerá durante a próxima janela de manutenção.
			Não ocorre uma inatividade durante esta alteração. No entanto, há um possível impacto na performance. Para obter mais informações, consulteConverte uma instância de banco de dados em uma implantação multi-AZ para o Amazon RDS	Todos os mecanismos de banco de dados

Configuração e descrição do console	Opção da CLI e parâmetro da API do RDS	Quando a alteração ocorre	Observações sobre tempo de inatividade	Mecanismos de banco de dados com suporte

Tipo de rede
 	
Os protocolos de endereçamento IP compatíveis com a instância de banco de dados.
 	
IPv4 para especificar que os recursos podem se comunicar com a instância de banco de dados somente por meio do protocolo de endereçamento Internet Protocol versão 4 (IPv4).
 	
Modo de pilha dupla para especificar que os recursos podem se comunicar com a instância de banco de dados por IPv4, Internet Protocol versão 6 (IPv6) ou ambos. Use o modo de pilha dupla se você tiver algum recurso que precise se comunicar com sua instância de banco de dados pelo protocolo de endereçamento IPv6. Além disso, associe um bloco CIDR IPv6 a todas as sub-redes no grupo de sub-redes de banco de dados especificado.
	Opção da CLI:
	
	`--network-type`
	
	Parâmetro da API do RDS:
	
	`NetworkType`
		Se você optar por aplicar a alteração imediatamente, ela ocorrerá imediatamente.
		
		Se você não optar por aplicar a alteração imediatamente, ela ocorrerá durante a próxima janela de manutenção.
			É possível que ocorra inatividade durante esta alteração.	Todos os mecanismos de banco de dados

Configuração e descrição do console	Opção da CLI e parâmetro da API do RDS	Quando a alteração ocorre	Observações sobre tempo de inatividade	Mecanismos de banco de dados com suporte
Para obter mais informações, consulte Endereçamento IP do Amazon RDS				

Nova senha mestre
 	
A senha da conta de usuário mestre. A senha deve conter de 8 a 41 caracteres alfanuméricos.
	Opção da CLI:
	
	`--master-user-password`
	
	Parâmetro da API do RDS:
	
	`MasterUserPassword`
		A alteração é aplicada de forma assíncrona logo que possível. Essa configuração ignora a configuração para aplicar imediatamente.	Não ocorre uma inatividade durante esta alteração.	Todos os mecanismos de banco de dados

Configuração e descrição do console	Opção da CLI e parâmetro da API do RDS	Quando a alteração ocorre	Observações sobre tempo de inatividade	Mecanismos de banco de dados com suporte

Option group
 	
O grupo de opções que você deseja associar à instância de banco de dados.
 	
Para obter mais informações, consulte Trabalhar com grupos de opções.
	Opção da CLI:
	
	`--option-group-name`
	
	Parâmetro da API do RDS:
	
	`OptionGroupName`
		Se você optar por aplicar a alteração imediatamente, ela ocorrerá imediatamente.
		
		Se você não optar por aplicar a alteração imediatamente, ela ocorrerá durante a próxima janela de manutenção.
			Não ocorre uma inatividade durante esta alteração. Uma exceção é adicionar o plug-in de auditoria do MariaDB a uma instância de banco de dados do RDS para MariaDB ou do RDS para MySQL, o que pode causar uma interrupção.	Todos os mecanismos de banco de dados

Configuração e descrição do console	Opção da CLI e parâmetro da API do RDS	Quando a alteração ocorre	Observações sobre tempo de inatividade	Mecanismos de banco de dados com suporte

Performance Insights
 	
Selecione Enable Performance Insights (Habilitar Performance Insights) para monitorar a carga da instância de banco de dados do para que você possa analisar e solucionar problemas relacionados à performance do banco de dados.
 	
O Performance Insights não está disponível para algumas versões de mecanismo de banco de dados e classes de instância de banco de dados. A seção Performance Insights não será exibida no console se não estiver disponível para sua instância de banco de dados.
 	
Para obter mais informações, consulte Monitorar a carga de banco de dados com o Performance Insights no Amazon RDS e O mecanismo de banco de dados do Amazon RDS, a região e a classe de instância são
	Opção da CLI:
	
	`--enable-performance-insights| --no-enable-performance-insights`
	
	Parâmetro da API do RDS:
	
	`EnablePerformanceInsights`
		A alteração ocorre imediatamente. Essa configuração ignora a configuração para aplicar imediatamente.	Não ocorre uma inatividade durante esta alteração.	Todos, exceto o Db2

Configuração e descrição do console	Opção da CLI e parâmetro da API do RDS	Quando a alteração ocorre	Observações sobre tempo de inatividade	Mecanismos de banco de dados com suporte
compatíveis com o Performance Insights.				

Performance Insights AWS KMS key
 	
O identificador de chave do AWS KMS para que a AWS KMS key criptografe dados do Performance Insights. O identificador da chave é o nome do recurso da Amazon (ARN), o identificador de chave do AWS KMS ou o alias da chave para a chave do KMS.
 	
Para obter mais informações, consulteAtivar e desativar o Insights de Performance do Amazon RDS
	Opção da CLI:
	
	`--performance-insights-kms-key-id`
	
	Parâmetro da API do RDS:
	
	`PerformanceInsightsKMSKeyId`
		A alteração ocorre imediatamente. Essa configuração ignora a configuração para aplicar imediatamente.	Não ocorre uma inatividade durante esta alteração.	Todos, exceto o Db2

Configuração e descrição do console	Opção da CLI e parâmetro da API do RDS	Quando a alteração ocorre	Observações sobre tempo de inatividade	Mecanismos de banco de dados com suporte

Performance Insights retention period (Período de retenção do Performance Insights)
 	
O período de tempo, em dias, de retenção dos dados do Performance Insights. A configuração de retenção no nível gratuito é Default (7 days) (Padrão (7 dias)). Para reter seus dados de performance por mais tempo, especifique entre 1 e 24 meses. Para obter mais informações sobre os períodos de retenção, consulte Preços e retenção de dados para o Performance Insights.
 	
Para obter mais informações, consulte Ativar e desativar o Insights de Performance do Amazon RDS.
	Opção da CLI:
	
	`--performance-insights-retention-period`
	
	Parâmetro da API do RDS:
	
	`PerformanceInsightsRetentionPeriod`
		A alteração ocorre imediatamente. Essa configuração ignora a configuração para aplicar imediatamente.	Não ocorre uma inatividade durante esta alteração.	Todos, exceto o Db2

Configuração e descrição do console	Opção da CLI e parâmetro da API do RDS	Quando a alteração ocorre	Observações sobre tempo de inatividade	Mecanismos de banco de dados com suporte

Processor features (Recursos do processador)
 	
O número de núcleos de CPU e o número de tópicos por núcleo para a classe da instância de banco de dados.
 	
Para obter mais informações, consulte Configurar o processador para uma classe de instância de banco de dados no RDS para Oracle.
	Opção da CLI:
	
	`--processor-features` e `--use-default-processor-features` | `--no-use-default-processor-features`
	
	Parâmetro da API do RDS:
	
	`ProcessorFeatures` e `UseDefaultProcessorFeatures`
		Se você optar por aplicar a alteração imediatamente, ela ocorrerá imediatamente.
		
		Se você não optar por aplicar a alteração imediatamente, ela ocorrerá durante a próxima janela de manutenção.
			Ocorre uma inatividade durante esta alteração.	Somente Oracle

Configuração e descrição do console	Opção da CLI e parâmetro da API do RDS	Quando a alteração ocorre	Observações sobre tempo de inatividade	Mecanismos de banco de dados com suporte

IOPS provisionadas
 	
As IOPS provisionadas (operações de E/S por segundo) para a instância de banco de dados. Essa configuração estará disponível somente se você escolher uma das seguintes opções para Storage type (Tipo de armazenamento):
 	
• General purpose SSD (gp3) (SSD de uso geral (gp3))
• Provisioned IOPS SSD (io1) (SSD de IOPS provisionadas (io1))
• SSD de IOPS provisionadas (io2)
 	
Para obter mais informações, consulte the section called “Armazenamento de IOPS provisionadas” e the section called “Armazenamento gp3 (recomendado)”.
	Opção da CLI:
	
	`--iops`
	
	Parâmetro da API do RDS:
	
	`Iops`
		Se você optar por aplicar a alteração imediatamente, ela ocorrerá imediatamente.
		
		Se você não optar por aplicar a alteração imediatamente, ela ocorrerá durante a próxima janela de manutenção.
			Não ocorre uma inatividade durante esta alteração.	Todos os mecanismos de banco de dados

Configuração e descrição do console	Opção da CLI e parâmetro da API do RDS	Quando a alteração ocorre	Observações sobre tempo de inatividade	Mecanismos de banco de dados com suporte

Acesso público
 	
Publicly accessible (Acessível publicamente) para fornecer à instância de banco de dados um endereço IP público, o que significa que ela é acessível fora da VPC. Para ser acessível publicamente, a instância de banco de dados também deve estar em uma sub-rede pública na VPC.
 	
Not publicly accessible (Não acessível publicamente) para tornar a instância de banco de dados acessível somente de dentro da VPC.
 	
Para obter mais informações, consulte Ocultar uma instância de banco de dados em uma VPC da Internet.
 	
Para se conectar a uma instância de banco de dados de fora de sua VPC, a instância de banco de dados deverá estar acessível publicamente. Além disso, o acesso deve ser concedido
	Opção da CLI:
	
	`--publicly-accessible | --no-publicly-accessible`
	
	Parâmetro da API do RDS:
	
	`PubliclyAccessible`
		A alteração ocorre imediatamente. Essa configuração ignora a configuração para aplicar imediatamente.	Não ocorre uma inatividade durante esta alteração.	Todos os mecanismos de banco de dados

Configuração e descrição do console	Opção da CLI e parâmetro da API do RDS	Quando a alteração ocorre	Observações sobre tempo de inatividade	Mecanismos de banco de dados com suporte
usando as regras de entrada do grupo de segurança da instância de banco de dados. Além disso, outros requisitos devem ser atendidos. Para obter mais informações, consulte Não é possível conectar-se à instância de banco de dados do Amazon RDS. Se sua instância de banco de dados não estiver acessível ao público, também será possível usar uma conexão AWS Site-to-Site VPN ou uma conexão do AWS Direct Connect para acessá-la de uma rede privada. Para obter mais informações, consulte Privacidade do tráfego entre redes.				

Configuração e descrição do console	Opção da CLI e parâmetro da API do RDS	Quando a alteração ocorre	Observações sobre tempo de inatividade	Mecanismos de banco de dados com suporte

Grupo de segurança
 	
O grupo de segurança da VPC a ser associado à instância de banco de dados.
 	
Para obter mais informações, consulte Controlar acesso com grupos de segurança.
	Opção da CLI:
	
	`--vpc-security-group-ids`
	
	Parâmetro da API do RDS:
	
	`VpcSecurityGroupIds`
		A alteração é aplicada de forma assíncrona logo que possível. Essa configuração ignora a configuração para aplicar imediatamente.	Não ocorre uma inatividade durante esta alteração.	Todos os mecanismos de banco de dados

Configuração e descrição do console	Opção da CLI e parâmetro da API do RDS	Quando a alteração ocorre	Observações sobre tempo de inatividade	Mecanismos de banco de dados com suporte

Storage autoscaling (Autoscaling do armazenamento)
 	
Enable storage autoscaling (Habilitar o autoscaling do armazenamento) para permitir que o Amazon RDS aumente automaticamente o armazenamento quando necessário para evitar que a instância de banco de dados seja executada sem espaço de armazenamento.
 	
Use Maximum storage threshold (Limite máximo de armazenamento) para definir o limite máximo para o Amazon RDS aumentar automaticamente o armazenamento da instância de banco de dados. O padrão é de 1.000 GiB.
 	
Para obter mais informações, consulte Gerenciar a capacidade automaticamente com a escalabilidade automática de armazenamento do Amazon RDS.
	Opção da CLI:
	
	`--max-allocated-storage`
	
	Parâmetro da API do RDS:
	
	`MaxAllocatedStorage`
		A alteração ocorre imediatamente. Essa configuração ignora a configuração para aplicar imediatamente.	Não ocorre uma inatividade durante esta alteração.	Todos os mecanismos de banco de dados

Configuração e descrição do console	Opção da CLI e parâmetro da API do RDS	Quando a alteração ocorre	Observações sobre tempo de inatividade	Mecanismos de banco de dados com suporte

Storage throughput (Throughput de armazenamento)
 	
O novo valor do throughput de armazenamento da instância de banco de dados. Essa configuração estará disponível somente se você escolher General purpose SSD (gp3) (SSD de uso geral (gp3)) para Storage type (Tipo de armazenamento).
 	
Para obter mais informações, consulte the section called “Armazenamento gp3 (recomendado)”.
	Opção da CLI:
	
	`--storage-throughput`
	
	Parâmetro da API do RDS:
	
	`StorageThroughput`
		Se você optar por aplicar a alteração imediatamente, ela ocorrerá imediatamente.
		
		Se você não optar por aplicar a alteração imediatamente, ela ocorrerá durante a próxima janela de manutenção.
			Não ocorre uma inatividade durante esta alteração.	Todos os mecanismos de banco de dados

Configuração e descrição do console	Opção da CLI e parâmetro da API do RDS	Quando a alteração ocorre	Observações sobre tempo de inatividade	Mecanismos de banco de dados com suporte

Tipo de armazenamento
 	
O tipo de armazenamento que você deseja usar.
 	
Se você selecionar General Purpose SSD (gp3) (SSD de uso geral (gp3)), poderá provisionar Provisioned IOPS (IOPS provisionadas) adicionais e Storage throughput (Throughput de armazenamento) em Advanced settings (Configurações avançadas).
 	
Se você selecionar SSD de IOPS provisionadas (io1) ou SSD de IOPS provisionadas (io2), insira o valor IOPS provisionadas.
 	
Assim que o Amazon RDS começar a modificar sua instância de banco de dados para alterar o tamanho ou o tipo de armazenamento, não será possível enviar outra solicitação para alterar o tamanho ou o tipo de armazenamento por 6 horas.
	Opção da CLI:
	
	`--storage-type`
	
	Parâmetro da API do RDS:
	
	`StorageType`
		Se você optar por aplicar a alteração imediatamente, ela ocorrerá imediatamente.
		
		Se você não optar por aplicar a alteração imediatamente, ela ocorrerá durante a próxima janela de manutenção.
			As seguintes alterações resultam em uma breve inatividade durante a inicialização do processo. Depois disso, você poderá usar seu banco de dados normalmente enquanto a alteração ocorre.
			
			• De General Purpose (SSD) (Uso geral (SSD)) ou Provisioned IOPS (SSD) (IOPS provisionadas (SSD)) para Magnetic (Magnético).
				Todos os mecanismos de banco de dados

Configuração e descrição do console	Opção da CLI e parâmetro da API do RDS	Quando a alteração ocorre	Observações sobre tempo de inatividade	Mecanismos de banco de dados com suporte
Para obter mais informações, consulte Tipos de armazenamento do Amazon RDS.			• De Magnetic (Magnético) para General Purpose (SSD)(Uso geral (SSD)) ou Provisioned IOPS (SSD) (IOPS provisionadas (SSD)).	

Grupo de sub-redes de banco de dados
 	
O grupo de sub-redes de banco de dados para a instância de banco de dados. Você pode usar essa configuração para mover sua instância de banco de dados para uma VPC diferente.
 	
Para obter mais informações, consulte Amazon VPC e Amazon RDS.
	Opção da CLI:
	
	`--db-subnet-group-name`
	
	Parâmetro da API do RDS:
	
	`DBSubnetGroupName`
		Se você optar por aplicar a alteração imediatamente, ela ocorrerá imediatamente.
		
		Se você não optar por aplicar a alteração imediatamente, ela ocorrerá durante a próxima janela de manutenção.
			Ocorre uma inatividade durante esta alteração.	Todos os mecanismos de banco de dados

Manutenção de uma instância de banco de dados

Periodicamente, o Amazon RDS realiza a manutenção em seus recursos.

Tópicos

- Visão geral das atualizações de manutenção da instância de banco de dados
- Visualizar atualizações de manutenção pendentes
- Aplicar atualizações para uma instância de banco de dados
- Manutenção para implantações multi-AZ
- A janela de manutenção do Amazon RDS
- Ajustar a janela de manutenção da instância de banco de dados preferencial
- Atualizações do sistema operacional para instâncias de banco de dados do RDS

Visão geral das atualizações de manutenção da instância de banco de dados

A manutenção geralmente envolve atualizações dos seguintes atributos na instância de banco de dados:

- Hardware subjacente
- Sistema operacional subjacente
- Versão do mecanismo de banco de dados

As atualizações no sistema operacional geralmente ocorrem para problemas de segurança. Recomendamos que você os faça o quanto antes. Para ter mais informações sobre os sistemas operacionais compatíveis, consulte Aplicar atualizações para uma instância de banco de dados.

Tópicos

- Recursos off-line durante as atualizações de manutenção
- Modificações adiadas da instância de banco de dados
- Consistência eventual da API DescribePendingMaintenanceActions

Recursos off-line durante as atualizações de manutenção

Alguns itens de manutenção exigem que o Amazon RDS coloque a instância de banco de dados off-line por um curto período. Entre os itens de manutenção que exigem um recurso esteja offline estão sistema operacional obrigatório ou patches de banco de dados. A aplicação obrigatória de patches é automaticamente programada somente para patches relacionados à segurança e à confiabilidade da instância. Essa correção ocorre com pouca frequência, normalmente uma vez a cada poucos meses. Raramente requer mais do que uma fração de sua janela de manutenção.

Modificações adiadas da instância de banco de dados

As modificações feitas na instância de banco de dados adiadas que você optou por não aplicar imediatamente são aplicadas durante a janela de manutenção. Por exemplo, convém optar por alterar a classe da instância de banco de dados ou o grupo de parâmetros durante a janela de manutenção. Essas modificações especificadas usando a configuração pending reboot (reinicialização pendente) não aparecem na lista Pending maintenance (Manutenção pendente). Para mais informações sobre a modificação de uma instância de banco de dados , consulte Modificar uma instância de banco de dados do Amazon RDS.

Para ver as modificações pendentes para a próxima janela de manutenção, use o comando describe-db-instances da AWS CLI e verifique o campo `PendingModifiedValues`.

Consistência eventual da API DescribePendingMaintenanceActions

A API `DescribePendingMaintenanceActions` do Amazon RDS segue o modelo de consistência eventual. Isso significa que o resultado do comando `DescribePendingMaintenanceActions` poderá não ser imediatamente visível para todos os comandos subsequentes do RDS. Lembre-se disso ao usar `DescribePendingMaintenanceActions` imediatamente após usar um comando de API anterior.

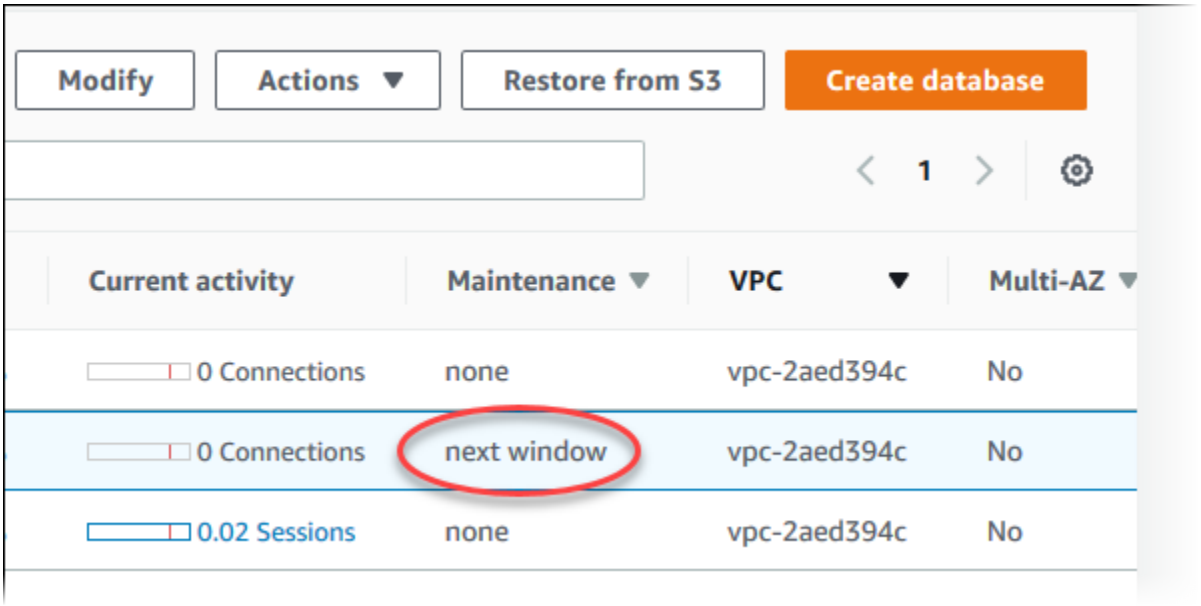
A consistência eventual pode afetar a maneira como você gerencia suas atualizações de manutenção. Por exemplo, se você executar o comando `ApplyPendingMaintenanceActions` para atualizar a versão do mecanismo de banco de dados para uma instância de banco de dados, ela ficará visível para `DescribePendingMaintenanceActions`. Nesse cenário, `DescribePendingMaintenanceActions` pode mostrar que a ação de manutenção não foi aplicada, embora tenha sido.

Para gerenciar a consistência eventual, é possível fazer o seguinte:

- Confirme o estado da instância de banco de dados antes de executar um comando para modificá-lo. Execute o comando `DescribePendingMaintenanceActions` apropriado usando um algoritmo de recuo exponencial para garantir tempo suficiente para que o comando anterior se propague pelo sistema. Para fazer isso, execute o comando `DescribePendingMaintenanceActions` repetidamente, começando com alguns segundos de espera e aumentando gradualmente até cinco minutos de espera.
- Adicione o tempo de espera entre os comandos subsequentes, mesmo que um comando `DescribePendingMaintenanceActions` exiba uma resposta precisa. Aplique um algoritmo de recuo exponencial começando com alguns segundos de espera e aumente gradualmente até cerca de cinco minutos de espera.

Visualizar atualizações de manutenção pendentes

Veja se uma atualização de manutenção está disponível para sua instância de banco de dados utilizando o console do RDS, a AWS CLI ou a API do RDS. Se estiver disponível, uma atualização será indicada na coluna Maintenance (Manutenção) da instância de banco de dados no console do Amazon RDS, conforme mostrado a seguir.



Se nenhuma atualização de manutenção estiver disponível para uma instância de banco de dados, o valor da coluna será none.

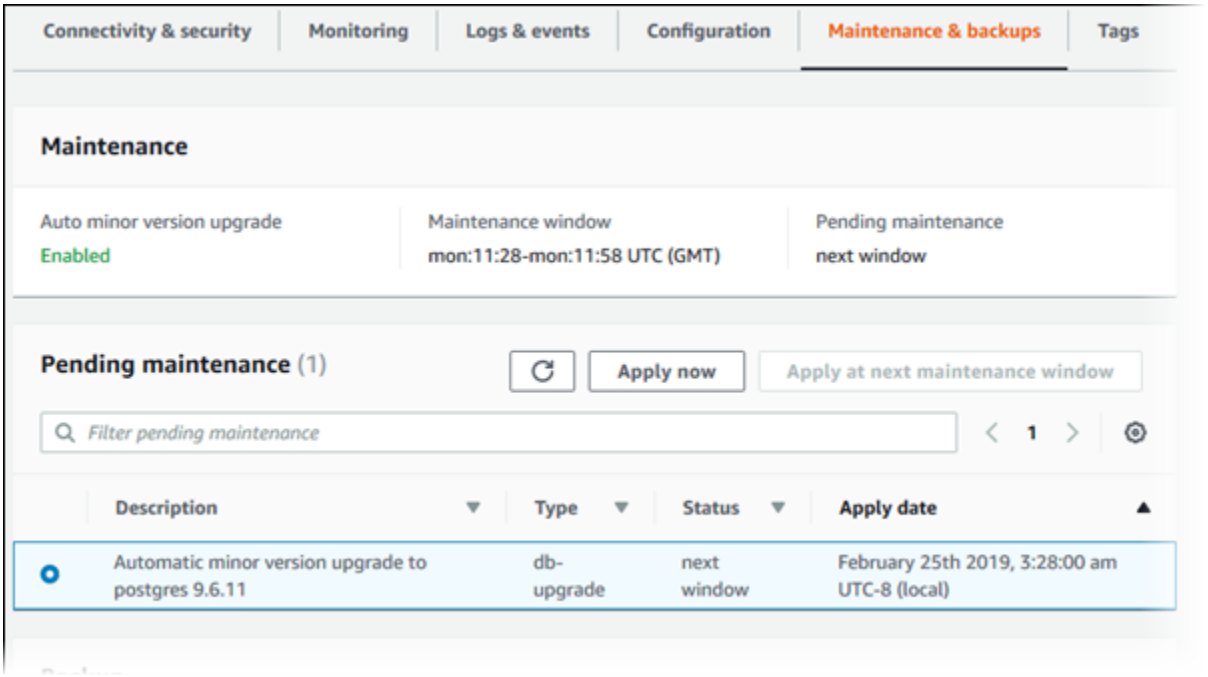
Se uma atualização de manutenção estiver disponível para uma instância de banco de dados, os seguintes valores de coluna serão possíveis:

- obrigatório – a ação de manutenção será aplicada ao recurso e não pode ser adiada indefinidamente.
- available (disponível) – a ação de manutenção está disponível, mas não será aplicada automaticamente ao recurso. Você pode aplicá-la manualmente.
- next window (próxima janela) – a ação de manutenção será aplicada ao recurso durante a próxima janela de manutenção.
- In progress (Em andamento) – a ação de manutenção está no processo de ser aplicado ao recurso.

Se uma atualização estiver disponível, você poderá seguir uma destas ações:

- Se o valor de manutenção for next window (próxima janela), adie os itens de manutenção escolhendo Defer upgrade (Adiar atualização) em Actions (Ações). Não é possível adiar uma ação de manutenção que já tiver sido iniciada.
- Aplicar os itens de manutenção imediatamente.
- Agendar os itens de manutenção para iniciar durante a próxima janela de manutenção.
- Não tome nenhuma ação.

Para executar uma ação, escolha a instância de banco de dados para mostrar seus detalhes e escolha Maintenance & backups (Manutenção e backups). Os itens de manutenção pendentes são exibidos.



A janela de manutenção determina quando as operações pendentes começam, mas não limita o tempo total de execução dessas operações. Não há garantia de que as operações de manutenção terminem antes de a janela de manutenção se encerrar, podendo continuar além do tempo de encerramento especificado. Para ter mais informações, consulte A janela de manutenção do Amazon RDS.

Você pode ver se uma atualização de manutenção está disponível para sua instância de banco de dados executando o comando `describe-pending-maintenance-actions` da AWS CLI.

Para ter informações sobre a aplicação de atualizações de manutenção, consulte Aplicar atualizações para uma instância de banco de dados.

Aplicar atualizações para uma instância de banco de dados

Com o Amazon RDS, você pode escolher quando aplicar operações de manutenção. Você pode decidir quando o Amazon RDS aplicará atualizações usando o AWS Management Console, a AWS CLI ou a API do RDS.

> **Note**
>
> Para o RDS for SQL Server, uma atualização do sistema operacional subjacente pode ser aplicada interrompendo e iniciando sua instância de banco de dados ou escalando sua classe de instância de banco de dados para cima e depois para baixo novamente.

Console

Para gerenciar uma atualização de uma instância de banco de dados

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Databases (Bancos de dados).
3. Escolha a instância de banco de dados que exige uma atualização obrigatória.
4. Em Actions (Ações), escolha uma das seguinte opções:
   - Aplicar patch agora
   - Aplicar patch na próxima janela

   > **Note**
   >
   > Se escolher Aplicar patch na próxima janela e depois quiser atrasar a atualização do sistema operacional, você poderá escolher Adiar atualização. Não é possível adiar uma ação de manutenção que já tiver sido iniciada.
   > Para cancelar uma ação de manutenção, modifique a instância de banco de dados e desative Auto minor version upgrade (Atualização automática da versão secundária).

AWS CLI

Para aplicar uma atualização pendente a uma instância de banco de dados, use o comando da AWS CLI apply-pending-maintenance-action.

Example

Para Linux, macOS ou Unix:

```
aws rds apply-pending-maintenance-action \
    --resource-identifier arn:aws:rds:us-west-2:001234567890:db:mysql-db \
    --apply-action system-update \
    --opt-in-type immediate
```

Para Windows:

```
aws rds apply-pending-maintenance-action ^
    --resource-identifier arn:aws:rds:us-west-2:001234567890:db:mysql-db ^
    --apply-action system-update ^
    --opt-in-type immediate
```

> **Note**
>
> Para adiar uma ação de manutenção, especifique `undo-opt-in` para `--opt-in-type`. Não será possível especificar `undo-opt-in` para `--opt-in-type` se a ação de manutenção já tiver sido iniciada.
> Para cancelar uma ação de manutenção, execute o comando modify-db-instance da AWS CLI e especifique `--no-auto-minor-version-upgrade`.

Para retornar uma lista de recursos que tenham pelo menos uma atualização pendente, use o comando describe-pending-maintenance-actions da AWS CLI.

Example

Para Linux, macOS ou Unix:

```
aws rds describe-pending-maintenance-actions \
    --resource-identifier arn:aws:rds:us-west-2:001234567890:db:mysql-db
```

Para Windows:

```
aws rds describe-pending-maintenance-actions ^
    --resource-identifier arn:aws:rds:us-west-2:001234567890:db:mysql-db
```

Você também pode retornar uma lista de recursos para uma instância de banco de dados especificando o parâmetro `--filters` do comando `describe-pending-maintenance-actions` da AWS CLI. O formato do comando `--filters` é `Name=`*`filter-name`*`,Value=`*`resource-id`*`,...`.

Os valores a seguir são os valores aceitos para o parâmetro `Name` de um filtro:

- `db-instance-id` – aceita uma lista de identificadores de instância de banco de dados ou nomes de recurso da Amazon (ARNs). A lista retornada inclui apenas ações de manutenção pendentes para as instâncias de bancos de dados identificadas por esses identificadores ou ARNs.
- `db-cluster-id` – aceita uma lista de identificadores de cluster de banco de dados ou ARNs para o Amazon Aurora. A lista retornada inclui apenas ações de manutenção pendentes para os clusters de bancos de dados identificados por esses identificadores ou ARNs.

Por exemplo, o exemplo a seguir retorna as ações de manutenção pendentes para as instâncias de banco de dados `sample-instance1` e `sample-instance2`.

Example

Para Linux, macOS ou Unix:

```
aws rds describe-pending-maintenance-actions \
 --filters Name=db-instance-id,Values=sample-instance1,sample-instance2
```

Para Windows:

```
aws rds describe-pending-maintenance-actions ^
 --filters Name=db-instance-id,Values=sample-instance1,sample-instance2
```

API do RDS

Para aplicar uma atualização a uma instância de banco de dados, chame a operação `ApplyPendingMaintenanceAction` da API do Amazon RDS.

Para retornar uma lista de recursos que tenham pelo menos uma atualização pendente, chame a operação `DescribePendingMaintenanceActions` da API do Amazon RDS.

Manutenção para implantações multi-AZ

A execução de uma instância de banco de dados como uma implantação multi-AZ pode reduzir ainda mais o impacto de um evento de manutenção. Esse resultado ocorre porque o Amazon RDS aplica atualizações do sistema operacional seguindo estas etapas:

1. Execute a manutenção no modo de espera.
2. Promova o modo de espera para o principal.
3. Execute manutenção no principal antigo, que se tornará o novo em modo de espera.

Se você atualiza o mecanismo de banco de dados de sua instância de banco de dados em uma implantação multi-AZ, o Amazon RDS modifica as instâncias de banco de dados primária e secundária ao mesmo tempo. Nesse caso, as instâncias de banco de dados primária e secundária na implantação multi-AZ não estão disponíveis durante a atualização. Essa operação causa tempo de inatividade até que a atualização seja concluída. A duração do tempo de inatividade varia de acordo com o tamanho de sua instância de banco de dados.

Se houver patches subjacentes do sistema operacional que precisem ser aplicados, será necessário um pequeno failover multi-AZ para aplicar os patches à instância de banco de dados primária. Esse failover geralmente dura menos de um minuto.

Se a instância de banco de dados executa o RDS para MySQL, o RDS para PostgreSQL ou o RDS para MariaDB, é possível minimizar o tempo de inatividade necessário para um upgrade usando uma implantação azul/verde. Para ter mais informações, consulte Usar implantações azul/verde do Amazon RDS para atualizações de banco de dados. Se você atualizar uma instância de banco de dados do RDS para SQL Server ou do RDS Custom para SQL Server em uma implantação multi-AZ, o Amazon RDS realizará atualizações contínuas, de modo que você enfrentará apenas uma interrupção durante um failover. Para ter mais informações, consulte Considerações Multi-AZ.

Se sua instância de banco de dados executa o RDS for SQL Server em uma implantação Multi-AZ, você poderá aplicar uma atualização ao sistema operacional subjacente usando um dos seguintes métodos:

- Modifique a classe da instância de banco de dados para um tamanho diferente e depois modifique-a de volta ao tamanho original.
- Aumente o tamanho da instância de banco de dados e depois reduza para o tamanho original.
- Modifique a instância de banco de dados de Multi-AZ para Single-AZ, pare e inicie a instância de banco de dados e, em seguida, altere a instância de volta para Multi-AZ.

Para ter mais informações sobre implantações Multi-AZ, consulte Configurar e gerenciar uma implantação multi-AZ para o Amazon RDS.

A janela de manutenção do Amazon RDS

As janelas de manutenção são um intervalo de tempo semanal durante o qual todas as alterações do sistema são aplicadas. Cada instância de banco de dados tem uma janela de manutenção semanal. A janela de manutenção é uma oportunidade de controlar quando as modificações e a aplicação de patches de software ocorrem. Para ter mais informações sobre como ajustar a janela de manutenção, consulte Ajustar a janela de manutenção da instância de banco de dados preferencial.

O RDS consome alguns dos recursos em sua instância de banco de dados enquanto a manutenção é aplicada. Você poderá observar um impacto mínimo na performance. Quanto a uma instância de banco de dados, em raras ocasiões, pode ser necessário realizar um failover Multi-AZ para concluir uma atualização de manutenção.

Se um evento de manutenção estiver programado para determinada semana, ele será iniciado durante a janela de manutenção de 30 minutos que você identificar. A maioria dos eventos de manutenção também é concluída durante a janela de manutenção de 30 minutos, embora os eventos de manutenção mais longos possam levar mais de 30 minutos para serem concluídos. A janela de manutenção é pausada quando a instância de banco de dados é interrompida .

A janela de manutenção de 30 minutos é selecionada aleatoriamente de um bloco de tempo de 8 horas por região. Se você não especificar uma janela de manutenção ao criar a instância de banco de dados, o RDS atribuirá uma janela de manutenção de 30 minutos em um dia da semana selecionado aleatoriamente.

A seguir, você pode encontrar os blocos de tempo de cada região dos quais as janelas de manutenção padrão são atribuídas.

Nome da região	Região	Bloco de hora
US East (Ohio)	us-east-2	De 03:00 a 11:00 UTC
US East (N. Virginia)	us-east-1	De 03:00 a 11:00 UTC
US West (N. Californi a)	us-west-1	De 06:00 a 14:00 UTC
US West (Oregon)	us-west-2	De 06:00 a 14:00 UTC

Nome da região	Região	Bloco de hora
Africa (Cape Town)	af-south-1	De 03:00 a 11:00 UTC
Asia Pacific (Hong Kong)	ap-east-1	De 06:00 a 14:00 UTC
Ásia-Pacífico (Hyderabad)	ap-south-2	06h30 a 14h30 UTC
Ásia-Pacífico (Jacarta)	ap-southeast-3	Das 08h às 16h UTC
Ásia-Pacífico (Melbourne)	ap-southeast-4	Das 11h às 19h UTC
Ásia-Pacífico (Mumbai)	ap-south-1	De 06:00 a 14:00 UTC
Asia Pacific (Osaka)	ap-northeast-3	De 22:00 a 23:59 UTC
Asia Pacific (Seoul)	ap-northeast-2	De 13:00 a 21:00 UTC
Ásia-Pacífico (Singapura)	ap-southeast-1	De 14:00 a 22:00 UTC
Asia Pacific (Sydney)	ap-southeast-2	De 12:00 a 20:00 UTC
Asia Pacific (Tokyo)	ap-northeast-1	De 13:00 a 21:00 UTC
Canada (Central)	ca-central-1	De 03:00 a 11:00 UTC
Oeste do Canadá (Calgary)	ca-west-1	Das 18h às 2h (UTC)
China (Pequim)	cn-north-1	De 06:00 a 14:00 UTC
China (Ningxia)	cn-northwest-1	De 06:00 a 14:00 UTC
Europe (Frankfurt)	eu-central-1	De 21:00 a 05:00 UTC

Nome da região	Região	Bloco de hora
Europe (Ireland)	eu-west-1	De 22:00 a 06:00 UTC
Europe (London)	eu-west-2	De 22:00 a 06:00 UTC
Europa (Milão)	eu-south-1	De 02:00 a 10:00 UTC
Europa (Paris)	eu-west-3	De 23:59 a 07:29 UTC
Europa (Espanha)	eu-south-2	De 02:00 a 10:00 UTC
Europe (Stockholm)	eu-north-1	De 23:00 a 07:00 UTC
Europa (Zurique)	eu-central-2	De 02:00 a 10:00 UTC
Israel (Tel Aviv)	il-central-1	De 03:00 a 11:00 UTC
Oriente Médio (Barém)	me-south-1	De 06:00 a 14:00 UTC
Oriente Médio (Emirados Árabes Unidos)	me-central-1	Das 5h às 13h UTC
América do Sul (São Paulo)	sa-east-1	De 00:00 a 08:00 UTC
AWS GovCloud (Leste dos EUA)	us-gov-east-1	De 17:00 a 01:00 UTC
AWS GovCloud (Oeste dos EUA)	us-gov-west-1	De 06:00 a 14:00 UTC

Ajustar a janela de manutenção da instância de banco de dados preferencial

A janela de manutenção deve ser definida no horário de menor utilização e, portanto, talvez precise ser modificada de vez em quando. Seu cluster de banco de dados estará indisponível durante esse

tempo somente se alterações do sistema, como uma alteração na classe de instância de banco de dados, estiverem sendo aplicadas e exigirem uma interrupção. Sua instância de banco de dados estará indisponível apenas pelo intervalo mínimo de tempo necessário para fazer as alterações necessárias.

No exemplo a seguir, ajuste a janela de manutenção preferencial para uma instância de banco de dados.

Para este exemplo, presume-se que a instância de banco de dados denominada mydbinstance exista e apresente uma janela de manutenção preferencial de "Sun:05:00-Sun:06:00" UTC.

Console

Para ajustar a janela de manutenção preferencial

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Databases (Bancos de dados) e selecione a instância de banco de dados que você deseja modificar.
3. Selecione Modify. A página Modify DB instance (Modificar instância de banco de dados) será exibida.
4. Na seção Maintenance (Manutenção), atualize a janela de manutenção.

   > **Note**
   >
   > A janela de manutenção e a janela de backup da instância de banco de dados não podem se sobrepor. Se você digitar um valor na janela de manutenção que sobreponha a janela de backup, será exibida uma mensagem de erro.

5. Escolha Continue.

   Na página de confirmação, revise suas alterações.
6. Para aplicar as alterações à janela de manutenção imediatamente, selecione Apply immediately (Aplicar imediatamente).
7. Selecione Modificar instância de banco de dados para salvar as alterações.

   Como alternativa, escolha Back (Voltar) para editar suas alterações ou escolha Cancel (Cancelar) para cancelar as alterações.

AWS CLI

Para ajustar a janela de manutenção preferencial, use o comando AWS CLI da `modify-db-instance` com os seguintes parâmetros:

- `--db-instance-identifier`
- `--preferred-maintenance-window`

Example

O exemplo de código a seguir define a janela de manutenção para terças, das 4:00 às 4:30 AM UTC.

Para Linux, macOS ou Unix:

```
aws rds modify-db-instance \
--db-instance-identifier mydbinstance \
--preferred-maintenance-window Tue:04:00-Tue:04:30
```

Para Windows:

```
aws rds modify-db-instance ^
--db-instance-identifier mydbinstance ^
--preferred-maintenance-window Tue:04:00-Tue:04:30
```

API do RDS

Para ajustar a janela de manutenção de sua preferência, use a operação da API `ModifyDBInstance` do Amazon RDS com os seguintes parâmetros:

- `DBInstanceIdentifier`
- `PreferredMaintenanceWindow`

Atualizações do sistema operacional para instâncias de banco de dados do RDS

Ocasionalmente, as instâncias de banco de dados do RDS para Db2, RDS para MariaDB, RDS para MySQL, RDS para PostgreSQL e RDS para Oracle exigem atualizações do sistema operacional. O Amazon RDS faz upgrade do sistema operacional para uma versão mais recente para melhorar o

desempenho do banco de dados e o procedimento de segurança geral dos clientes. Em geral, essas atualizações demoram cerca de dez minutos. As atualizações do sistema operacional não alteram a versão do mecanismo de banco de dados nem a classe de uma instância de banco de dados.

As atualizações do sistema operacional podem ser opcionais ou obrigatórias:

- Uma atualização opcional pode ser aplicada a qualquer momento. Embora essas atualizações sejam opcionais, recomendamos que você as aplique periodicamente para manter sua frota do RDS atualizada. O RDS não aplica essas atualizações automaticamente.

  Para ser notificado quando um novo patch opcional do sistema operacional estiver disponível, você poderá assinar o RDS-EVENT-0230 na categoria de evento de aplicação de patch de segurança. Para obter informações sobre como se inscrever em eventos do RDS, consulte Inscrever-se em notificações de eventos do Amazon RDS.

  > **Note**
  >
  > `RDS-EVENT-0230` não se aplica às atualizações de distribuição do sistema operacional.

  > **Note**
  >
  > Se você recebeu `RDS-EVENT-0230` uma instância de banco de dados RDS para SQL Server, a atualização do sistema operacional não pode ser aplicada usando a ação `apply-pending-maintenance`. Para ter mais informações, consulte Aplicar atualizações para uma instância de banco de dados.

- Uma atualização obrigatória é necessária e tem data de aplicação. Agende sua atualização para antes dessa data de aplicação. Após a data de aplicação especificada, o Amazon RDS atualiza automaticamente o sistema operacional de sua instância de banco de dados para a versão mais recente durante uma das janelas de manutenção atribuídas.

> **Note**
>
> A aplicação de todas as atualizações opcionais e obrigatórias pode ser necessária para cumprir várias obrigações de conformidade. Recomendamos que você aplique todas as atualizações disponibilizadas pelo RDS rotineiramente durante suas janelas de manutenção.

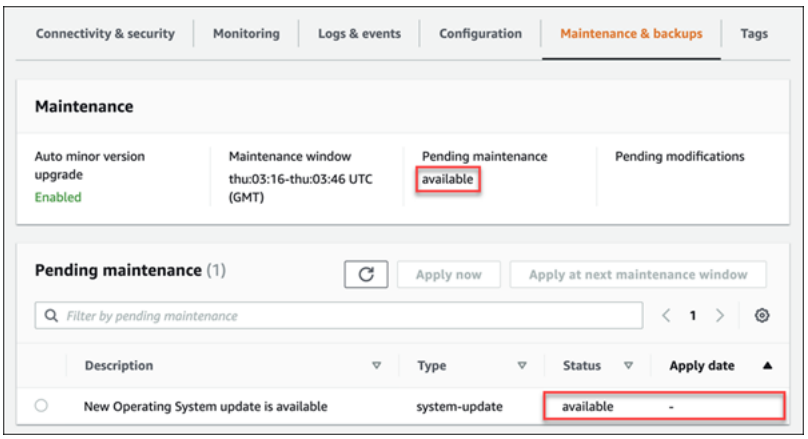
É possível usar o AWS Management Console ou a AWS CLI para obter informações sobre o tipo de atualização do sistema operacional.

Console

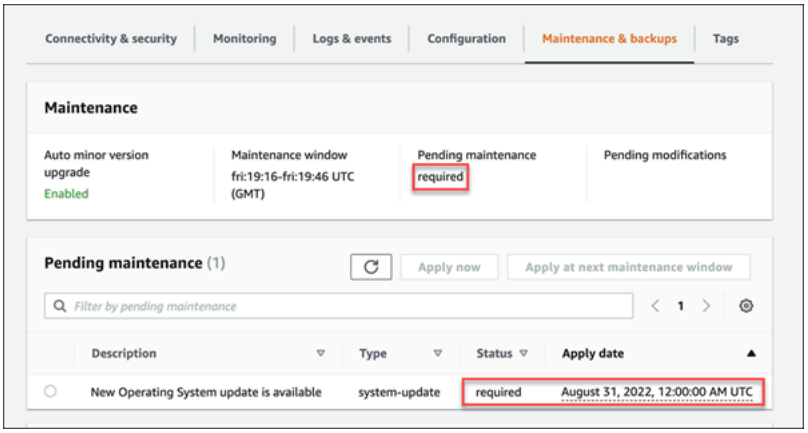
Como obter informações de atualização usando o AWS Management Console

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Databases (Bancos de dados) e, depois, a instância de banco de dados.
3. Escolha Maintenance & backups (Manutenção e backups).
4. Na seção Pending maintenance (Manutenção pendente), encontre a atualização do sistema operacional e confira o valor Status.

No AWS Management Console, uma atualização opcional tem o Status de sua manutenção definido como available (disponível) e não tem uma Apply date (Data de aplicação), conforme mostrado na imagem a seguir.



Uma atualização obrigatória tem o Status de sua manutenção definido como required (obrigatória) e tem uma Apply date (Data de aplicação), conforme mostrado na imagem a seguir.



AWS CLI

Para obter informações de atualização da AWS CLI, use o comando describe-pending-maintenance-actions.

```
aws rds describe-pending-maintenance-actions
```

Uma atualização obrigatória do sistema operacional inclui um valor `AutoAppliedAfterDate` e um valor `CurrentApplyDate`. Uma atualização opcional do sistema operacional não inclui esses valores.

A saída a seguir mostra uma atualização obrigatória do sistema operacional.

```
{
  "ResourceIdentifier": "arn:aws:rds:us-east-1:123456789012:db:mydb1",
  "PendingMaintenanceActionDetails": [
    {
      "Action": "system-update",
      "AutoAppliedAfterDate": "2022-08-31T00:00:00+00:00",
      "CurrentApplyDate": "2022-08-31T00:00:00+00:00",
      "Description": "New Operating System update is available"
    }
  ]
}
```

A saída a seguir mostra uma atualização opcional do sistema operacional.

```
{
  "ResourceIdentifier": "arn:aws:rds:us-east-1:123456789012:db:mydb2",
  "PendingMaintenanceActionDetails": [
```

```
    {
      "Action": "system-update",
      "Description": "New Operating System update is available"
    }
  ]
}
```

Disponibilidade de atualizações do sistema operacional

As atualizações do sistema operacional são específicas da versão do mecanismo de banco de dados e da classe de instância de banco de dados. Portanto, as instâncias de banco de dados recebem ou exigem atualizações em momentos diferentes. Quando uma atualização do sistema operacional estiver disponível para sua instância de banco de dados com base na versão do mecanismo e na classe de instância de banco de dados, essa atualização aparecerá no console. Ela também pode ser visualizada executando o comando describe-pending-maintenance-actions AWS CLI ou chamando a operação DescribePendingMaintenanceActions da API do RDS. Se houver uma atualização disponível para sua instância, você poderá atualizar o sistema operacional seguindo as instruções em Aplicar atualizações para uma instância de banco de dados.

Atualizar a versão de mecanismo de uma instância de banco de dados

O Amazon RDS fornece versões mais recentes de cada mecanismo de banco de dados com suporte, para que você possa manter em estado atualizado sua instância de banco de dados. Entre as versões mais novas podem estar correções de bugs, melhorias de segurança e outros aprimoramentos para o mecanismo de banco de dados. Quando o Amazon RDS oferecer suporte a uma nova versão de um mecanismo de banco de dados, escolha como e quando fazer upgrade de instâncias de banco de dados.

Há dois tipos de atualizações: atualizações de versão principal e atualizações de versão secundária. Em geral, uma atualização da versão do mecanismo principal pode apresentar alterações não compatíveis com aplicativos existentes. Por outro lado, uma atualização da versão secundária inclui apenas alterações compatíveis com versões anteriores dos aplicativos existentes.

Com relação a clusters de banco de dados multi-AZ, as atualizações de versão principal são comportadas apenas pelo RDS para PostgreSQL. As atualizações de versão secundária são comportadas por todos os mecanismos compatíveis com clusters de banco de dados multi-AZ. Para ter mais informações, consulte the section called “Atualização de um cluster de banco de dados multi-AZ”.

A sequência de numeração da versão é específica de cada mecanismo de banco de dados. Por exemplo, RDS for MySQL 5.7 e 8.0 são versões de mecanismo principais e a atualização de qualquer versão 5.7 para qualquer versão 8.0 é uma atualização da versão principal. RDS for MySQL versão 5.7.22 e 5.7.23 são versões secundárias e a atualização da versão 5.7.22 para a 5.7.23 é uma atualização da versão secundária.

> **Important**
>
> Você não pode modificar uma instância de banco de dados quando ela está sendo atualizada. Durante uma atualização, o status da instância de banco de dados é `upgrading`.

Para ter mais informações sobre atualizações de versões principais e secundárias de um mecanismo de banco de dados específico, consulte a seguinte documentação do mecanismo de banco de dados:

- Atualizações do mecanismo de banco de dados do MariaDB

- Atualizações do mecanismo de banco de dados do Microsoft SQL Server
- Atualizações do mecanismo de banco de dados do RDS para MySQL
- Fazer upgrade do mecanismo de banco de dados RDS para Oracle
- Atualizações do mecanismo de banco de dados do RDS para PostgreSQL

Para atualizações de versão principal, modifique manualmente a versão do mecanismo de banco de dados por meio do AWS Management Console, da AWS CLI ou da API do RDS. Nas atualizações de versão secundária, você pode modificar manualmente a versão do mecanismo ou optar por habilitar a opção Upgrade automático de versões secundárias.

> **Note**
>
> As atualizações do mecanismo de banco de dados necessitam de tempo de inatividade. Você pode minimizar o tempo de inatividade necessário para a atualização da instância de banco de dados utilizando uma implantação azul/verde. Para ter mais informações, consulte Usar implantações azul/verde do Amazon RDS para atualizações de banco de dados.

Tópicos

- Atualizar manualmente a versão do mecanismo
- Atualizar automaticamente a versão do mecanismo espelho

Atualizar manualmente a versão do mecanismo

Para atualizar manualmente a versão do mecanismo de uma instância de banco de dados, use o AWS Management Console, a AWS CLI ou a API do RDS.

Console

Para atualizar a versão do mecanismo de uma instância de banco de dados usando o console

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Databases (Bancos de dados) e a instância de banco de dados que você deseja atualizar.

3. Selecione Modify. A página Modify DB instance (Modificar instância de banco de dados) será exibida.
4. Em DB engine version (Versão do mecanismo de banco de dados), selecione a nova versão.
5. Selecione Continue (Continuar) e verifique o resumo de modificações.
6. Decida quando programar o upgrade:

   - Para colocar as alterações na fila de modificações pendentes, escolha Aplicar durante a próxima janela de manutenção programada. Durante a próxima janela de manutenção, o RDS aplica todas as alterações pendentes na fila.
   - Para aplicar as alterações imediatamente, escolha Apply immediately. Escolher essa opção pode causar uma interrupção em alguns casos. Para ter mais informações, consulte Usar a configuração de programação de modificações.
7. Na página de confirmação, revise suas alterações. Se estiverem corretas, escolha Modify DB Instance (Modificar instância de banco de dados) para salvar suas alterações.

   Como alternativa, escolha Back (Voltar) para editar suas alterações ou escolha Cancel (Cancelar) para cancelar as alterações.

AWS CLI

Para atualizar a versão do mecanismo de uma instância de banco de dados, use o comando modify-db-instance da CLI. Especifique os seguintes parâmetros:

- `--db-instance-identifier` – o nome da instância de banco de dados.
- `--engine-version` – o número da versão do mecanismo de banco de dados para a qual será feita a atualização.

  Para obter informações sobre versões de mecanismo válidas, use o comando AWS CLI describe-db-engine-versions.
- `--allow-major-version-upgrade` – para atualizar a versão principal.
- `--no-apply-immediately` – para aplicar alterações durante a próxima janela de manutenção. Para aplicar as alterações imediatamente, use `--apply-immediately`.

Example

Para Linux, macOS ou Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --engine-version new_version \
    --allow-major-version-upgrade \
    --no-apply-immediately
```

Para Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --engine-version new_version ^
    --allow-major-version-upgrade ^
    --no-apply-immediately
```

API do RDS

Para atualizar a versão do mecanismo de uma instância de banco de dados, use a ação ModifyDBInstance. Especifique os seguintes parâmetros:

- `DBInstanceIdentifier` – o nome da instância de banco de dados. Por exemplo *mydbinstance*.
- `EngineVersion` – o número da versão do mecanismo de banco de dados para a qual será feita a atualização. Para obter informações sobre versões de mecanismo válidas, use a operação DescribeDBEngineVersions.
- `AllowMajorVersionUpgrade` – para permitir ou não uma atualização de versão principal. Para isso, defina o valor como `true`.
- `ApplyImmediately` – se desejar aplicar as alterações imediatamente ou durante a próxima janela de manutenção. Para aplicar as alterações imediatamente, defina o valor como `true`. Para aplicar alterações durante a próxima janela de manutenção, defina o valor como `false`.

Atualizar automaticamente a versão do mecanismo espelho

As atualizações automáticas de versões secundárias atualizam periodicamente o banco de dados para versões recentes do respectivo mecanismo. No entanto, a atualização nem sempre inclui a versão mais recente do mecanismo de banco de dados. Se precisar manter seus bancos de dados em determinadas versões em momentos específicos, recomendamos atualizar manualmente para as versões de banco de dados necessárias de acordo com o cronograma exigido. Em casos de

problemas críticos de segurança ou quando uma versão atinge a data de fim do suporte, o Amazon RDS pode aplicar uma atualização de versão secundária, mesmo que você não tenha habilitado a opção Upgrade automático de versões secundárias. Para ter mais informações, consulte a documentação sobre atualização do seu mecanismo de banco de dados específico.

- Atualizações automáticas da versão secundária para o RDS para PostgreSQL
- Atualizações automáticas da versão secundária do RDS para MySQL
- Atualizações automáticas da versão secundária do RDS para MariaDB
- Atualizações de versões secundárias Oracle
- Atualizações do mecanismo de banco de dados do Microsoft SQL Server
- Db2 em versões do Amazon RDS

Tópicos

- Como atualizações da versão secundária automáticas funcionam
- Como ativar as atualizações automáticas de versões secundárias
- Determinar a disponibilidade das atualizações de manutenção
- Localizar destinos de atualização de versão secundária automática

Como atualizações da versão secundária automáticas funcionam

O destino da atualização é a versão do mecanismo de banco de dados para a qual o Amazon RDS atualiza seu banco de dados. Uma versão secundária do mecanismo é designada como o destino da atualização quando as seguintes condições são atendidas:

- O banco de dados está executando uma versão secundária do mecanismo de banco de dados inferior à versão secundária pretendida do mecanismo.

  Você pode encontrar a versão atual do mecanismo para a instância de banco de dados consultando a guia Configuração da página de detalhes do banco de dados ou executando o comando `describe-db-instances` da CLI.
- O banco de dados tem a atualização automática de versões secundárias habilitada.

O RDS programa as atualizações para serem executadas automaticamente na janela de manutenção. Durante a atualização, o RDS faz o seguinte:

1. Executa uma verificação prévia do sistema para garantir que o banco de dados esteja íntegro e pronto para ser atualizado.
2. Atualiza o mecanismo de banco de dados para a versão secundária do mecanismo pretendida.
3. Executa verificações pós-atualização
4. Marca a atualização do banco de dados como concluída

As atualizações automáticas geram tempo de inatividade. A duração do tempo de inatividade depende de vários fatores, como o tipo de mecanismo de banco de dados e o tamanho do banco de dados.

Como ativar as atualizações automáticas de versões secundárias

Controle se a atualização automática da versão secundária está habilitada para uma instância de banco de dados quando você realiza as seguintes tarefas:

- Criar uma instância de banco de dados
- Modificar uma instância de banco de dados
- Como criar uma réplica de leitura
- Restaurar uma instância de banco de dados a partir de um snapshot
- Restaurar uma instância de banco de dados para uma hora específica
- Importar uma instância de banco de dados do Amazon S3 (para um backup do MySQL no Amazon S3)

Ao realizar essas tarefas, controle se a atualização automática da versão secundária está habilitada para a instância de banco de dados das seguintes maneiras:

- Usando o console, defina a opção Auto minor version upgrade (Atualização automática da versão secundária).
- Usando a AWS CLI, defina a opção `--auto-minor-version-upgrade|--no-auto-minor-version-upgrade`.
- Usando a API do RDS, defina o parâmetro `AutoMinorVersionUpgrade`.

Determinar a disponibilidade das atualizações de manutenção

Para determinar se uma atualização de manutenção, como uma atualização da versão do mecanismo de banco de dados, está disponível para a instância de banco de dados, use o console, a AWS CLI ou a API do RDS. Também atualize manualmente a versão do mecanismo do banco de dados e ajuste a janela de manutenção. Para ter mais informações, consulte Manutenção de uma instância de banco de dados.

Localizar destinos de atualização de versão secundária automática

É possível usar o comando da AWS CLI a seguir para determinar a versão atual secundária de destino de atualização automática para uma versão de mecanismo de banco de dados secundária especificada em determinada Região da AWS. Você pode encontrar os valores `--engine` possíveis para este comando na descrição do parâmetro `Engine` em CreateDBInstance.

Para Linux, macOS ou Unix:

```
aws rds describe-db-engine-versions \
--engine engine \
--engine-version minor-version \
--region region \
--query "DBEngineVersions[*].ValidUpgradeTarget[*].
{AutoUpgrade:AutoUpgrade,EngineVersion:EngineVersion}" \
--output text
```

Para Windows:

```
aws rds describe-db-engine-versions ^
--engine engine ^
--engine-version minor-version ^
--region region ^
--query "DBEngineVersions[*].ValidUpgradeTarget[*].
{AutoUpgrade:AutoUpgrade,EngineVersion:EngineVersion}" ^
--output text
```

Por exemplo, o seguinte comando da AWS CLI determina o destino de upgrade secundário automático para a versão 8.0.11 secundária do MySQL na região da AWS Leste dos EUA (Ohio) (us-east-2).

Para Linux, macOS ou Unix:

```
aws rds describe-db-engine-versions \
--engine mysql \
--engine-version 8.0.11 \
--region us-east-2 \
--query "DBEngineVersions[*].ValidUpgradeTarget[*].
{AutoUpgrade:AutoUpgrade,EngineVersion:EngineVersion}" \
--output table
```

Para Windows:

```
aws rds describe-db-engine-versions ^
--engine mysql ^
--engine-version 8.0.11 ^
--region us-east-2 ^
--query "DBEngineVersions[*].ValidUpgradeTarget[*].
{AutoUpgrade:AutoUpgrade,EngineVersion:EngineVersion}" ^
--output table
```

A saída é semelhante à seguinte.

```
---------------------------------
|    DescribeDBEngineVersions    |
+--------------+-----------------+
|  AutoUpgrade |  EngineVersion  |
+--------------+-----------------+
|  False       |  8.0.15         |
|  False       |  8.0.16         |
|  False       |  8.0.17         |
|  False       |  8.0.19         |
|  False       |  8.0.20         |
|  False       |  8.0.21         |
|  True        |  8.0.23         |
|  False       |  8.0.25         |
+--------------+-----------------+
```

Neste exemplo, o valor de `AutoUpgrade` é `True` para o MySQL versão 8.0.23. Portanto, o destino de upgrade secundário automático é o MySQL versão 8.0.23, que está realçado na saída.

Important

Se você planeja migrar uma instância de banco de dados do RDS para PostgreSQL para um cluster de banco de dados do Aurora PostgreSQL no futuro próximo, recomendamos que você desative as atualizações automáticas de versão secundária para a instância de banco de dados no início da fase de planejamento. A migração para Aurora PostgreSQL pode ser adiada se a versão do RDS para PostgreSQL ainda não for compatível com o Aurora PostgreSQL. Para obter informações sobre Aurora PostgreSQL versões, consulte Versões de mecanismo para Amazon Aurora PostgreSQL.

Renomear uma instância de banco de dados

Você pode renomear uma instância de banco de dados usando o AWS Management Console, o comando `modify-db-instance` da AWS CLI ou a ação `ModifyDBInstance` da API do Amazon RDS. Renomear uma instância de banco de dados pode ter efeitos de longo alcance. Veja a seguir uma lista de considerações antes de renomear uma instância de banco de dados.

- Quando você renomeia uma instância de banco de dados, o endpoint dessa instância é alterado, pois o URL inclui o nome que você atribuiu a ela. Você sempre deve redirecionar o tráfego da URL antiga para a nova.
- Ao renomear uma instância de banco de dados, o nome DNS antigo que era usado pelo cluster é excluído imediatamente, embora ele possa permanecer armazenado em cache por alguns minutos. O novo nome DNS da instância de banco de dados renomeada torna-se efetivo em 10 minutos. A instância de banco de dados renomeada não ficará disponível até que o novo nome entre em vigor.
- Você não pode usar um nome de instância de banco de dados existente ao renomear uma instância.
- Todas as réplicas de leitura associadas a uma instância de banco de dados permanecem associadas a essa instância depois que ela é renomeada. Por exemplo, suponha que você tenha uma instância de banco de dados que serve seu banco de dados de produção e essa instância tem várias réplicas de leitura associadas. Se você renomear a instância de banco de dados e a substituir no ambiente de produção por um snapshot de banco de dados, a instância de banco de dados que você renomeou ainda terá as réplicas leitura associadas a ela.
- Métricas e eventos associados ao nome de uma instância de banco de dados serão mantidos se você reutilizar um nome de instância de banco de dados. Por exemplo, se você promover uma réplica de leitura e a renomear para que ela tenha o nome da instância de banco de dados primária anterior, os eventos e as métricas associados à instância de banco de dados primária serão associados à instância renomeada.
- Tags de instâncias de bancos de dados permanecem com a instância de banco de dados, independentemente de renomeação.
- snapshot de banco de dadoss são mantidos para uma instância de banco de dados renomeada.

> **Note**
>
> Uma instância de banco de dados é um ambiente de banco de dados isolado em execução na nuvem. Uma instância de banco de dados pode hospedar vários bancos de dados ou um único banco de dados Oracle com vários esquemas. Para obter informações sobre como alterar um nome de banco de dados, consulte a documentação do seu mecanismo de banco de dados.

Renomear para substituir uma instância de banco de dados existente

As razões mais comuns para renomear uma instância de banco de dados são porque você está promovendo uma réplica de leitura ou restaurando dados de um snapshot de banco de dados ou de uma Point-In-Time Recovery (PITR – Recuperação point-in-time). Ao renomear o banco de dados, você pode substituir a instância de banco de dados sem precisar alterar qualquer código de aplicativo que faça referência a ela. Nesses casos, faça o seguinte:

1. Pare todo o tráfego para a instância de banco de dados primária. Isso pode envolver redirecionar o tráfego, impedindo-o de acessar os bancos de dados na instância de banco de dados ou de alguma outra maneira que você queira impedir que o tráfego acesse seus bancos de dados na instância de banco de dados.
2. Renomeie a instância de banco de dados primária com um nome que indique que ela não é mais a instância de banco de dados primária, conforme descrito mais adiante neste tópico.
3. Crie uma nova instância de banco de dados primária restaurando de um snapshot de banco de dados ou promovendo uma réplica de leitura e dê à nova instância o nome da instância de banco de dados primária anterior.
4. Associe réplicas de leitura à nova instância de banco de dados primária.

Se você excluir a instância de banco de dados primária antiga, será responsável por excluir qualquer snapshot de banco de dados indesejado dessa instância primária antiga.

Para obter informações sobre como promover uma réplica de leitura, consulte Promoção de uma réplica de leitura a uma instância de banco de dados autônoma.

> **Important**
>
> A instância de banco de dados é reinicializada ao ser renomeada. Para instâncias multi-AZ do RDS para SQL Server com a opção “Sempre ativa” ou “Espelhamento” habilitada, é esperado um failover quando a instância é reiniciada após a operação de renomeação.

Console

Para renomear uma instância de banco de dados

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Databases (Bancos de dados).
3. Selecione a instância de banco de dados que você deseja renomear.
4. Selecione Modify.
5. Em Settings (Configurações), digite um novo nome para DB instance identifier (Identificador da instância de banco de dados).
6. Escolha Continue.
7. Para aplicar as alterações imediatamente, escolha Apply immediately. Escolher essa opção pode causar uma interrupção em alguns casos. Para obter mais informações, consulte Modificar uma instância de banco de dados do Amazon RDS.
8. Na página de confirmação, revise suas alterações. Se estiverem corretas, escolha Modify DB Instance (Modificar instância de banco de dados) para salvar suas alterações.

   Como alternativa, escolha Back (Voltar) para editar suas alterações ou escolha Cancel (Cancelar) para cancelar as alterações.

AWS CLI

Para renomear uma instância de banco de dados, use o comando AWS CLI da `modify-db-instance`. Forneça ao valor `--db-instance-identifier` atual e ao parâmetro `--new-db-instance-identifier` o novo nome da instância de banco de dados.

Example

Para Linux, macOS ou Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier DBInstanceIdentifier \
    --new-db-instance-identifier NewDBInstanceIdentifier
```

Para Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier DBInstanceIdentifier ^
    --new-db-instance-identifier NewDBInstanceIdentifier
```

API do RDS

Para renomear uma instância de banco de dados, chame a operação `ModifyDBInstance` da API do Amazon RDS com os seguintes parâmetros:

- `DBInstanceIdentifier` — nome existente da instância
- `NewDBInstanceIdentifier` — novo nome da instância

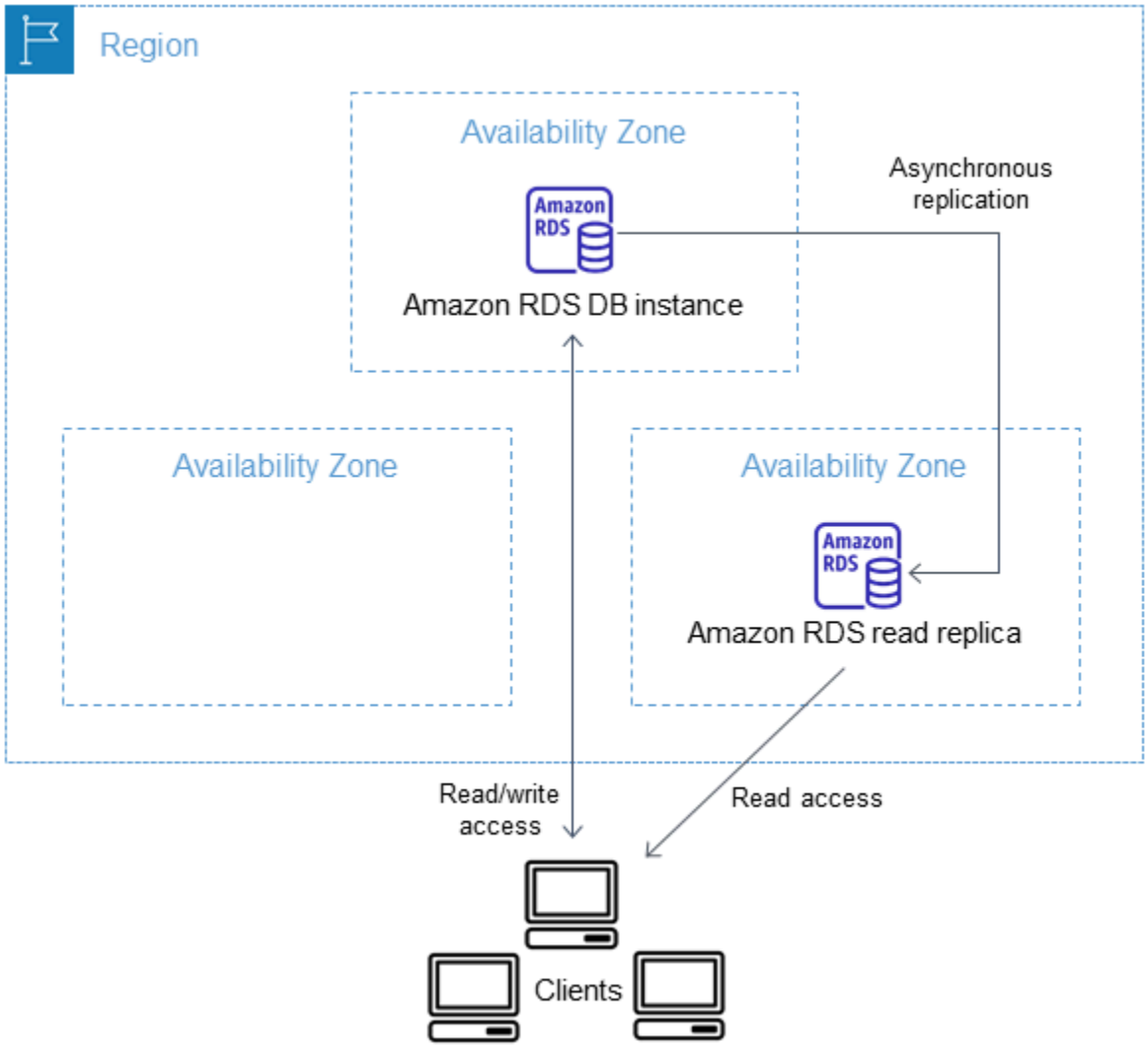
Trabalhar com réplicas de leitura de instância de banco de dados

Uma réplica de leitura é uma cópia somente para leitura de uma instância de banco de dados. É possível reduzir a carga na instância de banco de dados primária roteando as consultas das aplicações para a réplica de leitura. Dessa maneira, é possível aumentar a escala horizontalmente para além das limitações de capacidade de uma única instância de banco de dados para workloads de banco de dados com muita leitura.

Para criar uma réplica de leitura com base em uma instância de banco de dados de origem, o Amazon RDS utiliza os recursos de replicação integrada do mecanismo de banco de dados. Para obter informações sobre como usar réplicas de leitura com um mecanismo específico, consulte as seguintes seções:

- Trabalhar com réplicas para o Amazon RDS para Db2
- Trabalhar com réplicas de leitura do MariaDB
- Trabalhar com réplicas de leitura do Microsoft SQL Server no Amazon RDS
- Trabalhar com réplicas de leitura MySQL
- Trabalhar com réplicas de leitura do Amazon RDS para Oracle
- Trabalhar com réplicas de leitura do Amazon RDS para PostgreSQL

Depois de criar uma réplica de leitura com base em uma instância de banco de dados de origem, a origem se torna a instância de banco de dados primária. Quando você faz atualizações na instância de banco de dados primária, o Amazon RDS as copia de forma assíncrona para a réplica de leitura. O diagrama a seguir mostra uma instância de banco de dados de origem replicando para uma réplica de leitura em uma zona de disponibilidade (AZ) diferente. Os clientes têm acesso de leitura/gravação à instância de banco de dados primária e acesso somente de leitura à réplica.



As réplicas de leitura são cobradas como instâncias de banco de dados padrão com as mesmas taxas da classe de instância de banco de dados usada para a réplica. Você não é cobrado pela transferência de dados incorrida na replicação de dados entre a instância de banco de dados de origem e uma réplica de leitura dentro da mesma Região da AWS. Para obter mais informações, consulte Custos da replicação entre regiões e Faturamento da instância de banco de dados para Amazon RDS.

Tópicos

- Visão geral das réplicas de leitura do Amazon RDS
- Diferenças entre réplicas de leitura para mecanismos de banco de dados
- Como criar uma réplica de leitura
- Promoção de uma réplica de leitura a uma instância de banco de dados autônoma
- Monitoramento da replicação de leitura

- Criar uma réplica de leitura em uma Região da AWS diferente

Visão geral das réplicas de leitura do Amazon RDS

As seções a seguir abordam réplicas de leitura de instância de banco de dados. Para ter informações sobre réplicas de leitura de cluster de banco de dados multi-AZ, consulte the section called “Trabalhar com réplicas de leitura de cluster de banco de dados multi-AZ”.

Tópicos

- Casos de uso para réplicas de leitura
- Como as réplicas de leitura funcionam
- Réplicas de leitura em uma implantação multi-AZ
- Réplicas de leitura entre regiões
- Tipos de armazenamento da réplica de leitura
- Restrições para criar uma réplica com base em uma réplica
- Considerações ao excluir réplicas

Casos de uso para réplicas de leitura

Implantar uma ou mais réplicas de leitura a uma determinada instância de banco de dados de origem pode fazer sentido em inúmeros casos, inclusive os seguintes:

- Expandir além da capacidade computacional ou de E/S de uma única instância de banco de dados para workloads de leitura pesadas de banco de dados. Poder direcionar esse tráfego de leitura excessivo a uma ou mais réplicas de leitura.
- Atender ao tráfego de leitura enquanto a instância de banco de dados de origem está indisponível. Em alguns casos, sua instância de banco de dados de origem pode não ser capaz de fazer solicitações de E/S, por exemplo, devido à suspensão de E/S para backups ou manutenção programada. Nesses casos, é possível direcionar o tráfego de leitura para suas réplicas de leitura. Para esse tipo de caso de uso, lembre-se de que os dados na réplica de leitura podem estar "obsoletos", porque a instância de banco de dados de origem está indisponível.
- Casos de relatórios comerciais ou de data warehousing em que você pode desejar que as consultas de relatórios comerciais sejam executadas em relação a uma réplica de leitura, em vez de sua instância de banco de dados de produção.

- Implementação da recuperação de desastres. É possível promover uma réplica de leitura para uma instância autônoma como solução de recuperação de desastres, caso haja uma falha na instância de banco de dados primária.

Como as réplicas de leitura funcionam

Ao criar uma réplica de leitura, especifique uma instância de banco de dados existente como origem. Em seguida, o Amazon RDS faz um snapshot da instância de origem e cria uma instância apenas para leitura a partir do snapshot. O Amazon RDS usará o método de replicação assíncrona para que o mecanismo de banco de dados atualize a réplica de leitura sempre que houver alteração na instância de banco de dados primária.

A réplica de leitura funciona como uma instância de banco de dados que só permite conexões somente leitura. As exceções são os mecanismos de banco de dados RDS para Db2 e RDS para Oracle, que oferecem suporte a bancos de dados de réplica no modo de espera e no modo montado, respectivamente. Uma réplica de espera e uma réplica montada não aceitam conexões de usuário e, portanto, não podem atender a workloads somente leitura. O principal uso para réplicas montadas de espera e montadas é a recuperação de desastres entre regiões. Para obter mais informações, consulte Trabalhar com réplicas para o Amazon RDS para Db2 e Trabalhar com réplicas de leitura do Amazon RDS para Oracle.

As aplicações se conectam a uma réplica de leitura, assim como fazem com qualquer instância de banco de dados. O Amazon RDS replica todos os bancos de dados na instância de banco de dados de origem.

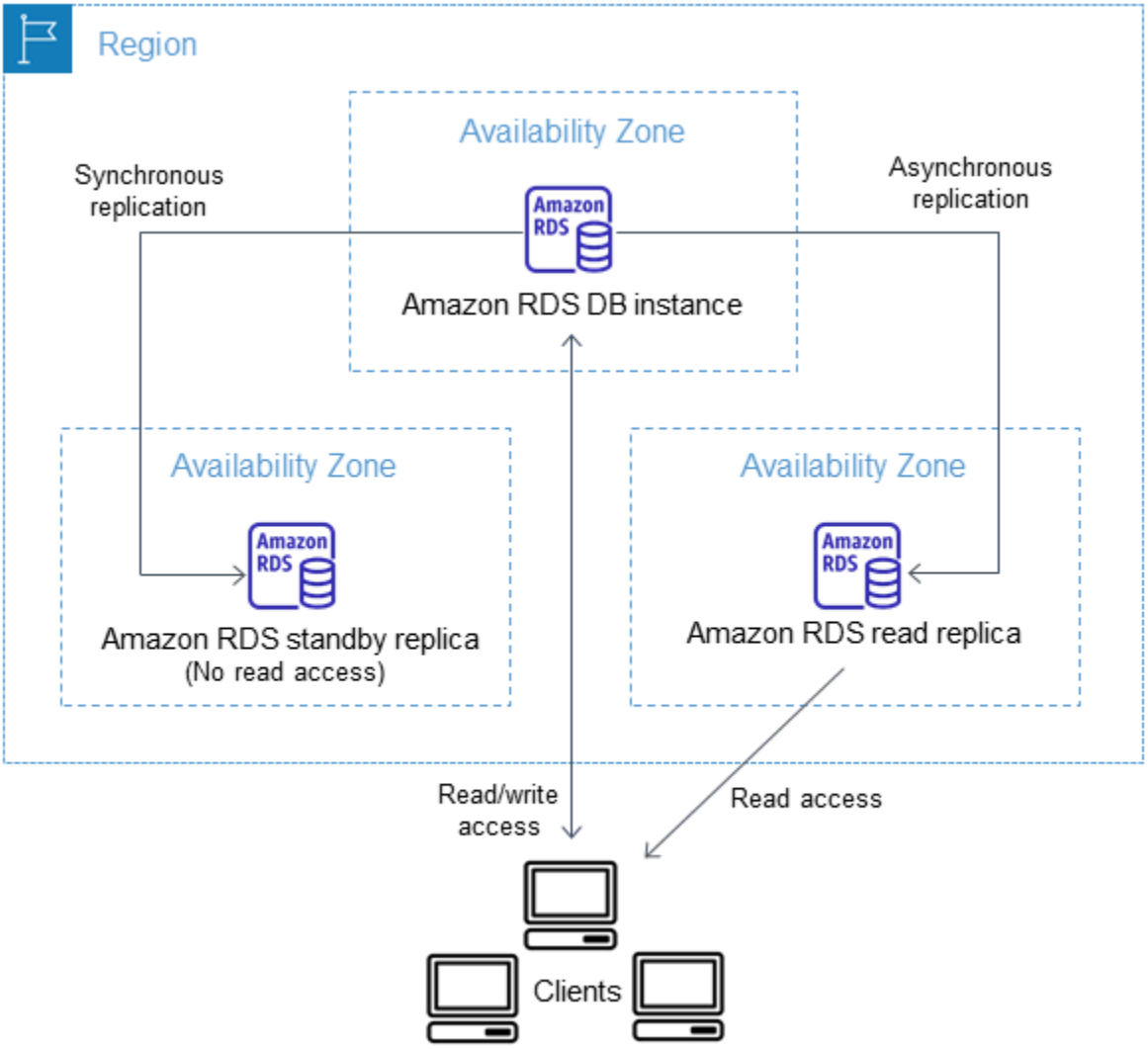
Você deve criar réplicas de leitura manualmente. O RDS não permite o ajuste de escala automático de réplicas de leitura, que é a adição ou remoção automática de réplicas de leitura conforme a demanda de leitura muda.

Réplicas de leitura em uma implantação multi-AZ

Você pode configurar uma réplica de leitura para uma instância de banco de dados que também tenha uma réplica de espera configurada para alta disponibilidade em uma implantação multi-AZ. A replicação com a réplica em espera é síncrona. Ao contrário de uma réplica de leitura, uma réplica em espera não pode fornecer tráfego de leitura.

No cenário a seguir, os clientes têm acesso de leitura/gravação a uma instância de banco de dados primária em uma AZ. A instância primária copia as atualizações de forma assíncrona em uma réplica

de leitura em uma segunda AZ e também as copia de forma síncrona em uma réplica em espera em uma terceira AZ. Os clientes têm acesso de leitura somente à réplica de leitura.



Para ter mais informações sobre réplicas de alta disponibilidade e de espera, consulte Configurar e gerenciar uma implantação multi-AZ para o Amazon RDS.

Réplicas de leitura entre regiões

Em alguns casos, uma réplica de leitura reside em uma Região da AWS diferente daquela da instância de banco de dados primária. Nesses casos, o Amazon RDS configura um canal de comunicação seguro entre a instância de banco de dados primária e a réplica de leitura. O Amazon RDS estabelece quaisquer configurações de segurança da AWS necessárias para permitir o canal seguro, como a inclusão de entradas de grupos de segurança. Para ter mais informações sobre réplicas de leitura entre regiões, consulte Criar uma réplica de leitura em uma Região da AWS diferente.

As informações neste capítulo aplicam-se à criação de réplicas de leitura do Amazon RDS na mesma Região da AWS que a instância de banco de dados de origem ou em uma Região da AWS separada. As informações a seguir não se aplicam à configuração da replicação com uma instância que esteja sendo executada em um instância do Amazon EC2 ou em um ambiente on-premises.

Tipos de armazenamento da réplica de leitura

Por padrão, uma réplica de leitura é criada com o mesmo tipo de armazenamento que a instância de banco de dados de origem. No entanto, é possível pode criar uma réplica de leitura que tenha um tipo diferente de armazenamento que o da instância de banco de dados de origem com base nas opções listadas na tabela a seguir.

Tipo de armazenamento da instância de banco de dados de origem	Alocação de armazenamento da instância de banco de dados de origem	Opções de tipo de armazenamento da réplica de leitura
Provisioned IOPS (IOPS provisionadas)	100 GiB–64 TiB	IOPS provisionadas, propósito geral, magnético
Finalidade geral	100 GiB–64 TiB	IOPS provisionadas, propósito geral, magnético
Finalidade geral	<100 GiB	Propósito geral, magnético
Magnético	100 GiB – 6 TiB	IOPS provisionadas, propósito geral, magnético
Magnético	<100 GiB	Propósito geral, magnético

> **Note**
>
> Quando você aumenta o armazenamento alocado de uma réplica de leitura, ela deve ser de pelo menos 10%. Ao tentar aumentar o valor em menos de 10%, você obtém um erro.

Restrições para criar uma réplica com base em uma réplica

O Amazon RDS não suporta a replicação circular. Você não pode configurar uma instância de banco de dados para servir como origem de replicação para uma instância de banco de dados existente. Só é possível criar uma nova réplica de leitura de uma instância de banco de dados existente. Por exemplo, se **`MySourceDBInstance`** replicar para **`ReadReplica1`**, não será possível configurar **`ReadReplica1`** para replicar de volta para **`MySourceDBInstance`**.

No RDS para MariaDB, no RDS para MySQL e em determinadas versões do RDS para PostgreSQL, é possível criar uma réplica de leitura de uma réplica de leitura existente. Por exemplo, é possível criar uma réplica de leitura **`ReadReplica2`** com base em uma réplica existente **`ReadReplica1`**. Para o RDS para Db2, RDS para Oracle e RDS para SQL Server, não é possível criar uma réplica de leitura de uma réplica de leitura existente.

Considerações ao excluir réplicas

O RDS não permite o ajuste de escala automático de réplicas de leitura. Por isso, o RDS não aumentará o número de réplicas quando a demanda aumentar nem diminuirá o número de réplicas quando a demanda diminuir. Se você não precisar mais das réplicas de leitura, exclua-as manualmente usando os mesmos mecanismos para excluir uma instância de banco de dados. Se você excluir uma instância de banco de dados de origem sem excluir suas réplicas de leitura na mesma Região da AWS, cada réplica será promovida a uma instância de banco de dados autônoma.

Para obter informações sobre como excluir uma instância de banco de dados, consulte Excluir uma instância de banco de dados. Para obter informações sobre a promoção de réplicas de leitura, consulte Promoção de uma réplica de leitura a uma instância de banco de dados autônoma. Para ter informações sobre como excluir a instância de banco de dados de origem de uma réplica de leitura entre regiões, consulte Considerações sobre replicação entre regiões.

Diferenças entre réplicas de leitura para mecanismos de banco de dados

Como os mecanismos de banco de dados do Amazon RDS implementam a replicação de maneira diferente, há várias diferenças significativas que você deve conhecer.

Db2

As réplicas do RDS para Db2 têm os seguintes atributos e comportamentos:

- Método de replicação – replicação física.

- Expurgação de logs de transações – o RDS para Db2 expurgará os logs da instância de banco de dados primária quando as seguintes condições forem atendidas:
  - Os logs têm pelo menos duas horas.
  - O período de retenção configurado para os logs de arquivo expirou.
  - O RDS para Db2 replicou com sucesso os logs para todas as instâncias de banco de dados de réplica.

  Isso se aplica tanto às instâncias de banco de dados da Região da AWS quanto às instâncias de banco de dados entre regiões. Para obter informações sobre como definir horas de retenção de log de arquivamento, consulte rdsadmin.set_archive_log_retention.
- Réplicas graváveis – uma réplica do Db2 é uma cópia física e o Db2 não permite gravações em uma réplica. É possível promover a réplica para torná-la gravável. A réplica promovida terá os dados replicados para o ponto quando a solicitação foi feita para promovê-la.
- Backups – backups automáticos e snapshots manuais são compatíveis com réplicas do RDS para Db2.
- Replicação paralela – os dados de log de arquivamento são sempre transmitidos em paralelo do banco de dados primário para todas as réplicas.
- Estado de espera – o principal uso para réplicas de espera é a recuperação de desastres entre regiões. Para mais informações, consulte Trabalhar com réplicas para o Amazon RDS para Db2.

MariaDB e MySQL

As réplicas de leitura do RDS para MariaDB e do RDS para MySQL têm os seguintes atributos e comportamentos:

- Método de replicação – replicação lógica.
- Limpeza dos logs de transação – o RDS para MariaDB e o RDS para MySQL mantêm todos os logs binários que não foram aplicados.
- Réplicas graváveis – é possível permitir que uma réplica de leitura do MariaDB ou do MySQL seja gravável.
- Backups – backups automáticos e snapshots manuais são compatíveis com réplicas de leitura do RDS para MariaDB ou do RDS para MySQL.
- Replicação paralela – todas as versões do MariaDB e do MySQL permitem threads de replicação paralelos.
- Estado montado – não suportado.

Oracle

As réplicas de leitura do RDS para Oracle têm os seguintes atributos e comportamentos:

- Método de replicação – replicação física.
- Limpeza dos logs de transação – se uma instância de banco de dados primária não tiver réplicas de leitura entre regiões, o Amazon RDS para Oracle manterá um mínimo de duas horas de logs de transação na instância de banco de dados de origem. Os logs serão limpos da instância de banco de dados de origem após duas horas ou após o tempo definido na configuração de horas de retenção do log de arquivamento, o que for maior. Os logs serão removidos da réplica de leitura após o tempo definido pela configurações horas de retenção do log de arquivamento somente se eles tiverem sido aplicados com êxito ao banco de dados.

  Em alguns casos, uma instância de banco de dados primária pode ter uma ou mais réplicas de leitura entre regiões. Nesse caso, o Amazon RDS for Oracle mantém os logs de transações na instância do banco de dados de origem até que eles tenham sido transmitidos e aplicados a todas as réplicas de leitura entre regiões.

  Para obter informações sobre como definir horas de retenção de log de arquivamento, consulte Retenção de logs redo arquivados.
- Réplicas graváveis – uma réplica de leitura do Oracle é uma cópia física, e o Oracle não permite gravações em uma réplica de leitura. É possível promover a réplica de leitura para torná-la gravável. A réplica de leitura promovida terá os dados replicados para o ponto quando a solicitação foi feita para promovê-la.
- Backups – backups automáticos e snapshots manuais são compatíveis com réplicas de leitura do RDS para Oracle.
- Replicação paralela – os dados de log redo são sempre transmitidos em paralelo do banco de dados primário para todas as réplicas de leitura.
- Estado montado – o principal uso para réplicas montadas é a recuperação de desastres entre regiões. Uma licença do Active Data Guard não é necessária para réplicas montadas. Para obter mais informações, consulte Trabalhar com réplicas de leitura do Amazon RDS para Oracle.

PostgreSQL

As réplicas de leitura do RDS para PostgreSQL têm os seguintes atributos e comportamentos:

- Método de replicação – replicação física.

- Limpeza dos logs de transação – o PostgreSQL tem o parâmetro `wal_keep_segments`, que determina quantos arquivos de log prévio de gravação (WAL) são mantidos para fornecer dados às réplicas de leitura. O valor de parâmetro especifica o número de logs a serem mantidos.
- Réplicas graváveis – uma réplica de leitura do PostgreSQL é uma cópia física e o PostgreSQL não permite que uma réplica de leitura torne-se gravável.
- Backups – snapshots manuais são compatíveis com réplicas de leitura do RDS for PostgreSQL. Os backups automatizados de réplicas de leitura são compatíveis apenas com o RDS para PostgreSQL 14.1 e versões superiores. Não é possível ativar backups automatizados de réplicas de leitura do PostgreSQL de versões do RDS para PostgreSQL anteriores à 14.1. Para o RDS para PostgreSQL 13 e versões anteriores, crie um instantâneo de uma réplica de leitura para criar um backup dele.
- Réplica paralela – o PostgreSQL tem um processo único que gerencia a replicação.
- Estado montado – não suportado.

SQL Server

As réplicas de leitura do RDS para SQL Server têm os seguintes atributos e comportamentos:

- Método de replicação – replicação física.
- Limpeza dos logs de transação – o arquivo de log virtual (VLF) do arquivo de log de transação na réplica primária poderá ficar truncado depois que ele não for mais necessário para as réplicas secundárias.

  O VLF só poderá ser marcado como inativo quando os registros de log tiverem sido reforçados nas réplicas. Independentemente da rapidez com que os subsistemas de disco estejam na réplica primária, o log de transações manterá os VLFs até que a réplica mais lenta o tenha reforçado.
- Réplicas graváveis – uma réplica de leitura do SQL Server é uma cópia física e também não permite gravações. É possível promover a réplica de leitura para torná-la gravável. A réplica de leitura promovida terá os dados replicados até o ponto quando a solicitação foi feita para promovê-la.
- Backups – backups automáticos e snapshots manuais não são compatíveis com réplicas de leitura do RDS para SQL Server.
- Replicação paralela – os dados de log redo são sempre transmitidos em paralelo do banco de dados primário para todas as réplicas de leitura.
- Estado montado – não suportado.

Como criar uma réplica de leitura

Você pode criar uma réplica de leitura a partir de uma instância de banco de dados existente usando o AWS Management Console, a AWS CLI ou a API do RDS. Crie uma réplica de leitura especificando o `SourceDBInstanceIdentifier`, que é o identificador da instância do banco de dados da instância de banco de dados de origem da qual você deseja replicar.

Ao criar uma réplica de leitura, o Amazon RDS faz um snapshot do banco de dados da sua instância de banco de dados de origem e inicia a replicação. A instância de banco de dados de origem sofre uma suspensão de E/S muito breve quando a operação de snapshot do banco de dados se inicia. A suspensão de E/S geralmente dura um segundo. Você pode evitar a suspensão de E/S se a instância de banco de dados de origem for uma implantação multi-AZ. Nesse caso, o snapshot é retirado da instância de banco de dados secundária.

Uma transação ativa e de longa duração pode atrasar o processo de criação da réplica de leitura. Recomendamos que você aguarde até que as transações de longa duração sejam concluídas antes de criar uma réplica de leitura. Se você criar várias réplicas de leitura em paralelo a partir da mesma instância de banco de dados de origem, o Amazon RDS fará somente um snapshot no início da primeira ação de criação.

Ao criar uma réplica de leitura, considere o seguinte. Primeiro, você deve habilitar backups automáticos na instância de banco de dados de origem, definindo o período de retenção de backup como um valor diferente de 0. Esse requisito também se aplica a uma réplica de leitura que seja a instância de banco de dados de origem de outra réplica de leitura. Para habilitar backups automáticos em uma réplica de leitura do RDS para MySQL, primeiro crie a réplica de leitura e modifique-a para habilitar backups automáticos.

> **Note**
>
> Em uma região Região da AWS, recomendamos criar todas as réplicas de leitura na mesma nuvem privada virtual (VPC) com base na mesma Amazon VPC da instância de banco de dados de origem. Se você criar uma réplica de leitura em uma VPC diferente da instância de banco de dados de origem, os intervalos de roteamento sem classe entre domínios (CIDR) podem se sobrepor entre a réplica e o sistema RDS. A sobreposição de CIDR torna a réplica instável, o que pode afetar negativamente as aplicações que se conectam a ela. Se você receber um erro ao criar a réplica de leitura, escolha um grupo de sub-redes de banco de dados de destino diferente. Para obter mais informações, consulte Trabalhar com uma instância de banco de dados em uma VPC.

Não há uma maneira direta de criar uma réplica de leitura em outra Conta da AWS usando o console ou a AWS CLI.

Console

Para criar uma réplica de leitura a partir de uma instância de banco de dados de origem

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Databases (Bancos de dados).
3. Escolha a instância de banco de dados que você deseja usar como a origem da réplica de leitura.
4. Em Actions (Ações), escolha Create read replica (Criar réplica de leitura).
5. Em DB instance identifier (Identificador da instância de banco de dados), insira um nome para a réplica de leitura.
6. Escolha a configuração da instância. Recomendamos que você use a classe de instância de banco de dados e o tipo de armazenamento iguais ou superiores aos da instância de banco de dados de origem para a réplica de leitura.
7. Para Região da AWS, especifique a região de destino da réplica de leitura.
8. Para Armazenamento, especifique o tamanho de armazenamento alocado e se deseja usar o ajuste de escala automático do armazenamento.

   Se sua instância de banco de dados de origem não estiver na configuração de armazenamento mais recente, a opção Atualizar a configuração do sistema de arquivos de armazenamento estará disponível. Você pode ativar essa configuração para atualizar o sistema de arquivos de armazenamento da réplica de leitura para a configuração preferida. Para obter mais informações, consulte the section called “Atualizando o sistema de arquivos de armazenamento”.
9. Em Disponibilidade, escolha se você quer criar uma versão de espera da réplica em outra zona de disponibilidade para que a réplica possa contar com failover.

> **Note**
>
> Você pode criar a réplica de leitura como uma instância de banco de dados multi-AZ independentemente de o banco de dados de origem ser ou não uma instância de banco de dados multi-AZ.

10. Especifique outras configurações de instância de banco de dados. Para obter informações sobre cada configuração disponível, consulte Configurações para instâncias de banco de dados.
11. Para criar uma réplica de leitura criptografada, expanda Configuração adicional e especifique as seguintes configurações:

    a. Selecione Habilitar criptografia.

    b. Em AWS KMS key, escolha o identificador da AWS KMS key da chave do KMS.

> **Note**
>
> A instância de banco de dados de origem deve ser criptografada. Para saber mais sobre a criptografia da instância de banco de dados de origem, consulte Criptografar recursos do Amazon RDS.

12. Escolha Create read replica (Criar réplica de leitura).

Depois que a réplica de leitura for criada, você poderá vê-la na página Databases (Bancos de dados) no console do RDS. Ele mostra Replica (Réplica) na coluna Role (Função).

AWS CLI

Para criar uma réplica de leitura de uma instância de banco de dados de origem, use o comando create-db-instance-read-replica da AWS CLI. Esse exemplo também define o tamanho de armazenamento alocado, habilita a autoescalabilidade do armazenamento e atualiza o sistema de arquivos para a configuração preferida.

Você pode especificar outras configurações. Para obter informações sobre cada configuração, consulte Configurações para instâncias de banco de dados.

> **Note**
>
> Para criar uma réplica em espera do RDS para Db2, no comando a seguir AWS CLI, inclua `--replica-mode mounted`.

Example

Para Linux, macOS ou Unix:

```
aws rds create-db-instance-read-replica \
    --db-instance-identifier myreadreplica \
    --source-db-instance-identifier mydbinstance \
    --allocated-storage 100 \
    --max-allocated-storage 1000 \
    --upgrade-storage-config
```

Para Windows:

```
aws rds create-db-instance-read-replica ^
    --db-instance-identifier myreadreplica ^
    --source-db-instance-identifier mydbinstance ^
    --allocated-storage 100 ^
    --max-allocated-storage 1000 ^
    --upgrade-storage-config
```

API do RDS

Para criar uma réplica de leitura a partir de uma instância de banco de dados MySQL, MariaDB, Oracle, PostgreSQL ou SQL Server de origem, chame a operação da API do Amazon RDS CreateDBInstanceReadReplica com os seguintes parâmetros obrigatórios:

- `DBInstanceIdentifier`
- `SourceDBInstanceIdentifier`

> **Note**
>
> Para criar uma réplica em espera do RDS para Db2, defina a operação opcional `ReplicaMode` como `mounted`.

Promoção de uma réplica de leitura a uma instância de banco de dados autônoma

Você pode promover uma réplica de leitura a uma instância de banco de dados autônoma. Se uma instância de banco de dados de origem tiver várias réplicas de leitura, promover uma das réplicas de leitura a uma instância de banco de dados não interferirá nas outras réplicas.

Ao promover uma réplica de leitura, o RDS reinicializa a instância de banco de dados antes de torná-la disponível. O processo de promoção pode levar vários minutos ou mais para ser concluído, dependendo do tamanho da réplica de leitura.



Casos de uso para promover uma réplica de leitura

Você pode querer promover uma réplica de leitura a uma instância de banco de dados autônoma por algum dos seguintes motivos:

- Implementação de recuperação de falhas – você poderá usar a promoção da réplica de leitura como um esquema de recuperação de dados se a instância de banco de dados primária falhar. Essa abordagem complementa a replicação síncrona, a detecção automática de falhas e o failover.

Se estiver ciente das ramificações e limitações da replicação assíncrona e ainda assim quiser usar a promoção de réplica de leitura para recuperação de dados, será possível fazer isso. Primeiramente, crie uma réplica de leitura e, depois, monitore a instância de banco de dados primária para identificar se há falhas. Em caso de falha, faça o seguinte:

1. Promova a réplica de leitura.
2. Direcione o tráfego de banco de dados para a instância de banco de dados promovida.
3. Crie uma réplica de leitura de substituição com a instância de banco de dados promovida como origem.

- Atualizando a configuração de armazenamento — Se sua instância de banco de dados de origem não estiver na configuração de armazenamento preferencial, você poderá criar uma réplica de leitura da instância e atualizar a configuração do sistema de arquivos de armazenamento. Essa opção migra o sistema de arquivos da réplica de leitura para a configuração preferida. Você pode promover uma réplica de leitura a uma instância de banco de dados autônoma.

  Você pode usar essa opção para superar as limitações de escalabilidade no armazenamento e no tamanho do arquivo para sistemas de arquivos mais antigos de 32 bits. Para obter mais informações, consulte the section called “Atualizando o sistema de arquivos de armazenamento”.

  Essa opção só está disponível se sua instância de banco de dados de origem não estiver na configuração de armazenamento mais recente ou se você estiver modificando a classe da instância de banco de dados na mesma solicitação.

- Estilhaçamento – a fragmentação incorpora a arquitetura "shared-nothing" e envolve essencialmente a quebra de um grande banco de dados em vários bancos de dados menores. Uma maneira comum de dividir um banco de dados é dividir tabelas que não estão juntas na mesma consulta em diferentes hosts. Outro método é duplicar uma tabela em vários hosts e depois usar um algoritmo de hash para determinar qual host recebe uma determinada atualização. É possível pode criar réplicas de leitura correspondentes a cada um de seus fragmentos (bancos de dados menores) e promovê-los quando decidir convertê-los em fragmentos individuais. Você pode, então, criar o espaço da chave (se você estiver dividindo linhas) ou a distribuição de tabelas para cada um dos fragmentos de acordo com seus requisitos.

- Execução de operações de DDL (somente MySQL e MariaDB) – As operações de DDL, como criação ou recriação de índices, podem levar tempo e causar um prejuízo significativo à performance da sua instância de banco de dados. É possível executar essas operações em uma réplica de leitura do MySQL ou MariaDB uma vez que a réplica de leitura esteja em sincronia com

sua instância de banco de dados primária. Então, será possível promover a réplica de leitura e direcionar seus aplicativos para usar a instância promovida.

> **Note**
>
> Se a réplica de leitura for uma instância de banco de dados do RDS para Oracle, você poderá realizar uma transição em vez de uma promoção. Em uma transição, a instância de banco de dados de origem se torna a nova réplica e a réplica se torna a nova instância de banco de dados de origem. Para obter mais informações, consulte Executar uma alternância do Oracle Data Guard.

Características de uma réplica de leitura promovida

Depois de promover a réplica de leitura, ela deixa de funcionar como réplica de leitura e se torna uma instância de banco de dados autônoma. A nova instância de banco de dados autônoma tem as seguintes características:

- A instância de banco de dados autônoma mantém o grupo de opções e o grupo de parâmetros da réplica de leitura de pré-promoção.
- É possível criar réplicas de leitura da instância de banco de dados autônoma e executar operações de recuperação para um ponto no tempo.
- Não é possível usar a instância de banco de dados como um destino de replicação, pois ela não é mais uma réplica de leitura.

Pré-requisitos para promover uma réplica de leitura

Antes de promover uma réplica de leitura, faça o seguinte:

- Revise sua estratégia de backup:
  - Recomendamos habilitar os backups e concluir pelo menos um backup. A duração de backup é uma função do número de alterações feitas ao banco de dados desde o backup anterior.
  - Se você habilitou backups em sua réplica de leitura, configure a janela de backup automatizado para que os backups diários não interfiram com a promoção da réplica de leitura.
  - Certifique-se de que a réplica de leitura não tenha o status `backing-up`. Você não pode promover uma réplica de leitura quando ela está nesse estado.

- Interrompa a gravação de todas as transações na instância de banco de dados primária e aguarde até que o RDS aplique todas as atualizações à réplica de leitura.

  As atualizações do banco de dados serão feitas nas réplicas de leitura depois que forem feitas na instância de banco de dados primária. O atraso na replicação pode variar significativamente. Use a métrica `Replica Lag` para determinar quando todas as atualizações foram feitas na réplica de leitura.

- (Somente para o MySQL e o MariaDB) Para fazer alterações na réplica de leitura do MySQL ou MariaDB antes de promovê-la, defina o parâmetro `read_only` como `0` no grupo de parâmetros de banco de dados para a réplica de leitura. Você então poderá executar todas as operações de DDL necessárias, como criar índices, na réplica de leitura. As ações realizadas na réplica de leitura não afetam a performance da instância de banco de dados primária.

Promover uma réplica de leitura: etapas básicas

As etapas a seguir mostram o processo geral para promover uma réplica de leitura a uma instância de banco de dados

1. Promova a réplica de leitura utilizando a opção Promote (Promover) no console do Amazon RDS, o comando da AWS CLI `promote-read-replica` ou a operação `PromoteReadReplica` da API do Amazon RDS.

   > **Note**
   >
   > O processo de promoção leva alguns minutos para ser concluído. Ao promover uma réplica de leitura, o RDS interrompe a replicação e reinicializa a réplica de leitura. Quando a reinicialização for concluída, a réplica de leitura estará disponível como uma nova instância de banco de dados.

2. (Opcional) Modifique a nova instância de banco de dados para ser uma implantação multi-AZ. Para ter mais informações, consulte Modificar uma instância de banco de dados do Amazon RDS e Configurar e gerenciar uma implantação multi-AZ para o Amazon RDS.

Console

Para promover uma réplica de leitura a uma instância de banco de dados autônoma

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No console do Amazon RDS, escolha Databases (Bancos de dados).

   O painel Databases (Bancos de dados) é exibido. Todas as réplicas de leitura exibem Replica (Réplica) na coluna Role (Função).
3. Escolha a réplica de leitura que você deseja promover.
4. Em Actions (Ações), selecione Promote (Promover).
5. Na página Promote Read Replica (Promover réplica de leitura), insira o período de retenção de backup e a janela de backup para a instância de banco de dados recém-promovida.
6. Quando estiver satisfeito com as configurações, escolha Continue (Continuar).
7. Na página de confirmação, escolha Promote Read Replica (Promover réplica de leitura).

AWS CLI

Para promover uma réplica de leitura para uma instância de banco de dados autônoma, use o comando da AWS CLI `promote-read-replica`.

Example

Para Linux, macOS ou Unix:

```
aws rds promote-read-replica \
    --db-instance-identifier myreadreplica
```

Para Windows:

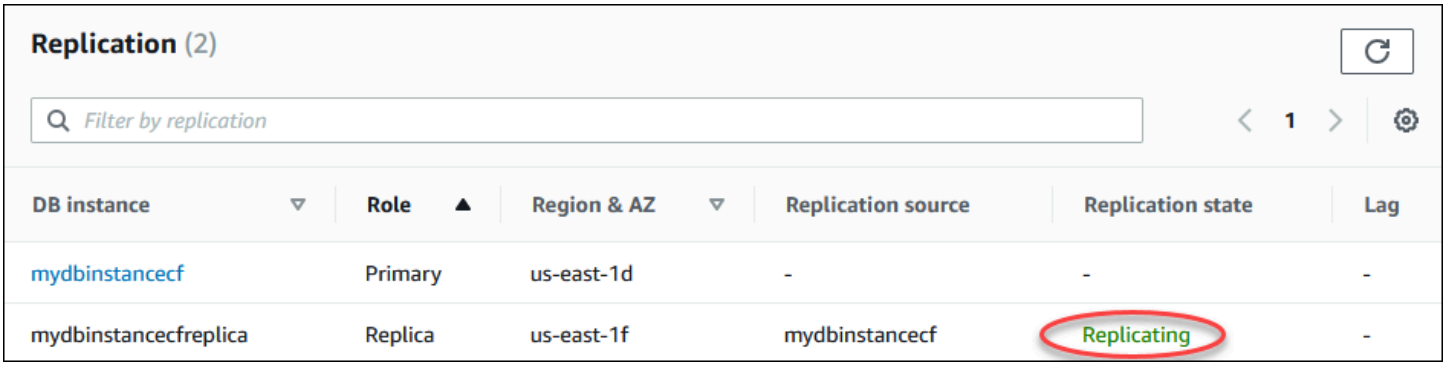
```
aws rds promote-read-replica ^
    --db-instance-identifier myreadreplica
```

API do RDS

Para promover uma réplica de leitura a uma instância de banco de dados autônoma, chame a operação da API do Amazon RDS `PromoteReadReplica` com o parâmetro obrigatório `DBInstanceIdentifier`.

Monitoramento da replicação de leitura

É possível monitorar o status de uma réplica de leitura de várias maneiras. O console do Amazon RDS mostra o status de uma réplica de leitura na seção Replication (Replicação) da guia Connectivity & security (Conectividade e segurança) nos detalhes da réplica de leitura. Para visualizar os detalhes de uma réplica de leitura, clique no nome dela na lista de instâncias de banco de dados no console do Amazon RDS.



Também é possível visualizar o status de uma réplica de leitura usando o comando da AWS CLI `describe-db-instances` ou a operação da API do Amazon RDS `DescribeDBInstances`.

O status de uma réplica de leitura pode ser um dos seguintes:

- replicating (replicando) – a réplica de leitura está sendo replicada com sucesso.
- replicação degradada (somente SQL Server) – as réplicas estão recebendo dados da instância primária, mas um ou mais bancos de dados podem não estar recebendo atualizações. Isso pode ocorrer, por exemplo, quando uma réplica está no processo de configuração de bancos de dados recém-criados. Também pode ocorrer quando alterações não suportadas de DDL ou objetos grandes são feitas no ambiente azul de uma implantação azul/verde.

  O status não fará a transição de `replication degraded` para `error`, a menos que ocorra um erro durante o estado degradado.
- error – ocorreu um erro com a replicação. Verifique o campo Replication Error (Erro de replicação) no console do Amazon RDS ou o log de eventos para determinar o erro exato. Para ter mais informações sobre como solucionar um erro de replicação, consulte Solucionar problemas de uma réplica de leitura do MySQL.
- terminated (encerrada) (somente para MariaDB, MySQL ou PostgreSQL) – a replicação foi encerrada. Isso ocorre se a replicação é interrompida por mais de 30 dias consecutivos, manualmente ou devido a um erro de replicação. Nesse caso, o Amazon RDS encerra a replicação entre a instância de banco de dados primária e todas as réplicas de leitura. O Amazon RDS faz

isso para evitar um aumento nos requisitos de armazenamento da instância de banco de dados de origem e nos tempos de failover prolongado.

A replicação interrompida pode afetar o armazenamento porque os logs podem aumentar de tamanho e número devido ao alto volume de mensagens de erro que estão sendo gravadas no log. A replicação interrompida também pode afetar a recuperação de falhas devido ao tempo que o Amazon RDS requer para manter e processar o grande número de logs durante a recuperação.

- terminated (encerrada) (somente para Oracle): a replicação é encerrada. Isso ocorrerá se a replicação for interrompida por mais de oito horas porque não há armazenamento suficiente restante na réplica de leitura. Nesse caso, o Amazon RDS encerra a replicação entre a instância de banco de dados primária e a réplica de leitura afetada. Esse status é terminal, e a réplica de leitura deve ser recriada.
- stopped (interrompida) (somente para MariaDB ou MySQL) – a replicação foi interrompida por causa de uma solicitação iniciada pelo cliente.
- replication stop point set (ponto de interrupção da replicação definido) (somente para MySQL) – um ponto de interrupção iniciado pelo cliente foi definido usando o procedimento armazenado mysql.rds_start_replication_until e a replicação está em andamento.
- replication stop point reached (ponto de interrupção da replicação alcançado) (somente para MySQL) – um ponto de interrupção iniciado pelo cliente foi definido usando o procedimento armazenado mysql.rds_start_replication_until e a replicação foi interrompida porque o ponto de interrupção foi alcançado.

Você pode ver onde uma instância de banco de dados está sendo replicada e, se for o caso, verifique o status de replicação. Na página Database (Banco de dados) no console do RDS, será exibido Primary (Primário) na coluna Role (Função). Escolha o nome da instância de banco de dados. Na página de detalhes, na guia Connectivity & security (Conectividade e segurança), seu status de replicação estará em Replication (Replicação).

Monitorar atraso da replicação

Você pode monitorar o atraso da replicação no Amazon CloudWatch visualizando a métrica `ReplicaLag` do Amazon RDS.

Para o Db2, a métrica `ReplicaLag` é o atraso máximo de bancos de dados que ficaram para trás, em segundos. Por exemplo, se dois bancos de dados atrasarem 5 segundos e 10 segundos respectivamente, `ReplicaLag` será de 10 segundos. Bancos de dados sem status de Recuperação de Desastres de Alta Disponibilidade (HADR) disponíveis não são incluídos no cálculo.

Para o MySQL e o MariaDB, a métrica `ReplicaLag` informa o valor do campo `Seconds_Behind_Master` do comando `SHOW REPLICA STATUS`. As causas comuns de atraso de replicação do MySQL e do MariaDB são as seguintes:

- Uma queda de rede.
- Gravar em tabelas com índices em uma réplica de leitura. Se o parâmetro `read_only` não estiver definido como 0 na réplica de leitura, isso poderá interromper a replicação.
- Uso de um mecanismo de armazenamento não transacional, como o MyISAM. A replicação é compatível apenas com o mecanismo de armazenamento InnoDB no MySQL e o mecanismo de armazenamento XtraDB no MariaDB.

> **Note**
>
> Versões anteriores do MariaDB usavam `SHOW SLAVE STATUS` em vez de `SHOW REPLICA STATUS`. Se você estiver usando uma versão do MariaDB anterior à 10.5, use `SHOW SLAVE STATUS`.

Quando a métrica `ReplicaLag` chega a 0, isso mostra que a réplica alcançou a instância do banco de dados primária. Se a métrica `ReplicaLag` retornar `-1`, então a replicação não está ativa no momento. `ReplicaLag = -1` é equivalente a `Seconds_Behind_Master = NULL`.

Para Oracle, a métrica `ReplicaLag` é a soma do valor de `Apply Lag` e a diferença entre a hora atual e o valor de `DATUM_TIME` do atrasado de aplicação. O valor de `DATUM_TIME` é a última vez em que a réplica de leitura recebeu dados de sua instância de banco de dados de origem. Para ter mais informações, consulte V$DATAGUARD_STATS na documentação da Oracle.

Para o SQL Server, a métrica `ReplicaLag` é o atraso máximo de bancos de dados que ficaram para trás, em segundos. Por exemplo, se você tiver dois bancos de dados com atraso de 5 segundos e 10 segundos respectivamente, `ReplicaLag` será de 10 segundos. A métrica `ReplicaLag` retorna o valor da consulta a seguir.

```
SELECT MAX(secondary_lag_seconds) max_lag FROM sys.dm_hadr_database_replica_states;
```

Para ter mais informações, consulte secondary_lag_seconds na documentação da Microsoft.

`ReplicaLag` retornará `-1` se o RDS não puder determinar o atraso, como durante a configuração da réplica ou quando a réplica de leitura estiver no estado `error`.

> **Note**
>
> Novos bancos de dados não serão incluídos no cálculo de atraso até que estejam acessíveis na réplica de leitura.

Para o PostgreSQL, a métrica `ReplicaLag` retorna o valor da consulta a seguir.

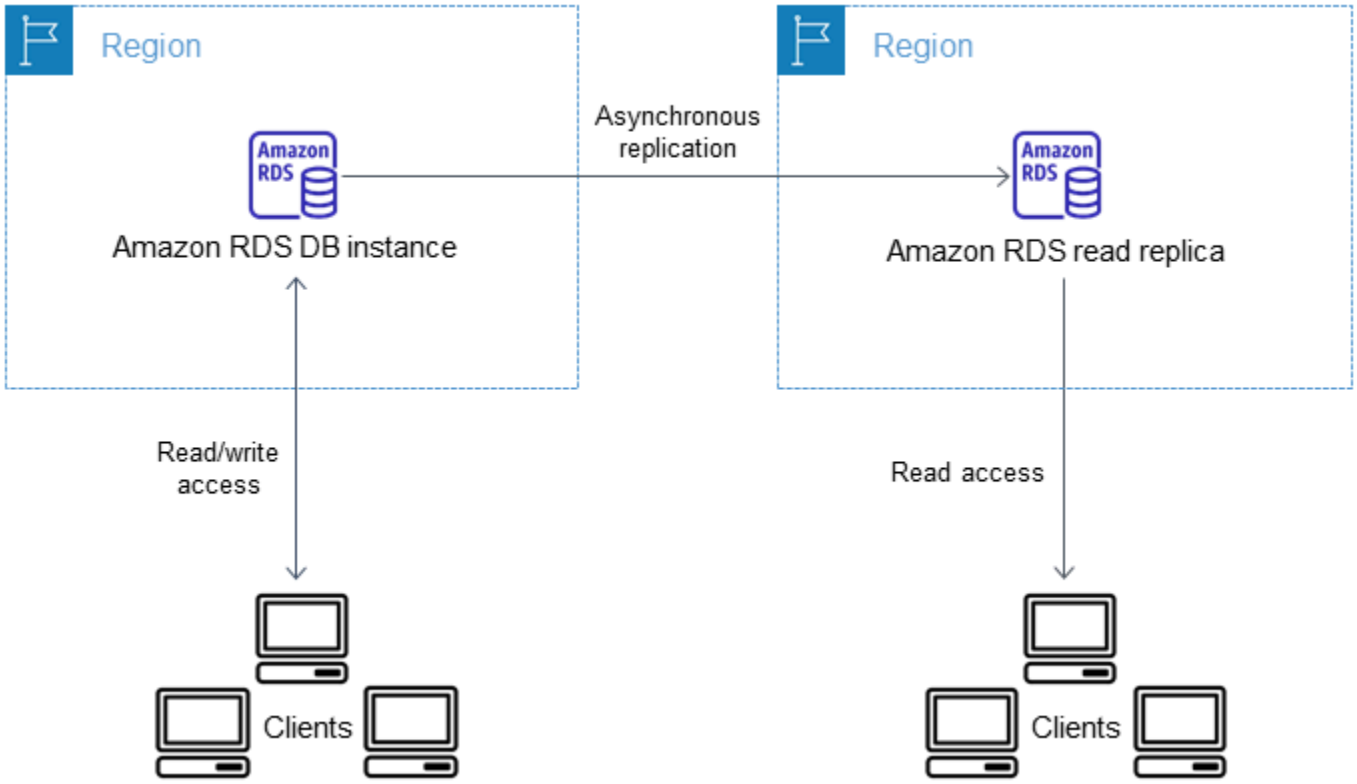
```
SELECT extract(epoch from now() - pg_last_xact_replay_timestamp()) AS reader_lag
```

O PostgreSQL versões 9.5.2 e posterior usa slots de replicação física para gerenciar a retenção do log de gravação antecipada (WAL) na instância de origem. Para cada instância da réplica de leitura entre regiões, o Amazon RDS cria um slot de replicação física e o associa à instância. Duas métricas do Amazon CloudWatch, `Oldest Replication Slot Lag` e `Transaction Logs Disk Usage`, mostram o quão longe está a réplica mais atrasada em termos de dados WAL recebidos e quanto tempo de armazenamento está sendo usado para dados WAL. O valor de `Transaction Logs Disk Usage` pode aumentar consideravelmente quando uma réplica de leitura entre regiões está muito atrasada.

Para ter mais informações sobre como monitorar uma instância de banco de dados com CloudWatch, consulte Monitorar métricas do Amazon RDS com o Amazon CloudWatch.

Criar uma réplica de leitura em uma Região da AWS diferente

Com o Amazon RDS, é possível criar uma réplica de leitura em uma Região da AWS diferente da instância de banco de dados de origem.



Crie uma réplica de leitura em uma Região da AWS diferente para fazer o seguinte:

- Melhorar suas capacidades de recuperação de desastres.
- Dimensionar operações de leitura em uma Região da AWS mais próxima de seus usuários.
- Facilitar a migração de um datacenter em uma Região da AWS para um datacenter em outra Região da AWS.

Criar uma réplica de leitura em uma Região da AWS diferente da instância de origem é semelhante à criação de uma réplica na mesma Região da AWS. É possível usar o AWS Management Console, executar o comando `create-db-instance-read-replica` ou chamar a operação de API `CreateDBInstanceReadReplica`.

> **Note**
>
> Para criar uma réplica de leitura criptografada em uma Região da AWS diferente da instância de banco de dados de origem, a instância de banco de dados de origem deve ser criptografada.

Tópicos

- Disponibilidade de região e versão
- Criação de uma réplica de leitura entre regiões
- Como o Amazon RDS faz a replicação entre regiões
- Considerações sobre replicação entre regiões
- Custos da replicação entre regiões

Disponibilidade de região e versão

A disponibilidade e a compatibilidade de recursos variam entre versões específicas de cada mecanismo de banco de dados e entre Regiões da AWS. Para ter mais informações sobre a disponibilidade de versões e regiões com a replicação entre regiões, consulte Regiões e mecanismos de banco de dados compatíveis com réplicas de leitura entre regiões no Amazon RDS.

Criação de uma réplica de leitura entre regiões

O procedimento a seguir mostra como criar uma réplica de leitura de uma instância de banco de dados de origem do MariaDB, do Microsoft SQL Server, do MySQL, do Oracle ou do PostgreSQL em uma Região da AWS diferente.

Console

É possível criar uma réplica de leitura entre Regiões da AWS usando o AWS Management Console.

Como criar uma réplica de leitura entre Regiões da AWS com o console

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Bancos de dados.
3. Selecione a instância de banco de dados do MariaDB, do Microsoft SQL Server, do MySQL, do Oracle ou do PostgreSQL que você deseja usar como a origem de réplica de leitura.
4. Em Actions (Ações), escolha Create read replica (Criar réplica de leitura).
5. Em DB instance identifier (Identificador da instância de banco de dados), insira um nome para a réplica de leitura.
6. Escolha a Destination Region (Região de destino).

7. Escolha as especificações da instância que você deseja usar. Recomendamos usar a mesma categoria de instância de banco de dados ou maior e o mesmo tipo de armazenamento para a réplica de leitura.
8. Para criar uma réplica de leitura criptografada em outra Região da AWS:

   a. Selecione Habilitar criptografia.

   b. Em AWS KMS key, escolha o identificador do AWS KMS key da chave do KMS na Região da AWS de destino.

> **Note**
>
> Para criar uma réplica de leitura criptografada, a instância do banco de dados de origem deve ser criptografada. Para saber mais sobre a criptografia da instância de banco de dados de origem, consulte Criptografar recursos do Amazon RDS.

9. Escolha outras opções, como autoscaling do armazenamento.
10. Escolha Create read replica (Criar réplica de leitura).

AWS CLI

Para criar uma réplica de leitura de uma instância de banco de dados de origem do MariaDB, do Microsoft SQL Server, do MySQL, do Oracle ou do PostgreSQL em uma Região da AWS diferente, é possível usar o comando `create-db-instance-read-replica`. Nesse caso, use `create-db-instance-read-replica` da Região da AWS em que você quer a réplica de leitura (região de destino) e especifique o nome do recurso da Amazon (ARN) da instância do banco de dados de origem. Um ARN identifica de forma exclusiva um recurso criado no Amazon Web Services.

Por exemplo, se a instância de banco de dados de origem estiver na região Leste dos EUA (N. da Virgínia), o ARN será semelhante a este exemplo:

```
arn:aws:rds:us-east-1:123456789012:db:mydbinstance
```

Para obter informações sobre ARNs, consulte Nomes de recurso da Amazon (ARNs) no Amazon RDS.

Para criar uma réplica de leitura em uma Região da AWS diferente da região da instância de banco de dados de origem, é possível usar o comando AWS CLI`create-db-instance-read-replica`

da Região da AWS de destino. Os seguintes parâmetros são necessários para criar uma réplica de leitura em outra Região da AWS:

- `--region`: a Região da AWS de destino em que a réplica de leitura é criada.
- `--source-db-instance-identifier` – o identificador de instância de banco de dados para a instância de origem. Esse identificador deve estar no formato do ARN da Região da AWS de origem.
- `--db-instance-identifier`: o identificador da réplica de leitura na Região da AWS de destino.

Example de uma réplica de leitura entre regiões

O código a seguir cria uma réplica de leitura na região Oeste dos EUA (Oregon) de uma instância de banco de dados de origem na região Leste dos EUA (N. da Virgínia).

Para Linux, macOS ou Unix:

```
aws rds create-db-instance-read-replica \
    --db-instance-identifier myreadreplica \
    --region us-west-2 \
    --source-db-instance-identifier arn:aws:rds:us-east-1:123456789012:db:mydbinstance
```

Para Windows:

```
aws rds create-db-instance-read-replica ^
    --db-instance-identifier myreadreplica ^
    --region us-west-2 ^
    --source-db-instance-identifier arn:aws:rds:us-east-1:123456789012:db:mydbinstance
```

Os seguintes parâmetros também são necessários para criar uma réplica de leitura criptografada em outra Região da AWS:

- `--kms-key-id`: o identificador da AWS KMS key da chave do KMS a ser usada para criptografar a réplica de leitura na Região da AWS de destino.

Example de uma réplica de leitura criptografada entre regiões

O código a seguir cria uma réplica de leitura criptografada na região Oeste dos EUA (Oregon) de uma instância de banco de dados de origem na região Leste dos EUA (N. da Virgínia).

Para Linux, macOS ou Unix:

```
aws rds create-db-instance-read-replica \
    --db-instance-identifier myreadreplica \
    --region us-west-2 \
    --source-db-instance-identifier arn:aws:rds:us-east-1:123456789012:db:mydbinstance
 \
    --kms-key-id my-us-west-2-key
```

Para Windows:

```
aws rds create-db-instance-read-replica ^
    --db-instance-identifier myreadreplica ^
    --region us-west-2 ^
    --source-db-instance-identifier arn:aws:rds:us-east-1:123456789012:db:mydbinstance
 ^
    --kms-key-id my-us-west-2-key
```

A opção `--source-region` é necessária quando você está criando uma réplica de leitura criptografada entre as regiões GovCloud (Leste dos EUA) da AWS e GovCloud (Oeste dos EUA) da AWS. Em `--source-region`, especifique a Região da AWS da instância de banco de dados de origem.

Se a `--source-region` não estiver especificada, especifique um valor de `--pre-signed-url`. Uma presigned URL é um URL que contém uma solicitação assinada do Signature Versão 4 para o comando `create-db-instance-read-replica` chamado na Região da AWS de origem. Para saber mais sobre a opção `pre-signed-url`, consulte create-db-instance-read-replica na Referência de comandos da AWS CLI.

API do RDS

Para criar uma réplica de leitura de uma instância de banco de dados de origem do MariaDB, do MySQL, do Microsoft SQL Server, do Oracle ou do PostgreSQL em uma Região da AWS diferente, você pode chamar a operação da API do Amazon RDS CreateDBInstanceReadReplica. Nesse caso, chame CreateDBInstanceReadReplica na Região da AWS em que você deseja a réplica de leitura (região de destino) e especifique o nome do recurso da Amazon (ARN) da instância de banco de dados de origem. Um ARN identifica de forma exclusiva um recurso criado no Amazon Web Services.

Para criar uma réplica de leitura criptografada em uma Região da AWS diferente da região da instância de banco de dados de origem, é possível usar a operação `CreateDBInstanceReadReplica` da API do Amazon RDS na Região da AWS de destino. Para criar uma réplica de leitura criptografada em outra Região da AWS, especifique um valor para `PreSignedURL`. A `PreSignedURL` deve conter uma solicitação para a operação `CreateDBInstanceReadReplica` a ser chamada na Região da AWS de origem em que a réplica de leitura foi criada. Para saber mais sobre `PreSignedUrl`, consulte `CreateDBInstanceReadReplica`.

Por exemplo, se sua instância de banco de dados de origem estiver na região Leste dos EUA (N. da Virgínia) region, o ARN é semelhante ao seguinte.

```
arn:aws:rds:us-east-1:123456789012:db:mydbinstance
```

Para obter informações sobre ARNs, consulte Nomes de recurso da Amazon (ARNs) no Amazon RDS.

Example

```
https://us-west-2.rds.amazonaws.com/
    ?Action=CreateDBInstanceReadReplica
    &KmsKeyId=my-us-east-1-key
    &PreSignedUrl=https%253A%252F%252Frds.us-west-2.amazonaws.com%252F
         %253FAction%253DCreateDBInstanceReadReplica
         %2526DestinationRegion%253Dus-east-1
         %2526KmsKeyId%253Dmy-us-east-1-key
         %2526SourceDBInstanceIdentifier%253Darn%25253Aaws%25253Ards%25253Aus-
west-2%123456789012%25253Adb%25253Amydbinstance
         %2526SignatureMethod%253DHmacSHA256
         %2526SignatureVersion%253D4%2526SourceDBInstanceIdentifier%253Darn%25253Aaws
%25253Ards%25253Aus-west-2%25253A123456789012%25253Ainstance%25253Amydbinstance
         %2526Version%253D2014-10-31
         %2526X-Amz-Algorithm%253DAWS4-HMAC-SHA256
         %2526X-Amz-Credential%253DAKIADQKE4SARGYLE%252F20161117%252Fus-west-2%252Frds
%252Faws4_request
         %2526X-Amz-Date%253D20161117T215409Z
         %2526X-Amz-Expires%253D3600
         %2526X-Amz-SignedHeaders%253Dcontent-type%253Bhost%253Buser-agent%253Bx-amz-
content-sha256%253Bx-amz-date
         %2526X-Amz-Signature
%253D255a0f17b4e717d3b67fad163c3ec26573b882c03a65523522cf890a67fca613
    &DBInstanceIdentifier=myreadreplica
```

```
    &SourceDBInstanceIdentifier=&region-arn;rds:us-east-1:123456789012:db:mydbinstance
    &Version=2012-01-15
    &SignatureVersion=2
    &SignatureMethod=HmacSHA256
    &Timestamp=2012-01-20T22%3A06%3A23.624Z
    &AWSAccessKeyId=<&AWS; Access Key ID>
    &Signature=<Signature>
```

Como o Amazon RDS faz a replicação entre regiões

O Amazon RDS usa o seguinte processo para criar uma réplica de leitura entre regiões. Dependendo das Regiões da AWS envolvidas e da quantidade de dados nos bancos de dados, esse processo pode levar horas para ser concluído. Use essas informações para determinar até que ponto o processo prosseguiu ao criar uma réplica de leitura entre regiões:

1. O Amazon RDS começa a configurar a instância de banco de dados de origem como uma origem de replicação e define o status como modifying (modificação).
2. O Amazon RDS começa a configurar a réplica de leitura especificada na Região da AWS de destino e define o status como criando.
3. O Amazon RDS cria um snapshot automatizado do banco de dados da instância de banco de dados de origem na Região da AWS. O formato do nome do snapshot de banco de dados é `rds:<InstanceID>-<timestamp>`, onde `<InstanceID>` é o identificador da instância de origem e `<timestamp>` é a data e a hora em que a cópia começou. Por exemplo, `rds:mysourceinstance-2013-11-14-09-24` foi criada a partir da instância `mysourceinstance` em `2013-11-14-09-24`. Durante a criação de um snapshot de banco de dados automatizado, o status da instância do banco de dados de origem permanece como modifying (modificação), o status da réplica de leitura permanece como creating (criação) e o status do snapshot de banco de dados é creating (criação). A coluna de andamento da página snapshot de banco de dados (Snapshot de banco de dados) no console informa o andamento da criação do snapshot de banco de dados. Quando o snapshot de banco de dados estiver completo, o status do snapshot de banco de dados da instância do banco de dados de origem serão definidos como available (disponível).
4. O Amazon RDS inicia uma cópia do snapshot entre regiões para a transferência de dados inicial. A cópia do snapshot é listada como um snapshot automático na Região da AWS de destino com um status de criando. Ele tem o mesmo nome que o snapshot de banco de dados de origem. A coluna de andamento da exibição do snapshot de banco de dados indica até que ponto a cópia evoluiu. Quando a cópia estiver completa, o status da cópia do snapshot de banco de dados será definido como available (disponível).

5. O Amazon RDS usa o snapshot de banco de dados copiado para a carga de dados inicial na réplica de leitura. Durante essa fase, a réplica de leitura estará na lista de instâncias de banco de dados no destino, com um status de creating (criação). Quando a carga estiver completa, o status da réplica de leitura será definido como available (disponível) e a cópia do snapshot de banco de dados será excluída.
6. Quando a réplica de leitura atingir o status disponível, o Amazon RDS começará a replicar as alterações feitas na instância de origem desde o início da operação de criação da réplica de leitura. Durante essa fase, o tempo de atraso da replicação para a réplica de leitura será maior que 0.

   Para obter informações sobre o tempo de atraso de replicação, consulte Monitoramento da replicação de leitura.

Considerações sobre replicação entre regiões

Todas as considerações para a realização da replicação dentro de uma Região da AWS aplicam-se à replicação entre regiões. As seguintes considerações adicionais são válidas ao replicar entre Regiões da AWS:

- Uma instância de banco de dados de origem pode ter réplicas de leitura entre várias Regiões da AWS. Devido ao limite no número de entradas da lista de controle de acesso (ACL) para a VPC de origem, o RDS não pode garantir mais de cinco instâncias de banco de dados de réplica de leitura entre regiões.
- Você pode replicar entre as regiões da GovCloud (Leste dos EUA) e GovCloud (Oeste dos EUA), mas não para dentro ou para fora da GovCloud (EUA).
- Para os seguintes mecanismos do RDS, é possível criar uma réplica de leitura entre regiões do Amazon RDS com base em uma instância de banco de dados de origem do Amazon RDS somente quando ela não é uma réplica de leitura de outra instância do banco de dados do Amazon RDS.
  - RDS para SQL Server
  - RDS para Oracle
  - RDS para PostgreSQL versões anteriores à 14.1

  Essa limitação não se aplica às instâncias de banco de dados que executam o RDS para PostgreSQL versão 14.1 e posterior, RDS para MariaDB e RDS para MySQL.

- É possível que você perceba uma quantidade maior de tempo de atraso para réplicas de leitura que estejam em uma Região da AWS diferente da instância de origem. Esse tempo de atraso vem dos canais de rede mais longos entre os datacenters regionais.
- Para replicações de leitura entre regiões, qualquer um dos comandos de criação de réplica de leitura que especifique o parâmetro `--db-subnet-group-name` deve especificar um grupo de sub-redes de banco de dados da mesma VPC.
- Na maioria dos casos, a réplica de leitura usa o grupo de parâmetros de banco de dados padrão e o grupo de opções de banco de dados do mecanismo de banco de dados especificado.

  Para o mecanismo de banco de dados do Db2, você deve especificar um grupo de parâmetros personalizado para a réplica. Para réplicas que usam o modelo BYOL, esse grupo de parâmetros personalizados deve incluir seu IBM Site ID e IBM Customer ID. Você pode especificar esse grupo de parâmetros personalizado para a réplica usando o AWS Management Console, a AWS CLI ou a API do RDS. Para obter mais informações, consulte IDs da IBM para traga a sua própria licença para o Db2.

  Para os mecanismos de banco de dados do MySQL e do Oracle, você pode especificar um grupo de parâmetros personalizado para a réplica de leitura na opção `--db-parameter-group-name` do comando AWS CLI da create-db-instance-read-replica. Você não pode especificar um grupo de parâmetros personalizado ao usar o AWS Management Console.
- A réplica de leitura usa o grupo de segurança padrão.
- Para instâncias de banco de dados do MariaDB, do Microsoft SQL Server, do MySQL e do Oracle, quando a instância de banco de dados de origem de uma réplica de leitura entre regiões é excluída, a réplica de leitura é promovida.
- Para instâncias de bancos de dados PostgreSQL, quando a instância de banco de dados de origem de uma réplica de leitura entre regiões é excluída, o status de replicação da réplica de leitura é definido como `terminated`. A réplica de leitura não é promovida.

  É necessário promover a réplica de leitura manualmente ou excluí-la.

Solicitação de uma réplica de leitura entre regiões

Para se comunicar com a região de origem e solicitar a criação de uma réplica de leitura entre regiões, o solicitante (função do IAM ou usuário do IAM) deve ter acesso à instância de banco de dados de origem e à região de origem.

Certas condições na política do IAM do solicitante podem fazer com que a solicitação falhe. Os exemplos a seguir pressupõem que a instância de banco de dados de origem está em Leste dos EUA (Ohio) e a réplica de leitura é criada em Leste dos EUA (N. da Virgínia). Estes exemplos mostram condições na política do IAM do solicitante que fazem com que a solicitação falhe:

- A política do solicitante tem uma condição para `aws:RequestedRegion`.

```
...
"Effect": "Allow",
"Action": "rds:CreateDBInstanceReadReplica",
"Resource": "*",
"Condition": {
    "StringEquals": {
        "aws:RequestedRegion": "us-east-1"
    }
}
```

  A solicitação falha porque a política não permite acesso à região de origem. Para uma solicitação bem-sucedida, especifique as regiões de origem e de destino.

```
...
"Effect": "Allow",
"Action": "rds:CreateDBInstanceReadReplica",
"Resource": "*",
"Condition": {
    "StringEquals": {
        "aws:RequestedRegion": [
            "us-east-1",
            "us-east-2"
        ]
    }
}
```

- A política do solicitante não permite acesso à instância de banco de dados de origem.

```
...
"Effect": "Allow",
"Action": "rds:CreateDBInstanceReadReplica",
"Resource": "arn:aws:rds:us-east-1:123456789012:db:myreadreplica"
...
```

Para uma solicitação bem-sucedida, especifique a instância de origem e a réplica.

```
...
"Effect": "Allow",
"Action": "rds:CreateDBInstanceReadReplica",
"Resource": [
    "arn:aws:rds:us-east-1:123456789012:db:myreadreplica",
    "arn:aws:rds:us-east-2:123456789012:db:mydbinstance"
]
...
```

- A política do solicitante nega `aws:ViaAWSService`.

```
...
"Effect": "Allow",
"Action": "rds:CreateDBInstanceReadReplica",
"Resource": "*",
"Condition": {
    "Bool": {"aws:ViaAWSService": "false"}
}
```

A comunicação com a região de origem é feita pelo RDS em nome do solicitante. Para uma solicitação bem-sucedida, não negue as chamadas feitas pelos serviços da AWS.

- A política do solicitante tem uma condição para `aws:SourceVpc` ou `aws:SourceVpce`.

Essas solicitações podem falhar, pois quando o RDS faz a chamada para a região remota, ela não é feita a partir da VPC ou VPC endpoint especificados.

Se você precisar usar uma das condições anteriores que causariam uma falha de uma solicitação, é possível incluir uma segunda instrução com `aws:CalledVia` em sua política, para fazer com que a solicitação seja bem-sucedida. Por exemplo, você pode usar `aws:CalledVia` com `aws:SourceVpce` como mostrado aqui:

```
...
"Effect": "Allow",
"Action": "rds:CreateDBInstanceReadReplica",
"Resource": "*",
"Condition": {
    "Condition" : {
        "ForAnyValue:StringEquals" : {
```

```
          "aws:SourceVpce": "vpce-1a2b3c4d"
        }
     }
},
{
    "Effect": "Allow",
    "Action": [
        "rds:CreateDBInstanceReadReplica"
    ],
    "Resource": "*",
    "Condition": {
        "ForAnyValue:StringEquals": {
            "aws:CalledVia": [
                "rds.amazonaws.com"
            ]
        }
    }
}
```

Para ter mais informações, consulte Políticas e permissões no IAM, no Guia do usuário do IAM.

Autorização de uma réplica de leitura

Após uma solicitação de criação de réplica de leitura de banco de dados entre regiões retornar `success`, o RDS inicia a criação da réplica em segundo plano. Uma autorização para o RDS acessar a instância de banco de dados de origem é criada. Essa autorização vincula a instância de banco de dados de origem à réplica de leitura e permite que o RDS copie apenas para a réplica de leitura especificada.

A autorização é verificada pelo RDS usando a permissão `rds:CrossRegionCommunication` na função do IAM vinculada ao serviço. Se a réplica estiver autorizada, o RDS se comunica com a região de origem e conclui a criação da réplica.

O RDS não tem acesso a instâncias de banco de dados que não foram autorizadas anteriormente por uma solicitação de `CreateDBInstanceReadReplica`. A autorização é revogada quando a criação da réplica de leitura é concluída.

O RDS usa a função vinculada ao serviço para verificar a autorização na região de origem. Se você excluir a função vinculada ao serviço durante o processo de criação de replicação, a criação falhará.

Para ter mais informações, consulte Usar funções vinculadas a serviço no Guia do usuário do IAM.

Uso de credenciais do AWS Security Token Service

Os tokens de sessão do endpoint global do AWS Security Token Service (AWS STS) são válidos apenas nas Regiões da AWS habilitadas por padrão (regiões comerciais). Se você usar credenciais da operação da API `assumeRole` no AWS STS, use o endpoint regional se a região de origem for uma região de aceitação. Caso contrário, haverá falha na solicitação. Isso acontece porque suas credenciais devem ser válidas em ambas as regiões, o que é verdadeiro para regiões opcionais somente quando o endpoint regional do AWS STS é usado.

Para usar o endpoint global, certifique-se de que ele esteja ativado para ambas as regiões nas operações. Defina o endpoint global como `Valid in all Regiões da AWS` nas configurações da conta do AWS STS.

A mesma regra se aplica às credenciais no parâmetro pre-signed URL.

Para ter mais informações, consulte Gerenciar o AWS STS em uma Região da AWS no Guia do usuário do IAM.

Custos da replicação entre regiões

Os dados transferidos para replicação entre regiões incorrem em cobranças de transferência de dados do Amazon RDS As ações de replicação entre regiões geram cobranças referentes aos dados transferidos para fora da Região da AWS de origem:

- Quando você cria uma réplica de leitura, o Amazon RDS cria um snapshot da instância de origem e transfere esse snapshot para a Região da AWS da réplica de leitura.
- Para cada modificação de dados feita nos bancos de dados de origem, o Amazon RDS transfere os dados da Região da AWS de origem para a Região da AWS da réplica de leitura.

Para ter mais informações sobre a definição e preço da transferência de dados, consulte Definição de preço do Amazon RDS.

Para instâncias do MySQL e do MariaDB, é possível reduzir seus custos de transferência de dados reduzindo o número de réplicas de leitura criadas entre regiões. Por exemplo, suponhamos que você tenha uma instância de banco de dados de origem em uma Região da AWS e queira manter três réplicas de leitura em outra Região da AWS. Nesse caso, crie somente uma das réplicas de leitura da instância do banco de dados de origem. Crie as outras duas réplicas da primeira réplica de leitura, e não da instância de banco de dados de origem.

Por exemplo, se você tiver a `source-instance-1` em uma Região da AWS, você poderá fazer o seguinte:

- Crie a `read-replica-1` na nova Região da AWS, especificando a `source-instance-1` como a origem.
- Crie `read-replica-2` de `read-replica-1`.
- Crie `read-replica-3` de `read-replica-1`.

Neste exemplo, serão cobrados de você somente os dados transferidos de `source-instance-1` para `read-replica-1`. A transferência dos dados transferidos da `read-replica-1` para as outras duas réplicas não será cobrada, pois elas estão todas na mesma Região da AWS. Se você criar todas as três réplicas diretamente de `source-instance-1` em outra Região da AWS, serão cobradas de você as transferências de dados das três réplicas.

Marcar recursos do Amazon RDS

Uma tag do Amazon RDS é um par de nome-valor que você define e associa a um recurso do Amazon RDS, como uma instância ou um snapshot de banco de dados. O nome é referido como chave. Opcionalmente, é possível fornecer um valor para a chave.

Você pode usar o AWS Management Console, a AWS CLI ou a API do Amazon RDS para adicionar, listar e excluir tags em recursos do Amazon RDS. Ao usar a CLI ou a API, forneça o nome do recurso da Amazon (ARN) do recurso do RDS com o qual deseja trabalhar. Para obter mais informações sobre a criação de um ARN, consulte Criar um ARN para o Amazon RDS.

É possível usar etiquetas para adicionar metadados aos recursos do Amazon RDS. Você pode usar as tags para adicionar suas próprias anotações sobre instâncias de banco de dados, snapshots, Aurora clusters e assim por diante. Fazer isso pode ajudar você a documentar os recursos do Amazon RDS. Você também pode usar as tags com procedimentos de manutenção automatizada.

Especificamente, você pode usar essas tags com políticas do IAM. É possível usá-las para gerenciar o acesso aos recursos do Amazon RDS e controlar quais ações podem ser aplicadas a esses recursos. Você também pode usar essas tags para monitorar custos agrupando despesas de recursos marcados com tags semelhantes.

É possível marcar os seguintes recursos do Amazon RDS:

- Instâncias de banco de dados
- clusters de banco de dados
- Clusters globais do Aurora
- Endpoints do cluster de banco de dados
- Réplicas de leitura
- DB snapshots
- Snapshots de cluster de banco de dados
- Instâncias de bancos de dados reservadas
- Assinaturas de eventos
- Grupos de opções de banco de dados
- Grupos de parâmetros do banco de dados
- Grupos de parâmetros de cluster de banco de dados

- Grupos de sub-redes de banco de dados
- RDS Proxies
- Endpoint do RDS Proxy
- Implantações azuis/verdes
- Integrações ETL zero

> **Note**
>
> Atualmente, você não pode etiquetar RDS Proxies e endpoints do RDS Proxy usando o AWS Management Console.

Tópicos

- Por que usar tags de recurso do Amazon RDS?
- Como as tags de recurso do Amazon RDS funcionam
- Práticas recomendadas para marcação de recursos do Amazon RDS
- Copiar tags para snapshots de banco de dados
- Adicionar e excluir tags no Amazon RDS
- Tutorial: especificar quais instâncias de banco de dados devem ser interrompidas usando tags

Por que usar tags de recurso do Amazon RDS?

É possível usar tags para fazer o seguinte:

- Categorize recursos do RDS por aplicação, projeto, departamento, ambiente e assim por diante. Por exemplo, é possível usar uma chave de tag para definir uma categoria, na qual o valor da tag é um item nessa categoria. É possível criar a tag `environment=prod`. Ou você pode definir uma chave de tag como `project` e um valor de tag de `Salix`, o que indica que um recurso do Amazon RDS está atribuído ao projeto Salix.
- Automatize tarefas de gerenciamento de recursos. Por exemplo, é possível criar uma janela de manutenção para instâncias marcadas com `environment=prod` que seja diferente da janela para instâncias marcadas com `environment=test`. Também é possível configurar snapshots de banco de dados automáticos para instâncias marcadas com `environment=prod`.

- Controle o acesso aos recursos do RDS em uma política do IAM. Faça isso usando a chave de condição global `aws:ResourceTag/tag-key`. Por exemplo, uma política pode permitir que somente usuários do grupo `DBAdmin` modifiquem instâncias de banco de dados marcadas com `environment=prod`. Consulte informações sobre como gerenciar o acesso a recursos marcados com políticas do IAM em Gerenciamento de identidade e acesso no Amazon RDS e em Controlar o acesso aos recursos da AWS no Guia do usuário do AWS Identity and Access Management.
- Monitore recursos com base em uma tag. Por exemplo, é possível criar um painel do Amazon CloudWatch para instâncias de banco de dados marcadas com `environment=prod`.
- Monitore custos agrupando despesas de recursos marcados com tags semelhantes. Por exemplo, se marcar os recursos do RDS associados ao projeto Salix com `project=Salix`, você poderá gerar relatórios de custos e alocar despesas para esse projeto. Para ter mais informações, consulte Como o faturamento da AWS funciona com tags no Amazon RDS.

Como as tags de recurso do Amazon RDS funcionam

AWSA não aplica nenhum significado semântico às suas tags. Tags são interpretadas estritamente como sequências de caracteres.

Tópicos

- Conjuntos de tags no Amazon RDS
- Estrutura de tags no Amazon RDS
- Recursos do Amazon RDS elegíveis para marcação
- Como o faturamento da AWS funciona com tags no Amazon RDS

Conjuntos de tags no Amazon RDS

Cada recurso do Amazon RDS tem um contêiner chamado conjunto de tags. O contêiner inclui todas as tags atribuídas ao recurso. Um recurso tem exatamente um conjunto de tags.

Um conjunto de tags contém de 0 a 50 tags. Se você adicionar uma tag a um recurso do RDS que tenha a mesma chave que uma tag existente no recurso, o novo valor substituirá o antigo.

Estrutura de tags no Amazon RDS

A estrutura de uma tag do RDS é a seguinte:

Chave de tag

A chave é o nome obrigatório da tag. O valor da string deve ter de 1 a 128 caracteres Unicode e não pode ter os prefixos `aws:` ou `rds:`. A string pode conter apenas o conjunto de letras, números e espaços em branco Unicode, _, ., :, /, =, +, - e @. O regex Java é `"^([\\p{L}\\p{Z}\\p{N}_.:/=+\\-@]*)$"`. As chaves da tag fazem distinção entre maiúsculas e minúsculas. Assim, as chaves `project` e `Project` são distintas.

Uma chave é exclusiva a um conjunto de tags. Por exemplo, não pode haver um par de chaves em um conjunto de tags com a mesma chave mas com valores diferentes, como `project=Trinity` e `project=Xanadu`.

Valor da tag

O valor é um valor de string opcional da tag. O valor da string deve ter de 1 a 256 caracteres Unicode. A string pode conter apenas o conjunto de letras, números e espaços em branco Unicode, _, ., :, /, =, +, - e @. O regex Java é `"^([\\p{L}\\p{Z}\\p{N}_.:/=+\\-@]*)$"`. Os valores de tags diferenciam maiúsculas de minúsculas. Assim, os valores `prod` e `Prod` são distintos.

Os valores não precisam ser exclusivos em um conjunto de tags e podem ser nulos. Por exemplo, você pode ter um par de valor-chave em um conjunto de tag de `project=Trinity` e `cost-center=Trinity`.

Recursos do Amazon RDS elegíveis para marcação

Você pode marcar os seguintes recursos do Amazon RDS:

- Instâncias de banco de dados
- clusters de banco de dados
- Endpoints do cluster de banco de dados
- Réplicas de leitura
- DB snapshots
- Snapshots de cluster de banco de dados
- Instâncias de bancos de dados reservadas
- Assinaturas de eventos
- Grupos de opções de banco de dados
- Grupos de parâmetros do banco de dados

- Grupos de parâmetros de cluster de banco de dados
- Grupos de sub-redes de banco de dados
- RDS Proxies
- Endpoint do RDS Proxy

> **Note**
>
> Atualmente, você não pode etiquetar RDS Proxies e endpoints do RDS Proxy usando o AWS Management Console.

- Implantações azuis/verdes
- Integrações ETL zero (pré-visualização)

Como o faturamento da AWS funciona com tags no Amazon RDS

Também é possível utilizar tags para organizar sua fatura da AWS para refletir sua própria estrutura de custo. Para fazer isso, inscreva-se para obter a fatura da sua Conta da AWS com os valores de chave de tag incluídos. Então, para ver o custo de recursos combinados, organize suas informações de faturamento de acordo com recursos com os mesmos valores de chave de tags. Por exemplo, é possível marcar vários recursos com um nome de aplicação específico, e depois organizar suas informações de faturamento para ver o custo total daquela aplicação em vários serviços. Para obter mais informações, consulte Usar tags de alocação de custos no Guia do usuário do AWS Billing.

Como as tags de alocação de custos funcionam com snapshots de de banco de dados

É possível adicionar uma tag a um snapshot dede banco de dados. No entanto, sua conta não refletirá esse agrupamento. Para que tags de alocação de custo sejam aplicadas a snapshots de de banco de dados, as seguintes condições devem ser atendidas:

- As tags devem ser anexadas à instância de banco de dados principal.
- A instância de banco de dados principal deve estar na mesma Conta da AWS que o snapshot do de banco de dados.
- A instância de banco de dados principal deve estar na mesma Região da AWS que o snapshot do de banco de dados.

Os snapshots do de banco de dados serão considerados órfãos se não estiverem na mesma região da instância de banco de dados principal ou se esta for excluída. Os snapshots de banco de dados

órfãos não comportam tags de alocação de custos. Os custos de snapshots órfãos são agregados em um único item de linha não marcado. Os snapshots do de banco de dados entre contas não são considerados órfãos quando as seguintes condições são atendidas:

- Eles estão na mesma região da instância de banco de dados principal.
- A instância de banco de dados principal pertence à conta de origem.

> **Note**
>
> Se a instância de banco de dados principal pertencer a uma conta diferente, as tags de alocação de custos não se aplicarão aos snapshots de várias contas na conta de destino.

Práticas recomendadas para marcação de recursos do Amazon RDS

Ao usar tags, sugerimos seguir estas práticas recomendadas:

- Convenções de documentos para uso de tags que são seguidas por todas as equipes da organização. Acima de tudo, os nomes devem ser descritivos e consistentes. Por exemplo, padronize o formato `environment:prod` em vez de marcar alguns recursos com `env:production`.

  > **Important**
  >
  > Não armazene informações de identificação pessoal (PII) nem outras informações confidenciais ou sigilosas em tags.

- Automatize a marcação para garantir a consistência. Por exemplo, é possível usar as seguintes técnicas:
  - Inclua tags em um modelo do AWS CloudFormation. Ao criar recursos com o modelo, eles são marcados automaticamente.
  - Defina e aplique tags usando funções do AWS Lambda.
  - Crie um documento do SSM que inclua etapas para adicionar tags aos recursos do RDS.
- Use tags somente quando necessário. É possível adicionar até 50 tags a um único recurso do RDS, mas a prática recomendada é evitar a proliferação e a complexidade desnecessárias de tags.

- Revise as tags periodicamente para verificar a relevância e a precisão. Remova ou modifique as tags desatualizadas conforme necessário.
- Considere criar tags com o Editor de Tags da AWS no AWS Management Console. É possível usar o Editor de Tags para adicionar tags a vários recursos compatíveis da AWS ao mesmo tempo, incluindo os recursos do RDS. Para obter mais informações, consulte o Tag Editor no Guia do usuário dos Grupos de recursos da AWS.

Copiar tags para snapshots de banco de dados

Ao criar ou restaurar uma instância de banco de dados, você pode especificar que as tags da instância de banco de dados sejam copiadas para snapshots da instância de banco de dados. A cópia de tags garante que os metadados dos snapshots de banco de dados correspondam aos da instância de banco de dados de origem. Também garante que quaisquer políticas de acesso dos snapshots de banco de dados também correspondam às da instância de banco de dados de origem.

Você pode especificar que as etiquetas sejam copiadas para snapshots de banco de dados para as seguintes ações:

- Criar uma instância de banco de dados
- Restauração de uma instância de banco de dados
- Como criar uma réplica de leitura
- Copiar um snapshot de banco de dados

Para copiar tags para as ações anteriores, escolha Copiar tags para snapshots no AWS Management Console ou especifique `--copy-tags-to-snapshot` na AWS CLI.

Na maioria dos casos, as tags não são copiadas por padrão. No entanto, quando você restaura uma instância de banco de dados de um snapshot do banco de dados, o RDS verifica se você deve especificar novas tags. Se sim, as novas tags serão adicionadas à instância de banco de dados restaurada. Se não houver novas tags, o RDS adicionará as tags da instância de banco de dados de origem no momento da criação do snapshot à instância de banco de dados restaurada.

Para evitar que tags da fonte de instâncias de banco de dados sejam adicionadas a instâncias de banco de dados restauradas, recomendamos que você especifique novas tags ao restaurar uma instância de banco de dados.

> **Note**
>
> Em alguns casos, você pode incluir um valor para o parâmetro `--tags` do comando create-db-snapshot da AWS CLI. Ou pode fornecer pelo menos uma tag à operação da API CreateDBSnapshot. Nesses casos, o RDS não copia tags da instância de banco de dados de origem para o novo snapshot de banco de dados. Essa funcionalidade é aplicável mesmo que a instância de banco de dados de origem tenha a opção `--copy-tags-to-snapshot` (`CopyTagsToSnapshot`) ativada.
>
> Se você seguir essa abordagem, poderá criar uma cópia de uma instância de banco de dados de um snapshot de banco de dados. Essa abordagem evita adicionar tags que não se aplicam à nova instância de banco de dados. Você cria o snapshot de banco de dados com o comando `create-db-snapshot` da AWS CLI (ou a operação `CreateDBSnapshot` da API do RDS). Depois de criar o snapshot de banco de dados, é possível adicionar tags conforme descrito posteriormente neste tópico.

Adicionar e excluir tags no Amazon RDS

Você pode fazer o seguinte:

- Crie tags ao criar um recurso, por exemplo, ao executar o comando `create-db-instance` da AWS CLI.
- Adicione tags a um recurso existente usando o comando `add-tags-to-resource`.
- Liste as tags associadas a um recurso específico usando o comando `list-tags-for-resource`.
- Atualize as tags usando o comando `add-tags-to-resource`.
- Remova as tags de um recurso usando o comando `remove-tags-from-resource`.

Os procedimentos a seguir mostram como executar operações típicas de marcação em recursos relacionados a instâncias de banco de dados e. Observe que as tags ficam armazenadas em cache para fins de autorização. Por esse motivo, ao adicionar ou atualizar tags nos recursos do Amazon RDS, pode demorar vários minutos para que as modificações sejam disponibilizadas.

Console

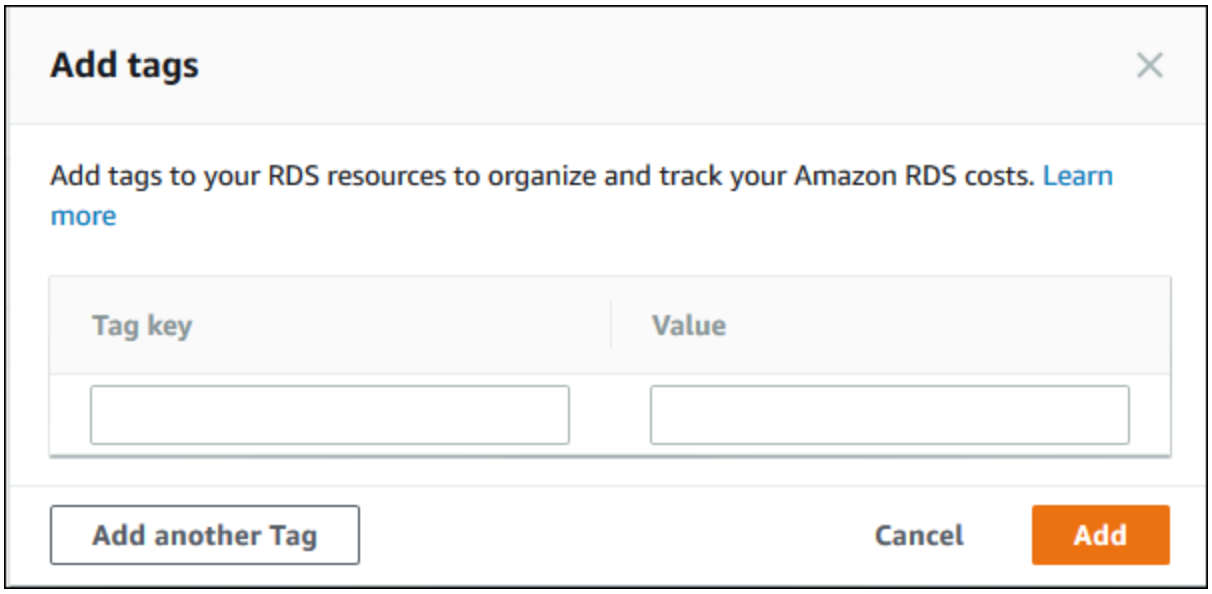
O processo para marcar um recurso do Amazon RDS é semelhante para todos os recursos. O procedimento a seguir mostra como marcar uma instância de banco de dados do Amazon RDS.

Para adicionar uma tag a uma instância de banco de dados

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Databases (Bancos de dados).

   > **Note**
   >
   > Para filtrar a lista de instâncias de bancos de dados no painel Databases (Bancos de dados), digite uma string de texto de Filter databases (Filtrar bancos de dados). Somente instâncias de banco de dados que contiverem a string aparecerão.

3. Escolha no nome da instância de banco de dados que você deseja marcar para mostrar os detalhes.
4. Na seção de detalhes, role para baixo até a seção Tags.
5. Escolha Adicionar. A janela Add tags (Adicionar tags) é exibida.



6. Digite um valor para Tag key (Chave de tag) e Value (Valor).
7. Para adicionar outra tag, escolha Add another Tag (Adicionar outra tag) e digite um valor para Tag key (Chave de tag) e Value (Valor).

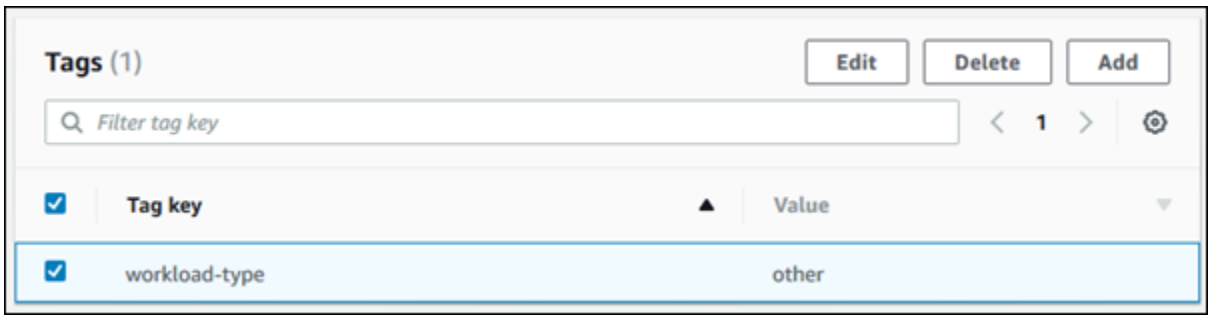
   Repita esta etapa quantas vezes for necessário.
8. Escolha Adicionar.

Para excluir uma tag de uma instância de banco de dados

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Databases (Bancos de dados).

> **Note**
>
> Para filtrar a lista de instâncias de bancos de dados no painel Databases (Bancos de dados), digite uma string de texto na caixa Filter databases (Filtrar bancos de dados). Somente instâncias de banco de dados que contiverem a string aparecerão.

3. Escolha o nome da instância de banco de dados para mostrar os detalhes.
4. Na seção de detalhes, role para baixo até a seção Tags.
5. Escolha a tag que você deseja excluir.



6. Selecione Delete (Excluir) e escolha Delete (Excluir) na janela Delete tags (Excluir tags).

AWS CLI

É possível adicionar, listar ou remover tags de uma instância de banco de dados usando a AWS CLI.

- Para adicionar uma ou mais etiquetas a um recurso do Amazon RDS, use o comando da AWS CLI `add-tags-to-resource`.
- Para listar as etiquetas em um recurso do Amazon RDS, use o comando da AWS CLI `list-tags-for-resource`.
- Para remover uma ou mais etiquetas de um recurso do Amazon RDS, use o comando da AWS CLI `remove-tags-from-resource`.

Para saber mais sobre como criar o ARN necessário, consulte Criar um ARN para o Amazon RDS.

API do RDS

É possível adicionar, listar ou remover tags de uma instância de banco de dados usando a API do Amazon RDS.

- Para adicionar uma tag a um recurso do Amazon RDS, use a operação `AddTagsToResource`.
- Para listar tags atribuídas a um recurso do Amazon RDS, use `ListTagsForResource`.
- Para remover tags de um recurso do Amazon RDS, use a operação `RemoveTagsFromResource`.

Para saber mais sobre como criar o ARN necessário, consulte Criar um ARN para o Amazon RDS.

As tags usam o seguinte esquema ao trabalhar com o XML usando a API do Amazon RDS:

```
<Tagging>
      <TagSet>
          <Tag>
              <Key>Project</Key>
              <Value>Trinity</Value>
          </Tag>
          <Tag>
              <Key>User</Key>
              <Value>Jones</Value>
          </Tag>
      </TagSet>
  </Tagging>
```

A tabela a seguir fornece uma lista das tags XML permitidas e suas características. Os valores de `Key` e `Value` diferenciam letras maiúsculas de minúsculas. Por exemplo, `project=Trinity` e `PROJECT=Trinity` são tags distintas.

Elemento de marcação por tag	Descrição
TagSet	Um conjunto de tags é um contêiner de todas as tags atribuídas a um recurso do Amazon RDS. Só pode haver um conjunto de tags por recurso. Você trabalha com um TagSet somente por meio da API do Amazon RDS.

Elemento de marcação por tag	Descrição
Tag	Uma tag é um par de chave-valor definido pelo usuário. Pode haver de 1 a 50 tags em um conjunto de tags.
Chave	Uma chave é o nome obrigatório da tag. Consulte as restrições em Estrutura de tags no Amazon RDS. O valor da string pode ter de 1 a 128 caracteres Unicode e não pode ter os prefixos `aws:` ou `rds:`. A string pode conter apenas o conjunto de letras em Unicode, dígitos, espaços em branco, '_', '.', '/', '=', '+', '-' (Java regex: "^([\\p{L}\\ \p{Z}\\p{N}_.:/=+\\-]*)$"). As chaves devem ser exclusivas a um conjunto de tags. Por exemplo, não pode haver um par de chaves em um conjunto de tags com a mesma chave mas com valores diferentes, como projeto/Trinity e projeto/Xanadu.
Valor	Um valor é o valor opcional da tag. Consulte as restrições em Estrutura de tags no Amazon RDS. O valor da string pode ter de 1 a 256 caracteres Unicode e não pode ter os prefixos `aws:` ou `rds:`. A string pode conter apenas o conjunto de letras em Unicode, dígitos, espaços em branco, '_', '.', '/', '=', '+', '-' (Java regex: "^([\\p{L}\\ \p{Z}\\p{N}_.:/=+\\-]*)$"). Os valores não têm que ser exclusivos em um conjunto de tags e podem ser nulos. Por exemplo, você pode ter um par de chave-valor em um conjunto de tags definido como projeto/Trinity e centro-custos/Trinity.

Tutorial: especificar quais instâncias de banco de dados devem ser interrompidas usando tags

Este tutorial pressupõe que você tenha várias instâncias de banco de dados em um ambiente de desenvolvimento ou teste. Você precisa manter essas instâncias de banco de dados por vários

dias. Algumas instâncias de banco de dados executam testes noturnos, enquanto outras podem ser interrompidas durante a noite e iniciadas novamente no dia seguinte.

O tutorial a seguir mostra como atribuir uma tag às instâncias de banco de dados que são adequadas para interrupção durante a noite. O tutorial mostra como um script pode detectar quais instâncias de banco de dados têm a tag e, então, interromper as instâncias marcadas. Neste exemplo, a parte de valor do par chave-valor não importa. A presença da tag `stoppable` significa que a instância de banco de dados tem essa propriedade definida pelo usuário.

No tutorial a seguir, os comandos e APIs de marcação funcionam com ARNs, o que permite que o RDS funcione perfeitamente em regiões da AWS, contas da AWS e diferentes tipos de recurso que possam ter nomes curtos idênticos. Você pode especificar o ARN em vez do ID da instância de banco de dados em comandos CLI que operam em instâncias de banco de dados.

Para especificar quais instâncias de banco de dados devem ser interrompidas

1. Primeiro, determine o ARN de uma instância de banco de dados que você queira designar como passível de ser interrompida.

   No exemplo a seguir, substitua o nome de suas próprias instâncias de banco de dados por *dev-test-db-instance*. Em comandos subsequentes que usam parâmetros ARN, substitua o ARN de sua própria instância de banco de dados. O ARN inclui seu próprio ID de conta daAWS e o nome da região daAWS onde sua instância de Bancos de Dados está localizada.

   ```
   $ aws rds describe-db-instances --db-instance-identifier dev-test-db-instance \
     --query "*[].{DBInstance:DBInstanceArn}" --output text
   arn:aws:rds:us-east-1:123456789102:db:dev-test-db-instance
   ```

2. Adicione a tag `stoppable` a essa instância de banco de dados.

   Selecione o nome dessa tag. Como esse exemplo trata a tag como um atributo presente ou ausente, ele omite a parte `Value=` do parâmetro `--tags`. Essa abordagem significa que você pode evitar a criação de uma convenção de nomenclatura que codifique todas as informações relevantes nos nomes. Nessa convenção, você pode codificar informações no nome da instância de banco de dados ou nos nomes de outros recursos.

   ```
   $ aws rds add-tags-to-resource \
     --resource-name arn:aws:rds:us-east-1:123456789102:db:dev-test-db-instance \
     --tags Key=stoppable
   ```

3. Confirme se a tag está presente na instância de banco de dados.

   Os comandos a seguir recuperam as informações de tag para a instância de banco de dados no formato JSON e em texto separado por tabulação simples.

```
$ aws rds list-tags-for-resource \
  --resource-name arn:aws:rds:us-east-1:123456789102:db:dev-test-db-instance
{
    "TagList": [
        {
            "Key": "stoppable",
            "Value": ""

        }
    ]
}
aws rds list-tags-for-resource \
  --resource-name arn:aws:rds:us-east-1:123456789102:db:dev-test-db-instance --
output text
TAGLIST stoppable
```

4. Interrompa todas as instâncias de banco de dados designadas como `stoppable`.

   O exemplo a seguir cria um arquivo de texto que lista todas as instâncias de banco de dados. O comando shell percorre a lista, verifica se cada instância de banco de dados está marcada com o atributo relevante e executa o comando `aws rds stop-db-instance` para cada instância de banco de dados.

```
$ aws rds describe-db-instances --query "*[].[DBInstanceArn]" --output text >/tmp/
db_instance_arns.lst
$ for arn in $(cat /tmp/db_instance_arns.lst)
do
  match="$(aws rds list-tags-for-resource --resource-name $arn --output text | grep
 stoppable)"
  if [[ ! -z "$match" ]]
  then
      echo "DB instance $arn is tagged as stoppable. Stopping it now."
# Note that you need to get the DB instance identifier from the ARN.
      dbid=$(echo $arn | sed -e 's/.*://')
      aws rds stop-db-instance --db-instance-identifier $dbid
  fi
done
```

```
DB instance arn:arn:aws:rds:us-east-1:123456789102:db:dev-test-db-instance is
 tagged as stoppable. Stopping it now.
{
    "DBInstance": {
        "DBInstanceIdentifier": "dev-test-db-instance",
        "DBInstanceClass": "db.t3.medium",
        ...
```

Você pode executar um script como o anterior no final de todos os dias para garantir que as instâncias de banco de dados não essenciais sejam interrompidas. Você também pode agendar um trabalho com um utilitário, como `cron`, para realizar essa verificação todas as noites. Por exemplo, você pode fazer isso caso algumas instâncias de banco de dados sejam deixadas em execução por engano. Nesse caso, você pode ajustar o comando que prepara a lista de instâncias de banco de dados para conferir.

O comando a seguir produz uma lista de suas instâncias de banco de dados, mas apenas as que estão no estado `available`. O script pode ignorar as instâncias de banco de dados que já estão paradas, pois elas terão valores de status diferentes, como `stopped` ou `stopping`.

```
$ aws rds describe-db-instances \
  --query '*[].{DBInstanceArn:DBInstanceArn,DBInstanceStatus:DBInstanceStatus}|[?
DBInstanceStatus == `available`]|[].{DBInstanceArn:DBInstanceArn}' \
  --output text
arn:aws:rds:us-east-1:123456789102:db:db-instance-2447
arn:aws:rds:us-east-1:123456789102:db:db-instance-3395
arn:aws:rds:us-east-1:123456789102:db:dev-test-db-instance
arn:aws:rds:us-east-1:123456789102:db:pg2-db-instance
```

> **Tip**
>
> Você pode usar a atribuição de tags e encontrar instâncias de banco de dados com essas tags a fim de reduzir custos de outras maneiras. Por exemplo, considere esse cenário com instâncias de banco de dados utilizadas para desenvolvimento e testes. Nesse caso, você pode designar que algumas instâncias de banco de dados sejam excluídas no final de cada dia. Ou você pode designar que suas instâncias de banco de dados sejam alteradas para classes de instância de banco de dados pequenas durante os períodos de baixa utilização esperada.

Nomes de recurso da Amazon (ARNs) no Amazon RDS

Os recursos criados na Amazon Web Services são identificados de forma exclusiva com um nome de recurso da Amazon (ARN). Para determinadas operações do Amazon RDS, você precisará identificar exclusivamente um recurso do Amazon RDS especificando seu ARN. Por exemplo, quando você cria uma réplica de leitura da instância de banco de dados do RDS, é necessário fornecer o ARN para a instância de banco de dados de origem.

Para ter informações sobre como criar um ARN e obter um ARN existente, consulte os tópicos a seguir.

Tópicos

- Criar um ARN para o Amazon RDS
- Obter um ARN existente para o Amazon RDS

Criar um ARN para o Amazon RDS

Os recursos criados na Amazon Web Services são identificados de forma exclusiva com um nome de recurso da Amazon (ARN). Você pode criar um ARN para um recurso do Amazon RDS usando a seguinte sintaxe.

```
arn:aws:rds:<region>:<account number>:<resourcetype>:<name>
```

Nome da região	Região	Endpoint	Protocolo	
Leste dos EUA (Ohio)	us-east-2	rds.us-east-2.amazonaws.com	HTTPS	
		rds-fips.us-east-2.api.aws	HTTPS	
		rds.us-east-2.api.aws	HTTPS	
		rds-fips.us-east-2.amazonaws.com	HTTPS	
Leste dos EUA (Norte da Virgínia)	us-east-1	rds.us-east-1.amazonaws.com	HTTPS	
		rds-fips.us-east-1.api.aws	HTTPS	
		rds-fips.us-east-1.amazonaws.com	HTTPS	

Nome da região	Região	Endpoint	Protocolo
		rds.us-east-1.api.aws	HTTPS
Oeste dos EUA (N. da Califórnia)	us-west-1	rds.us-west-1.amazonaws.com	HTTPS
		rds.us-west-1.api.aws	HTTPS
		rds-fips.us-west-1.amazonaws.com	HTTPS
		rds-fips.us-west-1.api.aws	HTTPS
Oeste dos EUA (Oregon)	us-west-2	rds.us-west-2.amazonaws.com	HTTPS
		rds-fips.us-west-2.amazonaws.com	HTTPS
		rds.us-west-2.api.aws	HTTPS
		rds-fips.us-west-2.api.aws	HTTPS
África (Cidade do Cabo)	af-south-1	rds.af-south-1.amazonaws.com	HTTPS
		rds.af-south-1.api.aws	HTTPS
Ásia-Pacífico (Hong Kong)	ap-east-1	rds.ap-east-1.amazonaws.com	HTTPS
		rds.ap-east-1.api.aws	HTTPS
Ásia-Pacífico (Hyderabad)	ap-south-2	rds.ap-south-2.amazonaws.com	HTTPS
		rds.ap-south-2.api.aws	HTTPS
Ásia-Pacífico (Jacarta)	ap-southeast-3	rds.ap-southeast-3.amazonaws.com	HTTPS
		rds.ap-southeast-3.api.aws	HTTPS

Nome da região	Região	Endpoint	Protocolo
Ásia-Pacífico (Malásia)	ap-southeast-5	rds.ap-southeast-5.amazonaws.com	HTTPS
Ásia-Pacífico (Melbourne)	ap-southeast-4	rds.ap-southeast-4.amazonaws.com rds.ap-southeast-4.api.aws	HTTPS HTTPS
Ásia-Pacífico (Mumbai)	ap-south-1	rds.ap-south-1.amazonaws.com rds.ap-south-1.api.aws	HTTPS HTTPS
Ásia-Pacífico (Osaka)	ap-northeast-3	rds.ap-northeast-3.amazonaws.com rds.ap-northeast-3.api.aws	HTTPS HTTPS
Ásia-Pacífico (Seul)	ap-northeast-2	rds.ap-northeast-2.amazonaws.com rds.ap-northeast-2.api.aws	HTTPS HTTPS
Ásia-Pacífico (Singapura)	ap-southeast-1	rds.ap-southeast-1.amazonaws.com rds.ap-southeast-1.api.aws	HTTPS HTTPS
Ásia-Pacífico (Sydney)	ap-southeast-2	rds.ap-southeast-2.amazonaws.com rds.ap-southeast-2.api.aws	HTTPS HTTPS
Ásia-Pacífico (Taipei)	ap-east-2	rds.ap-east-2.amazonaws.com	HTTPS

Nome da região	Região	Endpoint	Protocolo
Ásia-Pacífico (Tailândia)	ap-southeast-7	rds.ap-southeast-7.amazonaws.com	HTTPS
Ásia-Pacífico (Tóquio)	ap-northeast-1	rds.ap-northeast-1.amazonaws.com	HTTPS
		rds.ap-northeast-1.api.aws	HTTPS
Canadá (Central)	ca-central-1	rds.ca-central-1.amazonaws.com	HTTPS
		rds.ca-central-1.api.aws	HTTPS
		rds-fips.ca-central-1.api.aws	HTTPS
		rds-fips.ca-central-1.amazonaws.com	HTTPS
Oeste do Canadá (Calgary)	ca-west-1	rds.ca-west-1.amazonaws.com	HTTPS
		rds-fips.ca-west-1.amazonaws.com	HTTPS
Europa (Frankfurt)	eu-central-1	rds.eu-central-1.amazonaws.com	HTTPS
		rds.eu-central-1.api.aws	HTTPS
Europa (Irlanda)	eu-west-1	rds.eu-west-1.amazonaws.com	HTTPS
		rds.eu-west-1.api.aws	HTTPS
Europa (Londres)	eu-west-2	rds.eu-west-2.amazonaws.com	HTTPS
		rds.eu-west-2.api.aws	HTTPS
Europa (Milão)	eu-south-1	rds.eu-south-1.amazonaws.com	HTTPS
		rds.eu-south-1.api.aws	HTTPS

Nome da região	Região	Endpoint	Protocolo
Europa (Paris)	eu-west-3	rds.eu-west-3.amazonaws.com	HTTPS
		rds.eu-west-3.api.aws	HTTPS
Europa (Espanha)	eu-south-2	rds.eu-south-2.amazonaws.com	HTTPS
		rds.eu-south-2.api.aws	HTTPS
Europa (Estocolmo)	eu-north-1	rds.eu-north-1.amazonaws.com	HTTPS
		rds.eu-north-1.api.aws	HTTPS
Europa (Zurique)	eu-central-2	rds.eu-central-2.amazonaws.com	HTTPS
		rds.eu-central-2.api.aws	HTTPS
Israel (Tel Aviv)	il-central-1	rds.il-central-1.amazonaws.com	HTTPS
		rds.il-central-1.api.aws	HTTPS
México (Central)	mx-central-1	rds.mx-central-1.amazonaws.com	HTTPS
Oriente Médio (Barém)	me-south-1	rds.me-south-1.amazonaws.com	HTTPS
		rds.me-south-1.api.aws	HTTPS
Oriente Médio (Emirados Árabes Unidos)	me-central-1	rds.me-central-1.amazonaws.com	HTTPS
		rds.me-central-1.api.aws	HTTPS
América do Sul (São Paulo)	sa-east-1	rds.sa-east-1.amazonaws.com	HTTPS
		rds.sa-east-1.api.aws	HTTPS

Nome da região	Região	Endpoint	Protocolo
AWS GovCloud (Leste dos EUA)	us-gov-east-1	rds.us-gov-east-1.amazonaws.com rds.us-gov-east-1.api.aws	HTTPS HTTPS
AWS GovCloud (Oeste dos EUA)	us-gov-west-1	rds.us-gov-west-1.amazonaws.com rds.us-gov-west-1.api.aws	HTTPS HTTPS

A tabela a seguir mostra o formato que deve ser usado para criar um ARN para um tipo de recurso específico do Amazon RDS.

Tipo de recurso	Formato ARN
Instância de banco de dados	arn:aws:rds:*<region>*:*<account>* :db:*<name>* Por exemplo: `arn:aws:rds: us-east-2 :123456789012 :db:my-mysql-instance-1`
Cluster de banco de dados	arn:aws:rds:*<region>*:*<account>* :cluster:*<name>* Por exemplo: `arn:aws:rds: us-east-2 :123456789012 :cluster: my-aurora-cluster-1`
Assinatura de eventos	arn:aws:rds:*<region>*:*<account>* :es:*<name>* Por exemplo:

Tipo de recurso	Formato ARN
	`arn:aws:rds: us-east-2 :123456789012 :es:my-subscription`
Grupo de opções do banco de dados	arn:aws:rds:*<region>*:*<account>* :og:*<name>* Por exemplo: `arn:aws:rds: us-east-2 :123456789012 :og:my-og`
DB parameter group (Grupo de parâmetros de banco de dados)	arn:aws:rds:*<region>*:*<account>* :pg:*<name>* Por exemplo: `arn:aws:rds: us-east-2 :123456789012 :pg:my-param-enable-logs`
Parameter group do cluster de banco de dados	arn:aws:rds:*<region>*:*<account>* :cluster-pg:*<name>* Por exemplo: `arn:aws:rds: us-east-2 :123456789012 :cluster-pg: my-cluster-param-timezone`
Instância de banco de dados reservada	arn:aws:rds:*<region>*:*<account>* :ri:*<name>* Por exemplo: `arn:aws:rds: us-east-2 :123456789012 :ri:my-reserved-postgresql`
Grupo de segurança de banco de dados	arn:aws:rds:*<region>*:*<account>* :secgrp:*<name>* Por exemplo: `arn:aws:rds: us-east-2 :123456789012 :secgrp:my-public`

Tipo de recurso	Formato ARN
Snapshot de banco de dados automatizado	arn:aws:rds:*<region>*:*<account>* :snapshot:rds:*<name>* Por exemplo: `arn:aws:rds: us-east-2 :123456789012 :snapshot:rds: my-mysql-db-2019-07-22-07-23`
Snapshot de cluster de banco de dados automatizado	arn:aws:rds:*<region>*:*<account>* :cluster-snapshot:rds:*<name>* Por exemplo: `arn:aws:rds: us-east-2 :123456789012 :cluster-snapshot:rds: my-aurora-cluster-2019-07-22-16-16`
Snapshot de banco de dados manual	arn:aws:rds:*<region>*:*<account>* :snapshot:*<name>* Por exemplo: `arn:aws:rds: us-east-2 :123456789012 :snapshot: my-mysql-db-snap`
Snapshot de cluster de banco de dados manual	arn:aws:rds:*<region>*:*<account>* :cluster-snapshot:*<name>* Por exemplo: `arn:aws:rds: us-east-2 :123456789012 :cluster-snapshot: my-aurora-cluster-snap`
Grupo de sub-rede de banco de dados	arn:aws:rds:*<region>*:*<account>* :subgrp:*<name>* Por exemplo: `arn:aws:rds: us-east-2 :123456789012 :subgrp:my-subnet-10`

Obter um ARN existente para o Amazon RDS

Você pode obter o ARN de um recurso do RDS usando o AWS Management Console, a AWS Command Line Interface (AWS CLI) ou a API do RDS.

Console

Para obter um ARN do AWS Management Console, navegue até o recurso para o qual deseja um ARN e veja os detalhes desse recurso.

Por exemplo, é possível obter o ARN de uma instância de banco de dados da guia Configuração dos detalhes da instância de banco de dados.

AWS CLI

Para obter um ARN a partir da AWS CLI para um recurso do RDS específico, use o comando `describe` para esse recurso. A tabela a seguir mostra cada comando da AWS CLI e a propriedade do ARN usada com o comando para obter um ARN.

AWS CLI command	Propriedade do ARN
describe-event-subscriptions	EventSubscriptionArn
describe-certificates	CertificateArn
describe-db-parameter-groups	DBParameterGroupArn
describe-db-cluster-parameter-groups	DBClusterParameterGroupArn
describe-db-instances	DBInstanceArn
describe-db-security-groups	DBSecurityGroupArn
describe-db-snapshots	DBSnapshotArn
describe-events	SourceArn
describe-reserved-db-instances	ReservedDBInstanceArn
describe-db-subnet-groups	DBSubnetGroupArn

AWS CLI command	Propriedade do ARN
describe-option-groups	OptionGroupArn
describe-db-clusters	DBClusterArn
describe-db-cluster-snapshots	DBClusterSnapshotArn

Por exemplo, o seguinte comando da AWS CLI obtém o ARN para uma instância de banco de dados.

Example

Para Linux, macOS ou Unix:

```
aws rds describe-db-instances \
--db-instance-identifier DBInstanceIdentifier \
--region us-west-2 \
--query "*[].{DBInstanceIdentifier:DBInstanceIdentifier,DBInstanceArn:DBInstanceArn}"
```

Para Windows:

```
aws rds describe-db-instances ^
--db-instance-identifier DBInstanceIdentifier ^
--region us-west-2 ^
--query "*[].{DBInstanceIdentifier:DBInstanceIdentifier,DBInstanceArn:DBInstanceArn}"
```

A saída desse comando é semelhante à seguinte:

```
[
    {
        "DBInstanceArn": "arn:aws:rds:us-west-2:account_id:db:instance_id",
        "DBInstanceIdentifier": "instance_id"
    }
]
```

API do RDS

Para obter um ARN para um recurso do RDS específico, é possível chamar as seguintes operações da API do RDS e usar as propriedades do ARN mostradas a seguir.

Operação da API do RDS	Propriedade do ARN
DescribeEventSubscriptions	EventSubscriptionArn
DescribeCertificates	CertificateArn
DescribeDBParameterGroups	DBParameterGroupArn
DescribeDBClusterParameterGroups	DBClusterParameterGroupArn
DescribeDBInstances	DBInstanceArn
DescribeDBSecurityGroups	DBSecurityGroupArn
DescribeDBSnapshots	DBSnapshotArn
DescribeEvents	SourceArn
DescribeReservedDBInstances	ReservedDBInstanceArn
DescribeDBSubnetGroups	DBSubnetGroupArn
DescribeOptionGroups	OptionGroupArn
DescribeDBClusters	DBClusterArn
DescribeDBClusterSnapshots	DBClusterSnapshotArn

Trabalhar com armazenamento para instâncias de banco de dados do Amazon RDS

Para especificar como deseja armazenar seus dados no Amazon RDS, escolha um tipo de armazenamento e forneça um tamanho de armazenamento ao criar ou modificar uma instância de banco de dados. Mais tarde, você poderá aumentar a quantidade ou alterar o tipo de armazenamento modificando a instância de banco de dados. Para obter mais informações sobre qual tipo de armazenamento usar para sua workload, consulte Tipos de armazenamento do Amazon RDS.

Tópicos

- Aumento da capacidade de armazenamento da instância de banco de dados
- Gerenciar a capacidade automaticamente com a escalabilidade automática de armazenamento do Amazon RDS
- Atualizando o sistema de arquivos de armazenamento para uma instância de banco de dados
- Modificar as configurações de armazenamento SSD de IOPS provisionadas
- Alterações de armazenamento com alto consumo de E/S
- Modificar configurações do armazenamento SSD de uso geral (gp3)
- Usando um volume de registro dedicado (DLV)

Aumento da capacidade de armazenamento da instância de banco de dados

Se precisar de espaço para dados adicionais, é possível aumentar o armazenamento de uma instância de banco de dados existente. Para fazer isso, você pode usar o Console de Gerenciamento do Amazon RDS, a API do Amazon RDS ou a AWS Command Line Interface (AWS CLI) Para obter informações sobre limites de armazenamento, consulte Armazenamento de instâncias de banco de dados do Amazon RDS.

> **Note**
>
> Não é possível reduzir a quantidade de armazenamento para uma instância de banco de dados depois que ela foi alocada. Ao aumentar o armazenamento alocado, o aumento deve ser de pelo menos 10%. Ao tentar aumentar o valor em menos de 10%, você obtém um erro.

A escalabilidade do armazenamento de instâncias de banco de dados do RDS para SQL Server tem suporte somente para os tipos de armazenamento SSD de uso geral e SSD de IOPS provisionadas.

Para monitorar a quantidade de armazenamento livre para sua instância de banco de dados para poder responder quando necessário, recomendamos criar um alarme do Amazon CloudWatch. Para obter mais informações sobre como definir alarmes do CloudWatch, consulte Usar alarmes do CloudWatch.

A escalabilidade de armazenamento geralmente não gera nenhuma interrupção ou degradação na performance da instância de banco de dados. Depois de modificar o tamanho de armazenamento de uma instância de banco de dados, o status da instância de banco de dados é storage-optimization.

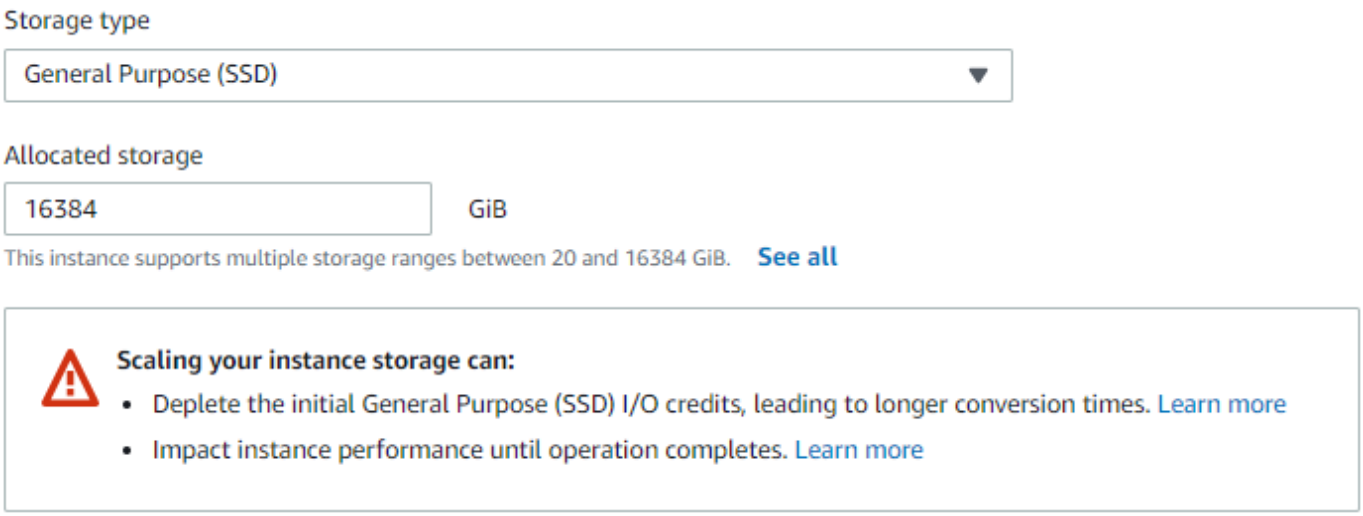
Note

A otimização de armazenamento pode demorar muitas horas. Não é possível fazer outras modificações de armazenamento por até seis (6) horas ou até a conclusão da otimização de armazenamento na instância, o que acontecer por último. Você pode ver o progresso da otimização do armazenamento no AWS Management Console ou usando o comando describe-db-instances da AWS CLI.

Console

Para aumentar o armazenamento para uma instância de banco de dados

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Databases (Bancos de dados).
3. Escolha a instância de banco de dados que você deseja modificar.
4. Selecione Modify.
5. Digite um novo valor em Allocated Storage (Armazenamento alocado). Ele deve ser maior que o valor atual.



6. Escolha Continue (Continuar) para ir para a próxima tela.
7. Escolha Apply immediately (Aplicar imediatamente) na seção Scheduling of modifications (Agendamento de modificações) para aplicar as alterações à instância de banco de dados imediatamente.

   Ou escolha Apply during the next scheduled maintenance window (Aplicar durante a próxima janela de manutenção programada) para aplicar as alterações na próxima janela de manutenção.
8. Quando as configurações estiverem de acordo com as suas preferências, escolha Modify DB instance (Modificar instância de banco de dados).

AWS CLI

Para aumentar o armazenamento de uma instância de banco de dados, use o comando da AWS CLI`modify-db-instance`. Defina os seguintes parâmetros:

- `--allocated-storage`: quantidade de armazenamento a ser alocada para a instância de banco de dados, em gibibytes.
- `--apply-immediately`: use `--apply-immediately` para aplicar as alterações de armazenamento de forma imediata.

  Ou use `--no-apply-immediately` (o padrão) para aplicar as alterações durante a próxima janela de manutenção. Ocorrerá uma interrupção imediata quando as alterações forem aplicadas.

Para obter mais informações sobre armazenamento, consulte Armazenamento de instâncias de banco de dados do Amazon RDS.

API do RDS

Para aumentar o armazenamento de uma instância de banco de dados, use a operação da API do Amazon RDS `ModifyDBInstance`. Defina os seguintes parâmetros:

- `AllocatedStorage`: quantidade de armazenamento a ser alocada para a instância de banco de dados, em gibibytes.
- `ApplyImmediately`: defina essa opção como `True` para aplicar imediatamente as alterações de armazenamento. Defina essa opção como `False` (o padrão) para aplicar as alterações durante a próxima janela de manutenção. Ocorrerá uma interrupção imediata quando as alterações forem aplicadas.

Para obter mais informações sobre armazenamento, consulte Armazenamento de instâncias de banco de dados do Amazon RDS.

Gerenciar a capacidade automaticamente com a escalabilidade automática de armazenamento do Amazon RDS

Se a sua workload for imprevisível, você poderá habilitar a escalabilidade automática para uma instância de banco de dados do Amazon RDS. Para fazer isso, você pode usar o console do Amazon RDS, a API do Amazon RDS ou a AWS CLI.

Por exemplo, você pode usar esse recurso para um novo aplicativo de jogos para dispositivos móveis que os usuários estejam adotando rapidamente. Nesse caso, uma workload que aumenta rapidamente pode exceder o armazenamento de banco de dados disponível. Para evitar ter que aumentar o armazenamento de banco de dados manualmente, você poderá usar escalabilidade automática do Amazon RDS.

Com a autoescalabilidade de armazenamento habilitada, quando o Amazon RDS detecta que seu espaço de banco de dados está acabando, ele aumenta automaticamente o armazenamento. O Amazon RDS inicia uma modificação de armazenamento para uma instância de banco de dados habilitada para a autoescalabilidade quando estes fatores se aplicam:

- O espaço livre for menor ou igual a 10% do espaço alocado.
- A condição de armazenamento baixo durar pelo menos 5 minutos.

- Ao menos seis horas se passaram desde a última modificação de armazenamento ou a conclusão da otimização de armazenamento na instância, o que tiver ocorrido há mais tempo.

O armazenamento adicional está em incrementos de qual destes for maior:

- 10 GiB
- 10% do armazenamento alocado atualmente
- Previsão de crescimento do armazenamento que excede o armazenamento atual alocado nas próximas sete horas, com base nas métricas de `FreeStorageSpace` da última hora. Para obter mais informações sobre métricas, consulte Monitoring with Amazon CloudWatch (Monitorar com o Amazon CloudWatch).

O limite máximo de armazenamento é o limite definido para autoscaling da instância de banco de dados. As seguintes restrições se aplicam:

- Você deve definir o limite máximo de armazenamento para pelo menos 10% a mais do que o armazenamento alocado atual. Recomendamos defini-lo no mínimo 26% acima para evitar o recebimento de uma notificação de evento avisando que o tamanho do armazenamento está se aproximando do limite máximo.

  Por exemplo, se você tiver uma instância de banco de dados com 1.000 GiB de armazenamento alocado, defina o limite máximo de armazenamento para pelo menos 1.100 GiB. Se não fizer isso, você receberá um erro como Invalid max storage size for *`engine_name`* (Tamanho máximo de armazenamento inválido para engine_name). No entanto, recomendamos definir o limite máximo de armazenamento para pelo menos 1.260 GiB para evitar a notificação de evento.
- Para uma instância de banco de dados que usa armazenamento de IOPS provisionadas (io1 ou io2 Block Express), a proporção de IOPS com relação ao limite máximo de armazenamento (em GiB) deve estar dentro de determinado intervalo. Para ter mais informações, consulte Armazenamento SSD de IOPS provisionadas.
- Não é possível definir o limite máximo de armazenamento para instâncias habilitadas para a escalabilidade automática como um valor maior que o armazenamento máximo alocado para o mecanismo de banco de dados e a classe de instância de banco de dados.

  Por exemplo, o SQL Server Standard Edition na db.m5.xlarge possui um armazenamento alocado padrão para a instância de 20 GiB (o mínimo) e um armazenamento alocado máximo de 16.384 GiB. O limite máximo de armazenamento padrão para escalabilidade automática é de 1.000 GiB.

Se você usar esse padrão, a instância não será dimensionada automaticamente acima de 1.000 GiB. Isso ocorre mesmo que o armazenamento máximo alocado para a instância seja 16.384 GiB.

> **Note**
>
> Recomendamos que você escolha cuidadosamente o limite máximo de armazenamento com base nos padrões de uso e nas necessidades do cliente. Se houver aberrações nos padrões de uso, o limite máximo de armazenamento poderá impedir a escalabilidade do armazenamento para um valor inesperadamente alto quando o autoscaling estimar um limite muito alto. Depois que uma instância de banco de dados é dimensionada automaticamente, seu armazenamento alocado não pode ser reduzido.

Tópicos

- Limitações do ajuste de escala automático de armazenamento
- Habilitar a escalabilidade automática de armazenamento para uma nova instância de banco de dados
- Alterar as configurações de escalabilidade automática de armazenamento para uma instância de banco de dados
- Como fazer a redução vertical ou horizontal da escala da instância de banco de dados de forma manual
- Desativar a escalabilidade automática de armazenamento para uma instância de banco de dados

Limitações do ajuste de escala automático de armazenamento

As seguintes limitações se aplicam à escalabilidade automática do armazenamento:

- A escalabilidade automática não ocorrerá se o limite máximo de armazenamento for excedido pelo incremento de armazenamento.
- Ao usar autoscaling, o RDS prevê o tamanho do armazenamento para operações de autoscaling subsequentes. Se houver previsão de que uma operação subsequente excederá o limite máximo de armazenamento, o RDS fará o autoscaling para o limite máximo de armazenamento.
- A autoescalabilidade não pode impedir totalmente situações de armazenamento completo para grandes cargas de dados. Isso ocorre porque não é possível fazer outras modificações de

armazenamento por até seis (6) horas ou até a conclusão da otimização de armazenamento na instância, o que ocorrer há mais tempo.

Se você executar um carregamento grande de dados e a escalabilidade automática não fornecer espaço suficiente, o banco de dados poderá permanecer no estado de armazenamento cheio por várias horas. Isso pode prejudicar o banco de dados.

- Se você iniciar uma operação de escalabilidade de armazenamento ao mesmo tempo que o Amazon RDS iniciar uma operação de escalabilidade automática, a modificação do armazenamento terá precedência. A operação de escalabilidade automática será cancelada.
- Não é possível diminuir o armazenamento alocado. Não é possível reduzir a quantidade de armazenamento para uma instância de banco de dados depois que ela foi alocada.
- A escalabilidade automática não pode ser usada com o armazenamento magnético.
- A escalabilidade automática não pode ser usada com as seguintes classes de instância de geração anterior que têm menos de 6 TiB de armazenamento ordenável: db.m3.large, db.m3.xlarge e db.m3.2xlarge.
- As operações de autoscaling não são registradas por AWS CloudTrail. Para obter mais informações sobre o CloudTrail, consulte Monitorar chamadas de API do Amazon RDSno AWS CloudTrail.

Embora a escalabilidade automática ajude a aumentar o armazenamento na sua Instância de banco de dados do Amazon RDS dinamicamente, você ainda deve configurar o armazenamento inicial de sua instância de banco de dados com um tamanho adequado para sua workload típica.

Habilitar a escalabilidade automática de armazenamento para uma nova instância de banco de dados

Ao criar uma nova instância de banco de dados do Amazon RDS, você pode optar pela escalabilidade automática de armazenamento. Também é possível definir um limite máximo de armazenamento que o Amazon RDS pode alocar para a instância de banco de dados.

> **Note**
>
> Quando você clonar uma instância de banco de dados do Amazon RDS que tenha a escalabilidade automática de armazenamento habilitada, essa configuração não será herdada automaticamente pela instância clonada. A nova instância de banco de dados terá a mesma quantidade de armazenamento alocado que a instância original. Você poderá

ativar a escalabilidade automática de armazenamento novamente para a nova instância se a instância clonada continuar a aumentar seus requisitos de armazenamento.

Console

Para habilitar a escalabilidade automática para uma nova instância de banco de dados

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No canto superior direito do console do Amazon RDS, escolha a região da AWS na qual você quer criar a instância de banco de dados.
3. No painel de navegação, escolha Databases (Bancos de dados).
4. Escolha Create database (Criar banco de dados). Na página Select engine (Selecionar mecanismo), escolha seu mecanismo de banco de dados e especifique as informações de sua instância de banco de dados conforme descrito em Conceitos básicos do Amazon RDS.
5. Na seção Storage autoscaling (Autoscaling do armazenamento), defina o valor de Maximum storage threshold (Limite máximo de armazenamento) para a instância de banco de dados.
6. Especifique as informações restantes da instância de banco de dados conforme descrito em Conceitos básicos do Amazon RDS.

AWS CLI

Para habilitar a escalabilidade automática de armazenamento para uma nova instância de banco de dados, use o comando `create-db-instance` da AWS CLI. Defina o seguinte parâmetro:

- `--max-allocated-storage`: ativa a autoescalabilidade de armazenamento e define o limite máximo de tamanho do armazenamento, em gibibytes.

Para verificar se a escalabilidade automática de armazenamento do Amazon RDS está disponível para sua instância de banco de dados, use o comando `describe-valid-db-instance-modifications` da AWS CLI. Para verificar com base na classe de instâncias antes de criar uma instância, use o comando `describe-orderable-db-instance-options`. Verifique o seguinte campo no valor retornado:

- `SupportsStorageAutoscaling`: indica se a instância de banco de dados ou a classe de instâncias oferece suporte à autoescalabilidade de armazenamento.

Para obter mais informações sobre armazenamento, consulte Armazenamento de instâncias de banco de dados do Amazon RDS.

API do RDS

Para habilitar a escalabilidade automática de armazenamento para uma nova instância de banco de dados, use a operação da API do Amazon RDS `CreateDBInstance`. Defina o seguinte parâmetro:

- `MaxAllocatedStorage`: ativa a autoescalabilidade de armazenamento do Amazon RDS e define o limite máximo de tamanho do armazenamento, em gibibytes.

Para verificar se a escalabilidade automática de armazenamento do Amazon RDS está disponível para a instância de banco de dados, use a operação `DescribeValidDbInstanceModifications` da API do Amazon RDS para uma instância existente ou a operação `DescribeOrderableDBInstanceOptions` antes de criar uma instância. Verifique o seguinte campo no valor retornado:

- `SupportsStorageAutoscaling`: indica se a instância de banco de dados oferece suporte à autoescalabilidade de armazenamento.

Para obter mais informações sobre armazenamento, consulte Armazenamento de instâncias de banco de dados do Amazon RDS.

Alterar as configurações de escalabilidade automática de armazenamento para uma instância de banco de dados

É possível ativar a escalabilidade automática de armazenamento para uma instância de banco de dados do Amazon RDS existente. Também é possível alterar o limite máximo de armazenamento que o Amazon RDS pode alocar para a instância de banco de dados.

Console

Para alterar as configurações de escalabilidade automática de armazenamento para uma instância de banco de dados

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Databases (Bancos de dados).

3. Selecione a instância de banco de dados que você deseja modificar e escolha Modify (Modificar). A página Modify DB instance (Modificar instância de banco de dados) será exibida.
4. Altere o limite de armazenamento na seção Autoscaling (Escalabilidade automática). Para obter mais informações, consulte Modificar uma instância de banco de dados do Amazon RDS.
5. Quando todas as alterações estiverem conforme o desejado, escolha Continue (Continuar) e verifique as modificações.
6. Na página de confirmação, revise suas alterações. Se estiverem corretas, escolha Modificar instância de banco de dados para salvar suas alterações. Se não estiverem corretas, escolha Back (Voltar) para editar as alterações ou escolha Cancel (Cancelar) para cancelar as alterações.

   A alteração do limite de escalabilidade automática do armazenamento ocorre imediatamente. Essa configuração ignora a configuração de Apply immediately.

AWS CLI

Para alterar as configurações de escalabilidade automática de armazenamento para uma instância de banco de dados, use o comando da `modify-db-instance` AWS CLI. Defina o seguinte parâmetro:

- `--max-allocated-storage`: define o limite máximo de tamanho do armazenamento, em gibibytes. Se o valor for maior que o parâmetro `--allocated-storage`, a escalabilidade automática de armazenamento será ativada. Se o valor for o mesmo que o parâmetro `--allocated-storage`, a escalabilidade automática de armazenamento será desativada

Para verificar se a escalabilidade automática de armazenamento do Amazon RDS está disponível para sua instância de banco de dados, use o comando da AWS CLI `describe-valid-db-instance-modifications`. Para verificar com base na classe de instâncias antes de criar uma instância, use o comando `describe-orderable-db-instance-options`. Verifique o seguinte campo no valor retornado:

- `SupportsStorageAutoscaling`: indica se a instância de banco de dados oferece suporte à autoescalabilidade de armazenamento.

Para obter mais informações sobre armazenamento, consulte Armazenamento de instâncias de banco de dados do Amazon RDS.

API do RDS

Para alterar as configurações de escalabilidade automática de armazenamento para uma instância de banco de dados, use a operação de API do Amazon RDS `ModifyDBInstance`. Defina o seguinte parâmetro:

- `MaxAllocatedStorage`: define o limite máximo de tamanho do armazenamento, em gibibytes.

Para verificar se a escalabilidade automática de armazenamento do Amazon RDS está disponível para a instância de banco de dados, use a operação `DescribeValidDbInstanceModifications` da API do Amazon RDS para uma instância existente ou a operação `DescribeOrderableDBInstanceOptions` antes de criar uma instância. Verifique o seguinte campo no valor retornado:

- `SupportsStorageAutoscaling`: indica se a instância de banco de dados oferece suporte à autoescalabilidade de armazenamento.

Para obter mais informações sobre armazenamento, consulte Armazenamento de instâncias de banco de dados do Amazon RDS.

Como fazer a redução vertical ou horizontal da escala da instância de banco de dados de forma manual

O Amazon RDS fornece escalabilidade automática de armazenamento para atender à demanda crescente. No entanto, existem limitações em relação à redução vertical ou horizontal da escala:

- Armazenamento do RDS: embora o RDS permita aumentar a escala do armazenamento verticalmente e de forma automática à medida que a demanda aumenta, a redução da escala vertical não é automática.
- Réplicas de leitura: o RDS não permite aumentar a escala horizontalmente (adição) ou reduzir a escala horizontalmente (exclusão) das réplicas de leitura. Você deve adicionar ou remover manualmente as réplicas de leitura de acordo com seus requisitos de carga.

Para reduzir a escala dos recursos do RDS verticalmente, execute as seguintes ações manuais:

- Com relação ao armazenamento, não é possível reduzir manualmente o armazenamento alocado de uma instância de banco de dados usando o comando `modify-db-instance`. Em vez disso, escolha uma das seguintes técnicas:

  - Use uma implantação azul/verde se seu mecanismo de banco de dados permitir. Crie um banco de dados verde com um tamanho de armazenamento menor e, em seguida, promova-o a banco de dados azul. Para ter mais informações, consulte Modificar as configurações de armazenamento e desempenho.
  - Crie uma instância de banco de dados com menor armazenamento alocado, migre manualmente os dados do banco de dados atual para a instância de banco de dados recém-criada e troque seus endpoints de banco de dados.
- Para réplicas de leitura, exclua manualmente todas as réplicas não utilizadas por meio da AWS CLI ou do console do RDS.

Desativar a escalabilidade automática de armazenamento para uma instância de banco de dados

Se não precisar mais que o Amazon RDS aumente automaticamente o armazenamento para uma instância de banco de dados do Amazon RDS, você poderá desativar a escalabilidade automática de armazenamento. Depois que fizer isso, você poderá aumentar manualmente a quantidade de armazenamento para sua instância de banco de dados.

Console

Para desativar a escalabilidade automática de armazenamento para uma instância de banco de dados

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Databases (Bancos de dados).
3. Selecione a instância de banco de dados que você deseja modificar e escolha Modify (Modificar). A página Modify DB instance (Modificar instância de banco de dados) será exibida.
4. Desmarque a caixa de seleção Enable storage autoscaling (Habilitar o autoscaling do armazenamento) na seção Storage autoscaling (Autoscaling do armazenamento). Para obter mais informações, consulte Modificar uma instância de banco de dados do Amazon RDS.
5. Quando todas as alterações estiverem conforme o desejado, escolha Continue (Continuar) e verifique as modificações.
6. Na página de confirmação, revise suas alterações. Se estiverem corretas, escolha Modificar instância de banco de dados para salvar suas alterações. Se não estiverem corretas, escolha

Back (Voltar) para editar as alterações ou escolha Cancel (Cancelar) para cancelar as alterações.

A alteração do limite de escalabilidade automática do armazenamento ocorre imediatamente. Essa configuração ignora a configuração de Apply immediately.

AWS CLI

Para desativar a escalabilidade automática para uma instância de banco de dados, use o comando da AWS CLI `modify-db-instance` e o seguinte parâmetro:

- `--max-allocated-storage`: especifique um valor igual à configuração `--allocated-storage` para evitar a autoescalabilidade de armazenamento do Amazon RDS adicional para a instância de banco de dados especificada.

Para obter mais informações sobre armazenamento, consulte Armazenamento de instâncias de banco de dados do Amazon RDS.

API do RDS

Para desativar a escalabilidade automática de armazenamento para uma instância de banco de dados, use a operação de API do Amazon RDS `ModifyDBInstance`. Defina o seguinte parâmetro:

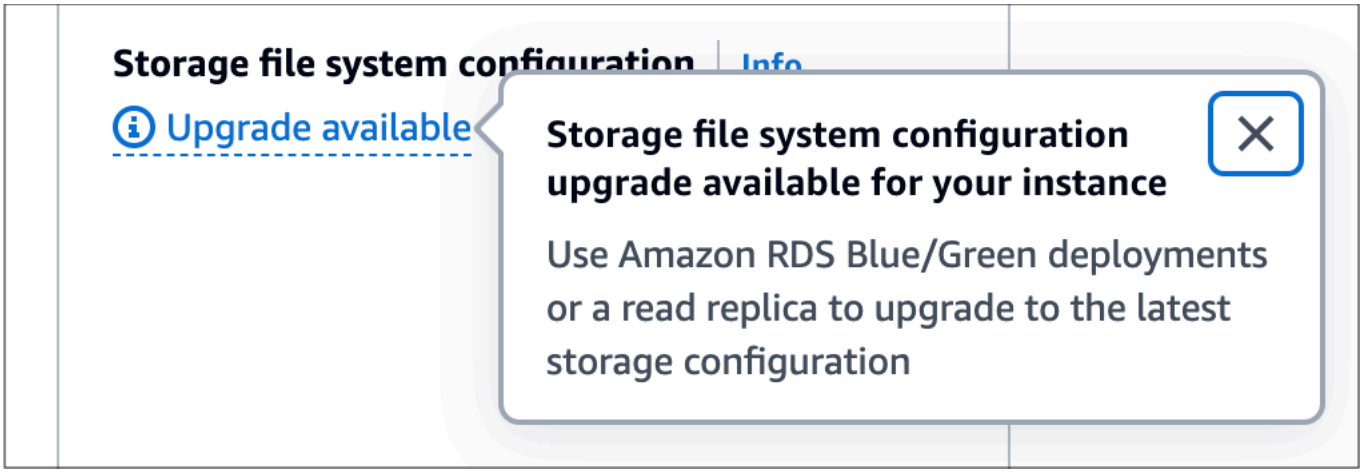
- `MaxAllocatedStorage`: especifique um valor igual à configuração `AllocatedStorage` para evitar a autoescalabilidade de armazenamento do Amazon RDS adicional para a instância de banco de dados especificada.

Para obter mais informações sobre armazenamento, consulte Armazenamento de instâncias de banco de dados do Amazon RDS.

Atualizando o sistema de arquivos de armazenamento para uma instância de banco de dados

A maioria das instâncias de banco de dados do RDS oferece um tamanho máximo de armazenamento de 64 TiB para bancos de dados do RDS para MariaDB, MySQL e PostgreSQL. Alguns sistemas de arquivos mais antigos de 32 bits podem ter capacidades de armazenamento mais baixas. Para determinar a capacidade de armazenamento da instância de banco de dados, use o comando describe-valid-db-instance-modifications da AWS CLI.

O RDS verifica se o sistema de armazenamento tem um tamanho de armazenamento de 16 TiB, um limite de tamanho de arquivo de 2 TiB ou gravações não otimizadas. Se suas instâncias de banco de dados atenderem a essas condições, o RDS alertará você de que a configuração do seu sistema de arquivos é elegível para uma atualização. Você pode verificar a elegibilidade de atualização da sua instância de banco de dados no painel Armazenamento da página de detalhes da instância de banco de dados.



Se sua instância de banco de dados for elegível para uma atualização do sistema de arquivos, use uma das seguintes técnicas:

- Crie uma implantação azul/verde e especifique Atualizar a configuração do sistema de arquivos de armazenamento.

  Essa opção atualiza o sistema de arquivos no ambiente verde para a configuração preferida. Em seguida, você pode fazer a transição da implantação azul/verde, o que promove o ambiente verde para o novo ambiente de produção. Para obter instruções detalhadas, consulte the section called “Criar uma implantação azul/verde”.
- Crie uma réplica de leitura da instância de banco de dados e especifique Atualizar a configuração do sistema de arquivos de armazenamento.

  Essa opção atualiza o sistema de arquivos da réplica de leitura para a configuração preferida. Você pode promover uma réplica de leitura a uma instância de banco de dados autônoma. Para obter instruções detalhadas, consulte the section called “Como criar uma réplica de leitura”.

Durante o processo de atualização do armazenamento, o mecanismo de banco de dados não fica disponível. A atualização da configuração de armazenamento é uma operação que exige muita E/S e leva a tempos de criação mais longos para réplicas de leitura e implantações azul/verdes. O

processo de atualização do armazenamento é mais rápido quando ambas as condições abaixo são atendidas:

- A instância de banco de dados de origem usa armazenamento SSD de IOPS provisionadas (io1 ou io2 Block Express).
- Você provisionou o ambiente verde ou a réplica de leitura com um tamanho de instância 4xlarge ou maior.

As atualizações de armazenamento que envolvem armazenamento SSD de uso geral (gp2) podem esgotar seu saldo de créditos de E/S, resultando em tempos de upgrade mais longos. Para ter mais informações, consulte the section called “Armazenamento da instância de banco de dados”.

Durante uma atualização do armazenamento, o RDS aumenta o tamanho de armazenamento alocado em 10% para a instância verde ou réplica de leitura se ambas as condições abaixo forem atendidas:

- O consumo de armazenamento na instância de banco de dados de origem é maior ou igual a 90% do tamanho de armazenamento alocado.
- O ajuste de escala automático está habilitado.

O RDS desativa o ajuste de escala automático quando o novo tamanho de armazenamento é maior ou igual ao armazenamento máximo alocado que foi definido para a instância. Se o ajuste de escala automático do armazenamento for desabilitado antes do início da atualização do armazenamento, o tamanho do armazenamento não aumentará durante a atualização.

Modificar as configurações de armazenamento SSD de IOPS provisionadas

Modifique as configurações para uma instância de banco de dados que usa o armazenamento SSD de IOPS provisionadas usando o console do Amazon RDS, a AWS CLI ou a API do Amazon RDS. Especifique o tipo de armazenamento, o armazenamento alocado e a quantidade de IOPS provisionadas necessários. O intervalo depende do mecanismo do banco de dados e do tipo da instância.

Embora seja possível reduzir a quantidade de IOPS provisionadas da instância, não é possível reduzir o tamanho do armazenamento.

Na maioria dos casos, dimensionar armazenamento não exige a interrupção nem afeta a performance do servidor. Depois de modificar as IOPS de armazenamento de uma instância de banco de dados, o status dessa instância será storage-optimization.

> **Note**
>
> A otimização de armazenamento pode demorar muitas horas. Não é possível fazer outras modificações de armazenamento por até seis (6) horas ou até a conclusão da otimização de armazenamento na instância, o que acontecer por último.

Para obter informações sobre os intervalos de armazenamento alocado e IOPS provisionadas disponíveis para cada mecanismo de banco de dados, consulte Armazenamento SSD de IOPS provisionadas.

Console

Para alterar as configurações de IOPS provisionadas para uma instância de banco de dados

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Databases (Bancos de dados).

   Para filtrar a lista de instâncias de banco de dados, em Filter databases (Filtrar bancos de dados), digite uma string de texto para que o Amazon RDS use nos filtros dos resultados. Serão exibidas apenas as instâncias de banco de dados cujos nomes contiverem a string.
3. Escolha a instância de banco de dados com IOPS provisionadas que você deseja modificar.
4. Selecione Modify.
5. Na página Modificar instância de banco de dados, selecione SSD de IOPS provisionadas (io1) ou SSD de IOPS provisionadas (io2) para Tipo de armazenamento.
6. Em Provisioned IOPS (IOPS provisionadas), insira um valor.

   Se o valor especificado para Allocated storage (Armazenamento alocado) ou Provisioned IOPS (IOPS provisionadas) estiver fora dos limites aceitos pelo outro parâmetro, uma mensagem de aviso será exibida. Essa mensagem fornece o intervalo de valores necessários para os outros parâmetros.
7. Escolha Continue.

8. Selecione Apply immediately (Aplicar imediatamente) na seção Scheduling of modifications (Agendamento de modificações) para aplicar as alterações à instância de banco de dados imediatamente. Ou escolha Apply during the next scheduled maintenance window (Aplicar durante a próxima janela de manutenção programada) para aplicar as alterações na próxima janela de manutenção.
9. Revise os parâmetros a serem alterados e escolha Modify DB instance (Modificar instância de banco de dados) para concluir a modificação.

   O novo valor para o armazenamento alocado ou para as IOPS provisionadas será exibido na coluna Status.

AWS CLI

Para alterar a configuração de IOPS provisionadas para uma instância de banco de dados, use o comando da AWS CLI `modify-db-instance`. Defina os seguintes parâmetros:

- `--storage-type`: defina como `io2` ou `io1` para IOPS provisionadas.
- `--allocated-storage`: quantidade de armazenamento a ser alocada para a instância de banco de dados, em gibibytes.
- `--iops`: a nova quantidade de IOPS provisionadas para a instância de banco de dados, expressa em operações de E/S por segundo.
- `--apply-immediately`: use `--apply-immediately` para aplicar as alterações imediatamente. Use `--no-apply-immediately` (o padrão) para aplicar as alterações durante a próxima janela de manutenção.

API do RDS

Para alterar as configurações de IOPS provisionadas para uma instância de banco de dados, use a operação de API do Amazon RDS `ModifyDBInstance`. Defina os seguintes parâmetros:

- `StorageType`: defina como `io2` ou `io1` para IOPS provisionadas.
- `AllocatedStorage`: quantidade de armazenamento a ser alocada para a instância de banco de dados, em gibibytes.
- `Iops`: a nova taxa de IOPS para a instância de banco de dados, expressa em operações de E/S por segundo.

- `ApplyImmediately`: defina essa opção como `True` para aplicar as alterações imediatamente. Defina essa opção como `False` (o padrão) para aplicar as alterações durante a próxima janela de manutenção.

Alterações de armazenamento com alto consumo de E/S

As instâncias de banco de dados do Amazon RDS utilizam volumes do Amazon Elastic Block Store (EBS) para armazenamento de log e banco de dados. Dependendo da quantidade de armazenamento solicitada, o RDS (exceto com RDS para SQL Server) cruza automaticamente múltiplos volumes de Amazon EBS para aprimorar a performance. As instâncias de banco de dados do RDS com tipos de armazenamento SSD são respaldadas por um ou quatro volumes cruzados do Amazon EBS em uma configuração RAID 0. Por padrão, as operações de modificação de armazenamento para uma instância de banco de dados do RDS têm um impacto mínimo nas operações contínuas do banco de dados.

Na maioria dos casos, as modificações na escalabilidade do armazenamento são completamente transferidas para a camada do Amazon EBS e são transparentes para o banco de dados. Normalmente, esse processo é concluído em alguns minutos. No entanto, alguns volumes de armazenamento do RDS mais antigos exigem um processo diferente para modificar o tamanho, as IOPS provisionadas ou o tipo de armazenamento. Isso envolve fazer uma cópia completa dos dados usando uma operação com potencial alto consumo de E/S.

A modificação do armazenamento usará uma operação com alto consumo de E/S se algum dos seguintes fatores se aplicar:

- O tipo de armazenamento da origem for magnético. O armazenamento magnético não oferecer suporte à modificação de volume elástico.
- A instância de banco de dados do RDS não estiver em um layout de um ou quatro volumes do Amazon EBS. Você pode visualizar o número de volumes do Amazon EBS em uso em suas instâncias de banco de dados do RDS usando métricas de monitoramento avançado. Para ter mais informações, consulte Como visualizar métricas do SO no console do RDS.
- O tamanho-alvo da solicitação de modificação aumenta o armazenamento alocado acima de 400 GiB para instâncias do RDS para MariaDB, MySQL e PostgreSQL, e 200 GiB para RDS para Oracle. As operações de autoescalabilidade do armazenamento têm o mesmo efeito quando aumentam o tamanho de armazenamento alocado da instância de banco de dados acima desses limites.

Se a modificação do armazenamento envolver uma operação com alto consumo de E/S, ela consumirá recursos de E/S e aumentará a carga da instância de banco de dados. As modificações de armazenamento com operações com alto consumo de E/S que envolvem armazenamento SSD de propósito geral (gp2) podem esgotar seu saldo de créditos de E/S, resultando em tempos de conversão mais longos.

Como prática recomendada, recomendamos programar essas solicitações de modificação de armazenamento fora dos horários de pico para ajudar a reduzir o tempo necessário para concluir a operação de modificação do armazenamento. Como alternativa, você pode criar uma réplica de leitura da instância de banco de dados e realizar a modificação do armazenamento na réplica de leitura. Depois, promova a réplica de leitura para que seja a instância de banco de dados primária. Para ter mais informações, consulte Trabalhar com réplicas de leitura de instância de banco de dados.

Para obter mais informações, consulte Por que uma instância de banco de dados do Amazon RDS fica travada no estado de modificação quando tento aumentar o armazenamento alocado?

Modificar configurações do armazenamento SSD de uso geral (gp3)

Modifique as configurações de uma instância de banco de dados que usa o armazenamento SSD de uso geral (gp3) utilizando o console do Amazon RDS, a AWS CLI ou a API do Amazon RDS. Especifique o tipo de armazenamento, o armazenamento alocado, a quantidade de IOPS provisionadas e o throughput de armazenamento necessários.

Embora seja possível reduzir a quantidade de IOPS provisionadas e de throughput de armazenamento da instância de banco de dados, não é possível reduzir o tamanho do armazenamento.

Na maioria dos casos, não é necessário realizar uma interrupção para escalar o armazenamento. Depois de modificar as IOPS de armazenamento de uma instância de banco de dados, o status dessa instância será storage-optimization. Você pode esperar latências elevadas, mas ainda no intervalo de milissegundo de um dígito, durante a otimização do armazenamento. A instância de banco de dados fica totalmente operacional após uma modificação de armazenamento.

> **Note**
>
> Não é possível fazer outras modificações de armazenamento até seis (6) horas após a conclusão da otimização de armazenamento na instância.

Para obter informações sobre os intervalos de armazenamento alocado, IOPS provisionadas e throughput disponíveis para cada mecanismo de banco de dados, consulte Armazenamento gp3 (recomendado).

Console

Como alterar as configurações de performance de armazenamento para uma instância de banco de dados

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Databases (Bancos de dados).

   Para filtrar a lista de instâncias de banco de dados, em Filter databases (Filtrar bancos de dados), digite uma string de texto para que o Amazon RDS use nos filtros dos resultados. Serão exibidas apenas as instâncias de banco de dados cujos nomes contiverem a string.
3. Selecione a instância de banco de dados com armazenamento gp3 que você deseja modificar.
4. Escolha Modificar.
5. Na página Modificar instância de banco de dados, selecione SSD de uso geral (gp3) em Tipo de armazenamento e faça o seguinte:

   a. Em Provisioned IOPS (IOPS provisionadas), selecione um valor.

      Se o valor especificado para Allocated storage (Armazenamento alocado) ou Provisioned IOPS (IOPS provisionadas) estiver fora dos limites aceitos pelo outro parâmetro, uma mensagem de aviso será exibida. Essa mensagem fornece o intervalo de valores necessários para os outros parâmetros.

   b. Em Storage throughput (Throughput de armazenamento), selecione um valor.

      Se o valor especificado para Provisioned IOPS (IOPS provisionadas) ou Storage throughput (Throughput de armazenamento) estiver fora dos limites aceitos pelo outro parâmetro, uma mensagem de aviso será exibida. Essa mensagem fornece o intervalo de valores necessários para os outros parâmetros.
6. Escolha Continue.
7. Selecione Apply immediately (Aplicar imediatamente) na seção Scheduling of modifications (Agendamento de modificações) para aplicar as alterações à instância de banco de dados imediatamente. Ou escolha Apply during the next scheduled maintenance window (Aplicar

durante a próxima janela de manutenção programada) para aplicar as alterações na próxima janela de manutenção.

8. Revise os parâmetros a serem alterados e escolha Modify DB instance (Modificar instância de banco de dados) para concluir a modificação.

   O novo valor de IOPS provisionadas é exibido na coluna Status.

AWS CLI

Para alterar as configurações de performance de armazenamento para uma instância de banco de dados, use o comando `modify-db-instance` da AWS CLI. Defina os seguintes parâmetros:

- `--storage-type`: defina como `gp3` para SSD de uso geral (gp3).
- `--allocated-storage`: quantidade de armazenamento a ser alocada para a instância de banco de dados, em gibibytes.
- `--iops`: a nova quantidade de IOPS provisionadas para a instância de banco de dados, expressa em operações de E/S por segundo.
- `--storage-throughput`: o novo throughput de armazenamento para a instância de banco de dados, expressa em MiBps.
- `--apply-immediately`: use `--apply-immediately` para aplicar as alterações imediatamente. Use `--no-apply-immediately` (o padrão) para aplicar as alterações durante a próxima janela de manutenção.

API do RDS

Para alterar as configurações de armazenamento para uma instância de banco de dados, utilize a operação de API do Amazon RDS `ModifyDBInstance`. Defina os seguintes parâmetros:

- `StorageType`: defina como `gp3` para SSD de uso geral (gp3).
- `AllocatedStorage`: quantidade de armazenamento a ser alocada para a instância de banco de dados, em gibibytes.
- `Iops`: a nova taxa de IOPS para a instância de banco de dados, expressa em operações de E/S por segundo.
- `StorageThroughput`: o novo throughput de armazenamento para a instância de banco de dados, expressa em MiBps.

- `ApplyImmediately`: defina essa opção como `True` para aplicar as alterações imediatamente. Defina essa opção como `False` (o padrão) para aplicar as alterações durante a próxima janela de manutenção.

Usando um volume de registro dedicado (DLV)

Use um volume dedicado de log (DLV) para uma instância de banco de dados que usa o armazenamento de IOPS provisionadas (PIOPS). Um DLV move os logs de transações do banco de dados do PostgreSQL e logs binários e redo logs do MySQL/MariaDB para um volume de armazenamento separado do volume que contém as tabelas do banco de dados. Um DLV torna o registro de gravação de transações mais eficiente e consistente. Os DLVs são ideais para bancos de dados com grande armazenamento alocado, altos requisitos de I/O por segundo (IOPS) ou cargas de trabalho sensíveis à latência.

Os DLVs são compatíveis com armazenamento PIOPS (io1 e io2 Block Express) e são criados com um tamanho fixo de 1.024 GiB e 3.000 IOPS provisionadas.

O Amazon RDS é compatível com DLVs em todas as Regiões da AWS para as seguintes versões:

- 10.6.7 e versões 10.6 posteriores
- MySQL 8.0.28 e versões posteriores a 8.0, MySQL 8.4.3 e versões posteriores a 8.4
- Todas as versões 11, 12 e 13, versões 10.4 e superior e versões 9.6.9 e superior do PostgreSQL

O Amazon RDS on é compatível com implantações multi-AZ. Quando você modifica ou cria uma instância multi-AZ, um DLV é criado tanto para a primária quanto para a secundária.

O RDS oferece suporte a DLVs com réplicas de leitura. Se a instância de banco de dados primária tiver um DLV ativado, todas as réplicas de leitura criadas após a ativação do DLV também terão um DLV. Qualquer réplica de leitura criada antes da mudança para o DLV não a terá habilitada, a menos que seja explicitamente modificada para isso. Recomendamos que todas as réplicas de leitura anexadas a uma instância primária antes da ativação do DLV também sejam modificadas manualmente para terem um DLV.

> **Note**
>
> Recomendamos DLVs para configurações de banco de dados de 5 TiB ou mais.

Para obter mais informações sobre os benefícios de DLVs, consulte as seguintes postagens no blog:

- Melhore o desempenho do banco de dados com volumes de log dedicados do Amazon RDS
- Benchmark do Amazon RDS para PostgreSQL com volumes de log dedicados
- Maximização do desempenho do AWS RDS para MySQL com volumes de log dedicados na documentação do Percona

Para obter informações sobre os intervalos de armazenamento alocado, IOPS provisionadas e throughput disponíveis para cada mecanismo de banco de dados, consulte Armazenamento SSD de IOPS provisionadas.

Tópicos

- Considerações ao habilitar e desabilitar o DLV
- Habilitar o DLV ao criar uma instância de banco de dados
- Habilitar o DLV em uma instância de banco de dados existente
- Monitorar o armazenamento DLV

Considerações ao habilitar e desabilitar o DLV

Habilitar e desabilitar o DLV pode ser demorado e causar tempo de inatividade. O processo envolve copiar todos os logs de transações ou redo logs e logs binários (dependendo do mecanismo de banco de dados) no novo volume ao habilitar ou de volta no armazenamento original ao desabilitar. A duração dessa operação é influenciada por vários fatores:

- Número de logs de transações:
  - Bancos de dados maiores com mais transações geram mais logs, aumentando o tempo necessário para cópia.
  - Os logs de transações poderão se acumular na instância do banco de dados primário se os slots de replicação estiverem inativos ou se a replicação estiver atrasada, aumentando o tempo necessário para a cópia. Verifique se a replicação está atualizada e remova todos os slots desnecessários.
- Configuração do armazenamento:
  - Largura de banda do EBS da instância de banco de dados: maior largura de banda permite uma transferência de dados mais rápida.

- Número de IOPS provisionadas: mais operações de entrada e saída por segundo (IOPS) podem acelerar o processo de cópia.
- Atividade do banco de dados: altos níveis de atividade do banco de dados durante a configuração podem retardar o processo.

Para minimizar o tempo de inatividade, recomendamos o planejamento e o agendamento durante períodos de baixa atividade ou janelas de manutenção.

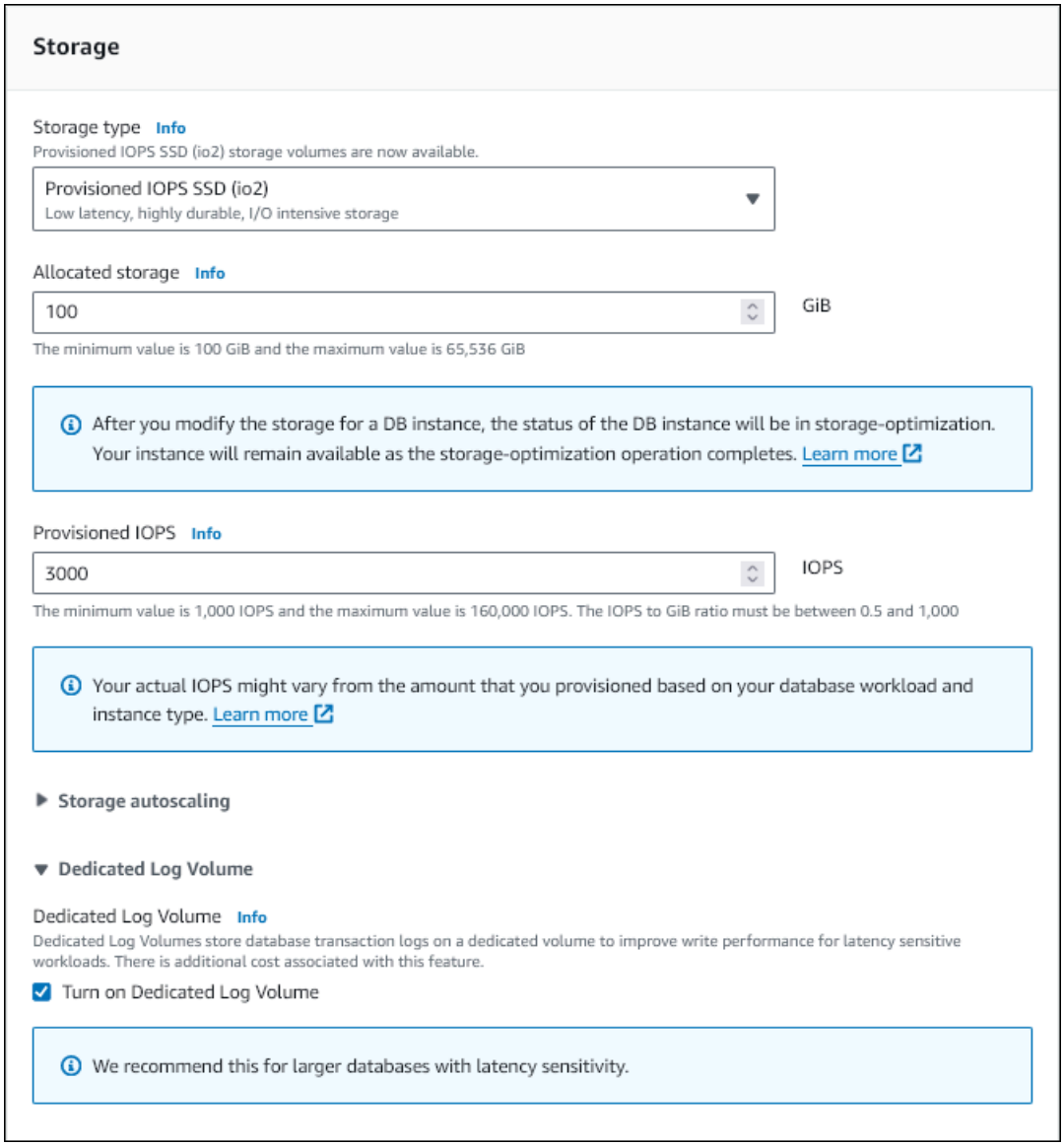
Habilitar o DLV ao criar uma instância de banco de dados

É possível usar o AWS Management Console, a AWS CLI ou a API do RDS para criar uma instância de banco de dados com o DLV habilitado.

Console

Como habilitar o DLV em uma nova instância de banco de dados

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. Escolha Create database (Criar banco de dados).
3. Na página Criar instância de banco de dados, escolha um mecanismo de banco de dados compatível com DLV.
4. Em Armazenamento:
    a. Escolha SSD de IOPS provisionadas (io1) ou SSD de IOPS provisionadas (io2).
    b. Insira o Armazenamento alocado e as IOPS provisionadas que você deseja.
    c. Expanda Volume dedicado de logs e selecione Ativar o volume dedicado de logs.



5. Selecione outras configurações, conforme necessário.
6. Selecione Criar banco de dados.

Depois que o banco de dados é criado, o valor do volume dedicado de logs aparece na guia Configuração da página de detalhes do banco de dados.

CLI

Para habilitar o DLV ao criar uma instância de banco de dados usando o armazenamento de IOPS provisionadas, use o comando create-db-instance da AWS CLI. Defina os seguintes parâmetros:

- `--dedicated-log-volume`: ativa um volume de logs dedicado.
- `--storage-type`: defina como `io2` ou `io1` para IOPS provisionadas.
- `--allocated-storage`: quantidade de armazenamento a ser alocada para a instância de banco de dados, em gibibytes.
- `--iops`: a quantidade de IOPS provisionadas para a instância de banco de dados, expressa em operações de E/S por segundo.

API do RDS

Para habilitar o DLV ao criar uma instância de banco de dados usando o armazenamento de IOPS provisionadas, use a operação de API CreateDBInstance do Amazon RDS. Defina os seguintes parâmetros:

- `DedicatedLogVolume`: defina como `true` para usar um volume de logs dedicado.
- `StorageType`: defina como `io2` ou `io1` para IOPS provisionadas.
- `AllocatedStorage`: quantidade de armazenamento a ser alocada para a instância de banco de dados, em gibibytes.
- `Iops`: a taxa de IOPS para a instância de banco de dados, expressa em operações de E/S por segundo.

Habilitar o DLV em uma instância de banco de dados existente

É possível usar o AWS Management Console, a AWS CLI ou a API do RDS para modificar uma instância de banco de dados para habilitar o DLV.

Depois de modificar a configuração de DLV para uma instância de banco de dados, é necessário reinicializar a instância de banco de dados.

Console

Como habilitar o DLV em uma instância de banco de dados existente

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Databases (Bancos de dados).

Para filtrar a lista de instâncias de banco de dados, em Filter databases (Filtrar bancos de dados), digite uma string de texto para que o Amazon RDS use nos filtros dos resultados. Serão exibidas apenas as instâncias de banco de dados cujos nomes contiverem a string.

3. Escolha a instância de banco de dados com o armazenamento de IOPS provisionadas que você deseja modificar.
4. Escolha Modificar.
5. Na página Modificar instância de banco de dados:
   - Em Armazenamento, expanda Volume dedicado de logs e selecione Ativar o volume dedicado de logs.
6. Escolha Continuar.
7. Escolha Aplicar imediatamente para aplicar as alterações à instância de banco de dados imediatamente. Ou escolha Apply during the next scheduled maintenance window (Aplicar durante a próxima janela de manutenção programada) para aplicar as alterações na próxima janela de manutenção.
8. Revise os parâmetros a serem alterados e escolha Modify DB instance (Modificar instância de banco de dados) para concluir a modificação.

O novo valor do volume dedicado de logs aparece na guia Configuração da página de detalhes do banco de dados.

CLI

Para habilitar ou desabilitar o DLV em uma instância de banco de dados existente usando o armazenamento de IOPS provisionadas, use o comando. AWS CLI `modify-db-instance` Defina os seguintes parâmetros:

- `--dedicated-log-volume`: ativa um volume de logs dedicado.

  Use `--no-dedicated-log-volume` (o padrão) para desabilitar um volume de logs dedicado.
- `--apply-immediately`: use `--apply-immediately` para aplicar as alterações imediatamente.

  Use `--no-apply-immediately` (o padrão) para aplicar as alterações durante a próxima janela de manutenção.

API do RDS

Para habilitar ou desabilitar o DLV em uma instância de banco de dados existente usando o armazenamento de IOPS provisionadas, use a operação de API do Amazon RDS. `ModifyDBInstance` Defina os seguintes parâmetros:

- `DedicatedLogVolume`: defina essa opção como `true` para habilitar um volume de logs dedicado.

  Defina essa opção como `false` para desabilitar um volume de logs dedicado. Este é o valor padrão.
- `ApplyImmediately`: defina essa opção como `True` para aplicar as alterações imediatamente.

  Defina essa opção como `False` (o padrão) para aplicar as alterações durante a próxima janela de manutenção.

Monitorar o armazenamento DLV

É possível monitorar o uso do armazenamento DLV usando a métrica `FreeStorageSpaceLogVolume` no CloudWatch.

É possível usar a seguinte consulta para RDS para PostgreSQL para encontrar o tamanho ocupado pelos logs de transações:

```
SELECT pg_size_pretty(COALESCE(sum(size), 0)) AS total_wal_generated_size
FROM pg_catalog.pg_ls_waldir();
```

Se o DLV ficar sem armazenamento, a instância de banco de dados entrará no estado `storage-full`, causando tempo de inatividade.

Excluir uma instância de banco de dados

Você pode excluir uma instância de banco de dados usando o AWS Management Console, a AWS CLI ou a API do RDS. Se você quiser excluir uma instância de banco de dados de um cluster de banco de dados do Aurora, consulte Excluir clusters e instâncias de banco de dados do Aurora.

Tópicos

- Pré-requisitos para excluir uma instância de banco de dados
- Considerações ao excluir uma instância de banco de dados
- Excluir uma instância de banco de dados

Pré-requisitos para excluir uma instância de banco de dados

Antes de tentar excluir a instância de banco de dados, garanta que a proteção contra exclusão esteja desativada. Por padrão, a proteção contra exclusão está ativada para uma instância de banco de dados criada com o console.

Se sua instância de banco de dados tiver a proteção contra exclusão ativada, você poderá desativá-la modificando as configurações da instância. Selecione Modificar na página de detalhes do banco de dados ou chame o comando modify-db-instance. Essa operação não causa uma interrupção. Para ter mais informações, consulte Configurações para instâncias de banco de dados.

Considerações ao excluir uma instância de banco de dados

A exclusão de uma instância de banco de dados afeta a capacidade de recuperação da instância, a disponibilidade do backup e o status da réplica de leitura. Considere os seguintes problemas:

- É possível decidir se deseja criar um snapshot de banco de dados final. Você tem as seguintes opções:
  - Se você criar um snapshot final, poderá usá-lo para restaurar a instância de banco de dados excluída. O RDS retém o snapshot final e todos os snapshots manuais que criados anteriormente. Não é possível criar um snapshot de banco de dados final de sua instância de banco de dados se ela não está no estado `Available`. Para ter mais informações, consulte Visualizar o status de uma instância de banco de dados Amazon RDS em um .
  - Se você não criar um snapshot final, a exclusão da instância será mais rápida. A desvantagem é que não existe um snapshot final que você possa restaurar posteriormente. Se você decidir

restaurar a instância de banco de dados excluída, retenha backups automatizados ou use um snapshot manual anterior para restaurá-la até o momento do snapshot anterior.

- É possível decidir se deseja reter backups automatizados. Você tem as seguintes opções:
  - Se você retiver backups automatizados, o RDS os manterá pelo período de retenção em vigor para a instância de banco de dados no momento em que você a excluir. É possível usar backups automatizados para restaurar a instância de banco de dados para um período durante, mas não após o período de retenção. Esse período de retenção entra em vigor independentemente de você criar um snapshot final do banco de dados. Para excluir um backup automatizado retido, consulte Excluir backups automatizados retidos.
  - Os backups automatizados retidos e os snapshots manuais incorrerão em cobranças até serem excluídos. Para ter mais informações, consulte Custos de retenção.
  - Se você não retiver backups automatizados, o RDS excluirá os backups automatizados que estiverem na mesma Região da AWS que sua instância de banco de dados. Não é possível recuperar esses backups. Se os backups automatizados tiverem sido replicados em outra Região da AWS, o RDS os manterá mesmo que você não opte por reter backups automatizados. Para ter mais informações, consulte Replicar backups automatizados para outra Região da AWS.

    > **Note**
    >
    > Normalmente, se você criar um snapshot final do banco de dados, não será necessário reter backups automatizados.

- Quando você exclui sua instância de banco de dados, o RDS não exclui snapshots de banco de dados manuais. Para ter mais informações, consulte Criar um snapshot de banco de dados para uma instância de banco de dados single-AZ para o Amazon RDS.
- Se você quiser excluir todos os recursos do RDS, observe que os seguintes recursos geram cobranças:
  - Instâncias de banco de dados
  - DB snapshots
  - Clusters do banco de dados

  Se você comprou instâncias reservadas, elas serão cobradas de acordo com o contrato que você celebrou ao comprar a instância. Para ter mais informações, consulte Instâncias de banco de dados reservadas para o Amazon RDS. Você pode ter informações de cobrança de todos os seus

recursos da AWS usando o AWS Cost Explorer. Para ter mais informações, consulte Analisar seus custos com oAWS Cost Explorer.

- Se você excluir uma instância de banco de dados que tem réplicas de leitura na mesma Região da AWS, cada réplica de leitura será promovida a uma instância de banco de dados independente. Para ter mais informações, consulte Promoção de uma réplica de leitura a uma instância de banco de dados autônoma. Se sua instância de banco de dados tiver réplicas de leitura em diferentes Regiões da AWS, consulte Considerações sobre replicação entre regiões para ter informações relacionadas à exclusão da instância de banco de dados de origem de uma réplica de leitura entre regiões.
- Quando o status de uma instância de banco de dados for `deleting`, seu valor de certificado CA não será exibido no console do RDS nem na saída de comandos da AWS CLI ou de operações de API do RDS. Para ter mais informações sobre certificados CA, consulte Usar SSL/TLS para criptografar uma conexão com uma instância ou um cluster de banco de dados.
- O tempo necessário para excluir a instância de banco de dados varia dependendo do período de retenção do backup (ou seja, quantos backups serão excluídos), da quantidade de dados excluídos e se um snapshot final será criado.

Excluir uma instância de banco de dados

Você pode excluir uma instância de banco de dados usando o AWS Management Console, a AWS CLI ou a API do RDS. Faça o seguinte:

- Forneça o nome da instância de banco de dados
- Habilitar ou desabilitar a opção para fazer um snapshot de banco de dados final da instância
- Habilitar ou desabilitar a opção para reter backups automatizados

> **Note**
>
> Não é possível excluir uma instância de banco de dados quando a proteção contra exclusão está ativada. Para ter mais informações, consulte Pré-requisitos para excluir uma instância de banco de dados.

Console

Para excluir uma instância de banco de dados

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Databases (Bancos de dados) e a instância de banco de dados que você deseja excluir.
3. Em Actions, escolha Delete.
4. Para criar um snapshot de banco de dados final para a instância de banco de dados, escolha Create final snapshot? (Criar snapshot final?).
5. Se você optar por criar um snapshot final, insira o Final snapshot name (Nome do snapshot final).
6. Para reter backups automatizados, escolha Retain automated backups (Reter backups automatizados).
7. Digite **`delete me`** na caixa.
8. Escolha Excluir.

AWS CLI

Para encontrar os IDs das instâncias de banco de dados em sua conta, chame o comando describe-db-instances:

```
aws rds describe-db-instances --query 'DBInstances[*].[DBInstanceIdentifier]' --output
 text
```

Para excluir uma instância de banco de dados usando a AWS CLI, chame o comando delete-db-instance com as seguintes opções:

- `--db-instance-identifier`
- `--final-db-snapshot-identifier` ou `--skip-final-snapshot`

Example Com um snapshot final e sem backups automatizados retidos

Para Linux, macOS ou Unix:

```
aws rds delete-db-instance \
```

```
    --db-instance-identifier mydbinstance \
    --final-db-snapshot-identifier mydbinstancefinalsnapshot \
    --delete-automated-backups
```

Para Windows:

```
aws rds delete-db-instance ^
    --db-instance-identifier mydbinstance ^
    --final-db-snapshot-identifier mydbinstancefinalsnapshot ^
    --delete-automated-backups
```

Example Com backups automatizados retidos e sem snapshot final

Para Linux, macOS ou Unix:

```
aws rds delete-db-instance \
    --db-instance-identifier mydbinstance \
    --skip-final-snapshot \
    --no-delete-automated-backups
```

Para Windows:

```
aws rds delete-db-instance ^
    --db-instance-identifier mydbinstance ^
    --skip-final-snapshot ^
    --no-delete-automated-backups
```

API do RDS

Para excluir uma instância de banco de dados usando a API do Amazon RDS, chame a operação `DeleteDBInstance` com os seguintes parâmetros:

- `DBInstanceIdentifier`
- `FinalDBSnapshotIdentifier` ou `SkipFinalSnapshot`

Tutorial: Gerenciar um ambiente de instância de banco de dados do MySQL do desenvolvimento à produção

Introdução

Um caso comum ao gerenciar uma instância de banco de dados do Amazon RDS envolve a supervisão do ciclo de vida, desde o desenvolvimento inicial até a implantação na produção. Este tutorial oferece orientação para lidar com as principais tarefas a fim de garantir que o banco de dados tenha a performance ideal e se adapte para atender às suas necessidades operacionais em evolução. Além disso, ele descreve as opções para sincronizar as alterações feitas entre seus ambientes de desenvolvimento e de produção para garantir consistência e confiabilidade.

Ao concluir essas etapas, você aprende a:

- Como realizar tarefas específicas com instâncias de banco de dados do MySQL, como adicionar e atualizar tags do Amazon RDS, expandir o armazenamento, criar réplicas de leitura e excluir recursos.
- Como sincronizar atualizações de um ambiente de produção para um ambiente de desenvolvimento para testes e validação abrangentes.

Para concluir este tutorial, realize as seguintes tarefas:

1. Crie uma instância de banco de dados do MySQL.
2. Adicione tags do Amazon RDS para categorizar sua instância de banco de dados como um ambiente de desenvolvimento.
3. Aumente a capacidade de armazenamento da instância de banco de dados para acomodar workloads maiores.
4. Crie réplicas de leitura para aprimorar a resiliência e a disponibilidade da instância de banco de dados.
5. Atualize as tags do Amazon RDS para categorizar a instância de banco de dados como um ambiente de produção.
6. Exclua a instância de banco de dados que não é mais necessária para que ela não gere custos adicionais.
7. Próximas etapas: sincronizar a instância de desenvolvimento com a produção para ter consistência em todos os ambientes

Pré-requisitos

Antes de começar, conclua as etapas nas seguintes seções:

- Inscrever-se para uma Conta da AWS
- Criar um usuário com acesso administrativo

Adicionar tags para categorizar sua instância de banco de dados como um ambiente de desenvolvimento

Para categorizar a instância de banco de dados como um ambiente de desenvolvimento, adicione uma tag do Amazon RDS à instância que você criou. Tag do Amazon RDS é um par de nome/valor que você define e associa a um recurso do Amazon RDS. Marcar seus recursos AWS ajuda a distinguir entre seus recursos de desenvolvimento e de produção da AWS. Para ter mais informações sobre tags do Amazon RDS, consulte Marcar recursos do Amazon RDS.

1. No console do Amazon RDS, escolha Databases (Bancos de dados).
2. Selecione a instância de banco de dados que você deseja marcar.
3. Na seção de detalhes, role até a seção Tags.
4. Selecione Gerenciar tags e Adicionar nova tag.
5. Digite um valor para Tag key (Chave de tag) e Value (Valor). Por exemplo, você pode usar o ambiente de chave de tag com o valor dev para especificar que a instância do banco de dados faz parte do ambiente de desenvolvimento.
6. Selecione Adicionar nova tag e Salvar alterações.

Sua instância de banco de dados agora está marcada como um ambiente de desenvolvimento. Isso facilita a identificação da instância de banco de dados e o gerenciamento de custos associados a esse recurso.

Aumente a capacidade de armazenamento de uma instância de banco de dados para acomodar as crescentes necessidades de dados.

Depois, modifique a capacidade de armazenamento da instância de banco de dados do MySQL para acomodar dados adicionais. Inicialmente, a capacidade de armazenamento da instância de banco de dados é definida para atender às necessidades imediatas da aplicação. No entanto, à medida que

os volumes de dados aumentam, talvez seja necessário ajustar as configurações de armazenamento para garantir a performance e a estabilidade contínuas do banco de dados. Esse processo envolve aumentar o armazenamento alocado de sua instância de banco de dados. Para ter mais informações sobre como modificar a capacidade de armazenamento da instância de banco de dados, consulte Trabalhar com armazenamento para instâncias de banco de dados do Amazon RDS.

1. No console do Amazon RDS, escolha Databases (Bancos de dados).
2. Escolha a instância de banco de dados que você deseja modificar.
3. Escolha Modificar.
4. Em Armazenamento, aumente o Armazenamento alocado. O valor do armazenamento modificado deve ser maior que o atual.
5. Escolha Continuar.
6. Em Programação das modificações, é possível escolher Aplicar imediatamente para aplicar as alterações de armazenamento à instância de banco de dados imediatamente ou escolher Aplicar durante a próxima janela de manutenção programada para aplicar as alterações durante a próxima janela de manutenção.
7. Quando as configurações estiverem de acordo com as suas preferências, escolha Modify DB instance (Modificar instância de banco de dados).

A capacidade de armazenamento da sua instância de banco de dados agora está maior. Isso permite que ele lide com eficiência com volumes de dados maiores e garanta performance e estabilidade contínuas à medida que as necessidades de dados da aplicação aumentam.

Criar réplicas de leitura para aprimorar a resiliência e a disponibilidade de uma instância de banco de dados

Crie uma réplica de leitura da instância de banco de dados do MySQL. Réplicas de leitura aprimoram a resiliência e a disponibilidade da instância de banco de dados. Para reduzir o tráfego de leitura na instância de banco de dados primária, crie uma réplica de leitura da instância de banco de dados. Isso encaminha as consultas para a réplica de leitura, o que pode ajudar a distribuir a carga e melhorar a performance geral do banco de dados. Para ter mais informações sobre réplicas de leitura da instância de banco de dados, consulte Trabalhar com réplicas de leitura de instância de banco de dados.

Para que uma instância de banco de dados do MySQL possa servir como uma origem de replicação, é necessário ativar backups automáticos na instância de banco de dados de origem. Para fazer isso,

defina o período de retenção de backup como um valor diferente de 0. Para ter mais informações sobre réplicas de leitura do MySQL, consulte Trabalhar com réplicas de leitura MySQL.

1. No console do Amazon RDS, escolha Databases (Bancos de dados).
2. Selecionar a instância de banco de dados que você deseja usar como a origem da réplica de leitura.
3. Em Ações, selecione Criar réplica de leitura.
4. Em Identificador da instância de banco de dados, insira um nome para a réplica de leitura em letras minúsculas.
5. Escolha a configuração da instância. Recomendamos que você use a classe de instância de banco de dados e o tipo de armazenamento iguais ou superiores aos da instância de banco de dados de origem para a réplica de leitura.
6. Para Região da AWS, especifique a região de destino da réplica de leitura.
7. Deixe as configurações padrão ou modifique-as conforme necessário.
8. Escolha Create read replica (Criar réplica de leitura).

A réplica de leitura aparece abaixo da instância de banco de dados de origem na página Bancos de dados, no console do RDS. Ele mostra Réplica na coluna Perfil.

Atualizar as tags para categorizar uma instância de banco de dados como um ambiente de produção

Quando sua instância de banco de dados estiver pronta para passar da fase de desenvolvimento para a produção, é importante atualizar suas tags para refletir sua transição. Para alinhar a instância de banco de dados às suas estratégias operacionais e de monitoramento, atualize as tags iniciais para indicar que a instância de banco de dados agora faz parte do ambiente de produção. Isso garante melhor visibilidade e gerenciamento do banco de dados.

1. No console do Amazon RDS, escolha Databases (Bancos de dados).
2. Selecionar a instância de banco de dados que você deseja atualizar
3. Na seção de detalhes, role até a seção Tags.
4. Selecione Gerenciar tags.
5. Remova sua tag inicial que significa um ambiente de desenvolvimento.
6. Selecione Adicionar nova tag.

7. Digite um novo valor para Chave de tag e Valor. Por exemplo, você pode usar o ambiente de chave de tag com o valor prod para especificar que a instância de banco de dados faz parte do ambiente de produção.
8. Selecione Adicionar nova tag e Salvar alterações.

A tag na sua instância de banco de dados é atualizada para significar a transição do banco de dados para um ambiente de produção.

Excluir uma instância de banco de dados quando ela não for mais necessária para evitar custos adicionais.

Antes do final deste tutorial, é fundamental abordar o gerenciamento de seus recursos. Caso você tenha algum recurso que não seja mais necessário, exclua-o para evitar custos adicionais e otimizar o ambiente de nuvem.

1. No console do Amazon RDS, escolha Databases (Bancos de dados).
2. Selecionar a instância de banco de dados que você deseja excluir
3. Em Ações, selecione Excluir. A exclusão de uma instância de banco de dados excluirá permanentemente a instância com todo o seu conteúdo e recursos relacionados.
4. Confirme a exclusão da instância de banco de dados e selecione Excluir.

Como alternativa, se você optar por manter a instância de banco de dados para uso futuro, poderá continuar gerenciando-a como parte do ambiente de produção. Isso envolve a manutenção de um ambiente de desenvolvimento sincronizado para facilitar testes e validação abrangentes. Para ter mais informações, consulte Próximas etapas: sincronizar a instância de desenvolvimento com a produção para ter consistência em todos os ambientes.

Próximas etapas: sincronizar a instância de desenvolvimento com a produção para ter consistência em todos os ambientes

Criar um ambiente de desenvolvimento

Para gerenciar um ambiente de produção, é importante manter um ambiente de desenvolvimento sincronizado para testes e validação abrangentes. Para criar um ambiente de desenvolvimento, primeiro crie um snapshot de banco de dados da instância de banco de dados de produção atual. Um snapshot de banco de dados captura toda a instância de banco de dados criando um snapshot do volume de armazenamento. Para ter instruções sobre como criar um snapshot de banco de dados

no console do Amazon RDS, consulte Criar um snapshot de banco de dados para uma instância de banco de dados single-AZ para o Amazon RDS.

Depois de criar o snapshot de banco de dados do seu ambiente de produção, crie uma instância de banco de dados para seu ambiente de desenvolvimento restaurando um snapshot de banco de dados. As instâncias de banco de dados restauradas são associadas automaticamente aos grupos de parâmetros e de opções de banco de dados padrão. Porém, você pode aplicar um grupo de parâmetros personalizado e um grupo de opções personalizado especificando-os durante uma restauração. Para ter instruções sobre como restaurar de um snapshot de banco de dados, consulte Tutorial: restaurar uma instância de banco de dados do Amazon RDS de um snapshot do banco de dados.

Por fim, designe a nova instância de banco de dados como seu novo ambiente de desenvolvimento atualizando as tags do Amazon RDS. Para ter orientação sobre como atualizar as tags do Amazon RDS para refletir essa alteração, consulte a seção Atualizar as tags para categorizar uma instância de banco de dados como um ambiente de produção anterior.

Agora você tem um novo ambiente de desenvolvimento que espelha a configuração do banco de dados do ambiente de produção.

Sincronizar um ambiente de desenvolvimento com o ambiente de produção

Depois que seu novo ambiente de desenvolvimento for estabelecido, é necessário mantê-lo sincronizado com todas as alterações que ocorrerem no ambiente de produção. Isso garante que seu ambiente de desenvolvimento reflita com precisão o estado atual da produção, o que é essencial para testes, validação e solução de problemas eficazes. O Amazon RDS oferece uma série de maneiras diferentes de manter o ambiente de desenvolvimento atualizado com o ambiente de produção. Para ter mais informações sobre essas opções, consulte Orchestrating database refreshes for Amazon RDS and Amazon Aurora.

Uma das principais formas de sincronizar os ambientes de desenvolvimento e de produção é criando e restaurando snapshots de banco de dados. Um snapshot de banco de dados permite que você crie um ambiente de desenvolvimento que reflita a configuração do banco de dados do ambiente de produção durante o período em que o snapshot foi criado. Para ter mais informações sobre snapshots de banco de dados, consulte Gerenciar backups manuais. Para ter informações sobre como restaurar uma instância de banco de dados, consulte Restaurar uma instância de banco de dados.

Os snapshots de banco de dados são particularmente valiosos para os casos de uso a seguir.

- Configuração inicial de um ambiente de desenvolvimento: os snapshots de banco de dados são úteis para criar o ambiente de desenvolvimento inicial para testes, pois fornecem uma referência consistente que reflete o estado exato do ambiente de produção no momento do snapshot.
- Aplicações de alto tráfego: em ambientes de produção em que a operação contínua é essencial, o uso de implantações multi-AZ para snapshots evita a suspensão de E/S no banco de dados principal, garantindo performance ininterrupta e alta disponibilidade.
- Compartilhamento de dados em diferentes contas do RDS: os snapshots de banco de dados podem ser compartilhados entre diferentes Contas da AWS, facilitando a transferência de dados entre contas ou regiões. Isso é útil para projetos colaborativos ou casos em que os dados precisam ser compartilhados para várias finalidades. Para ter mais informações, consulte Compartilhar um snapshot de banco de dados para o Amazon RDS.

Neste tutorial, você explorou tarefas essenciais para gerenciar a instância de banco de dados durante todo o ciclo de vida. Você aprendeu a criar uma instância de banco de dados, adicionar e atualizar tags do Amazon RDS, expandir o armazenamento e criar réplicas de leitura. Você também aprendeu maneiras de desenvolver essas operações fundamentais e gerenciar o ambiente de produção de forma eficaz. Isso incluiu estabelecer um ambiente de desenvolvimento para testes e sincronizá-lo com o ambiente de produção para fins de consistência. Essas tarefas ajudam a manter uma infraestrutura de banco de dados resiliente e escalável, garantindo que o ambiente do Amazon RDS funcione com eficiência.

Configurar e gerenciar uma implantação multi-AZ para o Amazon RDS

Implantações Multi-AZ podem ter uma ou duas instâncias de banco de dados em espera. Quando a implantação tem uma instância de banco de dados em espera, ela é chamada de implantação de instância de banco de dados Multi-AZ. Uma implantação de instância de banco de dados Multi-AZ tem uma instância de banco de dados em espera que fornece suporte para failover, mas não serve tráfego de leitura. Quando a implantação tem duas instâncias de banco de dados em espera, ela é chamada de implantação de cluster de banco de dados Multi-AZ. Uma implantação de cluster de banco de dados Multi-AZ tem instâncias de banco de dados em espera que fornecem suporte para failover e também podem servir tráfego de leitura.

Você pode usar o AWS Management Console para determinar se uma implantação multi-AZ é uma implantação de instância de banco de dados multi-AZ ou uma implantação de cluster de banco de dados multi-AZ. No painel de navegação, escolha Databases (Bancos de dados) e um DB identifier (Identificador de banco de dados).

- Uma implantação de instância de banco de dados multi-AZ tem as seguintes características:
  - Há apenas uma linha para a instância de banco de dados.
  - O valor de Role (Função) é Instance (Instância) ou Primary (Principal).
  - O valor de multi-AZ é Yes (Sim).
- Uma implantação de cluster de banco de dados multi-AZ tem as seguintes características:
  - Há uma linha em nível de cluster com três linhas de instâncias de banco de dados abaixo dela.
  - Para a linha no nível do cluster, o valor de Role (Função) é Multi-AZ DB cluster (Cluster de banco de dados multi-AZ).
  - Para cada linha no nível da instância, o valor de Role (Função) é Writer instance (Instância de gravador) ou Reader instance (Instância de leitor).
  - Para cada linha no nível da instância, o valor de Multi-AZ é 3 Zones (Três zonas).

Tópicos

- Implantações de instâncias de banco de dados multi-AZ para o Amazon RDS
- Implantações de cluster de banco de dados multi-AZ para o Amazon RDS

Além disso, os tópicos a seguir se aplicam tanto a instâncias de banco de dados quanto a clusters de banco de dados multi-AZ.

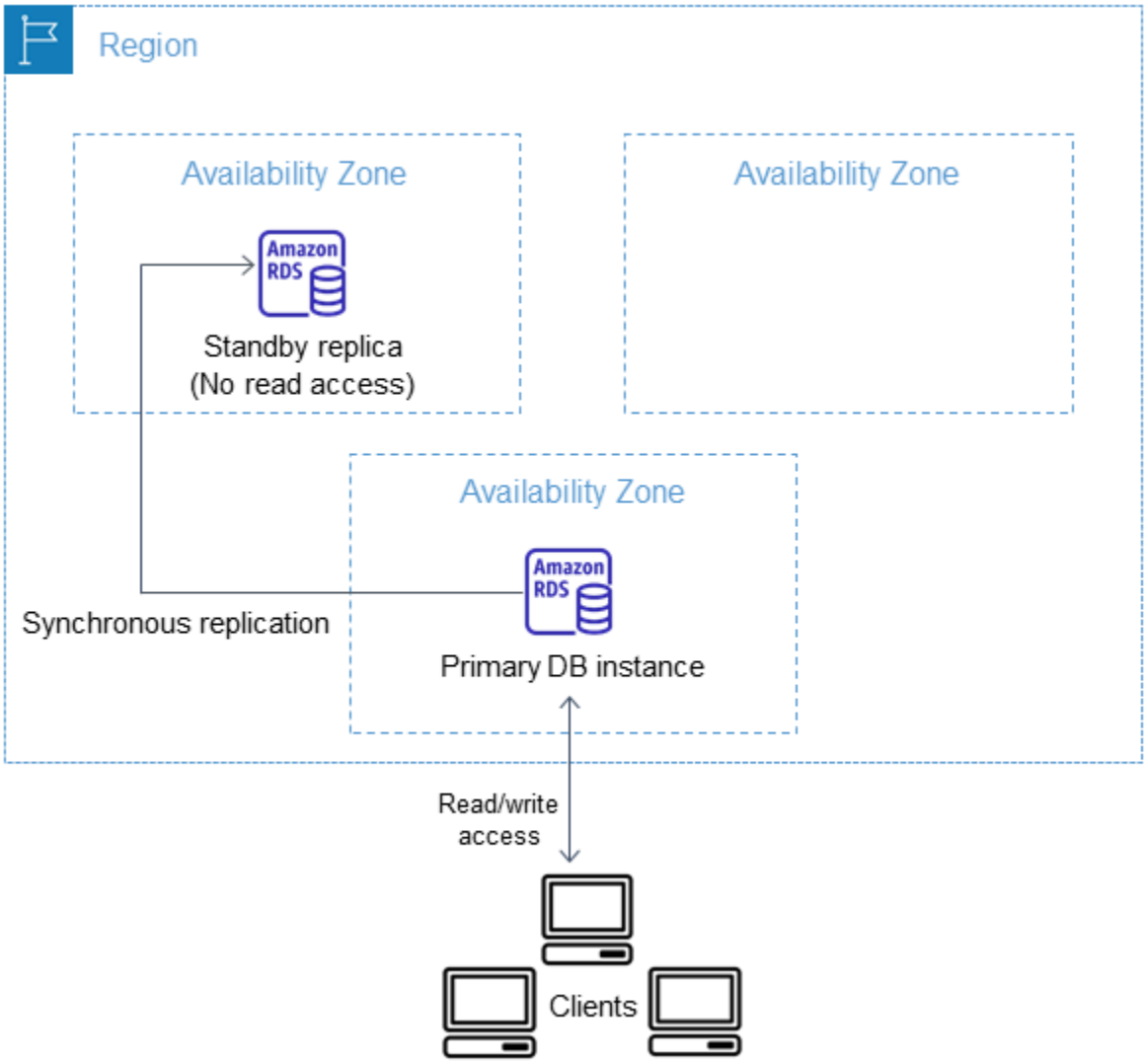
- the section called “Marcar recursos do RDS”
- the section called “ARNs no Amazon RDS”
- the section called “Trabalhar com armazenamento”
- the section called “Manutenção de uma instância de banco de dados”
- the section called “Atualizar a versão do mecanismo”

Implantações de instâncias de banco de dados multi-AZ para o Amazon RDS

O Amazon RDS oferece alta disponibilidade e suporte para failover em instâncias de banco de dados utilizando implantações multi-AZ com uma única instâncias de banco de dados em espera. Esse tipo de implantação é chamado de implantação de instância de banco de dados multi-AZ. O Amazon RDS usa várias tecnologias diferentes para fornecer esse suporte para failover. Implantações multi-AZ para instâncias de banco de dados MariaDB, MySQL, Oracle, PostgreSQL e RDS Custom para SQL Server usam a tecnologia de failover da Amazon. Instâncias de banco de dados do Microsoft SQL Server utilizam o SQL Server Database Mirroring (DBM) ou grupos de disponibilidade (AGs) Always On. Para obter informações sobre o suporte à versão do SQL Server para multi-AZ, consulte Implantações multi-AZ para o Amazon RDS for Microsoft SQL Server. Para ter informações sobre como trabalhar com o RDS Custom para multi-AZ, consulte Gerenciar uma implantação multi-AZ para o RDS Custom para SQL Server.

Em uma implantação de instância de banco de dados multi-AZ, o Amazon RDS provisiona e mantém automaticamente uma réplica em espera síncrona em outra zona de disponibilidade. A instância de banco de dados primária é replicada simultaneamente através de zonas de disponibilidade para uma réplica em espera, a fim de proporcionar a redundância de dados e minimizar os picos de latência durante os backups do sistema. Executar uma instância de banco de dados com alta disponibilidade pode aumentar a disponibilidade durante a manutenção planejada do sistema. Também pode ajudar a proteger bancos de dados contra falhas na instância de banco de dados e interrupção da zona de disponibilidade. Para ter mais informações sobre zonas de disponibilidade, consulte Regiões, zonas de disponibilidade e Local Zones.

> **Note**
>
> A opção de alta disponibilidade não é uma solução de escalabilidade para cenários somente leitura. Não é possível utilizar uma réplica em espera para servir tráfego de leitura. Para servir tráfego somente leitura, utilize um cluster de banco de dados multi-AZ ou uma réplica de leitura. Para ter mais informações sobre clusters de banco de dados multi-AZ, consulte Implantações de cluster de banco de dados multi-AZ para o Amazon RDS. Para ter mais informações sobre réplicas de leitura, consulte Trabalhar com réplicas de leitura de instância de banco de dados.



Com o console do RDS, é possível criar uma implantação de instância de banco de dados multi-AZ. Basta especificar multi-AZ ao criar essa instância. Você pode utilizar o console para converter instâncias de banco de dados existentes em implantações de instâncias de banco de dados multi-AZ, modificando a instância de banco de dados e especificando a opção multi-AZ. Também pode especificar uma implantação multi-AZ com a AWS CLI ou a API do Amazon RDS. Use o comando da CLI create-db-instance ou modify-db-instance ou a operação da API CreateDBInstance ou ModifyDBInstance.

O console do RDS mostra a zona de disponibilidade da réplica em espera (chamada AZ secundária). Você também pode usar o comando describe-db-instances da CLI ou a operação DescribeDBInstances da API para localizar a AZ secundária.

Instâncias de banco de dados que usam implantações de instância de banco de dados multi-AZ podem ter maior latência de gravação e confirmação em comparação com uma implantação

single-AZ. Isso pode acontecer devido à replicação de dados síncrona que ocorre. É possível ter uma alteração na latência se sua implantação falhar na réplica em espera, ainda que o AWS seja desenvolvido com conectividade de rede de baixa latência entre zonas de disponibilidade. Para uma aplicação de produção que exija performance de E/S rápida e consistente, recomendamos o armazenamento de IOPS provisionadas (operações de entrada/saída por segundo). Para ter mais informações sobre classes de instância de banco de dados, consulte Classes de instâncias de banco de dados .

Converter uma instância de banco de dados em uma implantação multi-AZ para o Amazon RDS

Modificar uma instância de banco de dados para uma implantação multi-AZ melhora a disponibilidade ao adicionar uma instância em espera em outra zona de disponibilidade. O processo envolve um tempo de inatividade mínimo e exige um planejamento cuidadoso em relação aos impactos no armazenamento e no desempenho. Essa alteração aumenta a tolerância a falhas e reduz o tempo de recuperação em caso de falhas, tornando-a ideal para ambientes de alta disponibilidade.

Se você tiver uma instância de banco de dados em uma implantação single-AZ e modificá-la para uma implantação de instância de banco de dados multi-AZ, o Amazon RDS realizará as seguintes ações:

1. Gera um snapshot dos volumes do Amazon Elastic Block Store (EBS) da instância de banco de dados primária.
2. Cria volumes para a réplica em espera baseados no snapshot. Esses volumes são inicializados em segundo plano e a performance máxima do volume é alcançada depois que os dados são totalmente inicializados.
3. Ativa a replicação síncrona no bloco entre os volumes das réplicas primária e em espera.

> **Important**
>
> Criar uma instância de banco de dados em espera com base em um snapshot durante uma conversão de single-AZ em multi-AZ evita o tempo de inatividade, mas pode afetar o desempenho, principalmente para workloads sensíveis à gravação. A replicação síncrona pode aumentar a latência de E/S, afetando o desempenho do banco de dados. Como prática recomendada, evite converter uma instância de banco de dados de produção em uma instância de banco de dados multi-AZ.

Em vez disso, crie uma réplica de leitura, habilite backups nela, converta-a em multi-AZ, carregue dados nos respectivos volumes e, depois, promova-os para a instância de banco de dados primário. Para ter mais informações, consulte Trabalhar com réplicas de leitura de instância de banco de dados.

Há duas formas de modificar uma instância de banco de dados para ser uma implantação de instância de banco de dados multi-AZ:

Tópicos

- Converter para uma implantação de instância de banco de dados multi-AZ com o console do RDS
- Modificar uma instância de banco de dados para ser uma implantação de instância de banco de dados multi-AZ

Converter para uma implantação de instância de banco de dados multi-AZ com o console do RDS

É possível usar o console do RDS para converter uma instância de banco de dados para uma implantação de instância de banco de dados multi-AZ.

É possível usar o console somente para concluir a conversão. Para usar o AWS CLI ou a API do RDS, siga as instruções em Modificar uma instância de banco de dados para ser uma implantação de instância de banco de dados multi-AZ.

Como converter para uma implantação de instância de banco de dados multi-AZ com o console do RDS

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Databases (Bancos de dados) e a instância de banco de dados que você deseja modificar.
3. Em Actions (Ações), selecione Convert to Multi-AZ deployment (Converter para implantação multi-AZ).
4. Na página de confirmação, selecione Apply Immediately (Aplicar imediatamente) para aplicar as alterações imediatamente. A escolha dessa opção não causa tempo de inatividade, mas pode causar um possível impacto na performance. Você também pode optar por aplicar a

atualização durante a próxima janela de manutenção. Para ter mais informações, consulte Usar a configuração de programação de modificações.

5. Selecione Convert to Multi-AZ (Converter em multi-AZ).

Modificar uma instância de banco de dados para ser uma implantação de instância de banco de dados multi-AZ

Você também pode modificar uma instância de banco de dados para ser uma implantação de instância de banco de dados multi-AZ das seguintes formas:

- Usando o console do RDS, modifique a instância de banco de dados e defina Multi-AZ deployment (Implantação Multi-AZ) como Yes (Sim).
- Usando o AWS CLI, chame o comando modify-db-instance e defina a opção `--multi-az`.
- Usando a API do RDS, chame a operação ModifyDBInstance e defina o parâmetro `MultiAZ` como `true`.

Para ter mais informações sobre como modificar uma instância de banco de dados , consulte Modificar uma instância de banco de dados do Amazon RDS. Após a conclusão da modificação, o Amazon RDS aciona um evento (RDS-EVENT-0025) que indica o término do processo. É possível monitorar eventos do Amazon RDS. Para ter mais informações sobre eventos do , consulte Trabalhar com a notificação de eventos do Amazon RDS.

Fazer failover de uma instância de banco de dados multi-AZ para o Amazon RDS

Se uma interrupção planejada ou não planejada da instância de banco de dados multi-AZ decorrer de um defeito de infraestrutura, o Amazon RDS alternará automaticamente para uma réplica em espera em outra zona de disponibilidade.

O tempo de conclusão do failover depende da atividade do banco de dados e de outras condições no momento em que a instância de banco de dados primária se tornou indisponível. Em geral, os tempos de failover variam de 60 a 120 segundos. No entanto, transações grandes ou um processo de recuperação longo podem aumentar o tempo de failover. Quando o failover é concluído, o console do RDS pode levar mais um tempo para refletir a nova zona de disponibilidade.

> **Note**
>
> Você pode forçar um failover manualmente ao reinicializar uma instância de banco de dados multi-AZ. Para ter mais informações, consulte Reinicializar uma instância de banco de dados .

O Amazon RDS processa os failovers automaticamente para que você possa retomar as operações de banco de dados o mais rápido possível e sem intervenção administrativa. A instância de banco de dados principal muda automaticamente para a réplica em espera se alguma das condições descritas na tabela a seguir ocorrer. Os motivos do failover podem ser visualizados no log de eventos.

Motivo do failover	Descrição
O sistema operacional subjacente à instância de banco de dados do RDS está sendo corrigido em uma operação offline.	Um failover foi acionado durante a janela de manutenção para um patch de SO ou uma atualização de segurança. Para ter mais informações, consulte Manutenção de uma instância de banco de dados.
O host principal da instância RDS multi-AZ não está íntegro.	A implantação de instância de banco de dados multi-AZ detectou uma instância de banco de dados primária danificada e executou failover.
O host principal da instância RDS multi-AZ está inacessível devido à perda de conectividade de rede.	O monitoramento do RDS detectou uma falha de alcançabilidade de rede na instância de banco de dados principal e acionou um failover.
A instância do RDS foi modificada pelo cliente.	Uma modificação da instância de banco de dados do RDS acionou um failover. Para ter mais informações, consulte Modificar uma instância de banco de dados do Amazon RDS.
A instância primária do RDS multi-AZ está ocupada e não responde.	A instância de banco de dados principal não responde. Recomendamos fazer o seguinte:

Motivo do failover	Descrição
	• Examine o evento e os logs do CloudWatch quanto ao uso excessivo de CPU, memória ou espaço de troca. Para ter mais informaçõ es, consulte Trabalhar com a notificação de eventos do Amazon RDS e Criar uma regra que é acionada em um evento do Amazon RDS. • Avalie sua workload para determinar se você está usando a classe de instância de banco de dados apropriada. Para ter mais informações, consulte Classes de instâncias de banco de dados . • Use o Monitoramento Avançado para métricas do sistema operacional em tempo real. Para ter mais informações, consulte Monitorar métricas do SO com o monitoram ento avançado. • Use o Performance Insights para ajudar a analisar quaisquer problemas que afetem a performance da sua instância de banco de dados. Para ter mais informações, consulte Monitorar a carga de banco de dados com o Performance Insights no Amazon RDS. Para ter mais informações sobre essas recomendações, consulte Ferramentas de monitoramento do Amazon RDS e Práticas recomendadas do Amazon RDS.

Motivo do failover	Descrição
O volume de armazenamento subjacente ao host principal da instância multi-AZ do RDS sofreu uma falha.	A implantação de instância de banco de dados multi-AZ detectou um problema de armazenam ento na instância de banco de dados primária e executou o failover.
O usuário solicitou um failover da instância de banco de dados.	Você reinicializou a instância de banco de dados e escolheu Reinicializar com failover. Para ter mais informações, consulteReinicial izar uma instância de banco de dados

Para determinar se ocorreu failover na instância de banco de dados multi-AZ, faça o seguinte:

- Configure assinaturas de eventos de banco de dados para notificar você por e-mail ou SMS de que um failover foi iniciado. Para ter mais informações sobre eventos do , consulte Trabalhar com a notificação de eventos do Amazon RDS.
- Visualize seus eventos de banco de dados usando o console do RDS ou operações de API.
- Visualize o estado atual da implantação de instância de banco de dados multi-AZ utilizando o console RDS ou operações de API.

Para obter informações sobre como você pode responder aos failovers, reduzir o tempo de recuperação e outras melhores práticas para o Amazon RDS, consulte Práticas recomendadas do Amazon RDS.

Definir o JVM TTL para pesquisas de nome DNS

O mecanismo de failover modifica automaticamente o registro de Domain Name System (DNS) da instância de banco de dados para apontar para a instância de banco de dados em espera. Como resultado, você precisará restabelecer todas as conexões existentes para sua instância de banco de dados. Em um ambiente de máquina virtual Java (JVM), devido à forma como o mecanismo de cache DNS do Java funciona, talvez seja necessário reconfigurar as configurações da JVM.

A JVM armazena em cache pesquisas de nome DNS. Ao resolver um nome de host para um endereço IP, a JVM armazena em cache o endereço IP por um período especificado, conhecido como Time-To-Live (TTL – Vida útil).

Como os recursos AWS usam entradas de nome DNS que acabam mudando, recomendamos configurar a JVM com um valor TTL de até 60 segundos. Isso garante que quando o endereço IP de um recurso mudar, seu aplicativo poderá receber e usar o novo endereço IP do recurso, consultando novamente o DNS.

Em algumas configurações do Java, o TTL padrão da JVM é definido de maneira que jamais atualiza entradas DNS até a JVM ser reiniciada. Por isso, se o endereço IP de um recurso AWS mudar enquanto a aplicação ainda estiver em execução, não será possível usar esse recurso até você reiniciar manualmente a JVM e as informações de IP armazenadas em cache serem atualizadas. Nesse caso, é crucial definir o TTL da JVM, de forma que ele atualize periodicamente as informações de IP armazenadas em cache.

Você pode obter o TTL padrão da JVM recuperando o valor da propriedade `networkaddress.cache.ttl`:

```
String ttl = java.security.Security.getProperty("networkaddress.cache.ttl");
```

> **Note**
>
> O TTL padrão pode variar de acordo com a versão da JVM e a possibilidade de um gerenciador de segurança estar instalado. Muitas JVMs oferecem um TTL padrão menor que 60 segundos. Se estiver usando uma JVM como essa, e não um gerenciador de segurança, será possível ignorar o restante deste tópico. Para ter mais informações sobre gerenciadores de segurança no Oracle, consulte The security manager (O gerenciador de segurança) na documentação do Oracle.

Para modificar o TTL da JVM, defina o valor da propriedade `networkaddress.cache.ttl`. Use um dos seguintes métodos, dependendo das necessidades:

- Para definir o valor da propriedade globalmente para todos os aplicativos que usam a JVM, defina `networkaddress.cache.ttl` no arquivo `$JAVA_HOME/jre/lib/security/java.security`.

  ```
  networkaddress.cache.ttl=60
  ```

- Para definir a propriedade localmente somente para seu aplicativo, defina `networkaddress.cache.ttl` no código de inicialização do aplicativo antes de quaisquer conexões de rede serem estabelecidas.

```
java.security.Security.setProperty("networkaddress.cache.ttl" , "60");
```

Implantações de cluster de banco de dados multi-AZ para o Amazon RDS

Uma implantação de cluster de banco de dados multi-AZ é um modo de implantação de alta disponibilidade semissíncrona do Amazon RDS com duas instâncias de banco de dados de réplica legíveis. Um cluster de bancos de dados multi-AZ tem uma instância de banco de dados de gravação e duas instâncias de banco de dados de leitura em três zonas de disponibilidade diferentes na mesma região da Região da AWS. Clusters de banco de dados multi-AZ oferecem alta disponibilidade, maior capacidade para workloads de leitura e menor latência do gravação quando comparados com implantação de instância de banco de dados multi-AZ.

Você pode importar dados de um banco de dados on-premises para um cluster de banco de dados multi-AZ seguindo as instruções em Importar dados para um banco de dados do Amazon RDS para MySQL com tempo de inatividade reduzido.

É possível comprar instâncias de banco de dados reservadas para um cluster de banco de dados multi-AZ. Para ter mais informações, consulte Instâncias de banco de dados reservadas para um cluster de banco de dados multi-AZ.

A disponibilidade e a compatibilidade de recursos variam entre versões específicas de cada mecanismo de banco de dados e entre Regiões da AWS. Para ter mais informações sobre a disponibilidade de versões e regiões do Amazon RDS com o clusters de banco de dados multi-AZ, consulte Regiões e mecanismos de banco de dados compatíveis com clusters de banco de dados multi-AZ no Amazon RDS.

Tópicos

- Disponibilidade de classe de instância para clusters de banco de dados multi-AZ
- Arquitetura de cluster de banco de dados multi-AZ
- Grupos de parâmetros para clusters de banco de dados multi-AZ
- RDS Proxy com clusters de banco de dados multi-AZ
- Atraso de réplica e clusters de banco de dados multi-AZ
- Snapshots de cluster de banco de dados multi-AZ
- Criar um cluster de banco de dados multi-AZ para o Amazon RDS
- Conectar-se a um cluster de banco de dados multi-AZ no Amazon RDS
- Conectar automaticamente um recurso de computação da AWS e um cluster de banco de dados multi-AZ para o Amazon RDS

- Modificar um cluster de banco de dados multi-AZ para o Amazon RDS
- Atualização da versão de mecanismo de um cluster de banco de dados multi-AZ para o Amazon RDS
- Renomear um cluster de banco de dados multi-AZ para o Amazon RDS
- Reinicializar um cluster de banco de dados multi-AZ e instâncias de banco de dados de leitor do Amazon RDS
- Fazer failover de um cluster de banco de dados multi-AZ para o Amazon RDS
- Configurar a replicação lógica do PostgreSQL com clusters de banco de dados multi-AZ para o Amazon RDS
- Trabalhar com réplicas de leitura de cluster de banco de dados multi-AZ para o Amazon RDS
- Configurar a replicação externa de clusters de banco de dados multi-AZ para o Amazon RDS
- Excluir um cluster de banco de dados multi-AZ para o Amazon RDS
- Limitações de clusters de banco de dados multi-AZ do Amazon RDS

Important

Clusters de banco de dados multi-AZ não são idênticos a clusters de bancos de dados Aurora. Para obter informações sobre clusters de bancos de dados Aurora, consulte o Guia do usuário do Amazon Aurora.

Disponibilidade de classe de instância para clusters de banco de dados multi-AZ

As implantações de clusters de banco de dados multi-AZ são compatíveis com as seguintes classes de instância de banco de dados: `db.m5d`, `db.m6gd`, `db.m6id`, `db.m6idn`, `db.r5d`, `db.r6gd`, `db.x2iedn`, `db.r6id`, `db.r6idn` e `db.c6gd`.

Note

As classes de instância c6gd são as únicas compatíveis com o tamanho de instância `medium`.

Para ter mais informações sobre classes de instância de banco de dados, consulte the section called “Classes da instância de banco de dados”.

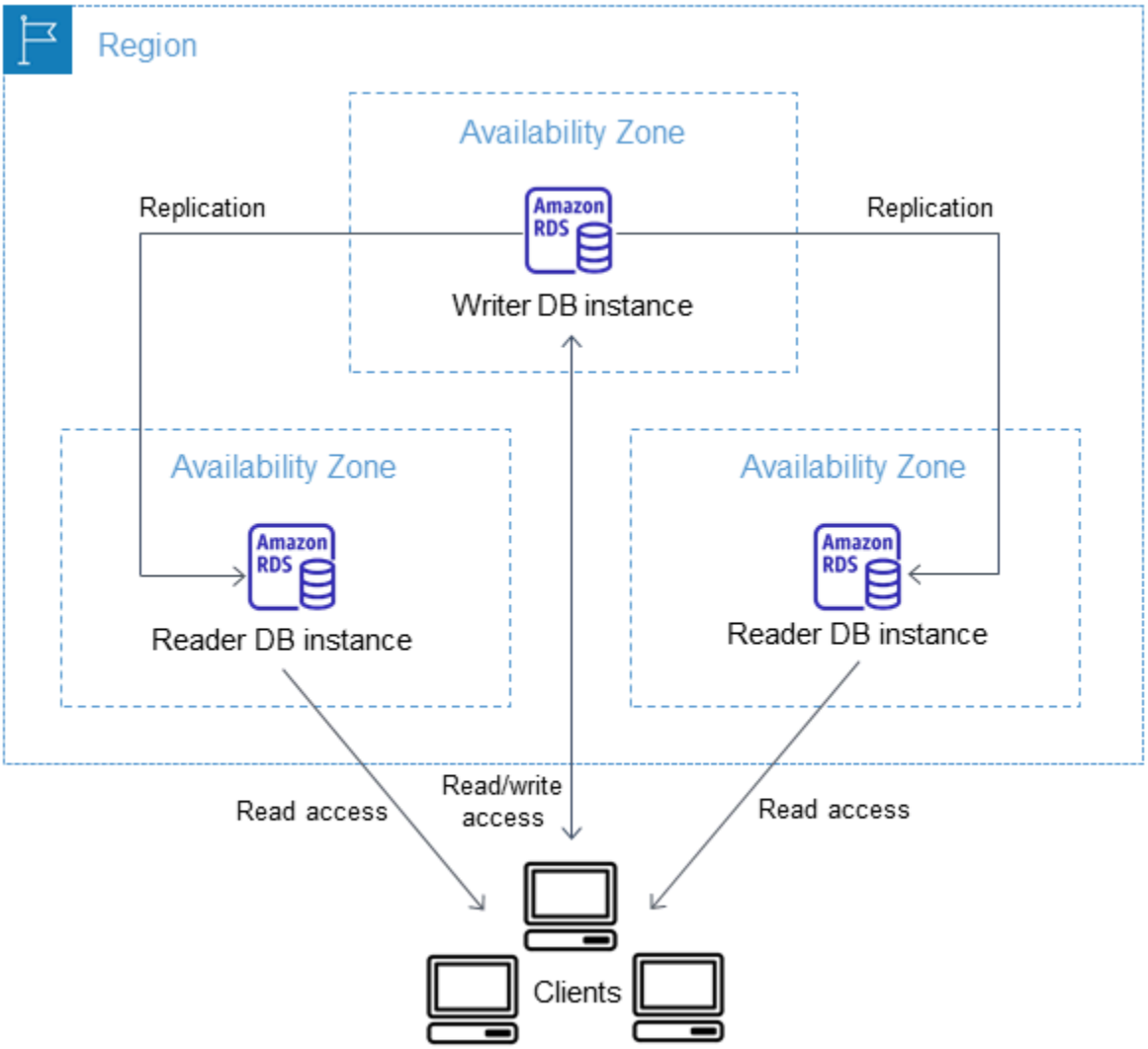
Arquitetura de cluster de banco de dados multi-AZ

Com um cluster de banco de dados multi-AZ, o Amazon RDS replica dados da instância de banco de dados de gravador para ambas as instâncias de banco de dados de leitor utilizando os recursos de replicação nativa do mecanismo de banco de dados. Quando uma alteração é feita na instância de banco de dados de gravador, ela é enviada a cada instância de banco de dados de leitor.

As implantações de cluster de banco de dados multi-AZ usam replicação semissíncrona, que requer reconhecimento de, pelo menos, uma instância de banco de dados de leitor para que uma alteração seja confirmada. Isso não exige o reconhecimento de que os eventos foram totalmente executados e confirmados em todas as réplicas.

Instâncias de banco de dados de leitor atuam como destinos de failover automático e também servem tráfego de leitura para aumentar a taxa de transferência de leitura da aplicação. Se ocorrer uma interrupção na instância de banco de dados de gravador, o RDS fará o gerenciamento do failover para uma das instâncias de banco de dados de leitor. O RDS faz isso com base em qual instância de banco de dados de leitor tem o registro de alteração mais recente.

O diagrama a seguir mostra um cluster de banco de dados multi-AZ.



Em geral, clusters de banco de dados multi-AZ têm menor latência de gravação quando comparados a implantações de instâncias de banco de dados multi-AZ. Ele também permite que workloads de somente leitura sejam executadas em instâncias de banco de dados do leitor. O console do RDS mostra a zona de disponibilidade da instância de banco de dados de gravador e as zonas de disponibilidade das instâncias de banco de dados de leitor. Você também pode usar o comando describe-db-clusters da CLI ou a operação de API DescribeDBClusters para encontrar essas informações.

> **Important**
>
> Para evitar erros de replicação em clusters de banco de dados multi-AZ do RDS para MySQL, é altamente recomendável que todas as tabelas tenham uma chave primária.

Grupos de parâmetros para clusters de banco de dados multi-AZ

Em um cluster de banco de dados multi-AZ, um grupo de parâmetros de cluster de banco de dados atua como um contêiner para valores de configuração de mecanismo que são aplicados a cada instância de banco de dados no cluster de banco de dados multi-AZ.

Em um cluster de banco de dados multi-AZ, um grupo de parâmetros de banco de dados é definido como o grupo de parâmetros de banco de dados padrão do mecanismo de banco de dados e da versão do mecanismo de banco de dados. As configurações no grupo de parâmetros do cluster de banco de dados são utilizadas para todas as instâncias de banco de dados do cluster.

Para obter informações sobre grupos de parâmetros, consulte the section called “Grupos de parâmetros de cluster de banco de dados”.

RDS Proxy com clusters de banco de dados multi-AZ

É possível usar o Amazon RDS Proxy para criar um proxy para os clusters de banco de dados multi-AZ. Com o RDS Proxy, as aplicações podem agrupar e compartilhar conexões de banco de dados para melhorar a capacidade de escala. Cada proxy também executa a multiplexação de conexões, também conhecida como reutilização de conexões. Com a multiplexação, o RDS Proxy executa todas as operações para uma transação usando uma conexão de banco de dados subjacente. O RDS Proxy também pode reduzir o tempo de inatividade de uma atualização de versão secundária de um cluster de banco de dados multi-AZ para um segundo ou menos. Para obter mais informações sobre os benefícios do &AS;, consulte Amazon RDS Proxy.

Para configurar um proxy para um cluster de banco de dados multi-AZ, escolha Criar um proxy RDS ao criar o cluster. Para obter instruções sobre como criar e gerenciar endpoints do RDS Proxy, consulte. the section called “Como trabalhar com endpoints do RDS Proxy”

Atraso de réplica e clusters de banco de dados multi-AZ

Atraso de réplica é a diferença de tempo entre a transação mais recente na instância de banco de dados do gravador e a transação aplicada mais recente em uma instância de banco de dados do leitor. A métrica do Amazon CloudWatch `ReplicaLag` representa essa diferença de tempo. Para ter mais informações sobre métricas do CloudWatch, consulte Monitorar métricas do Amazon RDS com o Amazon CloudWatch.

Embora os clusters de banco de dados multi-AZ permitam uma alta performance de gravação, o atraso de réplica ainda pode ocorrer devido à natureza da replicação baseada em mecanismo. Como qualquer failover deve primeiro resolver o atraso de réplica antes de promover uma nova instância

de banco de dados do gravador, monitorar e gerenciar esse atraso de réplica é algo a ser levado em consideração.

Para clusters de banco de dados multi-AZ do RDS para MySQL, o tempo de failover depende do atraso de réplica das duas instâncias de banco de dados de leitor. Ambas as instâncias de banco de dados do leitor devem aplicar transações não aplicadas antes que uma delas seja promovida para a nova instância de banco de dados de gravador.

Para clusters de banco de dados do RDS para PostgreSQL multi-AZ, o tempo de failover depende do menor atraso de réplica das duas instâncias de banco de dados de leitor restantes. A instância de banco de dados de leitor com o menor atraso de réplica deve aplicar as transações não aplicadas antes de ser promovida para a nova instância de banco de dados de gravador.

Para obter um tutorial que mostra como criar um alarme do CloudWatch quando o atraso de réplica excede um período de tempo definido, consulte Tutorial: criar um alarme do Amazon CloudWatch para atraso de réplica do cluster de banco de dados multi-AZ para o Amazon RDS.

Causas comuns de atraso de réplica

Em geral, o atraso de réplica ocorre quando o workload de gravação é muito alto para que as instâncias de banco de dados do leitor apliquem as transações de forma eficiente. Vários workloads podem incorrer em atraso de réplica temporário ou contínuo. Os seguintes exemplos demonstram as causas comuns:

- Alta simultaneidade de gravação ou atualização em lote pesado na instância de banco de dados do gravador, fazendo com que o processo de aplicação nas instâncias de banco de dados do leitor fique para trás.
- Workload de leitura pesada que usa recursos em uma ou mais instâncias de banco de dados do leitor. Executar consultas lentas ou grandes pode afetar o processo de aplicação e causar atraso de réplica.
- As transações que modificam grandes quantidades de dados ou instruções DDL às vezes podem causar um aumento temporário no atraso de réplica porque o banco de dados deve preservar a ordem de confirmação.

Diminuir o atraso de réplica

Para clusters de banco de dados multi-AZ para RDS para MySQL e RDS para PostgreSQL, você pode mitigar o atraso de réplica reduzindo a carga na instância de banco de dados do gravador.

Você também pode usar o controle de fluxo para reduzir o atraso da réplica. O controle de fluxo funciona controlando a utilização das gravações na instância de banco de dados do gravador, o que garante que o atraso de réplica não continue a crescer de forma não vinculada. O controle de utilização da gravação é realizado adicionando um atraso ao fim de uma transação, o que diminui a taxa de transferência de gravação na instância de banco de dados do gravador. Embora o controle de fluxo não garanta a eliminação do atraso, ele pode ajudar a reduzir o atraso geral em muitos workloads. As seções a seguir fornecem informações sobre como usar o controle de fluxo com o RDS para MySQL e o RDS para PostgreSQL.

Reduzir o atraso de réplica com controle de fluxo para o RDS para MySQL

Quando você está usando clusters de banco de dados multi-AZ do RDS para MySQL, o controle de transmissão é ativado por padrão usando o parâmetro dinâmico `rpl_semi_sync_master_target_apply_lag`. Esse parâmetro especifica o limite posterior que você deseja para o atraso de réplica. À medida que o atraso de réplica se aproxima desse limite configurado, o controle de transmissão controla a utilização das transações de gravação na instância de banco de dados de gravador para tentar conter o atraso de réplica abaixo do valor especificado. Em alguns casos, o atraso de réplica pode exceder o limite especificado. Por padrão, esse parâmetro é definido como 120 segundos. Para desativar o controle de transmissão, defina esse parâmetro como o valor máximo de 86.400 segundos (um dia).

Para exibir o atraso atual injetado pelo controle de transmissão, mostre o parâmetro `Rpl_semi_sync_master_flow_control_current_delay` executando a seguinte consulta.

```
SHOW GLOBAL STATUS like '%flow_control%';
```

O resultado deve ser semelhante ao seguinte:

```
+-----------------------------------------------+-------+
| Variable_name                                 | Value |
+-----------------------------------------------+-------+
| Rpl_semi_sync_master_flow_control_current_delay | 2010  |
+-----------------------------------------------+-------+
1 row in set (0.00 sec)
```

> **Note**
>
> O atraso é mostrado em microssegundos.

Quando você tem o Performance Insights ativado para um cluster de banco de dados multi-AZ do RDS para MySQL, você pode monitorar o evento de espera correspondente a uma instrução SQL indicando que as consultas foram atrasadas por um controle de fluxo. Quando um atraso foi introduzido por um controle de fluxo, você pode visualizar o evento de espera `/wait/synch/cond/semisync/semi_sync_flow_control_delay_cond` correspondente à instrução SQL no painel do Performance Insights. Para visualizar essas métricas, verifique se o Performance Schema está ativado. Para obter informações sobre o Performance Insights, consulte Monitorar a carga de banco de dados com o Performance Insights no Amazon RDS.

Reduzir o atraso de réplica com controle de fluxo para o RDS para PostgreSQL

Quando você está usando clusters de banco de dados do RDS para PostgreSQL multi-AZ, o controle e fluxo é implantado como uma extensão. Ela ativa um operador em segundo plano para todas as instâncias de banco de dados em um cluster de banco de dados. Por padrão, os operadores em segundo plano nas instâncias de banco de dados do leitor comunicam o atraso de réplica atual com o operador em segundo plano na instância de banco de dados do gravador. Se o atraso exceder dois minutos em qualquer instância de banco de dados do leitor, o operador em segundo plano na instância de banco de dados do gravador adicionará um atraso no final de uma transação. Para controlar o limite de atraso, use o parâmetro `flow_control.target_standby_apply_lag`.

Quando um controle de fluxo acelera um processo PostgreSQL, o evento de espera `Extension` em `pg_stat_activity` e o Performance Insights indicam isso. A função `get_flow_control_stats` exibe detalhes sobre quanto atraso está sendo adicionado no momento.

O controle de fluxo pode beneficiar a maioria das workloads de processamento de transações on-line (OLTP) que apresentam transações curtas, mas altamente simultâneas. Se o atraso for causado por transações de longa duração, como operações em lote, o controle de fluxo não proporcionará um benefício tão forte.

Você pode desativar o controle de fluxo removendo a extensão do `shared_preload_libraries` e reinicializando sua instância de banco de dados.

Snapshots de cluster de banco de dados multi-AZ

O Amazon RDS cria e salva backups automáticos do cluster de banco de dados multi-AZ durante a janela de backup configurada. O RDS cria um snapshot do volume de armazenamento do cluster de banco de dados fazendo backup de todo o cluster e não apenas das instâncias individuais.

Também é possível fazer backups manuais do cluster de banco de dados multi-AZ. Para backups de prazo muito longo, pense em exportar dados de snapshot para o Amazon S3. Para ter mais informações, consulte the section called “Criar um snapshot de cluster de banco de dados Multi-AZ”.

É possível restaurar seu cluster de banco de dados multi-AZ para um momento específico criando um cluster de banco de dados multi-AZ. Para obter instruções, consulte the section called “Restaurar um cluster de banco de dados multi-AZ para um horário especificado”.

Também é possível restaurar um snapshot de cluster de banco de dados multi-AZ para uma implantação single-AZ ou uma implantação de instância de banco de dados multi-AZ. Para obter instruções, consulte the section called “Restauração de um snapshot de cluster de banco de dados multi-AZ para uma instância de banco de dados”.

Criar um cluster de banco de dados multi-AZ para o Amazon RDS

Um cluster de banco de dados multi-AZ tem uma instância de banco de dados de gravador e duas instâncias de banco de dados de leitor em três zonas de disponibilidade separadas. Clusters de banco de dados multi-AZ oferecem alta disponibilidade, maior capacidade para workloads de leitura e menor latência quando comparados a implantações multi-AZ. Para ter mais informações sobre clusters de banco de dados multi-AZ, consulte Implantações de cluster de banco de dados multi-AZ para o Amazon RDS.

> **Note**
>
> Clusters de banco de dados multi-AZ têm suporte apenas para os mecanismos de banco de dados MySQL e PostgreSQL.

Pré-requisitos do cluster de banco de dados

> **Important**
>
> Antes de criar um cluster de banco de dados multi-AZ, você deve concluir as tarefas em Configuração do ambiente do Amazon RDS.

Os itens a seguir são pré-requisitos que devem ser cumpridos antes de criar um cluster de banco de dados multi-AZ.

Tópicos

- Configurar a rede para o cluster de banco de dados
- Pré-requisitos adicionais

Configurar a rede para o cluster de banco de dados

Só é possível criar um cluster de banco de dados multi-AZ em uma nuvem privada virtual (VPC) com base no serviço da Amazon VPC. Ele deve estar em uma Região da AWS que tenha pelo menos três zonas de disponibilidade. O grupo de sub-redes do banco de dados escolhido para o cluster de banco de dados deve incluir pelo menos três zonas de disponibilidade. Essa configuração garante que cada instância de banco de dados no cluster de banco de dados esteja em uma zona de disponibilidade diferente.

Para configurar a conectividade entre seu novo cluster de banco de dados e uma instância do Amazon EC2 na mesma VPC, faça isso durante a criação do cluster de banco de dados. Para se conectar ao cluster de banco de dados usando recursos que não sejam instâncias do EC2 na mesma VPC, configure as conexões de rede manualmente.

Tópicos

- Configurar a conectividade automática de rede com uma instância do EC2
- Configurar a rede manualmente

Configurar a conectividade automática de rede com uma instância do EC2

Ao criar um cluster de banco de dados multi-AZ, você pode usar o AWS Management Console para configurar a conectividade entre uma instância do EC2 e o novo cluster de banco de dados. Quando você faz isso, o RDS configura suas definições de VPC e rede automaticamente. O cluster de banco de dados é criado na mesma VPC da instância do EC2 para que a instância do EC2 possa acessar o cluster de banco de dados.

Confira a seguir os requisitos para conectar uma instância do EC2 ao cluster de banco de dados:

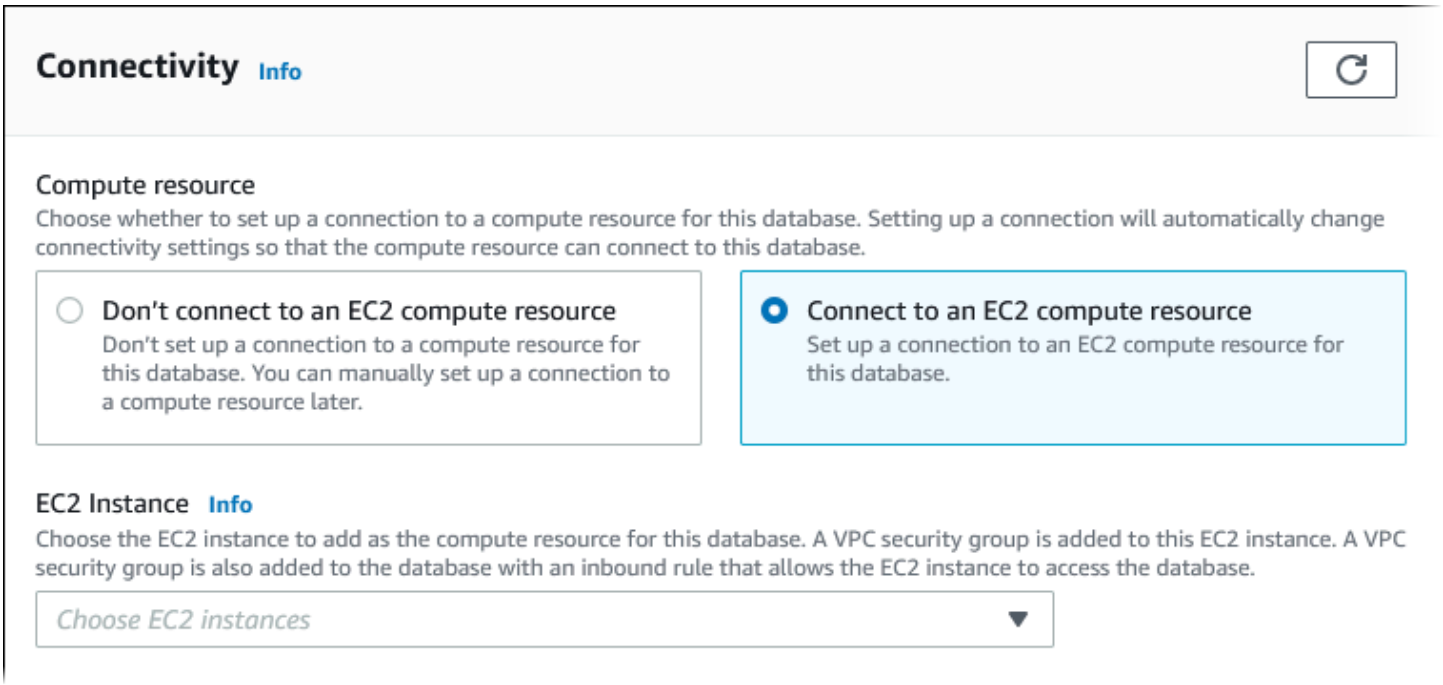
- A instância do EC2 deve existir na Região da AWS antes de criar o cluster de banco de dados.

  Se não houver nenhuma instância do EC2 na Região da AWS, o console fornecerá um link para que você crie uma.
- O usuário que está criando o cluster de banco de dados deve ter permissões para realizar as seguintes operações:
  - `ec2:AssociateRouteTable`
  - `ec2:AuthorizeSecurityGroupEgress`
  - `ec2:AuthorizeSecurityGroupIngress`
  - `ec2:CreateRouteTable`
  - `ec2:CreateSubnet`
  - `ec2:CreateSecurityGroup`
  - `ec2:DescribeInstances`
  - `ec2:DescribeNetworkInterfaces`
  - `ec2:DescribeRouteTables`
  - `ec2:DescribeSecurityGroups`

- `ec2:DescribeSubnets`
- `ec2:ModifyNetworkInterfaceAttribute`
- `ec2:RevokeSecurityGroupEgress`

Usar essa opção cria um cluster de banco de dados privado. O cluster de banco de dados usa um grupo de sub-redes de banco de dados somente com sub-redes privadas para restringir o acesso aos recursos da VPC.

Para conectar uma instância do EC2 ao cluster de banco de dados, escolha Connect to an EC2 compute resource (Conectar-se a um recurso de computação do EC2) na seção Connectivity (Conectividade) da página Create database (Criar banco de dados).



Quando você escolhe Connect to an EC2 compute resource (Conectar-se a um recurso de computação do EC2), o RDS define as opções a seguir automaticamente. Você não pode alterar essas configurações, a menos que opte por não configurar a conectividade com uma instância do EC2 escolhendo Don't connect to an EC2 compute resource (Não conectar a um recurso de computação do EC2).

Opção do console	Configuração automática
Virtual Private Cloud (VPC)	

Opção do console	Configuração automática
	O RDS define a VPC como aquela associada à instância do EC2.
DB subnet group (Grupo de subredes do banco de dados)	O RDS requer um grupo de sub-redes de banco de dados com uma sub-rede privada na mesma zona de disponibilidade da instância do EC2. Se existir um grupo de sub-redes de banco de dados que atenda a esse requisito, o RDS usará o grupo de sub-redes de banco de dados existente. Por padrão, essa opção está definida como Automatic setup (Configuração automática). Quando você escolhe Automatic setup (Configuração automátic a) e não há nenhum grupo de sub-redes de banco de dados que atenda a esse requisito, ocorre a ação a seguir. O RDS usa três sub-redes privadas disponíveis em três zonas de disponibi lidade, das quais uma é a mesma da instância do EC2. Se não houver uma sub-rede privada disponível em uma zona de disponibilidade, o RDS criará uma sub-rede privada na zona de disponibilidade. O RDS cria o grupo de sub-redes de banco de dados. Quando houver uma sub-rede privada disponível, o RDS usará a tabela de rotas associada a ela e adicionará todas as sub-redes que criar a essa tabela de rotas. Quando não houver nenhuma sub-rede privada disponível, o RDS criará uma tabela de rotas sem acesso ao gateway da Internet e adicionará as sub-redes que criar à tabela de rotas. O RDS também permite que você use grupos de sub-redes de banco de dados existentes. Selecione Choose existing (Selecion ar existente) se quiser usar um grupo de sub-redes de banco de dados existente de sua escolha.

Opção do console	Configuração automática
Acesso público	O RDS escolhe No (Não) para que o cluster de banco de dados não fique publicamente acessível. Por motivos de segurança, é uma prática recomendada manter o banco de dados privado e garantir que ele não possa ser acessado pela Internet.
VPC security group (firewall) [Grupo de segurança da VPC (firewall)]	O RDS cria um grupo de segurança associado ao cluster de banco de dados. O grupo de segurança é chamado de `rds-ec2-`*n*, em que *n* é um número. Esse grupo de segurança inclui uma regra de entrada com o grupo de segurança da VPC do EC2 (firewall) como origem. Esse grupo de segurança associado ao cluster de banco de dados permite que a instância do EC2 acesse o cluster de banco de dados. O RDS também cria um grupo de segurança associado à instância de banco de dados. O grupo de segurança é chamado de `ec2-rds-`*n*, em que *n* é um número. Esse grupo de segurança inclui uma regra de saída com o grupo de segurança da VPC do cluster de banco de dados como origem. Esse grupo de segurança permite que a instância do EC2 envie tráfego ao cluster de banco de dados. Para adicionar outro novo grupo de segurança, escolha Create new (Criar novo) e digite o nome do novo grupo de segurança. Para adicionar grupos de segurança existentes, escolha Choose existing (Escolher existente) e selecione os grupos de segurança que deseja adicionar.

Opção do console	Configuração automática
Zona de disponibilidade	O RDS escolhe a zona de disponibilidade da instância do EC2 para uma instância de banco de dados na implantação do cluster de banco de dados multi-AZ. O RDS escolhe aleatoria mente uma zona de disponibilidade diferente para as duas outras instâncias de banco de dados. A instância de banco de dados gravadora é criada na mesma zona de disponibi lidade que a instância do EC2. Existe a possibilidade de custos cruzados da zona de disponibilidade se ocorrer um failover e a instância de banco de dados gravadora estiver em uma zona de disponibilidade diferente.

Para ter mais informações sobre essas configurações, consulte Configurações para criar clusters de banco de dados multi-AZ.

Se você alterar essas configurações após a criação do cluster de banco de dados, as alterações poderão afetar a conexão entre a instância do EC2 e o cluster de banco de dados.

Configurar a rede manualmente

Para se conectar ao cluster de banco de dados usando recursos que não sejam instâncias do EC2 na mesma VPC, configure as conexões de rede manualmente. Se você utilizar o AWS Management Console para criar o cluster de banco de dados multi-AZ, poderá fazer o Amazon RDS criar automaticamente uma VPC para você. Como alternativa, você pode usar a VPC existente ou criar uma nova VPC para seu cluster de banco de dados multi-AZ. A VPC deve ter pelo menos uma sub-rede em pelo menos três das zonas de disponibilidade para você usá-la com um cluster de banco de dados multi-AZ. Para obter informações sobre as VPCs, consulte Amazon VPC e Amazon RDS.

Se você não tiver uma VPC padrão ou não tiver criado uma VPC e não planeja utilizar o console, faça o seguinte:

- Crie uma VPC com pelo menos uma sub-rede em pelo menos três das zonas de disponibilidade na região da AWS em que deseja implantar o cluster de banco de dados. Para ter mais informações, consulte Trabalhar com uma instância de banco de dados em uma VPC.

- Especifique um grupo de segurança da VPC que autorize conexões ao seu cluster de banco de dados. Para ter mais informações, consulte Fornecer acesso à instância de banco de dados na VPC criando um grupo de segurança e Controlar acesso com grupos de segurança.
- Especifique um grupo de sub-rede de banco de dados do RDS que defina pelo menos três sub-redes na VPC as quais possam ser usadas pelo cluster de banco de dados multi-AZ. Para ter mais informações, consulte Trabalhar com grupos de sub-redes de banco de dados.

Para obter informações sobre as limitações aplicadas aos clusters de banco de dados multi-AZ, consulte Limitações de clusters de banco de dados multi-AZ do Amazon RDS.

Se você quiser se conectar a um recurso que não esteja na mesma VPC do cluster de banco de dados multi-AZ, veja os cenários apropriados em Cenários para acessar uma instância de banco de dados em uma VPC.

Pré-requisitos adicionais

Antes de criar o cluster de banco de dados multi-AZ, considere os seguintes pré-requisitos adicionais:

- Para personalizar os parâmetros de configuração do cluster de banco de dados, especifique um grupo de parâmetros do cluster de banco de dados com as configurações de parâmetro necessárias. Para obter informações sobre como criar ou modificar um grupo de parâmetros de cluster de banco de dados, consulte Grupos de parâmetros para clusters de banco de dados multi-AZ.
- Determine o número de porta de TCP/IP a ser especificado para seu cluster de banco de dados. Em algumas empresas, firewalls bloqueiam conexões com as portas padrão. Se o firewall da sua empresa bloquear a porta padrão, escolha outra porta para o cluster de banco de dados. Todas as instâncias de banco de dados em um cluster de banco de dados utilizam a mesma porta.
- Se a versão principal do mecanismo do banco de dados atingir a data de fim do suporte padrão do RDS, você deverá usar a opção da CLI do Suporte estendido ou o parâmetro da API do RDS. Consulte mais informações em “RDS Extended Support” no Configurações para criar clusters de banco de dados multi-AZ.

Criar um cluster de banco de dados

Você pode criar um cluster de banco de dados multi-AZ usando o AWS Management Console, a AWS CLI ou a API do RDS.

Console

É possível criar um cluster de banco de dados multi-AZ escolhendo Multi-AZ DB cluster (Cluster de banco de dados multi-AZ), na seção Availability and durability (Disponibilidade e durabilidade).

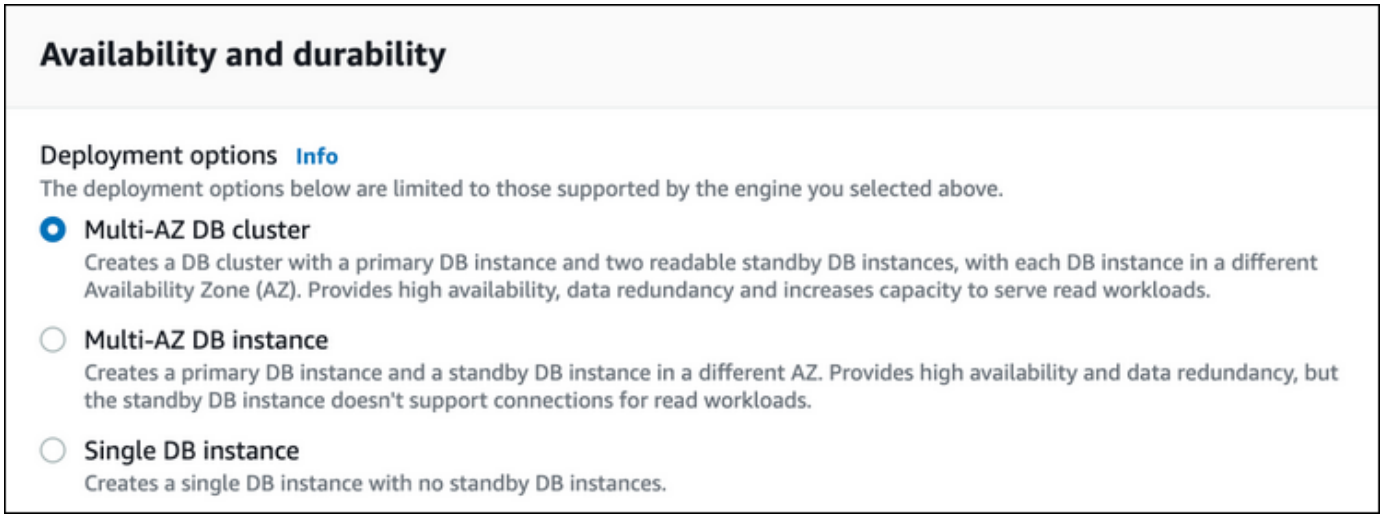
Para criar um cluster de banco de dados multi-AZ utilizando o console

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No canto superior direito do AWS Management Console, escolha a Região da AWS em que você deseja criar o cluster de banco de dados.

   Para obter informações sobre as Regiões da AWS compatíveis com clusters de banco de dados multi-AZ, consulte Limitações de clusters de banco de dados multi-AZ do Amazon RDS.
3. No painel de navegação, escolha Databases (Bancos de dados).
4. Selecione Criar banco de dados.

   Para criar um cluster de banco de dados multi-AZ, certifique-se de que a opção Standard Create (Criação padrão) esteja selecionada e não Easy Create (Criação fácil).
5. Em Engine type (Tipo de mecanismo), escolha MySQL ou PostgreSQL.
6. Em Version (Versão), escolha a versão do mecanismo de banco de dados.

   Para obter informações sobre as versões do mecanismo de banco de dados compatíveis com clusters de banco de dados multi-AZ, consulte Limitações de clusters de banco de dados multi-AZ do Amazon RDS.
7. Em Templates, escolha o template apropriado para a sua implantação.
8. Em Availability and durability (Disponibilidade e durabilidade), escolha Multi-AZ DB cluster (Cluster de banco de dados multi-AZ).



9. Em Identificador do cluster de banco de dados (DB cluster identifier), insira o identificador do cluster de banco de dados.
10. Em Master username (Nome de usuário primário), insira seu nome de usuário primário ou mantenha a configuração padrão.
11. Insira sua senha primária:
    a. Na seção Settings (Configurações), abra Credential Settings (Configurações de credencial).
    b. Se quiser especificar uma senha, desmarque a caixa Auto generate a password (Gerar uma senha automaticamente) se ela estiver marcada.
    c. (Opcional) Altere o valor Master username (Nome de usuário primário).
    d. Insira a mesma senha em Master password (Senha primária) e Confirm password (Confirmar senha).
12. Em Classe de instância do banco de dados, selecione a classe da instância de banco de dados. Para obter uma lista de classes de instâncias de banco de dados compatíveis, consulte the section called “Disponibilidade de classe de instância para clusters de banco de dados multi-AZ”.
13. (Opcional) Configure uma conexão com um recurso de computação para esse cluster de banco de dados.

    Você pode configurar a conectividade entre uma instância do Amazon EC2 e o novo cluster de banco de dados durante a criação do cluster de banco de dados. Para obter mais informações, consulte Configurar a conectividade automática de rede com uma instância do EC2.
14. Na seção Conectividade em Grupo de segurança da VPC (firewall), se você selecionar Criar, um grupo de segurança da VPC será criado com uma regra de entrada que permite que o endereço IP do computador local acesse o banco de dados.

15. Nas seções restantes, especifique suas configurações de cluster de banco de dados. Para obter informações sobre cada configuração, consulte Configurações para criar clusters de banco de dados multi-AZ.
16. Escolha Create database (Criar banco de dados).

    Se você optar por usar uma senha gerada automaticamente, o botão View credential details (Visualizar detalhes da credencial) será exibido na página Databases (Bancos de dados).

    Para visualizar o nome do usuário principal e a senha do cluster de banco de dados, escolha Visualizar detalhes da credencial.

    Para se conectar ao cluster de banco de dados como o usuário primário, use o nome de usuário e a senha exibidos.

    > **Important**
    >
    > Não é possível visualizar a senha do usuário mestre novamente.

17. Para Databases (Bancos de dados), escolha o nome do novo cluster de banco de dados.

No console do RDS, os detalhes do novo cluster de banco de dados são exibidos. O cluster de banco de dados terá um status de Creating (Criando) até o término da criação, quando ele estiver pronto para uso. Quando o status mudar para Available (Disponível), será possível conectar-se ao cluster de banco de dados. Dependendo da classe do cluster de banco de dados e do armazenamento alocado, pode levar alguns minutos até que o novo cluster de banco de dados fique disponível.

AWS CLI

Antes de criar um cluster de banco de dados multi-AZ utilizando a AWS CLI, atenda aos pré-requisitos necessários. Isso inclui a criação de uma VPC e um grupo de sub-redes de banco de dados do RDS. Para ter mais informações, consulte Pré-requisitos do cluster de banco de dados.

Para criar um cluster de banco de dados multi-AZ utilizando a AWS CLI, chame o comando create-db-cluster. Especifique `--db-cluster-identifier`. Para a opção `--engine`, especifique `mysql` ou `postgres`.

Para ter mais informações sobre cada opção, consulte Configurações para criar clusters de banco de dados multi-AZ.

Para obter informações sobre Regiões da AWS, mecanismos de banco de dados e as respectivas versões que são compatíveis com clusters de banco de dados multi-AZ, consulte Limitações de clusters de banco de dados multi-AZ do Amazon RDS.

O comando `create-db-cluster` cria a instância de banco de dados de gravador para o seu cluster de banco de dados e duas instâncias de banco de dados de leitor. Cada instância de banco de dados está em uma zona de disponibilidade diferente.

Por exemplo, o comando a seguir cria um cluster de banco de dados multi-AZ do MySQL 8.0 chamado `mysql-multi-az-db-cluster`.

Example

Para Linux, macOS ou Unix:

```
aws rds create-db-cluster \
   --db-cluster-identifier mysql-multi-az-db-cluster \
   --engine mysql \
   --engine-version 8.0.32  \
   --master-username admin \
   --manage-master-user-password  \
   --port 3306 \
   --backup-retention-period 1  \
   --db-subnet-group-name default \
   --allocated-storage 4000 \
   --storage-type io1 \
   --iops 10000 \
   --db-cluster-instance-class db.m5d.xlarge
```

Para Windows:

```
aws rds create-db-cluster ^
   --db-cluster-identifier mysql-multi-az-db-cluster ^
   --engine mysql ^
   --engine-version 8.0.32 ^
   --manage-master-user-password ^
   --master-username admin ^
   --port 3306 ^
   --backup-retention-period 1 ^
   --db-subnet-group-name default ^
   --allocated-storage 4000 ^
   --storage-type io1 ^
```

```
    --iops 10000 ^
    --db-cluster-instance-class db.m5d.xlarge
```

O comando a seguir cria um cluster de banco de dados multi-AZ do PostgreSQL 13.4 chamado `postgresql-multi-az-db-cluster`.

Example

Para Linux, macOS ou Unix:

```
aws rds create-db-cluster \
    --db-cluster-identifier postgresql-multi-az-db-cluster \
    --engine postgres \
    --engine-version 13.4 \
    --manage-master-user-password \
    --master-username postgres \
    --port 5432 \
    --backup-retention-period 1  \
    --db-subnet-group-name default \
    --allocated-storage 4000 \
    --storage-type io1 \
    --iops 10000 \
    --db-cluster-instance-class db.m5d.xlarge
```

Para Windows:

```
aws rds create-db-cluster ^
    --db-cluster-identifier postgresql-multi-az-db-cluster ^
    --engine postgres ^
    --engine-version 13.4 ^
    --manage-master-user-password ^
    --master-username postgres ^
    --port 5432 ^
    --backup-retention-period 1 ^
    --db-subnet-group-name default ^
    --allocated-storage 4000 ^
    --storage-type io1 ^
    --iops 10000 ^
    --db-cluster-instance-class db.m5d.xlarge
```

API do RDS

Antes de criar um cluster de banco de dados multi-AZ usando a API do RDS, certifique-se de atender aos pré-requisitos necessários, como criar uma VPC e um grupo de sub-redes de banco de dados do RDS. Para ter mais informações, consulte Pré-requisitos do cluster de banco de dados.

Para criar um cluster de banco de dados multi-AZ usando a API do RDS, chame a operação CreateDBCluster. Especifique `DBClusterIdentifier`. Para o parâmetro `Engine`, especifique `mysql` ou `postgresql`.

Para ter mais informações sobre cada opção, consulte Configurações para criar clusters de banco de dados multi-AZ.

A operação `CreateDBCluster` cria a instância de banco de dados de gravador para o seu cluster de banco de dados e duas instâncias de banco de dados de leitor. Cada instância de banco de dados está em uma zona de disponibilidade diferente.

Configurações para criar clusters de banco de dados multi-AZ

Para obter detalhes sobre as configurações que você escolhe ao criar um cluster de banco de dados multi-AZ, consulte a tabela a seguir. Para ter mais informações sobre as opções da AWS CLI, consulte create-db-cluster. Para ter mais informações sobre os parâmetros da API do RDS, consulte CreateDBCluster.

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS

Armazenamento alocado	O valor de armazenamento a ser alocado para cada instância de banco de dados no seu cluster de banco de dados (em gibibyte). Para ter mais informações, consulte Armazenamento de instâncias de banco de dados do Amazon RDS.	Opção da CLI:
		`--allocated-storage`
		Parâmetro da API:
		`AllocatedStorage`

Atualização da versão secundária automática	Selecione Enable auto minor version upgrade (Habilitar atualização automática da versão secundária) para que o cluster de banco	Opção da CLI: `--auto-minor-version-upgrade`

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS

	de dados receba atualizaç ões automáticas da versão do mecanismo de banco de dados secundárias preferidas quando elas forem disponibilizadas. O Amazon RDS executa atualizações automáticas de versões secundári as na janela de manutenção.	`--no-auto-minor-version-upg rade`
		
		Parâmetro da API:
		
		`AutoMinorVersionUpgrade`

Período de retenção de backup	O número de dias que você deseja que os backups automáticos do seu cluster de banco de dados fiquem retidos. Para um cluster de banco de dados multi-AZ, esse valor deve ser definido como **1** ou mais.
	
	Para ter mais informações, consulte Introdução aos backups.
		Opção da CLI:
		
		`--backup-retention-period`
		
		Parâmetro da API:
		
		`BackupRetentionPeriod`

Janela de backup	O período durante o qual o Amazon RDS faz um backup automático do seu cluster de banco de dados. A menos que você tenha um horário específico no qual deseja que o backup do banco de dados seja feito, utilize o padrão No Preference (Sem preferência).
	
	Para ter mais informações, consulte Introdução aos backups.
		Opção da CLI:
		
		`--preferred-backup-window`
		
		Parâmetro da API:
		
		`PreferredBackupWindow`

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS

Autoridade certificadora	A autoridade de certificação (CA) para o certificado do servidor usado pelo cluster de banco de dados.
	
	Para obter mais informações, consulte Usar SSL/TLS para criptografar uma conexão com uma instância ou um cluster de banco de dados.
		Opção da CLI:
		
		`--ca-certificate-identifier`
		
		Parâmetro da API do RDS:
		
		`CACertificateIdentifier`

Copiar tags para snapshots	Essa opção copia qualquer etiqueta do cluster de banco de dados para um snapshot do banco de dados quando você cria um snapshot.
	
	Para ter mais informações, consulte Marcar recursos do Amazon RDS.
		Opção da CLI:
		
		`-copy-tags-to-snapshot`
		
		`-no-copy-tags-to-snapshot`
		
		Parâmetro da API do RDS:
		
		`CopyTagsToSnapshot`

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS

Autenticação de banco de dados	A opção de autenticação de banco de dados que você deseja usar.
	
	Selecione Password authentication (Autenticação de senha) para autenticar usuários de banco de dados somente com senhas de banco de dados.
	
	Selecione Password and IAM DB authentication (Senha e autenticação de banco de dados do IAM) para autenticar usuários de banco de dados com senhas de banco de dados e credenciais de usuário por meio de perfis e usuários. Para obter mais informações, consulte Autenticação do banco de dados do IAM para MariaDB, MySQL e PostgreSQL.
		Opção da CLI:
		
		`--enable-iam-database-authentication`
		
		`--no-enable-iam-database-authentication`
		
		Parâmetro da API do RDS:
		
		`EnableIAMDatabaseAuthentication`

Porta de banco de dados	A porta pela qual você deseja acessar o cluster de banco de dados. A porta padrão é exibida.
	
	A porta não pode ser alterada após a criação do cluster de banco de dados.
	
	Em algumas empresas, firewalls bloqueiam conexões com as portas padrão. Se o firewall da sua empresa bloquear a porta padrão, escolha outra porta para o seu cluster de banco de dados.
		Opção da CLI:
		
		`--port`
		
		Parâmetro da API do RDS:
		
		`Port`

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS

Identificador do cluster de banco de dados	O nome do seu cluster de banco de dados. Forneça nomes para os seus clusters de banco de dados da mesma forma que você fornece nomes para os seus servidores on-premises. O identificador do cluster de banco de dados pode conter até 63 caracteres alfanuméricos e deve ser exclusivo para a conta na região da AWS que você escolher.	Opção da CLI:
		`--db-cluster-identifier`
		Parâmetro da API do RDS:
		`DBClusterIdentifier`

Classe de instância de banco de dados	A capacidade de computação e memória de cada instância de banco de dados no cluster de banco de dados multi-AZ, por exemplo `db.m5d.xlarge` .
	Se possível, escolha uma classe de instância de banco de dados grande o suficiente para um conjunto de trabalho de consulta típico pode ser sustentado na memória. Quando os conjuntos de trabalho são mantidos na memória o sistema pode evitar a gravação em disco, o que aprimora a performance.
	Para obter uma lista de classes de instâncias de banco de dados compatíveis, consulte the section called “Disponibilidade de classe de instância para clusters de banco de dados multi-AZ”.
		Opção da CLI:
		`--db-cluster-instance-class`
		Parâmetro da API do RDS:
		`DBClusterInstanceClass`

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS

Grupo de parâmetros do cluster de banco de dados	O parameter group do cluster de banco de dados que você deseja associar ao cluster de banco de dados.
	
	Para ter mais informações, consulte Grupos de parâmetros para clusters de banco de dados multi-AZ.
		Opção da CLI:
		
		`--db-cluster-parameter-group-name`
		
		Parâmetro da API do RDS:
		
		`DBClusterParameterGroupName`

DB engine version (Versão do mecanismo de banco de dados)	A versão do mecanismo de banco de dados que você deseja usar.	Opção da CLI:
		
		`--engine-version`
		
		Parâmetro da API do RDS:
		
		`EngineVersion`

Grupo de parâmetros do cluster de banco de dados	O grupo de parâmetros da instância de banco de dados que deve ser associado ao cluster de banco de dados. Para obter mais informações, consulte Grupos de parâmetros para clusters de banco de dados multi-AZ.	Opção da CLI: `--db-cluster-parameter-group-name` Parâmetro da API do RDS: **`DBClusterParameterGroupName`**

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS

DB subnet group (Grupo de subredes do banco de dados)	O grupo de sub-redes de banco de dados que você deseja usar para o cluster de banco de dados. Selecione Choose existing (Selecionar existente) para usar um grupo de sub-redes de banco de dados existente. Depois, escolha o grupo de sub-redes necessário na lista suspensa Existing DB subnet groups (Grupos de sub-redes de banco de dados existentes).
	
	Escolha Automatic setup (Configuração automática) para permitir que o RDS selecione um grupo de sub-redes de banco de dados compatível. Se não existir nenhum, o RDS criará um grupo de sub-redes para o cluster.
	
	Para ter mais informações, consulte Trabalhar com grupos de sub-redes de banco de dados.
		Opção da CLI:
		
		`--db-subnet-group-name`
		
		Parâmetro da API do RDS:
		
		`DBSubnetGroupName`

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS

Deletion protection (Proteção contra exclusão)	Selecione Enable deletion protection (Habilitar proteção contra exclusão) para impedir que seu cluster de banco de dados seja excluído. Por padrão, se você criar um cluster de banco de dados de produção com o console, a proteção contra exclusão será ativada.
	
	Para ter mais informações, consulte Excluir uma instância de banco de dados.
		Opção da CLI:
		
		`--deletion-protection`
		
		`--no-deletion-protection`
		
		Parâmetro da API do RDS:
		
		`DeletionProtection`

Criptografia	Enable Encryption (Habilitar criptografia) para habilitar a criptografia em repouso nesse cluster de banco de dados.
	
	A criptografia é ativada por padrão para clusters de banco de dados multi-AZ.
	
	Para ter mais informações, consulte Criptografar recursos do Amazon RDS.
		Opções da CLI:
		
		`--kms-key-id`
		
		`--storage-encrypted`
		
		`--no-storage-encrypted`
		
		Parâmetros da API do RDS:
		
		`KmsKeyId`
		
		`StorageEncrypted`

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS

Monitoramento avançado	Enable enhanced monitoring (Habilitar monitoramento avançado) para habilitar a coleta de métricas em tempo real do sistema operacional em que o cluster de banco de dados é executado.
	
	Para ter mais informações, consulte Monitorar métricas do SO com o monitoramento avançado.
		Opções da CLI:
		
		`--monitoring-interval`
		
		`--monitoring-role-arn`
		
		Parâmetros da API do RDS:
		
		`MonitoringInterval`
		
		`MonitoringRoleArn`

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS

Nome do banco de dados inicial	O nome do banco de dados no cluster de banco de dados. Se você não fornecer um nome, o Amazon RDS não criará um banco de dados no cluster de banco de dados para MySQL. No entanto, ele cria um banco de dados no cluster de banco de dados para o PostgreSQL. O nome não pode ser uma palavra reservada pelo mecanismo de banco de dados. Ele tem outras restrições, dependendo do mecanismo de banco de dados.
	
	MySQL:
	
	• Deve conter de 1 a 64 caracteres alfanuméricos.
	
	PostgreSQL:
	
	• Deve conter de 1 a 63 caracteres alfanuméricos.
	• Deve começar com uma letra ou um sublinhado. Os caracteres subsequentes podem ser letras, sublinhado ou dígitos (0 a 9).
	• O nome do banco de dados inicial é `postgres`.
		Opção da CLI:
		
		`--database-name`
		
		Parâmetro da API do RDS:
		
		`DatabaseName`

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS

Exportações de log	Os tipos de arquivos de log de bancos de dados a serem publicados no Amazon CloudWatch Logs.
	
	Para ter mais informações, consulte Publicação de logs de banco de dados no Amazon CloudWatch Logs.
		Opção da CLI:
		
		`-enable-cloudwatch-logs-exports`
		
		Parâmetro da API do RDS:
		
		`EnableCloudwatchLogsExports`

Janela de manutenção	A janela de 30 minutos na qual as modificações pendentes do seu cluster de banco de dados serão aplicadas. Se o período não for relevante, selecione No Preference (Sem preferência).
	
	Para ter mais informações, consulte A janela de manutenção do Amazon RDS.
		Opção da CLI:
		
		`--preferred-maintenance-window`
		
		Parâmetro da API do RDS:
		
		`PreferredMaintenanceWindow`

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS

Gerenciar credenciais principais no AWS Secrets Manager	Selecione Gerenciar credenciais principais no AWS Secrets Manager para gerenciar a senha do usuário principal em um segredo no Secrets Manager.
	
	Opcionalmente, selecione uma chave do KMS a ser usada para proteger o segredo. Escolha entre uma das chaves do KMS da sua conta ou insira a chave de uma conta distinta.
	
	Para ter mais informações, consulte Gerenciamento de senhas com Amazon RDS e AWS Secrets Manager.
		Opção da CLI:
		
		`--manage-master-user-password | --no-manage-master-user-password`
		
		`--master-user-secret-kms-key-id`
		
		Parâmetro da API do RDS:
		
		`ManageMasterUserPassword`
		
		`MasterUserSecretKmsKeyId`

Senha mestre	A senha da conta de usuário principal.	Opção da CLI:
		
		`--master-user-password`
		
		Parâmetro da API do RDS:
		
		`MasterUserPassword`

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS

Nome do usuário principal	O nome que você usa como o nome de usuário primário para fazer login no cluster de banco de dados com todos os privilégios do banco de dados.
	• Pode conter 1–16 caracteres alfanuméricos e sublinhados.
	• O primeiro caractere deve ser uma letra.
	• Não pode ser uma palavra reservada pelo mecanismo de banco de dados.
	
	Não é possível alterar o nome de usuário principal depois que o cluster de banco de dados multi-AZ é criado.
	
	Para ter mais informações sobre os privilégios concedidos ao usuário primário, consulte Privilégios da conta de usuário mestre.
		Opção da CLI:
		
		`--master-username`
		
		Parâmetro da API do RDS:
		
		`MasterUsername`

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS

Performance Insights	Selecione Enable Performance Insights (Habilitar Performance Insights) para monitorar a carga d cluster de banco de dados, para que você possa analisar e solucionar problemas relacionados à performance do banco de dados.
	
	Escolha um período de retenção para determinar a quantidade de histórico de dados do Performance Insights que deve ser mantida. A configuração de retenção no nível gratuito é Default (7 days) (Padrão (7 dias)). Para reter seus dados de performance por mais tempo, especifique entre 1 e 24 meses. Para obter mais informações sobre os períodos de retenção, consulte Preços e retenção de dados para o Performance Insights.
	
	Escolha uma chave do KMS a ser usada para proteger a chave usada para criptografar esse volume de banco de dados. Escolha entre uma das chaves do KMS da sua conta ou insira a chave de uma conta distinta.
	
	Para obter mais informações, consulteMonitorar a carga de banco de dados com o Performance Insights no Amazon RDS
		Opções da CLI:
		
		`--enable-performance-insights`
		
		`--no-enable-performance-insights`
		
		`--performance-insights-retention-period`
		
		`--performance-insights-kms-key-id`
		
		Parâmetros da API do RDS:
		
		`EnablePerformanceInsights`
		
		`PerformanceInsightsRetentionPeriod`
		
		`PerformanceInsightsKMSKeyId`

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS

IOPS provisionadas	A quantidade de IOPS (operações de entrada/saída por segundo) provisionadas a serem inicialmente alocadas ao cluster de banco de dados.	Opção da CLI:
		`--iops`
		Parâmetro da API do RDS:
		`Iops`

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS

Acesso público	Publicly accessible (Acessível publicamente) para fornecer ao cluster de banco de dados um endereço IP público, o que significa que ele é acessível fora da VPC. Para ser acessível publicame nte, o cluster de banco de dados também deve estar em uma sub-rede pública na VPC.
	
	Not publicly accessible (Não acessível publicamente) para tornar o cluster de banco de dados acessível somente de dentro da VPC.
	
	Para ter mais informações, consulte Ocultar uma instância de banco de dados em uma VPC da Internet.
	
	Para se conectar a um cluster de banco de dados de fora de sua VPC, o cluster de banco de dados deve ser acessível publicamente. Além disso, deve ser concedido acesso utilizando as regras de entrada do grupo de segurança do cluster de banco de dados, e outros requisitos devem ser atendidos. Para ter mais informações, consulte Não é possível conectar-se à instância de banco de dados do Amazon RDS.
		Opção da CLI:
		
		`--publicly-accessible`
		
		`--no-publicly-accessible`
		
		Parâmetro da API do RDS:
		
		`PubliclyAccessible`

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS
	Se o cluster de banco de dados não estiver acessível publicamente, será possível usar uma conexão AWS Site-to-Site VPN ou uma conexão do AWS Direct Connect para acessá-la de uma rede privada. Para ter mais informaçõ es, consulte Privacidade do tráfego entre redes.	

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS

Suporte estendido do RDS	Selecione Habilitar Suporte estendido do RDS para permitir que as versões principais do mecanismo compatíveis continuem funcionando após a data de término do suporte padrão do RDS.
	
	Quando você cria um cluster de banco de dados, o Amazon RDS usa como padrão o Suporte estendido do RDS. Para evitar a criação de um cluster de banco de dados após a data de fim do suporte padrão do RDS e para evitar cobranças pelo Suporte estendido do RDS, desabilite essa configuração. Os clusters de banco de dados existentes não incorrerão em cobranças até a data de início dos preços do Suporte estendido do RDS.
	
	Para obter mais informações, consulte Suporte estendido do Amazon RDS com Amazon RDS.
		Opção da CLI:
		
		`--engine-lifecycle-support`
		
		Parâmetro da API do RDS:
		
		`EngineLifecycleSupport`

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS
Storage throughput (Throughput de armazenamento)	O valor do throughput de armazenamento do cluster de banco de dados. Essa configuração ficará visível somente se você escolher o SSD de uso geral (gp3) como o tipo de armazenamento. Essa configuração não é configurável e é definida automaticamente com base na IOPS que você especifica. Para obter mais informações, consulte Armazenamento gp3 (recomendado).	Esse valor é calculado automaticamente e não tem uma opção de CLI.
RDS Proxy	Selecione Create an RDS Proxy (Criar um proxy RDS) para criar um proxy para seu cluster de banco de dados. O Amazon RDS cria automaticamente um perfil do IAM e um segredo do Secrets Manager para o proxy.	Não disponível ao criar um cluster de banco de dados.

Tipo de armazenamento	O tipo de armazenamento do seu cluster de banco de dados.
	
	Somente os armazenamentos SSD de uso geral (gp3), IOPS provisionadas (io1) e SSD de IOPS provisionadas (io2) são compatíveis.
	
	Para obter mais informações, consulte Tipos de armazenamento do Amazon RDS.
		Opção da CLI:
		
		`--storage-type`
		
		Parâmetro da API do RDS:
		
		`StorageType`

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS
Virtual Private Cloud (VPC)	Uma VPC baseada no serviço da Amazon VPC para associar a esse cluster de banco de dados. Para ter mais informações, consulte Amazon VPC e Amazon RDS.	Para a CLI e a API, especifique os IDs do grupo de segurança da VPC.

VPC security group (firewall) [Grupo de segurança da VPC (firewall)]	O grupo de segurança a ser associado ao cluster de banco de dados.
	
	Para ter mais informações, consulte Visão geral dos grupos de segurança de VPC.
		Opção da CLI:
		
		`--vpc-security-group-ids`
		
		Parâmetro da API do RDS:
		
		`VpcSecurityGroupIds`

Configurações não aplicáveis ao criar clusters de banco de dados Multi-AZ

As seguintes configurações no comando `create-db-cluster` da AWS CLI e na operação `CreateDBCluster` da API do RDS não se aplicam a clusters de banco de dados multi-AZ.

Você também não pode especificar essas configurações para clusters de banco de dados multi-AZ no console.

Configuração da AWS CLI	Configuração da API do RDS
`--availability-zones`	`AvailabilityZones`
`--backtrack-window`	`BacktrackWindow`
`--character-set-name`	`CharacterSetName`
`--domain`	`Domain`
`--domain-iam-role-name`	`DomainIAMRoleName`

Configuração da AWS CLI	Configuração da API do RDS
`--enable-global-write-forwarding | --no-enable-global-write-forwarding`	`EnableGlobalWriteForwarding`
`--enable-http-endpoint | --no-enable-http-endpoint`	`EnableHttpEndpoint`
`--global-cluster-identifier`	`GlobalClusterIdentifier`
`--option-group-name`	`OptionGroupName`
`--pre-signed-url`	`PreSignedUrl`
`--replication-source-identifier`	`ReplicationSourceIdentifier`
`--scaling-configuration`	`ScalingConfiguration`

Conectar-se a um cluster de banco de dados multi-AZ no Amazon RDS

Um cluster de banco de dados multi-AZ tem três instâncias de banco de dados em vez de uma única instância de banco de dados. Cada conexão é processada por uma instância de banco de dados específica. Quando você se conecta a um cluster de banco de dados multi-AZ, o nome de host e a porta especificados apontam para um nome de domínio totalmente qualificado chamado de endpoint. O cluster de banco de dados multi-AZ utiliza o mecanismo de endpoint para abstrair essas conexões para que você não precise especificar exatamente a qual instância de banco de dados no cluster de banco de dados se conectar. Por isso, você não precisa codificar todos os nomes de host ou escrever a própria lógica para reorganizar conexões quando algumas instâncias de banco de dados não estão disponíveis.

O endpoint do gravador conecta-se à instância de banco de dados de gravador do cluster de banco de dados, que oferece suporte a operações de leitura e gravação. O endpoint leitor se conecta a qualquer uma das duas instâncias de banco de dados de leitor, que aceitam apenas operações de leitura.

Usando endpoints, você pode mapear todas as conexões para a instância de banco de dados apropriada ou o grupo de instâncias de banco de dados com base no seu caso de uso. Por exemplo, para realizar instruções DDL e DML, conecte-se à instância de banco de dados que atua como gravador. Para realizar consultas, você pode se conectar ao endpoint leitor, com o cluster de banco de dados multi-AZ gerenciando automaticamente as conexões entre as instâncias de banco de dados de leitor. Para diagnósticos ou ajustes, conecte-se a um endpoint de instância de banco de dados específico para examinar detalhes sobre uma instância de banco de dados específica.

Para saber mais sobre como se conectar à sua instância de banco de dados, consulte Conectar a uma instância de banco de dados do Amazon RDS.

Consulte mais informações sobre como se conectar a clusters de banco de dados nos tópicos a seguir.

Tópicos

- Endpoints do cluster
- Endpoints de leitor
- Endpoints da instância
- Conexões de alta disponibilidade
- Conectar-se a clusters de banco de dados multi-AZ com os drivers da AWS para o Amazon RDS

Tipos de endpoints de cluster de banco de dados multi-AZ

Um endpoint é representado por um identificador exclusivo que contém um endereço de host. Os tipos de endpoints a seguir estão disponíveis em um cluster de banco de dados multi-AZ:

Endpoint do cluster

Um endpoint de cluster (ou endpoint de gravador) de um cluster de banco de dados multi-AZ se conecta à instância de banco de dados de gravador atual desse cluster de banco de dados. Esse endpoint é o único capaz de realizar operações de gravação, como instruções DDL e DML. Esse endpoint também pode realizar operações de leitura.

Cada cluster de banco de dados multi-AZ tem um único endpoint de cluster e uma única instância de banco de dados de gravador.

Use o endpoint cluster em todas as operações de gravação no cluster de banco de dados, inclusive inserções, atualizações, exclusões e alterações DDL. Você também pode usar o endpoint de cluster para operações de leitura, como consultas.

Se a instância de banco de dados de gravador atual de um cluster de banco de dados falhar, o cluster de banco de dados multi-AZ fará failover automático para uma nova instância de banco de dados de gravador. Durante um failover, o cluster de banco de dados continua atendendo a solicitações de conexão para o endpoint de cluster pela nova instância de banco de dados de gravador, com interrupção mínima de serviço.

O exemplo a seguir ilustra um endpoint de cluster para um cluster de banco de dados multi-AZ.

```
mydbcluster.cluster-123456789012.us-east-1.rds.amazonaws.com
```

Consulte mais informações sobre como se conectar endpoints de cluster em Endpoints do cluster.

Endpoint de leitor

Um endpoint leitor de um cluster de banco de dados multi-AZ é compatível com balanceamento de carga para conexões somente leitura com o cluster de banco de dados. Use o endpoint do leitor para operações de leitura, como consultas `SELECT`. Ao processar essas instruções nas instâncias de banco de dados de leitor, esse endpoint reduz a sobrecarga na instância de banco de dados de gravador. Ele também ajuda o cluster a escalar a capacidade de processar consultas `SELECT` simultâneas. Cada cluster de banco de dados multi-AZ tem um único endpoint de leitor.

O endpoint leitor envia cada solicitação de conexão para uma das instâncias de banco de dados de leitor. Quando você usa o endpoint de leitor para uma sessão, apenas é possível executar instruções somente leitura, como `SELECT`, nessa sessão.

O exemplo a seguir ilustra um endpoint de leitor para um cluster de banco de dados multi-AZ. A intenção somente leitura de um endpoint leitor é indicada por `-ro` no nome do endpoint do cluster.

```
mydbcluster.cluster-ro-123456789012.us-east-1.rds.amazonaws.com
```

Consulte mais informações sobre como se conectar a endpoints de leitor em Endpoints de leitor.

Endpoint da instância

Um endpoint de instância conecta-se a uma instância de banco de dados específica dentro de um cluster de banco de dados multi-AZ. Cada instância de banco de dados em um cluster de banco de dados, tem o próprio endpoint de instância exclusivo. Portanto, há um endpoint de instância para a instância de banco de dados de gravador atual do cluster de banco de dados e há um endpoint de instância para cada uma das instâncias de banco de dados de leitor no cluster de banco de dados.

O endpoint de instância oferece controle direto sobre as conexões com o cluster de banco de dados. Esse controle pode ajudar a resolver cenários nos quais talvez não seja apropriado utilizar o endpoint de cluster ou o endpoint de leitor. Por exemplo, o aplicativo cliente pode exigir um balanceamento de carga mais refinado com base no tipo de workload. Nesse caso, é possível configurar vários clientes para se conectarem a instâncias de banco de dados diferentes em um cluster de banco de dados com o objetivo de distribuir workloads de leitura.

O exemplo a seguir ilustra um endpoint de instância para uma instância de banco de dados em um cluster de banco de dados multi-AZ.

```
mydbinstance.123456789012.us-east-1.rds.amazonaws.com
```

Consulte mais informações sobre como se conectar a endpoints de instância em Endpoints da instância.

Visualizar endpoints

Use o console, a AWS CLI ou a API do Amazon RDS para visualizar os endpoints de cluster, de leitor e da instância.

Console

No AWS Management Console, você vê o endpoint de cluster e o endpoint de leitor na página de detalhes de cada cluster de banco de dados multi-AZ. Você vê o endpoint de instância na página de detalhes de cada instância de banco de dados.

AWS CLI

Com a AWS CLI, você vê os endpoints de gravador e leitor na saída do comando describe-db-clusters. Por exemplo, o comando a seguir mostra os atributos de endpoint para todos os clusters na região atual da AWS.

```
aws rds describe-db-cluster-endpoints
```

Amazon RDS API

Com a API do Amazon RDS, você recupera os endpoints chamando a ação DescribeDBClusterEndpoints. Essa saída também mostra endpoints de cluster de bancos de dados Amazon Aurora, se houver.

Endpoints do cluster

Cada cluster de banco de dados multi-AZ tem um único endpoint de cluster integrado, cujo nome e outros atributos são gerenciados pelo Amazon RDS. Não crie, exclua nem modifique esse tipo de endpoint.

Use o endpoint de cluster ao administrar seu cluster de banco de dados, realizar operações Extract, Transform, Load (ETL – Extração, transformação, carregamento) ou desenvolver e testar aplicações. O endpoint de cluster conecta-se à instância de banco de dados de gravador do cluster. A instância de banco de dados de gravador é a única em que você cria tabelas e índices, executa instruções `INSERT` e realiza outras operações DDL e DML.

O endereço IP físico apontado pelo endpoint de cluster muda quando o mecanismo de failover promove uma nova instância de banco de dados como a instância de banco de dados de gravador do cluster. Caso você use alguma forma de agrupamento de conexões ou outra multiplexação, prepare-se para enviar ou reduzir a vida útil para todas as informações DNS armazenadas em cache. Isso garante que você não tente estabelecer uma conexão de leitura/gravação com uma instância de banco de dados que fique indisponível ou seja somente leitura após um failover.

Endpoints de leitor

Use o endpoint de leitor em conexões somente leitura com o seu cluster de banco de dados multi-AZ. Esse endpoint ajuda o cluster de banco de dados a lidar com uma workload com uso intensivo de consulta. O endpoint leitor é o endpoint fornecido para aplicações que geram relatórios ou fazem outra operações somente leitura sobre o cluster. O endpoint leitor envia as conexões para instâncias de banco de dados de leitor disponíveis em um cluster de banco de dados multi-AZ.

Cada cluster multi-AZ tem um único endpoint de leitor integrado, cujo nome e outros atributos são gerenciados pelo Amazon RDS. Não crie, exclua nem modifique esse tipo de endpoint.

Endpoints da instância

Cada instância de banco de dados em um cluster de banco de dados multi-AZ tem seu próprio endpoint de instância integrado, cujo nome e outros atributos são gerenciados pelo Amazon RDS. Não crie, exclua nem modifique esse tipo de endpoint. Com um cluster de banco de dados multi-AZ, você normalmente usa os endpoints de gravador e leitor com mais frequência do que os endpoints de instância.

Nas operações diárias, a principal maneira de usar endpoints de instância é diagnosticar problemas de capacidade ou performance que afetam uma instância de banco de dados específica em um cluster de banco de dados multi-AZ. Conectado a uma instância de banco de dados específica, examine as variáveis de status, as métricas etc. Fazer isso pode ajudar a determinar o que está acontecendo nessa instância de banco de dados diferente do que está acontecendo com outras instâncias de banco de dados no cluster.

Conexões de alta disponibilidade

Para clusters de banco de dados multi-AZ em que a alta disponibilidade é importante, utilize o endpoint de gravador para conexões de leitura/gravação ou de uso geral e o endpoint de leitor para conexões somente leitura. Os endpoints de leitor e de gravador gerenciam o failover da instância de banco de dados melhor do que os endpoints de instância. Ao contrário dos endpoints de instância, os endpoints de leitor e de gravador alteram automaticamente a qual instância de banco de dados eles se conectam caso uma instância de banco de dados no cluster fique indisponível.

Se a instância de banco de dados de gravador de um cluster de banco de dados falhar, o Amazon RDS fará failover automaticamente para uma nova instância de banco de dados de gravador. Ele faz isso promovendo uma instância de banco de dados do leitor para uma nova instância de banco de dados de gravador. Se ocorrer um failover, será possível utilizar o endpoint de gravador para se reconectar à instância de banco de dados de gravador recém-promovida. Ou você pode usar o

endpoint de leitor para se reconectar a uma das instâncias de banco de dados de leitor no cluster de banco de dados. Durante um failover, o endpoint de leitor pode direcionar conexões à nova instância de banco de dados de gravador de um cluster de banco de dados por um curto período depois que uma instância de banco de dados de leitor é promovida para a nova instância de banco de dados de gravador. Se você projeta sua própria lógica de aplicação para gerenciar conexões de endpoint de instância, poderá descobrir manual ou programaticamente o conjunto resultante de instâncias de banco de dados disponíveis no cluster de banco de dados.

Conectar-se a clusters de banco de dados multi-AZ com os drivers da AWS para o Amazon RDS

O pacote de drivers da AWS foram projetados para comportar tempos mais rápidos de transição e de failover, além de autenticação com o AWS Secrets Manager, o AWS Identity and Access Management (IAM) e identidades federadas. Os drivers da AWS dependem do monitoramento do status do cluster de banco de dados e do conhecimento da topologia do cluster para determinar o novo gravador. Essa abordagem reduz os tempos de transição e de failover para segundos de um dígito, em comparação com dezenas de segundos para drivers de código aberto.

Como novos recursos do serviço são introduzidos, o objetivo do pacote de drivers da AWS é ter suporte integrado para esses recursos do serviço.

Conectar-se a clusters de banco de dados multi-AZ com o driver JDBC da Amazon Web Services (AWS)

O driver JDBC da Amazon Web Services (AWS) foi projetado como um wrapper JDBC avançado para ajudar as aplicações a aproveitar os recursos dos bancos de dados em cluster. Esse wrapper é complementar e amplia a funcionalidade do driver JDBC existente. O driver é compatível de forma intercambiável com os seguintes drivers de comunidade:

- MySQL Connector/J
- MariaDB Connector/J
- pgJDBC

Para instalar o driver JDBC da AWS, anexe o arquivo .jar do driver JDBC da AWS (localizado na aplicação `CLASSPATH`) e mantenha referências ao respectivo driver da comunidade. Atualize o respectivo prefixo do URL de conexão da seguinte forma:

- `jdbc:mysql://` para `jdbc:aws-wrapper:mysql://`

- `jdbc:mariadb://` para `jdbc:aws-wrapper:mariadb://`
- `jdbc:postgresql://` para `jdbc:aws-wrapper:postgresql://`

Consulte mais informações sobre o driver JDBC da AWS e siga as instruções para usá-lo em Amazon Web Services (AWS) JDBC Driver GitHub repository.

Conectar-se a clusters de banco de dados multi-AZ com o driver Python da Amazon Web Services (AWS)

O driver Python da Amazon Web Services (AWS) foi projetado como um wrapper Python avançado. Esse wrapper é complementar e amplia a funcionalidade do driver Psycopg de código aberto. O driver Python da AWS é compatível com as versões 3.8 e posterior do Python. É possível instalar o pacote `aws-advanced-python-wrapper` usando o comando `pip`, bem como os pacotes de código aberto `psycopg`.

Para ter mais informações sobre o driver Python da AWS e instruções completas para usá-lo, consulte Amazon Web Services (AWS) Python Driver GitHub repository.

Conectar automaticamente um recurso de computação da AWS e um cluster de banco de dados multi-AZ para o Amazon RDS

Você pode conectar automaticamente um cluster de banco de dados multi-AZ e recursos de computação da AWS, como instâncias do Amazon Elastic Compute Cloud (Amazon EC2) e funções do AWS Lambda.

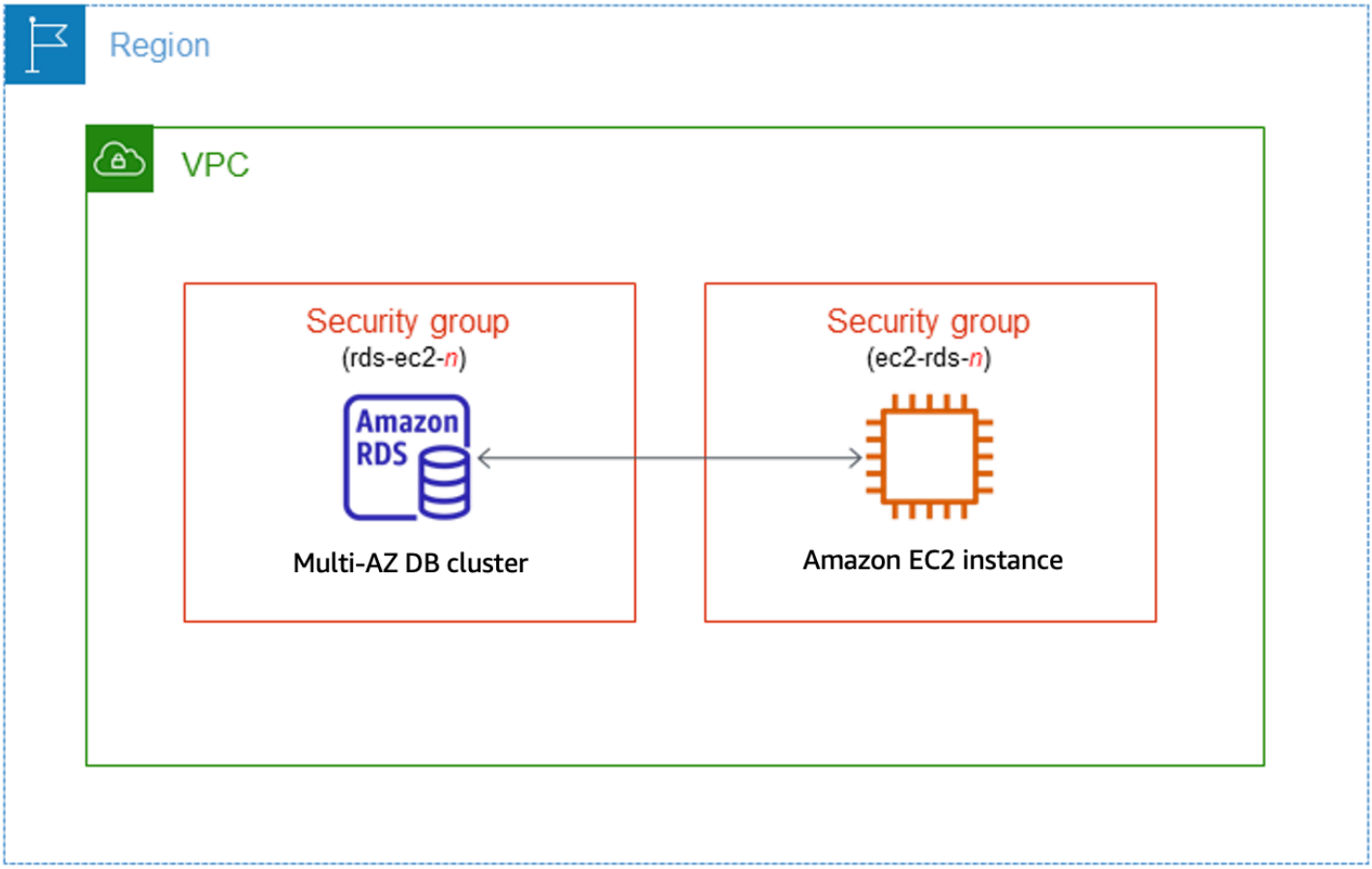
Os tópicos a seguir apresentam instruções detalhadas para definir configurações de rede, grupos de segurança e parâmetros de conexão e estabelecer conexões confiáveis com instâncias de banco de dados do Amazon RDS em uma implantação de cluster de banco de dados multi-AZ. Eles se concentram em otimizar a conectividade e o desempenho da rede para aplicações que interagem com o cluster de banco de dados multi-AZ, garantindo operações de dados seguras e eficientes.

Tópicos

- Conectar automaticamente uma instância do EC2 e um cluster de banco de dados multi-AZ
- Conectar automaticamente uma função do Lambda e um cluster de banco de dados multi-AZ

Conectar automaticamente uma instância do EC2 e um cluster de banco de dados multi-AZ

Você pode usar o console do Amazon RDS para simplificar a configuração de uma conexão entre uma instância do Amazon Elastic Compute Cloud (Amazon EC2) e um cluster de banco de dados multi-AZ. Geralmente, seu cluster de banco de dados multi-AZ está em uma sub-rede privada e sua instância do EC2 está em uma sub-rede pública em uma VPC. Você pode usar um cliente SQL em sua instância do EC2 para se conectar ao seu cluster de banco de dados multi-AZ. A instância do EC2 também pode executar servidores web ou aplicações que acessam seu cluster de banco de dados multi-AZ privado.



Se você quiser se conectar a uma instância do EC2 que não esteja na mesma VPC do cluster de banco de dados multi-AZ, veja os cenários em the section called “Cenários para acessar uma instância de banco de dados em uma VPC”.

Tópicos

- Visão geral da conectividade automática com uma instância do EC2
- Conectar automaticamente uma instância do EC2 e um cluster de banco de dados multi-AZ
- Visualizar recursos computacionais conectados

Visão geral da conectividade automática com uma instância do EC2

Quando você configura automaticamente uma conexão entre uma instância do EC2 e um cluster de banco de dados multi-AZ, o Amazon RDS configura o grupo de segurança da VPC para a instância do EC2 e para o cluster de banco de dados.

Confira a seguir os requisitos para conectar uma instância do EC2 a um cluster de banco de dados multi-AZ:

- A instância do EC2 deve existir na mesma VPC do cluster de banco de dados multi-AZ.

  Se não houver nenhuma instância do EC2 na mesma VPC, o console fornecerá um link para que você crie uma.

- O usuário que está configurando a conectividade deve ter permissões para realizar as seguintes operações do EC2:
  - `ec2:AuthorizeSecurityGroupEgress`
  - `ec2:AuthorizeSecurityGroupIngress`
  - `ec2:CreateSecurityGroup`
  - `ec2:DescribeInstances`
  - `ec2:DescribeNetworkInterfaces`
  - `ec2:DescribeSecurityGroups`
  - `ec2:ModifyNetworkInterfaceAttribute`
  - `ec2:RevokeSecurityGroupEgress`

Quando você configura uma conexão com uma instância do EC2, o Amazon RDS atua de acordo com a configuração atual dos grupos de segurança associados ao cluster de banco de dados multi-AZ e à instância do EC2, conforme descrito na tabela a seguir.

Configuração atual do grupo de segurança do RDS	Configuração atual do grupo de segurança do EC2	Ação do RDS
Há um ou mais grupos de segurança associados ao cluster de banco de dados multi-AZ com um nome que corresponde ao padrão `rds-ec2-`*n* (em que *n* é um número). Um grupo de segurança que corresponde ao padrão não foi modificado. Esse grupo de segurança tem apenas uma regra de saída com o grupo de segurança	Há um ou mais grupos de segurança associados à instância do EC2 com um nome que corresponde ao padrão `rds-ec2-`*n* (em que *n* é um número). Um grupo de segurança que corresponde ao padrão não foi modificado. Esse grupo de segurança tem apenas uma regra de saída com o grupo de segurança da	O Amazon RDS não realiza nenhuma ação. Uma conexão já foi configura da automaticamente entre a instância do EC2 e o cluster de banco de dados multi-AZ. Como já existe uma conexão entre a instância do EC2 e o banco de dados do RDS, os grupos de segurança não são modificados.

Configuração atual do grupo de segurança do RDS	Configuração atual do grupo de segurança do EC2	Ação do RDS
da VPC da instância do EC2 como origem.	VPC do cluster de banco de dados multi-AZ como origem.	

Configuração atual do grupo de segurança do RDS	Configuração atual do grupo de segurança do EC2	Ação do RDS
Qualquer uma das seguintes condições se aplica: • Não há nenhum grupo de segurança associado ao cluster de banco de dados multi-AZ com um nome que corresponde ao padrão `rds-ec2-`*n*. • Há um ou mais grupos de segurança associados ao cluster de banco de dados multi-AZ com um nome que corresponde ao padrão `rds-ec2-`*n*. No entanto, nenhum desses grupos de segurança pode ser usado para a conexão com a instância do EC2. Não será possível usar um grupo de segurança se não tiver uma regra de entrada no grupo de segurança da VPC da instância do EC2 como origem. Um grupo de segurança também não poderá ser usado se tiver sido modificado. São exemplos de modificação a adição de uma regra ou a alteração da porta de uma regra existente.	Qualquer uma das seguintes condições se aplica: • Não há um grupo de segurança associado à instância do EC2 com um nome que corresponde ao padrão `ec2-rds-`*n*. • Há um ou mais grupos de segurança associados à instância do EC2 com um nome que corresponde ao padrão `ec2-rds-`*n*. No entanto, nenhum desses grupos de segurança pode ser usado para a conexão com o cluster de banco de dados multi-AZ. Não será possível usar um grupo de segurança se não tiver uma regra de saída no grupo de segurança da VPC do cluster de banco de dados multi-AZ como origem. Um grupo de segurança também não poderá ser usado se tiver sido modificado.	RDS action: create new security groups

Configuração atual do grupo de segurança do RDS	Configuração atual do grupo de segurança do EC2	Ação do RDS
Há um ou mais grupos de segurança associados ao cluster de banco de dados multi-AZ com um nome que corresponde ao padrão `rds-ec2-`*n*. Um grupo de segurança que corresponde ao padrão não foi modificado. Esse grupo de segurança tem apenas uma regra de saída com o grupo de segurança da VPC da instância do EC2 como origem.	Há um ou mais grupos de segurança associados à instância do EC2 com um nome que corresponde ao padrão `ec2-rds-`*n*. No entanto, nenhum desses grupos de segurança pode ser usado para a conexão com o cluster de banco de dados multi-AZ. Não será possível usar um grupo de segurança se não tiver uma regra de saída no grupo de segurança da VPC do cluster de banco de dados multi-AZ como origem. Um grupo de segurança também não poderá ser usado se tiver sido modificado.	RDS action: create new security groups

Configuração atual do grupo de segurança do RDS	Configuração atual do grupo de segurança do EC2	Ação do RDS
Há um ou mais grupos de segurança associados ao cluster de banco de dados multi-AZ com um nome que corresponde ao padrão `rds-ec2-`*n*. Um grupo de segurança que corresponde ao padrão não foi modificado. Esse grupo de segurança tem apenas uma regra de saída com o grupo de segurança da VPC da instância do EC2 como origem.	Existe um grupo de segurança do EC2 válido para a conexão, mas ele não está associado à instância do EC2. Esse grupo de segurança tem um nome que corresponde ao padrão `rds-ec2-`*n*. Não foi modificado. Ele tem apenas uma regra de saída com o grupo de segurança da VPC do cluster de banco de dados multi-AZ como origem.	RDS action: associate EC2 security group

Configuração atual do grupo de segurança do RDS	Configuração atual do grupo de segurança do EC2	Ação do RDS
Qualquer uma das seguintes condições se aplica: • Não há nenhum grupo de segurança associado ao cluster de banco de dados multi-AZ com um nome que corresponde ao padrão `rds-ec2-`*n*. • Há um ou mais grupos de segurança associados ao cluster de banco de dados multi-AZ com um nome que corresponde ao padrão `rds-ec2-`*n*. No entanto, nenhum desses grupos de segurança pode ser usado para a conexão com a instância do EC2. Não será possível usar um grupo de segurança se não tiver uma regra de entrada no grupo de segurança da VPC da instância do EC2 como origem. Um grupo de segurança também não poderá ser usado se tiver sido modificado.	Há um ou mais grupos de segurança associados à instância do EC2 com um nome que corresponde ao padrão `rds-ec2-`*n*. Um grupo de segurança que corresponde ao padrão não foi modificado. Esse grupo de segurança tem apenas uma regra de saída com o grupo de segurança da VPC do cluster de banco de dados multi-AZ como origem.	RDS action: create new security groups

Ação do RDS: criar grupos de segurança

O Amazon RDS realiza as seguintes ações:

- Cria um grupo de segurança que corresponde ao padrão `rds-ec2-`*n*. Esse grupo de segurança tem uma regra de entrada com o grupo de segurança da VPC da instância do EC2 como origem. Esse grupo de segurança está associado ao cluster de banco de dados multi-AZ e permite que a instância do EC2 acesse o cluster de banco de dados multi-AZ.
- Cria um grupo de segurança que corresponde ao padrão `ec2-rds-`*n*. Esse grupo de segurança tem uma regra de saída com o grupo de segurança da VPC do cluster de banco de dados multi-AZ como origem. Esse grupo de segurança está associado à instância do EC2 e permite que ela envie tráfego ao cluster de banco de dados multi-AZ.

Ação do RDS: associar o grupo de segurança do EC2

O Amazon RDS associa o grupo de segurança do EC2 existente e válido à instância do EC2. Esse grupo de segurança permite que a instância do EC2 envie tráfego ao cluster de banco de dados multi-AZ.

Conectar automaticamente uma instância do EC2 e um cluster de banco de dados multi-AZ

Antes de configurar uma conexão entre uma instância do EC2 e um banco de dados do RDS, atenda aos requisitos descritos em Visão geral da conectividade automática com uma instância do EC2.

Se você alterar esses grupos de segurança depois de configurar a conectividade, as alterações poderão afetar a conexão entre a instância do EC2 e o banco de dados do RDS.

> **Note**
>
> Você só pode configurar automaticamente uma conexão entre uma instância do EC2 e um banco de dados do RDS utilizando o AWS Management Console. Você não pode configurar uma conexão automaticamente com a AWS CLI nem a API do RDS.

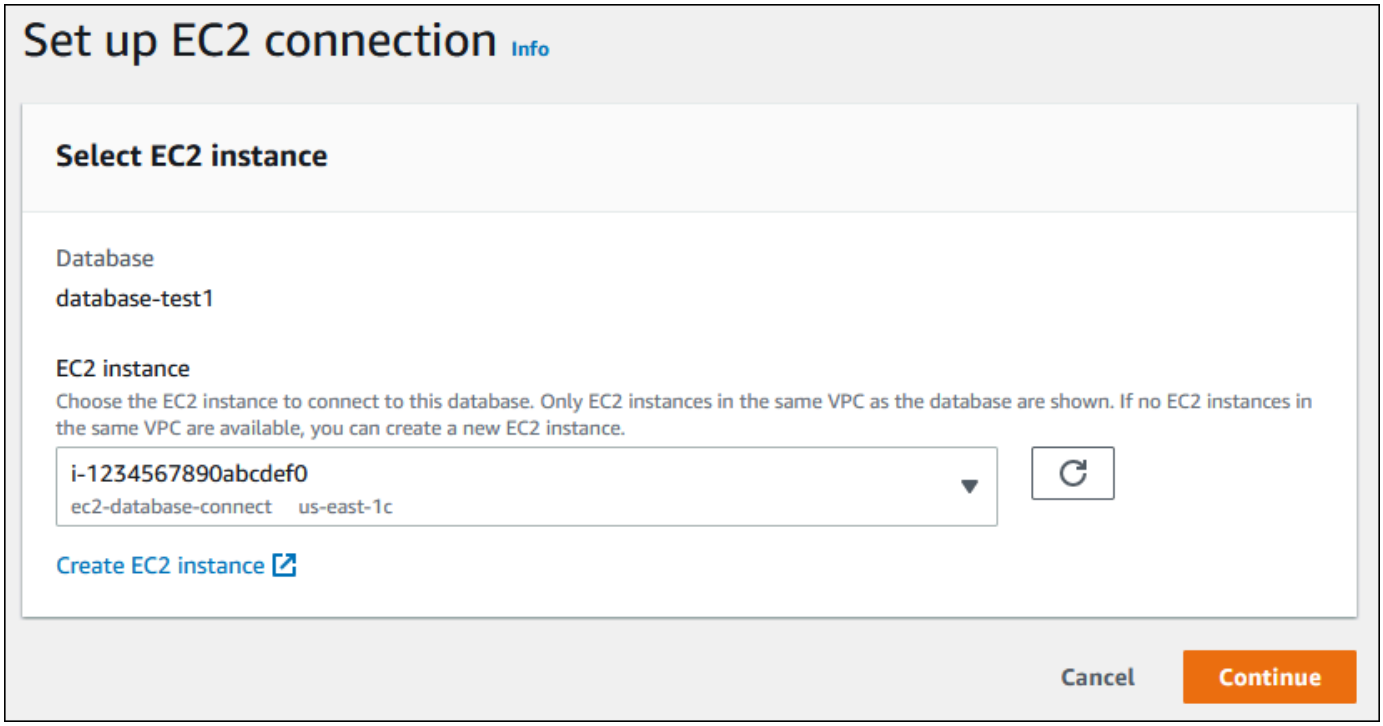
Como conectar automaticamente uma instância do EC2 e um banco de dados do RDS

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, selecione Databases (Bancos de dados) e depois selecione banco de dados do RDS.

3. Em Ações, selecione Configurar conexão do EC2.

   A página Set up EC2 connection (Configurar conexão do EC2) é exibida.

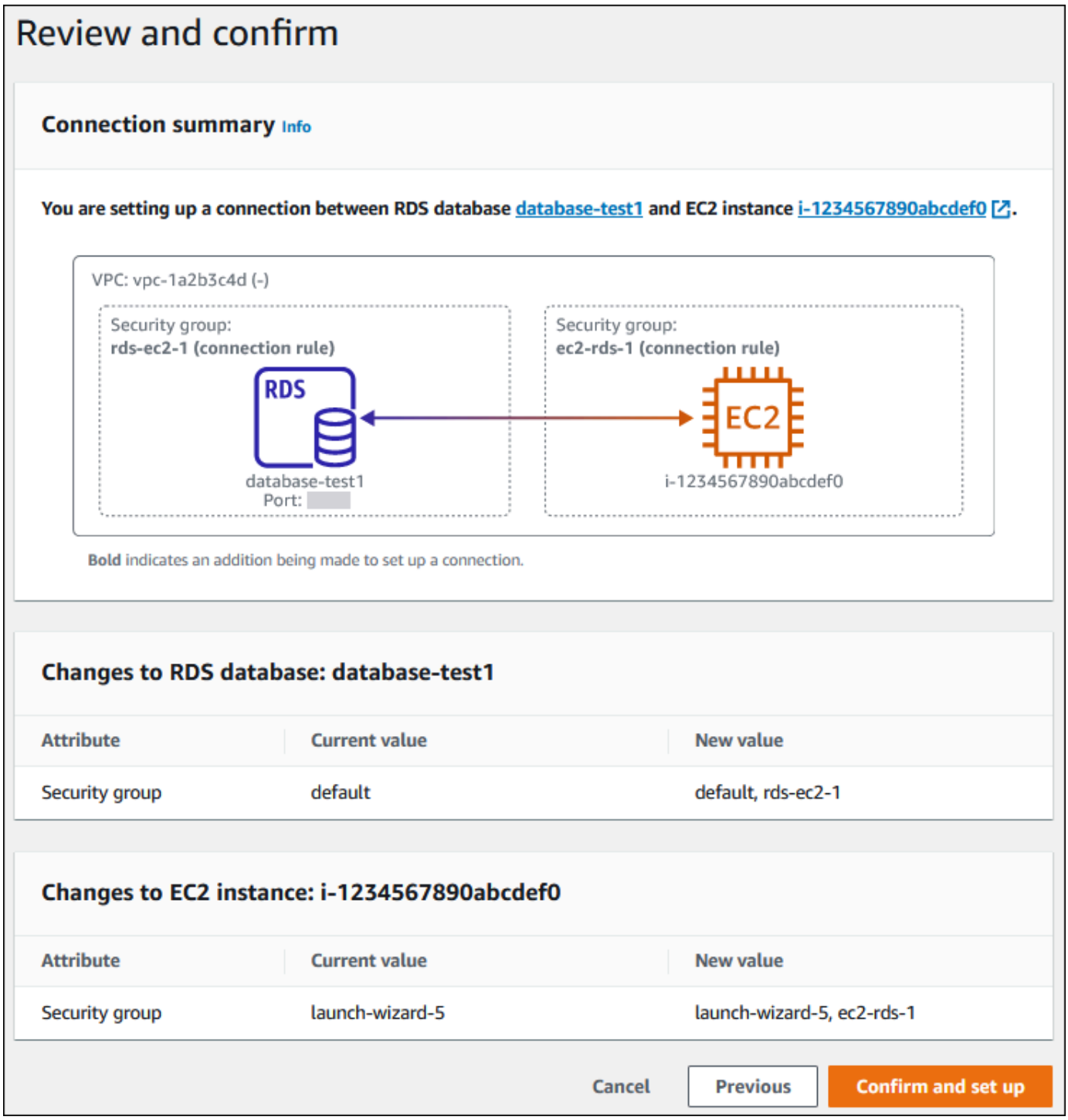
4. Na página Set up EC2 connection (Configurar conexão do EC2), selecione a instância do EC2.



   Se não houver nenhuma instância do EC2 na mesma VPC, selecione Create EC2 instance (Criar instância do EC2) para criar uma. Nesse caso, a nova instância do EC2 deve estar na mesma VPC do banco de dados do RDS.

5. Escolha Continuar.

   A página Review and confirm (Revisar e confirmar) é exibida.



6. Na página Review and confirm (Revisar e confirmar), analise as alterações que o RDS fará para configurar a conectividade com a instância do EC2.

   Se as alterações estiverem corretas, selecione Confirmar e configurar.

   Se as alterações não estiverem corretas, selecione Previous (Anterior) ou Cancel (Cancelar).

Visualizar recursos computacionais conectados

Você pode usar o AWS Management Console para visualizar os recursos computacionais conectados a um banco de dados do RDS. Os recursos mostrados incluem conexões de recursos computacionais que foram configuradas automaticamente. Você pode configurar a conectividade com recursos computacionais automaticamente das seguintes maneiras:

- Você pode selecionar o recurso computacional ao criar o banco de dados.

  Para ter mais informações, consulte Criar uma instância de banco de dados do Amazon RDS e Criar um cluster de banco de dados multi-AZ para o Amazon RDS.
- Você pode configurar a conectividade entre um banco de dados existente e um recurso computacional.

  Para ter mais informações, consulte Conectar automaticamente uma instância do EC2 e um banco de dados RDS.

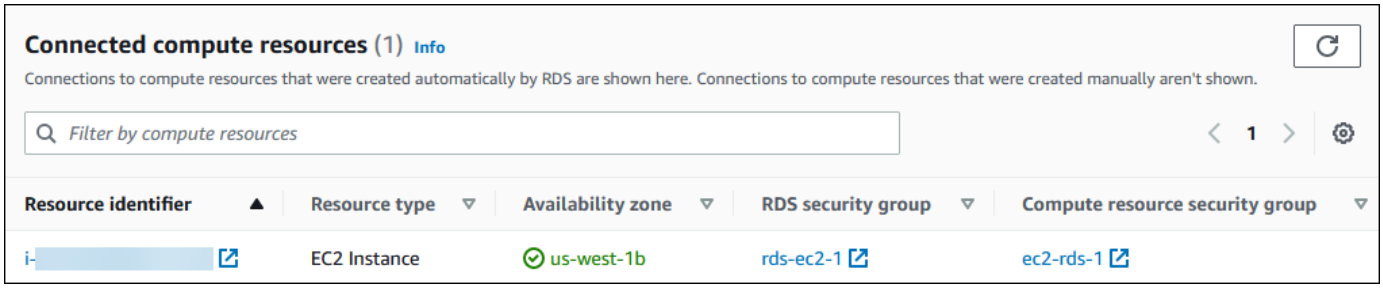
Os recursos computacionais listados não incluem aqueles que foram conectados manualmente ao banco de dados. Por exemplo, você pode permitir que um recurso computacional acesse um banco de dados manualmente adicionando uma regra ao grupo de segurança da VPC associado ao banco de dados.

Para que um recurso computacional seja listado, as seguintes condições devem ser atendidas:

- O nome do grupo de segurança associado ao recurso de computação corresponde ao padrão `ec2-rds-`*n* (em que *n* é um número).
- O grupo de segurança associado ao recurso de computação tem uma regra de saída com o intervalo de portas definido como a porta usada pelo banco de dados RDS.
- O grupo de segurança associado ao recurso computacional tem uma regra de saída com o intervalo de portas definido como um grupo de segurança associado ao banco de dados do RDS.
- O nome do grupo de segurança associado ao banco de dados RDS corresponde ao padrão `rds-ec2-`*n* (em que *n* é um número).
- O grupo de segurança associado ao banco de dados RDS tem uma regra de entrada com o intervalo de portas definido como a porta usada pelo banco de dados RDS.
- O grupo de segurança associado ao banco de dados RDS tem uma regra de entrada com a fonte definida como um grupo de segurança associado ao recurso computacional.

Como visualizar os recursos de computação conectados a um banco de dados do RDS

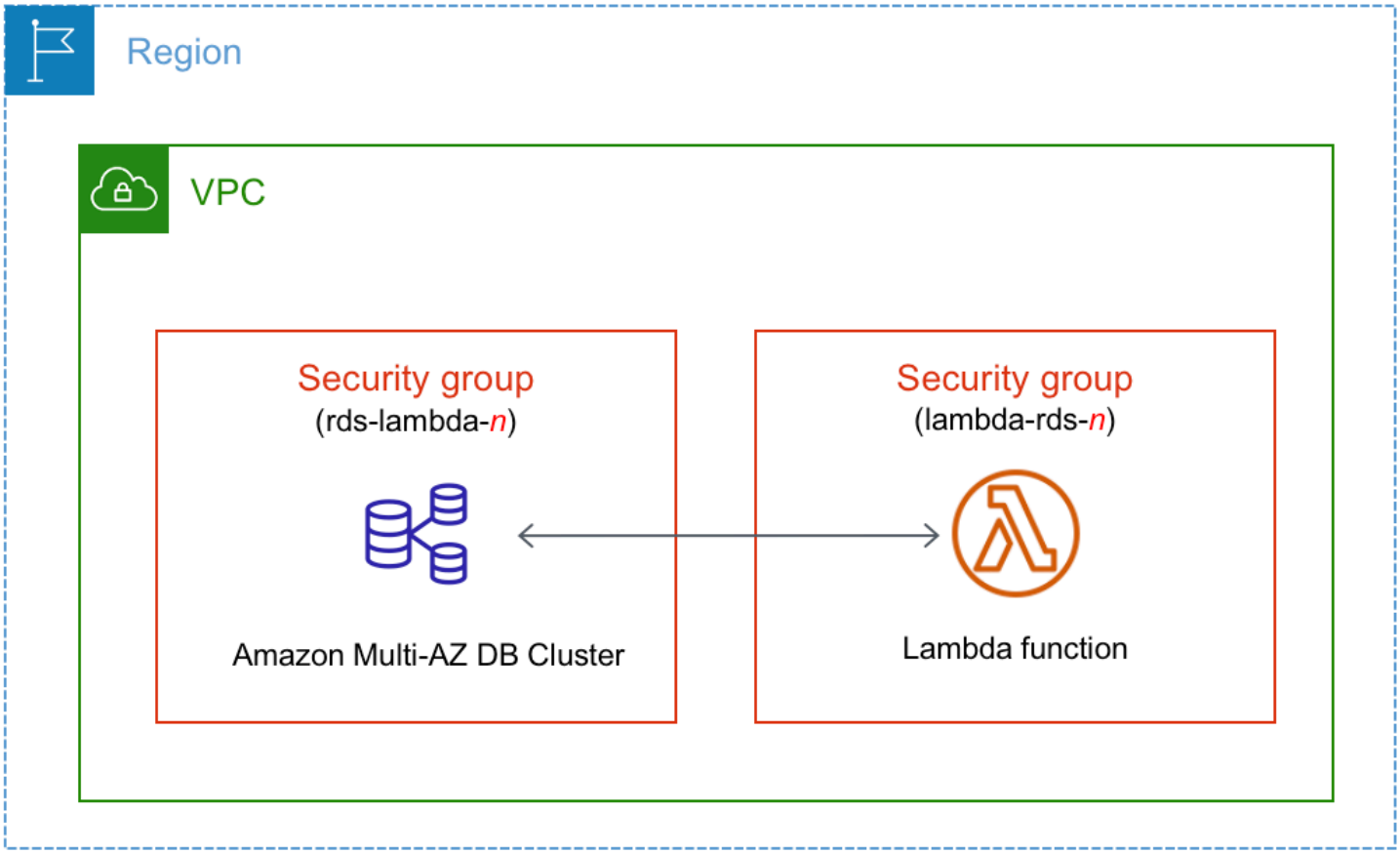
1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, selecione Databases (Bancos de dados) e depois selecione o nome do banco de dados RDS.
3. Na guia Connectivity & security (Conectividade e segurança), veja os recursos computacionais em Connected compute resources (Recursos computacionais conectados).



Conectar automaticamente uma função do Lambda e um cluster de banco de dados multi-AZ

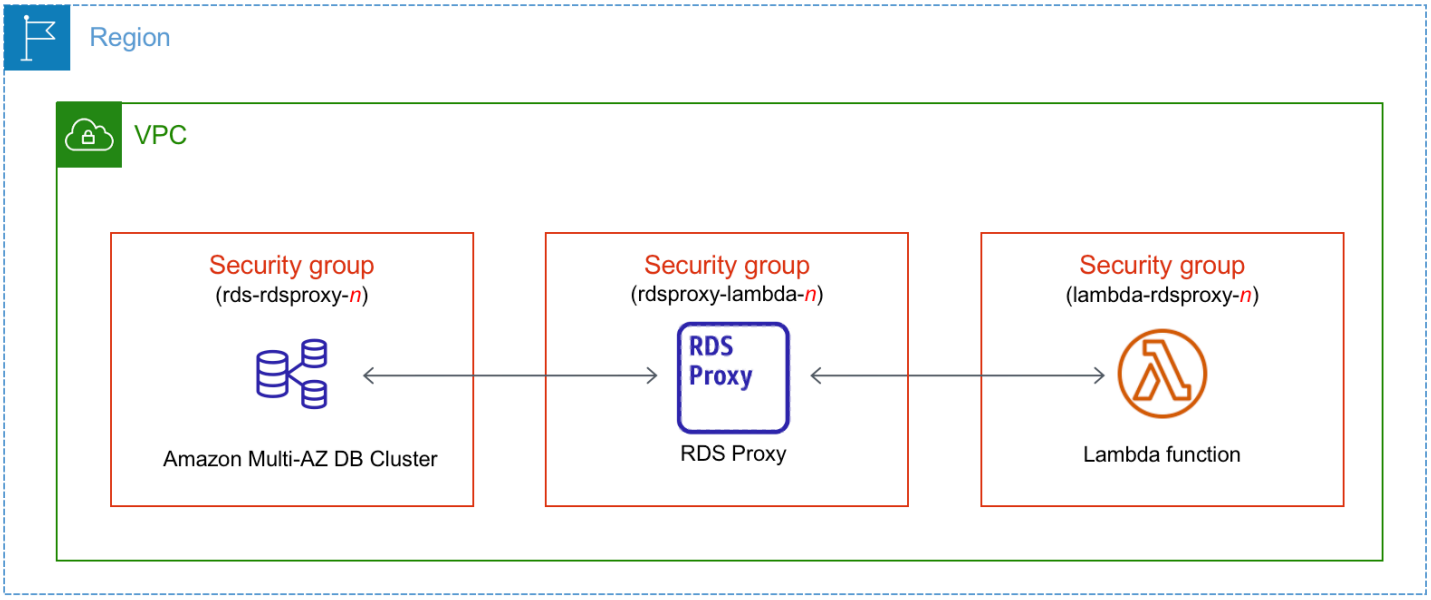
Você pode usar o console do RDS para simplificar a configuração de uma conexão entre uma função do Lambda e um cluster de banco de dados multi-AZ. Você pode usar o console do RDS para simplificar a configuração de uma conexão entre uma função do Lambda e um cluster de banco de dados multi-AZ. Muitas vezes, o cluster de banco de dados multi-AZ está em uma sub-rede privada dentro de uma VPC. A função do Lambda pode ser usada por aplicações para acessar o cluster de banco de dados multi-AZ privado.

A imagem a seguir mostra uma conexão direta entre o cluster de banco de dados multi-AZ e a função do Lambda.



Você pode configurar a conexão entre a função do Lambda e o banco de dados por meio do RDS Proxy para melhorar a performance e a resiliência do banco de dados. Em geral, as funções do Lambda fazem conexões curtas frequentes com o banco de dados que se beneficiam do grupo de conexões oferecido pelo RDS Proxy. É possível aproveitar qualquer autenticação do IAM que você já tenha para funções do Lambda, em vez de gerenciar credenciais de banco de dados no código de aplicação do Lambda. Para obter mais informações, consulte Amazon RDS Proxy.

Você pode usar o console para criar automaticamente um proxy para sua conexão. Também é possível selecionar proxies existentes. O console atualiza o grupo de segurança do proxy para permitir conexões do banco de dados e da função do Lambda. Você pode inserir suas credenciais do banco de dados ou selecionar o segredo do Secrets Manager necessário para acessar o banco de dados.



Tópicos

- Visão geral da conectividade automática com uma função do Lambda
- Conectar automaticamente uma função do Lambda e um cluster de banco de dados multi-AZ
- Visualizar recursos de computação conectados

Visão geral da conectividade automática com uma função do Lambda

Quando você configura uma conexão entre uma função do Lambda e um cluster de banco de dados multi-AZ, o Amazon RDS configura o grupo de segurança da VPC para a instância do Lambda e para o cluster de banco de dados.

Confira a seguir os requisitos para conectar uma instância do Lambda a um cluster de banco de dados multi-AZ:

- A função do Lambda deve existir na mesma VPC que o cluster de banco de dados multi-AZ.

  Se não houver nenhuma função do Lambda na mesma VPC, o console fornecerá um link para que você crie uma.
- O usuário que configura a conectividade deve ter permissões para realizar as seguintes operações do Amazon RDS, do Amazon EC2, do Lambda, do Secrets Manager e do IAM:
  - Amazon RDS
    - `rds:CreateDBProxies`

    - rds:DescribeDBInstances
    - rds:DescribeDBProxies
    - rds:ModifyDBInstance
    - rds:ModifyDBProxy
    - rds:RegisterProxyTargets
- Amazon EC2
    - ec2:AuthorizeSecurityGroupEgress
    - ec2:AuthorizeSecurityGroupIngress
    - ec2:CreateSecurityGroup
    - ec2:DeleteSecurityGroup
    - ec2:DescribeSecurityGroups
    - ec2:RevokeSecurityGroupEgress
    - ec2:RevokeSecurityGroupIngress
- Lambda
    - lambda:CreateFunctions
    - lambda:ListFunctions
    - lambda:UpdateFunctionConfiguration
- Secrets Manager
    - sercetsmanager:CreateSecret
    - secretsmanager:DescribeSecret
- IAM
    - iam:AttachPolicy
    - iam:CreateRole
    - iam:CreatePolicy
- AWS KMS
    - kms:describeKey

Quando você configura uma conexão entre uma função do Lambda e um cluster de banco de dados multi-AZ, o Amazon RDS configura o grupo de segurança da VPC para sua instância e para o cluster de banco de dados multi-AZ. Se você usar o RDS Proxy, o Amazon RDS também configura o grupo de segurança da VPC para o proxy. O Amazon RDS atua de acordo com a configuração atual dos

grupos de segurança associados ao cluster de banco de dados multi-AZ, à função do Lambda e ao proxy, conforme descrito na tabela a seguir.

Configuração atual do grupo de segurança do RDS	Configuração do grupo de segurança do Lambda atual	Configuração do grupo de segurança de proxy atual	Ação do RDS
O Amazon RDS não executa nenhuma ação porque os grupos de segurança de todos os recursos seguem o padrão de nomenclatura correto e têm as regras corretas de entrada e saída.	Há um ou mais grupos de segurança associados ao cluster de banco de dados multi-AZ com um nome correspondente ao padrão `rds-lambda-` *n* (em que *n* é um número) ou se `TargetHealth` de um proxy associado estiver `AVAILABLE`. Um grupo de segurança que corresponde ao padrão não foi modificado. Esse grupo de segurança tem apenas uma regra de entrada com o grupo de segurança da VPC da função do Lambda ou do proxy como origem.	Há um ou mais grupos de segurança associados à função do Lambda com um nome correspondente ao padrão `lambda-rds-` *n* ou `lambda-rdsproxy-` *n* (em que *n* é um número). Um grupo de segurança que corresponde ao padrão não foi modificado. Esse grupo de segurança tem apenas uma regra de saída com o grupo de segurança da VPC do cluster de banco de dados multi-AZ ou do proxy como destino.	Há um ou mais grupos de segurança associados ao proxy com um nome correspondente ao padrão `rdsproxy-lambda-` *n* (em que *n* é um número). Um grupo de segurança que corresponde ao padrão não foi modificado. Esse grupo de segurança tem regras de entrada e saída com os grupos de segurança da VPC da função do Lambda e do cluster de banco de dados multi-AZ.
		Qualquer uma das seguintes condições se aplica:	RDS action: create new security groups

Configuração atual do grupo de segurança do RDS	Configuração do grupo de segurança do Lambda atual	Configuração do grupo de segurança de proxy atual	Ação do RDS
Qualquer uma das seguintes condições se aplica: • Não há nenhum grupo de segurança associado ao cluster de banco de dados multi-AZ com um nome correspondente ao padrão `rds-lambda-` *n* ou se o `TargetHealth` de um proxy associado estiver `AVAILABLE`. • Há um ou mais grupos de segurança associados ao cluster de banco de dados multi-AZ com um nome correspondente ao padrão `rds-lambda-` *n* ou se o `TargetHealth` de um proxy associado	Qualquer uma das seguintes condições se aplica: • Não há um grupo de segurança associado à função do Lambda com um nome correspondente ao padrão `lambda-rds-` *n* ou `lambda-rdsproxy-` *n*. • Há um ou mais grupos de segurança associados à função do Lambda com um nome correspondente ao padrão `lambda-rds-` *n* ou `lambda-rdsproxy-` *n*. No entanto, o Amazon RDS não pode usar nenhum desses grupos de segurança para a conexão com o cluster de banco de dados multi-AZ.	• Não há um grupo de segurança associado ao proxy com um nome correspondente ao padrão `rdsproxy-lambda-` *n*. • Há um ou mais grupos de segurança associados ao proxy com um nome correspondente ao `rdsproxy-lambda-` *n*. No entanto, o Amazon RDS não pode usar nenhum desses grupos de segurança para a conexão com o cluster de banco de dados multi-AZ ou a função do Lambda. O Amazon RDS não pode usar um grupo	

Configuração atual do grupo de segurança do RDS	Configuração do grupo de segurança do Lambda atual	Configuração do grupo de segurança de proxy atual	Ação do RDS
estiver `AVAILABLE`. No entanto, o Amazon RDS não pode usar nenhum desses grupos de segurança para a conexão com a função do Lambda. O Amazon RDS não pode usar um grupo de segurança que não tenha uma regra de entrada no grupo de segurança da VPC da função do Lambda ou do proxy como origem. O Amazon RDS também não pode usar um grupo de segurança que tenha sido modificado. São exemplos de modificação a adição de uma regra ou a alteração da porta de uma regra existente.	O Amazon RDS não poderá usar um grupo de segurança se não tiver uma regra de saída com o grupo de segurança da VPC do cluster de banco de dados multi-AZ como origem. O Amazon RDS também não pode usar um grupo de segurança que tenha sido modificado.	de segurança que não tenha regras de entrada e saída com o grupo de segurança da VPC do cluster de banco de dados multi-AZ ou a função do Lambda. O Amazon RDS também não pode usar um grupo de segurança que tenha sido modificado.	

Configuração atual do grupo de segurança do RDS	Configuração do grupo de segurança do Lambda atual	Configuração do grupo de segurança de proxy atual	Ação do RDS
Há um ou mais grupos de segurança associados ao cluster de banco de dados multi-AZ com um nome correspon dente ao padrão `rds-lambda-` *n* ou se o `TargetHea lth` de um proxy associado estiver `AVAILABLE` . Um grupo de segurança que corresponde ao padrão não foi modificado. Esse grupo de segurança tem apenas uma regra de entrada com o grupo de segurança da VPC da função do Lambda ou do proxy como origem.	Há um ou mais grupos de segurança associados à função do Lambda com um nome correspondente ao padrão `lambda-rds-` *n* ou `lambda-rdsproxy-` *n*. No entanto, o Amazon RDS não pode usar nenhum desses grupos de segurança para a conexão com o cluster de banco de dados multi-AZ. O Amazon RDS não pode usar um grupo de segurança que não tenha uma regra de saída com o grupo de segurança da VPC do cluster de banco de dados multi-AZ ou do proxy como destino. O Amazon RDS também não pode usar um grupo de segurança que tenha sido modificad o.	Há um ou mais grupos de segurança associados ao proxy com um nome correspondente ao padrão `rdsproxy-lambda-` *n*. No entanto, o Amazon RDS não pode usar nenhum desses grupos de segurança para a conexão com o cluster de banco de dados multi-AZ ou a função do Lambda. O Amazon RDS não pode usar um grupo de segurança que não tenha regras de entrada e saída com o grupo de segurança da VPC do cluster de banco de dados multi-AZ ou a função do Lambda. O Amazon RDS também não pode usar um grupo de segurança que tenha sido modificad o.	RDS action: create new security groups

Configuração atual do grupo de segurança do RDS	Configuração do grupo de segurança do Lambda atual	Configuração do grupo de segurança de proxy atual	Ação do RDS
Há um ou mais grupos de segurança associados ao cluster de banco de dados multi-AZ com um nome correspondente ao padrão `rds-lambda-` *n* ou se o `TargetHealth` de um proxy associado estiver `AVAILABLE`. Um grupo de segurança que corresponde ao padrão não foi modificado. Esse grupo de segurança tem apenas uma regra de entrada com o grupo de segurança da VPC da função do Lambda ou do proxy como origem.	Existe um grupo de segurança do Lambda válido para a conexão, mas ele não está associado à função do Lambda. Esse grupo de segurança tem um nome correspondente ao padrão `lambda-rds-` *n* ou `lambda-rdsproxy-` *n*. Não foi modificado. Ele tem apenas uma regra de saída com o grupo de segurança da VPC do cluster de banco de dados multi-AZ ou do proxy como destino.	Existe um grupo de segurança do proxy válido para a conexão, mas ele não está associado ao proxy. Esse grupo de segurança tem um nome que corresponde ao padrão `rdsproxy-lambda-` *n*. Não foi modificado. Ele tem regras de entrada e saída com os grupos de segurança da VPC do cluster de banco de dados multi-AZ e da função do Lambda.	RDS action: associate Lambda security group

Configuração atual do grupo de segurança do RDS	Configuração do grupo de segurança do Lambda atual	Configuração do grupo de segurança de proxy atual	Ação do RDS
Qualquer uma das seguintes condições se aplica: • Não há nenhum grupo de segurança associado ao cluster de banco de dados multi-AZ com um nome correspondente ao padrão `rds-lambda-` *n* ou se o `TargetHealth` de um proxy associado estiver `AVAILABLE`. • Há um ou mais grupos de segurança associados ao cluster de banco de dados multi-AZ com um nome correspondente ao padrão `rds-lambda-` *n* ou se o `TargetHealth` de um	Há um ou mais grupos de segurança associados à função do Lambda com um nome correspondente ao padrão `lambda-rds-` *n* ou `lambda-rdsproxy-` *n*. Um grupo de segurança que corresponde ao padrão não foi modificado. Esse grupo de segurança tem apenas uma regra de saída com o grupo de segurança da VPC do cluster de banco de dados multi-AZ ou do proxy como destino.	Há um ou mais grupos de segurança associados ao proxy com um nome correspondente ao padrão `rdsproxy-lambda-` *n*. Um grupo de segurança que corresponde ao padrão não foi modificado. Esse grupo de segurança tem regras de entrada e saída com os grupos de segurança da VPC do cluster de banco de dados multi-AZ e da função do Lambda.	RDS action: create new security groups

Configuração atual do grupo de segurança do RDS	Configuração do grupo de segurança do Lambda atual	Configuração do grupo de segurança de proxy atual	Ação do RDS
proxy associado estiver `AVAILABLE`. No entanto, o Amazon RDS não pode usar nenhum desses grupos de segurança para a conexão com a função do Lambda ou o proxy. O Amazon RDS não pode usar um grupo de segurança que não tenha uma regra de entrada no grupo de segurança da VPC da função do Lambda ou do proxy como origem. O Amazon RDS também não pode usar um grupo de segurança que tenha sido modificado.			

Configuração atual do grupo de segurança do RDS	Configuração do grupo de segurança do Lambda atual	Configuração do grupo de segurança de proxy atual	Ação do RDS
Há um ou mais grupos de segurança associados ao cluster de banco de dados multi-AZ com um nome que correspon de ao padrão `rds-rdsproxy-` *n* (em que *n* é um número).	Qualquer uma das seguintes condições se aplica: • Não há um grupo de segurança associado à função do Lambda com um nome correspondente ao padrão `lambda-rds-` *n* ou `lambda-rdsproxy-` *n*. • Há um ou mais grupos de segurança associados à função do Lambda com um nome correspondente ao padrão `lambda-rds-` *n* ou `lambda-rdsproxy-` *n*. No entanto, o Amazon RDS não pode usar nenhum desses grupos de segurança para a conexão com o cluster de banco de dados multi-AZ.	Qualquer uma das seguintes condições se aplica: • Não há um grupo de segurança associado ao proxy com um nome correspondente ao padrão `rdsproxy-lambda-` *n*. • Há um ou mais grupos de segurança associados ao proxy com um nome correspondente ao `rdsproxy-lambda-` *n*. No entanto, o Amazon RDS não pode usar nenhum desses grupos de segurança para a conexão com o cluster de banco de dados multi-AZ ou a função do Lambda.	RDS action: create new security groups

Configuração atual do grupo de segurança do RDS	Configuração do grupo de segurança do Lambda atual	Configuração do grupo de segurança de proxy atual	Ação do RDS
	O Amazon RDS não pode usar um grupo de segurança que não tenha uma regra de saída com o grupo de segurança da VPC do cluster de banco de dados multi-AZ ou do proxy como destino. O Amazon RDS também não pode usar um grupo de segurança que tenha sido modificado.	O Amazon RDS não pode usar um grupo de segurança que não tenha regras de entrada e saída com o grupo de segurança da VPC do cluster de banco de dados multi-AZ ou a função do Lambda. O Amazon RDS também não pode usar um grupo de segurança que tenha sido modificado.	

Ação do RDS: criar grupos de segurança

O Amazon RDS realiza as seguintes ações:

- Cria um grupo de segurança correspondente ao padrão `rds-lambda-`*n*. Esse grupo de segurança tem uma regra de entrada com o grupo de segurança da VPC da função do Lambda ou do proxy como origem. Esse grupo de segurança está associado ao cluster de banco de dados multi-AZ e permite que a função ou o proxy acesse o cluster de banco de dados multi-AZ.
- Cria um grupo de segurança que corresponde ao padrão `lambda-rds-`*n*. Esse grupo de segurança tem uma regra de saída com o grupo de segurança da VPC do cluster de banco de dados multi-AZ ou do proxy como destino. Esse grupo de segurança está associado à função do Lambda e permite que ela envie tráfego para o cluster de banco de dados multi-AZ ou envie tráfego por meio de um proxy.
- Cria um grupo de segurança que corresponde ao padrão `rdsproxy-lambda-`*n*. Esse grupo de segurança tem regras de entrada e saída com os grupos de segurança da VPC do cluster de banco de dados multi-AZ e da função do Lambda.

Ação do RDS: associar o grupo de segurança do Lambda

O Amazon RDS associa o grupo de segurança do Lambda válido e existente à função do Lambda. Esse grupo de segurança permite que a função envie tráfego para o cluster de banco de dados multi-AZ ou envie tráfego por meio de um proxy.

Conectar automaticamente uma função do Lambda e um cluster de banco de dados multi-AZ

Você pode usar o console do Amazon RDS para conectar automaticamente uma função do Lambda ao cluster de banco de dados multi-AZ. Isso simplifica o processo de configuração de uma conexão entre esses recursos.

Você também pode usar o RDS Proxy para incluir um proxy em sua conexão. As funções do Lambda fazem conexões curtas frequentes com o banco de dados que se beneficiam do grupo de conexões oferecido pelo RDS Proxy. Também é possível usar qualquer autenticação do IAM que você já tenha para funções do Lambda, em vez de gerenciar credenciais de banco de dados no código da aplicação do Lambda.

Você pode conectar um cluster de banco de dados multi-AZ existente a funções do Lambda novas e existentes usando a página Configurar conexão do Lambda. O processo de configuração define automaticamente os grupos de segurança necessários para você.

Antes de configurar uma conexão entre uma função do Lambda e um cluster de banco de dados multi-AZ, garanta que:

- A função do Lambda e o cluster de banco de dados multi-AZ estejam na mesma VPC.
- Você tenha as permissões corretas para sua conta do usuário. Para receber mais informações sobre os requisitos, consulte Visão geral da conectividade automática com uma função do Lambda.

Se você alterar os grupos de segurança depois de configurar a conectividade, as alterações poderão afetar a conexão entre a função do Lambda e o cluster de banco de dados multi-AZ.

> **Note**
>
> Você possa configurar automaticamente uma conexão entre um cluster de banco de dados multi-AZ e uma função do Lambda somente no AWS Management Console. Para conectar uma função do Lambda, todas as instâncias no cluster de banco de dados multi-AZ devem estar no estado Disponível.

Como conectar automaticamente uma função do Lambda e um cluster de banco de dados multi-AZ

<result>

Depois de confirmar a configuração, o Amazon RDS inicia o processo de conexão da função do Lambda, do RDS Proxy (se você usou um proxy) e do cluster de banco de dados multi-AZ. O console mostra a caixa de diálogo Detalhes da conexão, que lista as alterações do grupo de segurança que permitem conexões entre seus recursos.
</result>

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, selecione Bancos de dados e, depois, o cluster de banco de dados multi-AZ que você deseja conectar a uma função do Lambda.
3. Em Ações, selecione Configurar conexão do Lambda.
4. Na página Configurar conexão do Lambda, em Selecionar função do Lambda, faça o seguinte:
   - Se você tiver uma função do Lambda na mesma VPC que o cluster de banco de dados multi-AZ, escolha Selecionar função existente e, depois, escolha a função.
   - Se você não tiver uma função do Lambda na mesma VPC, selecione Criar função do Lambda e, depois, insira um Nome da função. O runtime padrão é definido como Nodejs.18. Você pode modificar as configurações de sua nova função do Lambda no console do Lambda depois de concluir a configuração da conexão.
5. (Opcional) Em RDS Proxy, selecione Conectar usando o RDS Proxy e, depois, faça o seguinte:
   - Se você tiver um proxy que deseja usar, escolha Selecionar um proxy existente e, depois, escolha o proxy.
   - Se você não tiver um proxy e quiser que o Amazon RDS crie um automaticamente para você, selecione Criar um proxy. Então, para Credenciais de banco de dados, faça o seguinte:

     a. Selecione Nome de usuário e senha do banco de dados e, depois, insira o Nome do usuário e a Senha para o cluster de banco de dados multi-AZ.

     b. Selecione Segredo do Secrets Manager. Então, em Selecionar segredo, escolha um segredo do AWS Secrets Manager. Se você não tiver um segredo do Secrets Manager, selecione Criar um segredo do Secrets Manager para criar um segredo. Depois de criar o segredo, em Selecionar segredo, escolha o novo segredo.

Depois de criar o proxy, escolha Selecionar proxy existente e, depois, escolha o proxy. Observe que pode levar algum tempo até que o proxy esteja disponível para conexão.

6. (Opcional) Expanda Resumo da conexão e verifique as atualizações destacadas para seus recursos.
7. Escolha Set up (Configurar).

Visualizar recursos de computação conectados

Você pode usar o AWS Management Console para visualizar os recursos de computação conectados a um cluster de banco de dados multi-AZ. Os recursos mostrados incluem conexões de recursos de computação que o Amazon RDS configurou automaticamente.

Os recursos de computação listados não incluem aqueles que são conectados manualmente ao cluster de banco de dados multi-AZ. Por exemplo, você pode permitir que um recurso de computação acesse o cluster de banco de dados multi-AZ manualmente adicionando uma regra ao grupo de segurança da VPC associado ao cluster.

Para que o console liste uma função do Lambda, as seguintes condições devem ser aplicadas:

- O nome do grupo de segurança associado ao recurso de computação corresponde ao padrão `lambda-rds-`*n* ou `lambda-rdsproxy-`*n* (em que *n* é um número).
- O grupo de segurança associado ao recurso de comutação tem uma regra de saída com o intervalo de portas definido como a porta do cluster de banco de dados multi-AZ ou de um proxy associado. O destino da regra de saída deve ser definido como um grupo de segurança associado ao cluster de banco de dados multi-AZ ou a um proxy associado.
- O nome do grupo de segurança anexado ao proxy associado ao banco de dados corresponde ao padrão `rds-rdsproxy-`*n* (em que *n* é um número).
- O grupo de segurança associado à função tem uma regra de saída com a porta definida como a porta que o cluster de banco de dados multi-AZ ou o proxy associado usa. O destino deve ser definido como um grupo de segurança associado ao cluster de banco de dados multi-AZ ou ao proxy associado.

Como visualizar os recursos de computação conectados automaticamente a um cluster de banco de dados multi-AZ

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, selecione Bancos de dados e, depois, o cluster de banco de dados multi-AZ.
3. Na guia Conectividade e segurança, veja os recursos de computação em Recursos de computação conectados.

Modificar um cluster de banco de dados multi-AZ para o Amazon RDS

Um cluster de banco de dados multi-AZ tem uma instância de banco de dados de gravador e duas instâncias de banco de dados de leitor em três zonas de disponibilidade separadas. Clusters de banco de dados multi-AZ oferecem alta disponibilidade, maior capacidade para workloads de leitura e menor latência quando comparados a implantações multi-AZ. Para obter mais informações sobre clusters de banco de dados multi-AZ, consulte Implantações de cluster de banco de dados multi-AZ para o Amazon RDS.

Você pode modificar um cluster de banco de dados multi-AZ para alterar suas configurações. Você também pode realizar operações em um cluster de banco de dados multi-AZ, como obter snapshots.

> **Important**
>
> Não é possível modificar as instâncias de banco de dados dentro de um cluster de banco de dados multi-AZ. Todas as modificações devem ser feitas no nível do cluster de banco de dados. A única operação que é possível realizar em uma instância de banco de dados dentro de um cluster de banco de dados multi-AZ é reinicializá-la.

Você pode modificar um cluster de banco de dados multi-AZ usando o AWS Management Console, a AWS CLI ou a API do RDS.

Console

Para modificar um cluster de banco de dados multi-AZ

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Bancos de dados e o cluster de banco de dados multi-AZ que você deseja modificar.
3. Escolha Modify (Modificar). A página Modify DB cluster (Modificar cluster de banco de dados) é exibida.
4. Altere qualquer uma das configurações desejadas. Para obter informações sobre cada configuração, consulte Configurações para modificar clusters de banco de dados multi-AZ.
5. Quando todas as alterações estiverem conforme o desejado, escolha Continue (Continuar) e verifique o resumo das modificações.

6. (Opcional) Escolha Apply immediately (Aplicar imediatamente) para aplicar as alterações imediatamente. Escolher essa opção pode causar uma inatividade em alguns casos. Para ter mais informações, consulte Aplicar alterações imediatamente.
7. Na página de confirmação, revise suas alterações. Se estiverem corretas, escolha Modify DB cluster (Modificar cluster de banco de dados) para salvar as alterações.

   Ou escolha Back (Voltar) para editar as alterações ou Cancel (Cancelar) para cancelar as alterações.

AWS CLI

Para modificar um cluster de banco de dados multi-AZ utilizando a AWS CLI, chame o comando modify-db-cluster. Especifique o identificador de instância de banco de dados e os valores para as configurações que você deseja modificar. Para obter mais informações sobre cada opção, consulte Configurações para modificar clusters de banco de dados multi-AZ.

Example

O código a seguir modifica `my-multi-az-dbcluster` configurando o período de retenção de backup como 1 semana (7 dias). O código ativa a proteção contra exclusão, utilizando `--deletion-protection`. Para desativar a proteção contra exclusão, utilize `--no-deletion-protection`. As alterações serão aplicadas durante a janela de manutenção usando `--no-apply-immediately`. Use `--apply-immediately` para aplicar as alterações imediatamente. Para ter mais informações, consulte Aplicar alterações imediatamente.

Para Linux, macOS ou Unix:

```
aws rds modify-db-cluster \
    --db-cluster-identifier my-multi-az-dbcluster \
    --backup-retention-period 7 \
    --deletion-protection \
    --no-apply-immediately
```

Para Windows:

```
aws rds modify-db-cluster ^
    --db-cluster-identifier my-multi-az-dbcluster ^
    --backup-retention-period 7 ^
    --deletion-protection ^
```

```
    --no-apply-immediately
```

API do RDS

Para modificar um cluster de banco de dados multi-AZ usando a API do Amazon RDS, chame a operação ModifyDBCluster. Especifique o identificador de cluster de banco de dados e os parâmetros para as configurações que você deseja modificar. Para obter informações sobre cada parâmetro, consulte Configurações para modificar clusters de banco de dados multi-AZ.

Aplicar alterações imediatamente

Quando você modifica um cluster de banco de dados multi-AZ, pode aplicar as alterações imediatamente. Para aplicar as alterações imediatamente, escolha a opção Apply Immediately (Aplicar imediatamente) no AWS Management Console. Se preferir, use a opção `--apply-immediately` ao chamar AWS CLI ou defina o parâmetro `ApplyImmediately` como `true` ao usar a API do Amazon RDS.

Se você não optar por aplicar as alterações imediatamente, elas serão colocadas na fila de modificações pendentes. Durante a próxima janela de manutenção, todas as alterações pendentes na fila serão aplicadas. Se você optar por aplicar as alterações imediatamente, as novas alterações e todas as alterações na fila de modificações pendentes serão aplicadas.

> **Important**
>
> Se alguma das modificações pendentes exigir que o cluster de banco de dados esteja temporariamente indisponível (inatividade), a escolha da opção para aplicar imediatamente poderá causar tempo de inatividade inesperado.
> Quando você opta por aplicar uma alteração imediatamente, todas as modificações pendentes também são aplicadas de imediato, em vez de durante a próxima janela de manutenção.
> Se não quiser que uma alteração pendente seja aplicada na próxima janela de manutenção, é possível modificar a instância de banco de dados para reverter a alteração. Você pode fazer isso usando a AWS CLI e especificando a opção `--apply-immediately`.

As alterações em algumas configurações do banco de dados são aplicadas imediatamente, mesmo que você opte por adiá-las. Para ver como as diferentes configurações do banco de dados interagem com a configuração Aplicar imediatamente, consulte Configurações para modificar clusters de banco de dados multi-AZ.

Configurações para modificar clusters de banco de dados multi-AZ

Para obter detalhes sobre as configurações que é possível utilizar para modificar um cluster de banco de dados multi-AZ, consulte a tabela a seguir. Para obter mais informações sobre as opções da AWS CLI, consulte modify-db-cluster. Para obter mais informações sobre os parâmetros da API do RDS, consulte ModifyDBCluster.

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS	Quando a alteração ocorre	Observações sobre tempo de inatividade

Armazenamento alocado	O valor de armazenamento a ser alocado para cada instância de banco de dados no seu cluster de banco de dados (em gibibyte). Para ter mais informações, consulte Armazenamento de instâncias de banco de dados do Amazon RDS.	Opção da CLI:
		`--allocated-storage`
		Parâmetro da API do RDS:
		`AllocatedStorage`
			Se você optar por aplicar a alteração imediatamente, ela ocorrerá imediatamente.
			Se você não optar por aplicar a alteração imediatamente, ela ocorrerá durante a próxima janela de manutenção.
				Não ocorre uma inatividade durante esta alteração.

Atualização da versão secundária automática	Selecione Enable auto minor version upgrade (Habilitar atualização automática da versão secundária) para que o cluster de banco de dados receba	Opção da CLI:
		`--auto-minor-version-upgrade`
		`--no-auto-minor-version-upgrade`
			A alteração ocorre imediatamente. Essa configuração ignora a configuração para aplicar imediatamente.	Ocorre uma inatividade durante esta alteração.

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS	Quando a alteração ocorre	Observações sobre tempo de inatividade
	atualizações automáticas da versão do mecanismo de banco de dados secundárias preferidas quando elas forem disponibilizadas. O Amazon RDS executa atualizações automáticas de versões secundárias na janela de manutenção.	Parâmetro da API do RDS: `AutoMinorVersionUpgrade`		

Período de retenção de backup	O número de dias durante os quais os backups automáticos do cluster de banco de dados ficam retidos. Este valor deve ser maior que zero.
	Para ter mais informações, consulte Introdução aos backups.
		Opção da CLI:
		`--backup-retention-period`
		Parâmetro da API do RDS:
		`BackupRetentionPeriod`
			Se você optar por aplicar a alteração imediatamente, ela ocorrerá imediatamente.
			Se você não optar por aplicar a alteração imediatamente, ela ocorrerá durante a próxima janela de manutenção.
				Não ocorre uma inatividade durante esta alteração.

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS	Quando a alteração ocorre	Observações sobre tempo de inatividade

Janela de backup	O período durante o qual o Amazon RDS faz um backup automático do seu cluster de banco de dados. A menos que você tenha um horário específico no qual deseja que o backup do banco de dados seja feito, utilize o padrão No Preference (Sem preferência).
	
	Para ter mais informações, consulte Introdução aos backups.
		Opção da CLI:
		
		`--preferred-backup-window`
		
		Parâmetro da API do RDS:
		
		`PreferredBackupWindow`
			A alteração é aplicada de forma assíncrona logo que possível.	Não ocorre uma inatividade durante esta alteração.

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS	Quando a alteração ocorre	Observações sobre tempo de inatividade

Autoridade certificadora	A autoridade de certificação (CA) para o certificado do servidor usado pelo cluster de banco de dados.
	
	Para ter mais informações, consulte Usar SSL/TLS para criptografar uma conexão com uma instância ou um cluster de banco de dados.
		Opção da CLI:
		
		`--ca-certificate-identifier`
		
		Parâmetro da API do RDS:
		
		`CACertificateIdentifier`
			Se você optar por aplicar a alteração imediatamente, ela ocorrerá imediatamente.
			
			Se você não optar por aplicar a alteração imediatamente, ela ocorrerá durante a próxima janela de manutenção.
				Tempo de inatividade ocorrerá somente se o mecanismo de banco de dados não for compatível com alternância sem reinicialização. Você pode usar o comando da AWS CLI describe-db-engine-versions para determinar se o mecanismo de banco de dados é compatível com alternância sem reinicialização.

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS	Quando a alteração ocorre	Observações sobre tempo de inatividade

Copiar tags para snapshot	Essa opção copia qualquer etiqueta do cluster de banco de dados para um snapshot do banco de dados quando você cria um snapshot.
	
	Para ter mais informações, consulte Marcar recursos do Amazon RDS.
		Opção da CLI:
		
		`-copy-tags-to-snapshot`
		
		`-no-copy-tags-to-snapshot`
		
		Parâmetro da API do RDS:
		
		`CopyTagsToSnapshot`
			A alteração ocorre imediatamente. Essa configuração ignora a configuração para aplicar imediatamente.	Não ocorre uma inatividade durante esta alteração.

Autenticação de banco de dados	Para clusters de banco de dados multi-AZ, somente a autenticação por senha tem suporte.	Nenhum, pois a autenticação por senha é o padrão.	Se você optar por aplicar a alteração imediatamente, ela ocorrerá imediatamente. Se você não optar por aplicar a alteração imediatamente, ela ocorrerá durante a próxima janela de manutenção.	Não ocorre uma inatividade durante esta alteração.

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS	Quando a alteração ocorre	Observações sobre tempo de inatividade

Identificador do cluster de banco de dados	O identificador de cluster de banco de dados. Esse valor é armazenado como uma string em minúsculas.
	
	Quando você altera o identificador de cluster de banco de dados, o endpoint de cluster de banco de dados é alterado. Os identificadores e os endpoints das instâncias de banco de dados no cluster de banco de dados também mudam. O nome do novo cluster de banco de dados deve ser exclusivo. O tamanho máximo é de 63 caracteres.
	
	Os nomes das instâncias de
		Opção da CLI:
		
		`--new-db-cluster-identifier`
		
		Parâmetro da API do RDS:
		
		`NewDBClusterIdentifier`
			Se você optar por aplicar a alteração imediatamente, ela ocorrerá imediatamente.
			
			Se você não optar por aplicar a alteração imediatamente, ela ocorrerá durante a próxima janela de manutenção.
				Não ocorre uma inatividade durante esta alteração.

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS	Quando a alteração ocorre	Observações sobre tempo de inatividade
	banco de dados no cluster de banco de dados são alterados para corresponder ao novo nome do cluster de banco de dados. O nome de uma nova instância de banco de dados não pode ser igual ao nome de uma instância de banco de dados existente. Por exemplo, se você alterar o nome do cluster de banco de dados para `maz`, o nome de uma instância de banco de dados poderá ser alterado para `maz-instance-1` . Nesse caso, não pode haver uma instância de banco de dados existente			

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS	Quando a alteração ocorre	Observações sobre tempo de inatividade
	chamada `maz-instance-1` . Para ter mais informações, consulte Renomear um cluster de banco de dados multi-AZ para o Amazon RDS.			

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS	Quando a alteração ocorre	Observações sobre tempo de inatividade

Classe de instância de cluster de banco de dados	A capacidade de computação e memória de cada instância de banco de dados no cluster de banco de dados multi-AZ, por exemplo `db.r6gd.xlarge` .
	
	Se possível, escolha uma classe de instância de banco de dados grande o suficiente para um conjunto de trabalho de consulta típico pode ser sustentado na memória. Quando os conjuntos de trabalho são mantidos na memória o sistema pode evitar a gravação em disco, o
		Opção da CLI:
		
		`--db-cluster-instance-class`
		
		Parâmetro da API do RDS:
		
		`DBClusterInstanceClass`
			Se você optar por aplicar a alteração imediatamente, ela ocorrerá imediatamente.
			
			Se você não optar por aplicar a alteração imediatamente, ela ocorrerá durante a próxima janela de manutenção.
				Ocorre uma inatividade durante esta alteração.

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS	Quando a alteração ocorre	Observações sobre tempo de inatividade
	que aprimora a performance. Para ter mais informações, consulte the section called “Disponibilidade de classe de instância para clusters de banco de dados multi-AZ”.			

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS	Quando a alteração ocorre	Observações sobre tempo de inatividade

Grupo de parâmetros do cluster de banco de dados	O parameter group do cluster de banco de dados que você deseja associar ao cluster de banco de dados.
	
	Para ter mais informações, consulte Grupos de parâmetros para clusters de banco de dados multi-AZ.
		Opção da CLI:
		
		`--db-cluster-parameter-group-name`
		
		Parâmetro da API do RDS:
		
		`DBClusterParameterGroupName`
			A alteração de grupo de parâmetros ocorre imediatamente.	Não ocorre uma inatividade durante esta alteração. Quando você altera o grupo de parâmetros, as alterações feitas em alguns parâmetros são aplicadas às instâncias de banco de dados no cluster de banco de dados multi-AZ imediatamente sem reinicialização. Alterações em outros parâmetros são aplicadas somente depois que as instâncias de banco de dados forem reinicializadas.

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS	Quando a alteração ocorre	Observações sobre tempo de inatividade

DB engine version (Versão do mecanismo de banco de dados)	A versão do mecanismo de banco de dados que você deseja usar.	Opção da CLI:
		`--engine-version`
		Parâmetro da API do RDS:
		`EngineVersion`
			Se você optar por aplicar a alteração imediatamente, ela ocorrerá imediatamente.
			Se você não optar por aplicar a alteração imediatamente, ela ocorrerá durante a próxima janela de manutenção.
				Ocorre uma inatividade durante esta alteração.

Deletion protection (Proteção contra exclusão)	Selecione Enable deletion protection (Habilitar proteção contra exclusão) para impedir que seu cluster de banco de dados seja excluído.
	Para ter mais informações, consulte Excluir uma instância de banco de dados.
		Opção da CLI:
		`--deletion-protection`
		`--no-deletion-protection`
		Parâmetro da API do RDS:
		`DeletionProtection`
			A alteração ocorre imediatamente. Essa configuração ignora a configuração para aplicar imediatamente.	Não ocorre uma inatividade durante esta alteração.

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS	Quando a alteração ocorre	Observações sobre tempo de inatividade

Janela de manutenção	A janela de 30 minutos na qual as modificações pendentes do seu cluster de banco de dados serão aplicadas. Se o período não for relevante, selecione No Preference (Sem preferência).
	
	Para ter mais informações, consulte A janela de manutenção do Amazon RDS.
		Opção da CLI:
		
		`--preferred-maintenance-window`
		
		Parâmetro da API do RDS:
		
		`PreferredMaintenanceWindow`
			A alteração ocorre imediatamente. Essa configuração ignora a configuração para aplicar imediatamente.	Se houver uma ou mais ações pendentes que provoquem uma inatividade, e a janela de manutenção for alterada para incluir a hora atual, essas ações pendentes serão aplicadas imediatamente e ocorrerá uma inatividade.

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS	Quando a alteração ocorre	Observações sobre tempo de inatividade

Gerenciar credenciais principais no AWS Secrets Manager	Selecione Gerenciar credenciais principais no AWS Secrets Manager para gerenciar a senha do usuário principal em um segredo no Secrets Manager.
	
	Opcionalmente, selecione uma chave do KMS a ser usada para proteger o segredo. Escolha entre uma das chaves do KMS da sua conta ou insira a chave de uma conta distinta.
	
	Se o RDS já estiver gerenciando a senha do usuário principal para o cluster de banco de dados, você poderá alternar
		Opção da CLI:
		
		`--manage-master-user-password | --no-manage-master-user-password`
		
		`--master-user-secret-kms-key-id`
		
		`--rotate-master-user-password | --no-rotate-master-user-password`
		
		Parâmetro da API do RDS:
		
		`ManageMasterUserPassword`
		
		`MasterUserSecretKmsKeyId`
		
		`RotateMasterUserPassword`
			Se você estiver ativando ou desativando o gerenciamento automático de senhas de usuário principal, a alteração ocorrerá imediatamente. Essa configuração ignora a configuração para aplicar imediatamente.
			
			Se você estiver alternando a senha do usuário principal, deverá especificar que a alteração seja aplicada imediatamente.
				Não ocorre uma inatividade durante esta alteração.

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS	Quando a alteração ocorre	Observações sobre tempo de inatividade
	a senha do usuário principal selecionando Rotate secret immediately (Alternar segredo imediatamente). Para ter mais informações, consulte Gerenciamento de senhas com Amazon RDS e AWS Secrets Manager.			

Nova senha mestre	A senha da conta de usuário principal.	Opção da CLI:
		
		`--master-user-password`
		
		Parâmetro da API do RDS:
		
		`MasterUserPassword`
			A alteração é aplicada de forma assíncrona logo que possível. Essa configuração ignora a configuração para aplicar imediatamente.	Não ocorre uma inatividade durante esta alteração.

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS	Quando a alteração ocorre	Observações sobre tempo de inatividade

IOPS provisionadas	A quantidade de IOPS (operações de entrada/saída por segundo) provisionadas a serem inicialmente alocadas ao cluster de banco de dados.	Opção da CLI:
		`--iops`
		Parâmetro da API do RDS:
		`Iops`
			Se você optar por aplicar a alteração imediatamente, ela ocorrerá imediatamente.
			Se você não optar por aplicar a alteração imediatamente, ela ocorrerá durante a próxima janela de manutenção.
				Não ocorre uma inatividade durante esta alteração.

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS	Quando a alteração ocorre	Observações sobre tempo de inatividade
Acesso público	Publicly accessible (Acessível publicamente) para fornecer ao cluster de banco de dados um endereço IP público, o que significa que ele é acessível fora da nuvem privada virtual (VPC). Para ser acessível publicamente, o cluster de banco de dados também deve estar em uma sub-rede pública na VPC. Not publicly accessible (Não acessível publicamente) para tornar o cluster de banco de dados acessível somente de dentro da VPC.	Não disponível ao modificar uma instância de banco de dados.	A alteração ocorre imediatamente. Essa configuração ignora a configuração para aplicar imediatamente.	Não ocorre uma inatividade durante esta alteração.

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS	Quando a alteração ocorre	Observações sobre tempo de inatividade
	Para ter mais informações, consulte Ocultar uma instância de banco de dados em uma VPC da Internet. Para se conectar a um cluster de banco de dados de fora de sua VPC, o cluster de banco de dados deve ser acessível publicamente. Além disso, deve ser concedido acesso utilizando as regras de entrada do grupo de segurança do cluster de banco de dados, e outros requisitos devem ser atendidos. Para ter mais informações, consulte Não é possível conectar-se à			

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS	Quando a alteração ocorre	Observações sobre tempo de inatividade
	instância de banco de dados do Amazon RDS. Se o cluster de banco de dados não estiver acessível publicamente, será possível usar uma conexão AWS Site-to-Site VPN ou uma conexão do AWS Direct Connect para acessá-la de uma rede privada. Para ter mais informações, consulte Privacidade do tráfego entre redes.			

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS	Quando a alteração ocorre	Observações sobre tempo de inatividade

Tipo de armazenamento	O tipo de armazenamento do seu cluster de banco de dados.
	
	Somente os armazenamentos SSD de uso geral (gp3), IOPS provisionadas (io1) e SSD de IOPS provisionadas (io2) são compatíveis.
	
	Para ter mais informações, consulte Tipos de armazenamento do Amazon RDS.
		Opção da CLI:
		
		`--storage-type`
		
		Parâmetro da API do RDS:
		
		`StorageType`
			Se você optar por aplicar a alteração imediatamente, ela ocorrerá imediatamente.
			
			Se você não optar por aplicar a alteração imediatamente, ela ocorrerá durante a próxima janela de manutenção.
				Não ocorre uma inatividade durante esta alteração.

Grupo de segurança de VPC	O grupo de segurança a ser associado ao cluster de banco de dados.
	
	Para ter mais informações, consulte Visão geral dos grupos de segurança de VPC.
		Opção da CLI:
		
		`--vpc-security-group-ids`
		
		Parâmetro da API do RDS:
		
		`VpcSecurityGroupIds`
			A alteração é aplicada de forma assíncrona logo que possível. Essa configuração ignora a configuração para aplicar imediatamente.	Não ocorre uma inatividade durante esta alteração.

Configurações não aplicáveis ao modificar clusters de banco de dados multi-AZ

As seguintes configurações no comando da AWS CLI modify-db-cluster e na operação ModifyDBCluster da API do RDS não se aplicam a clusters de banco de dados multi-AZ.

Você também não pode modificar essas configurações para clusters de banco de dados multi-AZ no console.

Configuração da AWS CLI	Configuração da API do RDS
`--backtrack-window`	`BacktrackWindow`
`--cloudwatch-logs-export-configuration`	`CloudwatchLogsExportConfiguration`
`--copy-tags-to-snapshot | --no-copy-tags-to-snapshot`	`CopyTagsToSnapshot`
`--db-instance-parameter-group-name`	`DBInstanceParameterGroupName`
`--domain`	`Domain`
`--domain-iam-role-name`	`DomainIAMRoleName`
`--enable-global-write-forwarding | --no-enable-global-write-forwarding`	`EnableGlobalWriteForwarding`
`--enable-http-endpoint | --no-enable-http-endpoint`	`EnableHttpEndpoint`
`--option-group-name`	`OptionGroupName`
`--port`	`Port`
`--scaling-configuration`	`ScalingConfiguration`
`--storage-type`	`StorageType`

Atualização da versão de mecanismo de um cluster de banco de dados multi-AZ para o Amazon RDS

O Amazon RDS fornece versões mais recentes de cada mecanismo de banco de dados compatível, para que você possa manter o cluster de banco de dados multi-AZ atualizado. Este tópico explica o processo de atualização de um cluster de banco de dados multi-AZ para versões mais recentes.

A atualização de um cluster de banco de dados multi-AZ exige a seleção de uma nova versão de mecanismo compatível e o planejamento de um possível tempo de inatividade. O processo garante o mínimo de interrupção ao utilizar os recursos de failover da arquitetura multi-AZ. Durante a atualização, a instância primária é atualizada primeiro, seguida de um failover para a instância em espera para manter a disponibilidade. As práticas recomendadas incluem a realização de atualizações durante períodos de baixo tráfego, testes em ambientes que não sejam de produção e verificação da compatibilidade da aplicação com a nova versão.

Tópicos

- Atualizações de versões secundárias
- Atualizações da versão principal
- Atualização de um cluster de banco de dados multi-AZ
- Atualização de réplicas de leitura de cluster de banco de dados multi-AZ

Atualizações de versões secundárias

Uma atualização de versão secundária inclui somente alterações compatíveis com versões anteriores dos aplicativos existentes. Quando você inicia uma atualização de versão secundária, o Amazon RDS primeiro atualiza as instâncias de banco de dados de leitor, uma por vez. Depois, uma das instâncias de banco de dados de leitor passa a ser a nova instância de banco de dados de gravador. Depois, o Amazon RDS atualiza a antiga instância de gravador (que agora é uma instância de leitor).

O tempo de inatividade durante a atualização é limitado ao tempo necessário para que uma das instâncias de banco de dados de leitor se torne a nova instância de banco de dados de gravador. Esse tempo de inatividade funciona como um failover automático. Para ter mais informações, consulte the section called “Fazer o failover de um cluster de banco de dados multi-AZ”. Observe que o atraso da réplica do cluster de banco de dados multi-AZ pode afetar o tempo de inatividade. Para ter mais informações, consulte the section called “Atraso de réplica e clusters de banco de dados multi-AZ”.

Para réplicas de leitura de cluster de banco de dados multi-AZ do RDS para PostgreSQL, o Amazon RDS atualiza as instâncias membros do cluster uma por vez. Os perfis do cluster de leitor e gravador não mudam durante a atualização. Portanto, o cluster de banco de dados pode passar por tempo de inatividade enquanto o Amazon RDS atualiza a instância de gravador de cluster.

> **Note**
>
> O tempo de inatividade para uma atualização da versão secundária de cluster de banco de dados multi-AZ é geralmente de 35 segundos. Quando usado com o RDS Proxy, é possível reduzir ainda mais o tempo de inatividade para um segundo ou menos. Para ter mais informações, consulte Amazon RDS Proxy. Como alternativa, é possível usar um proxy de banco de dados de código aberto, como ProxySQL, PgBouncer ou driver avançado de encapsulamento JDBC da AWS.

Atualizações da versão principal

Um upgrade de versão principal pode apresentar alterações não compatíveis com aplicações existentes.

Ao iniciar uma atualização de versão principal de um cluster de banco de dados multi-AZ do RDS para PostgreSQL, o Amazon RDS atualiza simultaneamente as instâncias de leitor e gravador. Portanto, o cluster de banco de dados pode não estar disponível até que a atualização seja concluída.

Quando você inicia uma atualização de versão principal de um cluster de banco de dados multi-AZ do RDS para MySQL, o Amazon RDS atualiza as instâncias membros do cluster uma por vez, para que a replicação ocorra de uma versão de mecanismo anterior para uma posterior. É importante garantir que a workload seja compatível com as versões do mecanismo de origem e de destino durante uma atualização de versão principal, pois as versões do mecanismo podem diferir em sintaxe e recursos.

> **Note**
>
> Assim como nas atualizações de versão secundária, o tempo de inatividade para uma atualização de versão principal do RDS para MySQL é geralmente de 35 segundos. Quando usado com o RDS Proxy, é possível reduzir ainda mais o tempo de inatividade para um segundo ou menos. Para ter mais informações, consulte Amazon RDS Proxy.

Atualização de um cluster de banco de dados multi-AZ

O processo de atualização da versão do mecanismo de um cluster de banco de dados multi-AZ é o mesmo processo de atualização de uma versão do mecanismo de instância de banco de dados. Para obter instruções, consulte the section called “Atualizar a versão do mecanismo”. A única diferença é que, ao usar a AWS Command Line Interface (AWS CLI), você usa o comando modifique-db-cluster e especifica o parâmetro `--db-cluster-identifier` (bem como o parâmetro `--allow-major-version-upgrade`).

Para ter mais informações sobre atualizações de versões principais e secundárias, consulte a seguinte documentação do mecanismo de banco de dados:

- the section called “Atualizações do mecanismo de banco de dados do PostgreSQL”
- the section called “Atualizações do mecanismo de banco de dados do MySQL”

Atualização de réplicas de leitura de cluster de banco de dados multi-AZ

O Amazon RDS não atualiza automaticamente réplicas de leitura de clusters de banco de dados multi-AZ. Com relação às atualizações de versão secundária, primeiro é necessário atualizar manualmente todas as réplicas de leitura e, depois, atualizar o cluster. Caso contrário, a atualização será bloqueada. Quando você realiza uma atualização de versãoprincipal de um cluster, o estado da replicação de todas as réplicas de leitura muda para Encerrado. Você deve excluir e recriar as réplicas de leitura após a conclusão da atualização. Para ter mais informações, consulte the section called “Monitoramento da replicação de leitura”.

Renomear um cluster de banco de dados multi-AZ para o Amazon RDS

Você pode renomear um cluster de banco de dados multi-AZ utilizando o AWS Management Console, comando `modify-db-cluster` da AWS CLI ou a operação `ModifyDBCluster` da API do Amazon RDS. Renomear um cluster de banco de dados multi-AZ pode ter efeitos significativos. Veja a seguir uma lista de considerações antes de renomear um cluster de banco de dados multi-AZ.

- Quando você renomeia um cluster de banco de dados multi-AZ, os endpoints de cluster do cluster de banco de dados multi-AZ são alterados. Esses endpoints mudam porque incluem o nome que você atribuiu ao cluster de banco de dados multi-AZ. Você pode redirecionar o tráfego de um endpoint antigo para um novo. Para obter mais informações sobre endpoints de cluster de banco de dados multi-AZ, consulte Conectar-se a um cluster de banco de dados multi-AZ no Amazon RDS.
- Ao renomear um cluster de banco de dados multi-AZ, o nome DNS antigo que era usado pelo cluster é excluído, embora ele possa permanecer armazenado em cache por alguns minutos. O novo nome DNS do cluster de banco de dados multi-AZ renomeado entra em vigor em, aproximadamente, dois minutos. O cluster de banco de dados multi-AZ renomeado não fica disponível até que o novo nome entre em vigor.
- Você não pode usar um nome de cluster de banco de dados multi-AZ existente ao renomear um cluster.
- Métricas e eventos associados ao nome de um cluster de banco de dados multi-AZ serão mantidos se você reutilizar um nome de cluster de banco de dados.
- As tags de cluster de banco de dados multi-AZ permanecem no cluster de banco de dados multi-AZ, independentemente da renomeação.
- Snapshots de cluster de banco de dados são retidos para um cluster de banco de dados multi-AZ renomeado.

> **Note**
>
> Um cluster de banco de dados multi-AZ é um ambiente de banco de dados isolado em execução na nuvem. Um cluster de banco de dados multi-AZ pode hospedar vários bancos de dados. Para obter informações sobre como alterar um nome de banco de dados, consulte a documentação do seu mecanismo de banco de dados.

Renomear para substituir um cluster de banco de dados multi-AZ existente

As situações mais comuns para renomear um cluster de banco de dados multi-AZ incluem restaurar dados de um snapshot de cluster de banco de dados ou realizar uma recuperação a um ponto anterior no tempo (PITR). Ao renomear o cluster de banco de dados multi-AZ, você pode substituir o cluster de banco de dados multi-AZ sem alterar nenhum código de aplicação que faça referência ao cluster de banco de dados multi-AZ. Nesses casos, conclua as seguintes etapas:

1. Interrompa todo o tráfego para o cluster de banco de dados multi-AZ. É possível redirecionar o tráfego, impedindo-o de acessar os bancos de dados no cluster de banco de dados multi-AZ ou selecionar outra forma de impedir que o tráfego acesse seus bancos de dados no cluster de banco de dados multi-AZ.
2. Renomeia o cluster de banco de dados multi-AZ existente.
3. Crie um cluster de banco de dados multi-AZ restaurando-o de um snapshot de banco de dados ou recuperando-o para um momento específico. Depois, dê ao cluster de banco de dados multi-AZ o nome do cluster de banco de dados multi-AZ anterior.

Se você excluir o cluster de banco de dados multi-AZ antigo, será responsável por excluir qualquer snapshot de banco de dados indesejado do cluster de banco de dados multi-AZ antigo.

Console

Como renomear um cluster de banco de dados multi-AZ

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Bancos de dados.
3. Selecione o cluster de banco de dados multi-AZ que você deseja renomear.
4. Escolha Modificar.
5. Em Settings (Configurações), digite um novo nome para DB cluster identifier (Identificador do cluster de banco de dados).
6. Escolha Continue.
7. Para aplicar as alterações imediatamente, escolha Apply immediately. Escolher essa opção pode causar uma interrupção em alguns casos. Para ter mais informações, consulte Aplicar alterações imediatamente.

8. Na página de confirmação, revise suas alterações. Se estiverem corretas, escolha Modify cluster (Modificar cluster) para salvar as alterações.

   Como alternativa, selecione Back (Voltar) para editar suas alterações ou Cancel (Cancelar) para cancelar as alterações.

AWS CLI

Para renomear um cluster de banco de dados multi-AZ, use o comando modify-db-cluster da AWS CLI. Forneça ao valor `--db-cluster-identifier` atual e ao parâmetro `--new-db-cluster-identifier` o novo nome do cluster de banco de dados multi-AZ.

Example

Para Linux, macOS ou Unix:

```
aws rds modify-db-cluster \
    --db-cluster-identifier DBClusterIdentifier \
    --new-db-cluster-identifier NewDBClusterIdentifier
```

Para Windows:

```
aws rds modify-db-cluster ^
    --db-cluster-identifier DBClusterIdentifier ^
    --new-db-cluster-identifier NewDBClusterIdentifier
```

API do RDS

Para renomear um cluster de bancos de dados multi-AZ, chame a operação ModifyDBCluster da API do Amazon RDS com os seguintes parâmetros:

- `DBClusterIdentifier`: o nome existente do cluster de banco de dados.
- `NewDBClusterIdentifier`: o novo nome do cluster de banco de dados.

Reinicializar um cluster de banco de dados multi-AZ e instâncias de banco de dados de leitor do Amazon RDS

Talvez você precise reiniciar seu cluster de banco de dados Multi-AZ, geralmente por razões de manutenção. Por exemplo, se você fizer certas modificações ou alterar o grupo de parâmetros do cluster de banco de dados associado a um cluster de banco de dados, reinicialize o cluster de banco de dados. Isso faz com que as alterações entrem em vigor.

Se um cluster de banco de dados não estiver usando as alterações mais recentes no grupo de parâmetros de cluster de banco de dados associado, o AWS Management Console mostrará o grupo de parâmetros do cluster de banco de dados com um status pending-reboot. O status pending-reboot (reinicialização pendente) do grupo de parâmetros não resultará em uma reinicialização automática durante a próxima janela de manutenção. Para aplicar as alterações de parâmetro mais recentes a esse cluster de banco de dados, reinicie manualmente o cluster de banco de dados. Para ter mais informações sobre parameter groups, consulte Grupos de parâmetros para clusters de banco de dados multi-AZ.

A reinicialização de um cluster de banco de dados reinicia o serviço do mecanismo de banco de dados. A reinicialização de um cluster de banco de dados resulta em uma paralisação momentânea, durante a qual o status do cluster é definido como rebooting.

Você não poderá reiniciar seu cluster de banco de dados se ele não estiver no estado Available. Seu banco de dados pode ficar indisponível devido a vários motivos, como um backup em andamento, uma modificação solicitada anteriormente ou uma ação de janela de manutenção.

O tempo necessário para reinicializar o cluster de banco de dados depende do processo de recuperação de falhas, da atividade do banco de dados no momento da reinicialização e do comportamento do cluster de banco de dados específico. Para melhorar o tempo de reinicialização, recomendamos reduzir as atividades do banco de dados o máximo possível durante o processo de reinicialização. A redução das atividades do banco de dados reduz as atividades de reversão para transações em trânsito.

> **Important**
>
> Clusters de banco de dados Multi-AZ não oferecem suporte para reinicialização com um failover. Quando você reinicializa a instância de gravador de um cluster de banco de dados Multi-AZ, ela não afeta as instâncias de banco de dados de leitor nesse cluster de banco de dados e nenhum failover ocorre. Quando você reinicializa uma instância de banco de dados

de leitor, nenhum failover ocorre. Para fazer failover de um cluster de banco de dados Multi-AZ, escolha Failover no console, chame o comando da AWS CLI `failover-db-cluster` ou chame a operação de API `FailoverDBCluster`.

Console

Para reinicializar um cluster de banco de dados

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Databases (Bancos de dados) e depois o cluster de banco de dados Multi-AZ que você deseja reinicializar.
3. Em Actions (Ações), escolha Reboot (Reiniciar).

   A página Reboot DB cluster (Reinicializar cluster de banco de dados) é exibida.
4. Escolha Reboot (Reinicializar) para reinicializar seu cluster de banco de dados.

   Ou escolha Cancel (Cancelar).

AWS CLI

Para reinicializar um cluster de banco de dados multi-AZ utilizando a AWS CLI, chame o comando reboot-db-cluster.

```
aws rds reboot-db-cluster --db-cluster-identifier mymultiazdbcluster
```

API do RDS

Para reinicializar um cluster de banco de dados Multi-AZ usando a API do Amazon RDS, chame a operação RebootDBCluster.

Fazer failover de um cluster de banco de dados multi-AZ para o Amazon RDS

Se houver uma interrupção planejada ou não planejada da sua instância de banco de dados de gravador em um cluster de banco de dados multi-AZ, o Amazon RDS fará o failover automaticamente para uma instância de banco de dados de leitor em uma zona de disponibilidade diferente. Isso garante alta disponibilidade com o mínimo de interrupções. Os failovers podem ocorrer durante falhas de hardware, problemas de rede ou solicitações manuais. O tópico descreve a detecção automática de falhas, a sequência de eventos durante o failover e seu impacto nas operações de leitura e gravação. Ele também apresenta práticas recomendadas para monitorar e minimizar os tempos de failover.

O tempo de conclusão do failover depende da atividade do banco de dados e de outras condições no momento em que a instância de banco de dados de gravador se tornou indisponível. Em geral, os tempos de failover ficam abaixo de 35 segundos. O failover será concluído quando ambas as instâncias de banco de dados de leitor tiverem aplicado transações pendentes do gravador com falha. Quando o failover é concluído, o console do RDS pode levar mais um tempo para refletir a nova zona de disponibilidade.

Tópicos

- Failovers automáticos
- Fazer o failover manual de cluster de banco de dados multi-AZ
- Determinar se um cluster de banco de dados multi-AZ fez failover
- Definir o JVM TTL para pesquisas de nome DNS

Failovers automáticos

O Amazon RDS processa os failovers automaticamente para que você possa retomar as operações de banco de dados o mais rápido possível e sem intervenção administrativa. Para fazer failover, a instância de banco de dados de gravador alterna automaticamente para uma instância de banco de dados de leitor.

Fazer o failover manual de cluster de banco de dados multi-AZ

Se você fizer failover de um cluster de banco de dados multi-AZ manualmente, o RDS primeiro encerrará a instância de banco de dados primária. Depois, o sistema de monitoramento interno

detecta que a instância de banco de dados primária não está íntegra e promove uma instância de banco de dados de réplica legível. Em geral, os tempos de failover ficam abaixo de 35 segundos.

Você pode fazer failover de um cluster de banco de dados multi-AZ manualmente usando o AWS Management Console, a AWS CLI ou a API do RDS.

Console

Para fazer failover de um cluster de banco de dados multi-AZ manualmente

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Bancos de dados.
3. Escolha o cluster de banco de dados multi-AZ do qual você deseja fazer failover.
4. Em Actions (Ações), selecione Failover.

   A página Cluster de banco de dados de failover é exibida.
5. Selecione Failover para confirmar o failover manual.

AWS CLI

Para fazer failover de um cluster de banco de dados multi-AZ manualmente, utilize o comando da AWS CLI failover-db-cluster.

Example

```
aws rds failover-db-cluster --db-cluster-identifier mymultiazdbcluster
```

API do RDS

Para fazer failover de um cluster de banco de dados multi-AZ manualmente, chame a API FailoverDBCluster do Amazon RDS e especifique o `DBClusterIdentifier`.

Determinar se um cluster de banco de dados multi-AZ fez failover

Para determinar se ocorreu failover no cluster de banco de dados multi-AZ, faça o seguinte:

- Configure assinaturas de eventos de banco de dados para notificar você por e-mail ou SMS de que um failover foi iniciado. Para ter mais informações sobre eventos do , consulte Trabalhar com a notificação de eventos do Amazon RDS.

- Visualize seus eventos de banco de dados usando o console do Amazon RDS ou operações de API.
- Veja o estado atual do seu cluster de banco de dados multi-AZ utilizando o console do Amazon RDS, a AWS CLI e a API do RDS.

Para obter informações sobre como você pode responder aos failovers, reduzir o tempo de recuperação e outras melhores práticas para o Amazon RDS, consulte Práticas recomendadas do Amazon RDS.

Definir o JVM TTL para pesquisas de nome DNS

O mecanismo de failover modifica automaticamente o registro de Domain Name System (DNS) da instância de banco de dados para apontar para a instância de banco de dados de leitor. Como resultado, você precisará restabelecer todas as conexões existentes para sua instância de banco de dados. Em um ambiente de máquina virtual Java (JVM), devido à forma como o mecanismo de cache DNS do Java funciona, talvez seja necessário reconfigurar as configurações da JVM.

A JVM armazena em cache pesquisas de nome DNS. Ao resolver um nome de host para um endereço IP, a JVM armazena em cache o endereço IP por um período especificado, conhecido como Time-To-Live (TTL – Vida útil).

Como os recursos AWS usam entradas de nome DNS que acabam mudando, recomendamos configurar a JVM com um valor TTL de até 60 segundos. Isso garante que quando o endereço IP de um recurso mudar, seu aplicativo poderá receber e usar o novo endereço IP do recurso, consultando novamente o DNS.

Em algumas configurações do Java, o TTL padrão da JVM é definido de maneira que jamais atualiza entradas DNS até a JVM ser reiniciada. Por isso, se o endereço IP de um recurso AWS mudar enquanto a aplicação ainda estiver em execução, não será possível usar esse recurso até você reiniciar manualmente a JVM e as informações de IP armazenadas em cache serem atualizadas. Nesse caso, é crucial definir o TTL da JVM, de forma que ele atualize periodicamente as informações de IP armazenadas em cache.

> **Note**
>
> O TTL padrão pode variar de acordo com a versão da JVM e a possibilidade de um gerenciador de segurança estar instalado. Muitas JVMs oferecem um TTL padrão menor que 60 segundos. Se estiver usando uma JVM como essa, e não um gerenciador de segurança,

será possível ignorar o restante deste tópico. Para ter mais informações sobre gerenciadores de segurança no Oracle, consulte The security manager (O gerenciador de segurança) na documentação do Oracle.

Para modificar o TTL da JVM, defina o valor da propriedade `networkaddress.cache.ttl`. Use um dos seguintes métodos, dependendo das necessidades:

- Para definir o valor da propriedade globalmente para todos os aplicativos que usam a JVM, defina `networkaddress.cache.ttl` no arquivo `$JAVA_HOME/jre/lib/security/java.security`.

```
networkaddress.cache.ttl=60
```

- Para definir a propriedade localmente somente para seu aplicativo, defina `networkaddress.cache.ttl` no código de inicialização do aplicativo antes de quaisquer conexões de rede serem estabelecidas.

```
java.security.Security.setProperty("networkaddress.cache.ttl" , "60");
```

Configurar a replicação lógica do PostgreSQL com clusters de banco de dados multi-AZ para o Amazon RDS

Ao usar a replicação lógica do PostgreSQL com seu cluster de banco de dados multi-AZ, você pode replicar e sincronizar tabelas individuais em vez de toda a instância do banco de dados. A replicação lógica usa um modelo de publicação e de assinatura para replicar as alterações de uma fonte para um ou mais destinatários. Ela funciona usando registros de alterações do log de gravação antecipada (WAL) do PostgreSQL. Para ter mais informações, consulte the section called “Executar replicação lógica”.

Quando você cria um slot de replicação lógica na instância de banco de dados do gravador de um cluster de banco de dados multi-AZ, o slot é copiado de forma assíncrona para cada instância de banco de dados do leitor no cluster. Os slots nas instâncias de banco de dados do leitor são continuamente sincronizados com os da instância de banco de dados do gravador.

A replicação lógica é compatível com clusters de banco de dados multi-AZ que executam o RDS para PostgreSQL versão 14.8-R2 ou posterior e versão 15.3-R2 ou posterior.

> **Note**
>
> Além do atributo nativo de replicação lógica do PostgreSQL, os clusters de banco de dados multi-AZ que executam o RDS para PostgreSQL também são compatíveis com a extensão `pglogical`.

Para obter mais informações sobre a replicação lógica do PostgreSQL, consulte Logical replication na documentação do PostgreSQL.

Tópicos

- Pré-requisitos
- Configurar a replicação lógica
- Limitações e recomendações

Pré-requisitos

Para configurar a replicação lógica do PostgreSQL para clusters de banco de dados multi-AZ, você deve cumprir os pré-requisitos a seguir.

- Sua conta de usuário deve ser membro do grupo `rds_superuser` e ter privilégios de `rds_superuser`. Para ter mais informações, consulte the section called “Noções básicas de perfis e permissões do PostgreSQL”.
- Seu cluster de banco de dados multi-AZ deve estar associado a um grupo de parâmetros de cluster de banco de dados personalizado para que você possa configurar os valores dos parâmetros descritos no procedimento a seguir. Para ter mais informações, consulte the section called “Grupos de parâmetros de cluster de banco de dados”.

Configurar a replicação lógica

Para configurar a replicação lógica para um cluster de banco de dados multi-AZ, habilite parâmetros específicos dentro do grupo de parâmetros do cluster de banco de dados associado, depois crie slots de replicação lógica.

> **Note**
>
> A partir da versão 16 do PostgreSQL, é possível usar as instâncias de banco de dados de leitor do cluster de banco de dados multi-AZ para replicação lógica.

Como configurar a replicação lógica para um cluster de banco de dados multi-AZ do RDS para PostgreSQL

1. Abra o grupo de parâmetros de cluster de banco de dados personalizado associado ao seu cluster de banco de dados multi-AZ do RDS para PostgreSQL.
2. No campo de pesquisa Parâmetros, encontre o parâmetro estático `rds.logical_replication` e defina seu valor como 1. Essa alteração de parâmetro pode aumentar a geração de WAL, portanto habilite-o somente quando for usar slots lógicos.
3. Como parte dessa alteração, configure os seguintes parâmetros do cluster de banco de dados.

   - `max_wal_senders`
   - `max_replication_slots`
   - `max_connections`

   Dependendo do uso esperado, talvez você também precise alterar os valores dos parâmetros a seguir. No entanto, em muitos casos, os valores padrão são suficientes.

- `max_logical_replication_workers`
- `max_sync_workers_per_subscription`

4. Reinicialize o cluster de banco de dados multi-AZ para que os valores dos parâmetros entrem em vigor. Para obter instruções, consulte the section called “Reinicializar um cluster de banco de dados multi-AZ”.
5. Crie um slot de replicação lógica na instância de banco de dados do gravador do cluster de banco de dados multi-AZ, conforme explicado em the section called “Como trabalhar com slots de replicação lógica”. Esse processo requer que você especifique um plug-in de decodificação. No momento, o RDS para PostgreSQL é compatível com os plug-ins `test_decoding`, `wal2json` e `pgoutput` que acompanham o PostgreSQL.

   O slot é copiado de forma assíncrona para cada instância de banco de dados do leitor no cluster.
6. Verifique o estado do slot em todas as instâncias de banco de dados do leitor do cluster de banco de dados multi-AZ. Para fazer isso, inspecione a exibição de `pg_replication_slots` em todas as instâncias de banco de dados do leitor e verifique se o estado de `confirmed_flush_lsn` está progredindo enquanto a aplicação está consumindo ativamente as alterações lógicas.

   Os comandos a seguir demonstram como inspecionar o estado da replicação nas instâncias de banco de dados do leitor.

```
% psql -h test-postgres-instance-2.abcdefabcdef.us-west-2.rds.amazonaws.com

postgres=> select slot_name, slot_type, confirmed_flush_lsn from
 pg_replication_slots;
  slot_name   | slot_type | confirmed_flush_lsn
--------------+-----------+---------------------
 logical_slot | logical   | 32/D0001700
(1 row)

postgres=> select slot_name, slot_type, confirmed_flush_lsn from
 pg_replication_slots;
  slot_name   | slot_type | confirmed_flush_lsn
--------------+-----------+---------------------
 logical_slot | logical   | 32/D8003628
(1 row)
```

```
% psql -h test-postgres-instance-3.abcdefabcdef.us-west-2.rds.amazonaws.com

postgres=> select slot_name, slot_type, confirmed_flush_lsn from
 pg_replication_slots;
  slot_name   | slot_type | confirmed_flush_lsn
--------------+-----------+---------------------
 logical_slot | logical   | 32/D0001700
(1 row)

postgres=> select slot_name, slot_type, confirmed_flush_lsn from
 pg_replication_slots;
  slot_name   | slot_type | confirmed_flush_lsn
--------------+-----------+---------------------
 logical_slot | logical   | 32/D8003628
(1 row)
```

Depois de concluir as tarefas de replicação, interrompa o processo de replicação, elimine os slots de replicação e desative a replicação lógica. Para desativar a replicação lógica, modifique o grupo de parâmetros do cluster de banco de dados e defina o valor de `rds.logical_replication` de volta para `0`. Reinicialize o cluster para que a alteração do parâmetro entre em vigor.

Limitações e recomendações

As seguintes limitações e recomendações se aplicam ao uso da replicação lógica com clusters de banco de dados multi-AZ que executam o PostgreSQL versão 16:

- É possível usar somente instâncias de banco de dados de gravador para criar ou descartar slots de replicação lógica. Por exemplo, o comando `CREATE SUBSCRIPTION` deve usar o endpoint de gravador de cluster na string de conexão do host.
- É necessário usar o endpoint de gravador de cluster durante qualquer sincronização ou nova sincronização de tabelas. Por exemplo, é possível usar os seguintes comandos para sincronizar novamente uma tabela recém-adicionada:

```
Postgres=>ALTER SUBSCRIPTION subscription-name CONNECTION host=writer-endpoint
Postgres=>ALTER SUBSCRIPTION subscription-name REFRESH PUBLICATION
```

- É necessário aguardar a conclusão da sincronização da tabela antes de usar as instâncias de banco de dados de leitor para replicação lógica. É possível usar a tabela de catálogos do `pg_subscription_rel` para monitorar a sincronização das tabelas. A sincronização da tabela será concluída quando a coluna `srsubstate` for definida como pronta (`r`).

- Recomendamos usar endpoints de instância para conexão de replicação lógica quando a sincronização inicial da tabela for concluída. O comando a seguir reduz a carga na instância de banco de dados de gravador ao transferir a replicação para uma das instâncias de banco de dados de leitor:

  ```
  Postgres=>ALTER SUBSCRIPTION subscription-name CONNECTION host=reader-instance-endpoint
  ```

  Não é possível usar o mesmo slot em mais de uma instância de banco de dados ao mesmo tempo. Quando duas ou mais aplicações estão replicando alterações lógicas de instâncias de banco de dados diferentes no cluster, algumas alterações podem ser perdidas devido a um failover do cluster ou um problema de rede. Nessas situações, é possível usar endpoints de instância para replicação lógica na string de conexão do host. A outra aplicação usando a mesma configuração mostrará a seguinte mensagem de erro:

  ```
  replication slot slot_name is already active for PID x providing immediate feedback.
  ```

- Ao usar a extensão `pglogical`, você só pode usar o endpoint do gravador de cluster. A extensão tem limitações conhecidas que podem criar slots de replicação lógica não utilizados durante a sincronização da tabela. Os slots de replicação obsoletos reservam arquivos de log de gravação antecipada (WAL) e podem causar problemas de espaço em disco.

Trabalhar com réplicas de leitura de cluster de banco de dados multi-AZ para o Amazon RDS

Uma réplica de leitura de cluster de banco de dados é um tipo especial de cluster que você cria de uma instância de banco de dados de origem. Depois de criar uma réplica de leitura, todas as atualizações feitas na instância de banco de dados primária são copiadas de forma assíncrona na réplica de leitura do cluster de banco de dados multi-AZ. É possível reduzir a carga na instância de banco de dados primária roteando as consultas de leitura dos aplicativos para a réplica de leitura. Com o uso de réplicas de leitura, é possível expandir de maneira elástica para além das limitações de capacidade de uma única instância de banco de dados para workloads de banco de dados com muita leitura.

Também é possível criar uma ou mais réplicas de leitura de instância de banco de dados de um cluster de banco de dados multi-AZ. As réplicas de leitura de instâncias de banco de dados permitem que você escale além da capacidade de computação ou de E/S do cluster de banco de dados multi-AZ de origem, direcionando o excesso de tráfego de leitura para as réplicas de leitura. Atualmente, não é possível criar uma réplica de leitura de cluster de banco de dados multi-AZ por meio de um cluster de banco de dados multi-AZ existente.

Ao escolher entre migrar para um cluster de banco de dados multi-AZ utilizando uma réplica de leitura ou criar uma réplica de leitura de instância de banco de dados com base em um cluster de banco de dados multi-AZ, considere seu caso de uso e os requisitos de desempenho específicos.

Migrar para um cluster de banco de dados multi-AZ utilizando uma réplica de leitura

> Essa abordagem é ideal quando você precisa aprimorar a disponibilidade e a durabilidade do banco de dados e, ao mesmo tempo, minimizar o tempo de inatividade. Ao usar uma réplica de leitura na transição para um cluster de banco de dados multi-AZ, é possível garantir a operação contínua e a consistência de dados. Esse método é particularmente útil para ambientes de produção em que essencial manter a disponibilidade e reduzir o impacto nas workloads ativas.

Criar uma réplica de leitura de instância de banco de dados de um cluster de banco de dados multi-AZ

> Esse método é adequado quando você deseja escalar as operações de leitura ou descarregar o tráfego de leitura da instância do banco de dados primário. Ao criar uma réplica de leitura de um cluster de banco de dados multi-AZ existente, é possível distribuir workloads intensas em leitura e melhorar o desempenho sem afetar a estabilidade da instância primária.

Escolher a abordagem certa depende se sua prioridade é garantir alta disponibilidade e durabilidade ou escalar o desempenho de leitura. Avalie as características da workload e os requisitos operacionais para tomar uma decisão embasada.

Tópicos

- Migrar para um cluster de banco de dados multi-AZ utilizando uma réplica de leitura
- Criar uma réplica de leitura de instância de banco de dados de um cluster de banco de dados multi-AZ

Migrar para um cluster de banco de dados multi-AZ utilizando uma réplica de leitura

Para migrar uma implantação single-AZ ou uma implantação de instância de banco de dados multi-AZ para uma implantação de cluster de banco de dados multi-AZ com tempo de inatividade reduzido, você pode criar uma réplica de leitura de cluster de banco de dados multi-AZ. Para a origem, especifique a instância de banco de dados na implantação single-AZ ou a instância de banco de dados primária na implantação da instância de banco de dados multi-AZ. A instância de banco de dados pode processar transações de gravação durante a migração para um cluster de banco de dados multi-AZ.

Considere o seguinte antes de criar uma réplica de leitura do cluster de banco de dados multi-AZ:

- A instância de banco de dados de origem deve estar em uma versão que seja compatível com clusters de banco de dados multi-AZ. Para ter mais informações, consulte Regiões e mecanismos de banco de dados compatíveis com clusters de banco de dados multi-AZ no Amazon RDS.
- A réplica de leitura do cluster de banco de dados multi-AZ deve estar na mesma versão principal da origem e na mesma versão secundária ou superior.
- Você deve ativar backups automáticos na instância de banco de dados de origem, definindo o período de retenção de backup como um valor diferente de 0.
- O armazenamento alocado da instância de banco de dados de origem deve ser de 100 GiB ou mais.
- Para o RDS para MySQL, os parâmetros `gtid-mode` e `enforce_gtid_consistency` devem ser definidos como `ON` para a instância de banco de dados de origem. Você deve usar um grupo de parâmetros personalizado, não o grupo de parâmetros padrão. Para ter mais informações, consulte the section called “Grupos de parâmetros do banco de dados”.

- Uma transação ativa e de longa duração pode atrasar o processo de criação da réplica de leitura. Recomendamos que você aguarde até que as transações de longa duração sejam concluídas antes de criar uma réplica de leitura.
- Se você excluir a instância de banco de dados de origem para uma réplica de leitura de cluster de banco de dados multi-AZ, a réplica de leitura será promovida para um cluster de banco de dados multi-AZ autônomo.

Criar e promover a réplica de leitura do cluster de banco de dados multi-AZ

Você pode criar e promover um cluster de banco de dados multi-AZ usando o AWS Management Console, a AWS CLI ou a API do RDS.

> **Note**
>
> É altamente recomendável criar todas as réplicas de leitura na mesma nuvem privada virtual (VPC) com base na mesma Amazon VPC da instância de banco de dados de origem.
> Se você criar uma réplica de leitura em uma VPC diferente da instância de banco de dados de origem, os intervalos de Encaminhamento Entre Domínios Sem Classificação (CIDR) podem se sobrepor entre a réplica e o sistema do Amazon RDS. A sobreposição de CIDR torna a réplica instável, o que pode afetar negativamente as aplicações que se conectam a ela. Se você receber um erro ao criar a réplica de leitura, escolha um grupo de sub-redes de banco de dados de destino diferente. Para ter mais informações, consulte Trabalhar com uma instância de banco de dados em uma VPC.

Console

Para migrar uma implantação single-AZ ou uma implantação de instância de banco de dados multi-AZ para um cluster de banco de dados multi-AZ usando uma réplica de leitura, realize as etapas a seguir utilizando o AWS Management Console.

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. Crie a réplica de leitura do cluster de banco de dados multi-AZ.
    a. No painel de navegação, escolha Databases (Bancos de dados).
    b. Escolha a instância de banco de dados que você deseja usar como a origem da réplica de leitura.

c. Em Actions (Ações), escolha Create read replica (Criar réplica de leitura).

d. Em Availability and durability (Disponibilidade e durabilidade), escolha Multi-AZ DB cluster (Cluster de banco de dados multi-AZ).

e. Em DB instance identifier (Identificador da instância de banco de dados), insira um nome para a réplica de leitura.

f. Nas seções restantes, especifique suas configurações de cluster de banco de dados. Para obter informações sobre a configuração, consulte Configurações para criar clusters de banco de dados multi-AZ.

g. Escolha Create read replica (Criar réplica de leitura).

3. Quando estiver com tudo pronto, promova a réplica de leitura para que se torne um cluster de banco de dados multi-AZ autônomo:

a. Interrompa a gravação de todas as transações na instância de banco de dados de origem e aguarde até que todas as atualizações sejam feitas na réplica de leitura.

As atualizações do banco de dados serão feitas nas réplicas de leitura depois que forem feitas na instância de banco de dados primária. Esse atraso na replicação pode variar significativamente. Use a métrica `ReplicaLag` para determinar quando todas as atualizações foram feitas na réplica de leitura. Para ter mais informações sobre o atraso na replicação, consulte Monitoramento da replicação de leitura.

b. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.

c. No console do Amazon RDS, escolha Databases (Bancos de dados).

O painel Databases (Bancos de dados) é exibido. Todas as réplicas de leitura exibem Replica (Réplica) na coluna Role (Função).

d. Escolha a réplica de leitura do cluster de banco de dados multi-AZ que você deseja promover.

e. Em Actions (Ações), selecione Promote (Promover).

f. Na página Promote read replica (Promover réplica de leitura), insira o período de retenção de backup e a janela de backup para o cluster de banco de dados multi-AZ recém-promovido.

g. Quando estiver satisfeito com as configurações, escolha Promote read replica (Promover réplica de leitura).

h. Aguarde o status do cluster de banco de dados multi-AZ promovido ficar `Available`.

i. Direcione suas aplicações para que usem o cluster de banco de dados multi-AZ promovido.

Opcionalmente, exclua a implantação single-AZ ou a implantação da instância de banco de dados multi-AZ se não for mais necessária. Para obter instruções, consulte Excluir uma instância de banco de dados.

AWS CLI

Para migrar uma implantação single-AZ ou uma implantação de instância de banco de dados multi-AZ para um cluster de banco de dados multi-AZ usando uma réplica de leitura, realize as etapas a seguir utilizando o AWS CLI.

1. Crie a réplica de leitura do cluster de banco de dados multi-AZ.

   Para criar uma réplica de leitura da instância de banco de dados de origem, use o comando `create-db-cluster` da AWS CLI. Para `--replication-source-identifier`, especifique o nome do recurso da Amazon (ARN) da instância de banco de dados de origem.

   Para Linux, macOS ou Unix:

```
aws rds create-db-cluster \
  --db-cluster-identifier mymultiazdbcluster \
  --replication-source-identifier arn:aws:rds:us-
east-2:123456789012:db:mydbinstance
  --engine postgres \
  --db-cluster-instance-class db.m5d.large \
  --storage-type io1 \
  --iops 1000 \
  --db-subnet-group-name defaultvpc \
  --backup-retention-period 1
```

   Para Windows:

```
aws rds create-db-cluster ^
  --db-cluster-identifier mymultiazdbcluster ^
  --replication-source-identifier arn:aws:rds:us-
east-2:123456789012:db:mydbinstance
  --engine postgres ^
  --db-cluster-instance-class db.m5d.large ^
  --storage-type io1 ^
```

```
  --iops 1000 ^
  --db-subnet-group-name defaultvpc ^
  --backup-retention-period 1
```

2. Interrompa a gravação de todas as transações na instância de banco de dados de origem e aguarde até que todas as atualizações sejam feitas na réplica de leitura.

   As atualizações do banco de dados serão feitas nas réplicas de leitura depois que forem feitas na instância de banco de dados primária. Esse atraso na replicação pode variar significativamente. Use a métrica `Replica Lag` para determinar quando todas as atualizações foram feitas na réplica de leitura. Para ter mais informações sobre o atraso na replicação, consulte Monitoramento da replicação de leitura.

3. Quando estiver com tudo pronto, promova a réplica de leitura para que se torne um cluster de banco de dados multi-AZ autônomo.

   Para promover uma réplica de leitura de cluster de banco de dados multi-AZ, use o comando `promote-read-replica-db-cluster` da AWS CLI. Para `--db-cluster-identifier`, especifique o identificador da réplica de leitura do cluster de banco de dados multi-AZ.

   ```
   aws rds promote-read-replica-db-cluster --db-cluster-identifier mymultiazdbcluster
   ```

4. Aguarde o status do cluster de banco de dados multi-AZ promovido ficar `Available`.
5. Direcione suas aplicações para que usem o cluster de banco de dados multi-AZ promovido.

Opcionalmente, exclua a implantação single-AZ ou a implantação da instância de banco de dados multi-AZ se não for mais necessária. Para obter instruções, consulte Excluir uma instância de banco de dados.

API do RDS

Para migrar uma implantação single-AZ ou implantação de instância de banco de dados multi-AZ para um cluster de banco de dados multi-AZ usando uma réplica de leitura, realize as etapas a seguir utilizando a API do RDS.

1. Crie a réplica de leitura do cluster de banco de dados multi-AZ.

   Para criar uma réplica de leitura de cluster de banco de dados multi-AZ, use a operação `CreateDBCluster` com o parâmetro necessário `DBClusterIdentifier`. Para

`ReplicationSourceIdentifier`, especifique o nome do recurso da Amazon (ARN) da instância de banco de dados de origem.

2. Interrompa a gravação de todas as transações na instância de banco de dados de origem e aguarde até que todas as atualizações sejam feitas na réplica de leitura.

   As atualizações do banco de dados serão feitas nas réplicas de leitura depois que forem feitas na instância de banco de dados primária. Esse atraso na replicação pode variar significativamente. Use a métrica `Replica Lag` para determinar quando todas as atualizações foram feitas na réplica de leitura. Para ter mais informações sobre o atraso na replicação, consulte Monitoramento da replicação de leitura.

3. Quando estiver com tudo pronto, promova a réplica de leitura para que se torne um cluster de banco de dados multi-AZ autônomo.

   Para promover uma réplica de leitura de cluster de banco de dados multi-AZ, use a operação `PromoteReadReplicaDBCluster` com o parâmetro necessário `DBClusterIdentifier`. Especifique o identificador da réplica de leitura do cluster de banco de dados multi-AZ.

4. Aguarde o status do cluster de banco de dados multi-AZ promovido ficar `Available`.
5. Direcione suas aplicações para que usem o cluster de banco de dados multi-AZ promovido.

Opcionalmente, exclua a implantação single-AZ ou a implantação da instância de banco de dados multi-AZ se não for mais necessária. Para obter instruções, consulte Excluir uma instância de banco de dados.

Limitações para a criação de uma réplica de leitura de cluster de banco de dados multi-AZ

As limitações a seguir se aplicam à criação de uma réplica de leitura de cluster de banco de dados multi-AZ de uma implantação single-AZ ou uma implantação de instância de banco de dados multi-AZ.

- Não é possível criar uma réplica de leitura de cluster de banco de dados multi-AZ em uma Conta da AWS diferente da Conta da AWS que tem a instância de banco de dados de origem.
- Não é possível criar uma réplica de leitura de cluster de banco de dados multi-AZ em uma Região da AWS diferente da região da instância de banco de dados de origem.
- Não é possível recuperar uma réplica de leitura de cluster de banco de dados multi-AZ para um ponto anterior no tempo.
- A criptografia de armazenamento deve ter as mesmas configurações na instância de banco de dados de origem e no cluster de banco de dados multi-AZ.

- Se a instância de banco de dados de origem estiver criptografada, a réplica de leitura do cluster de banco de dados multi-AZ deverá ser criptografada usando a mesma chave do KMS.
- Se a instância de banco de dados de origem usar o armazenamento SSD de uso geral (gp3) e tiver menos de 400 GiB de armazenamento alocado, você não poderá modificar as IOPS provisionadas para a réplica de leitura do cluster de banco de dados multi-AZ.
- Para realizar uma atualização de versão secundária na instância de banco de dados de origem, primeiro você deve realizar a atualização da versão secundária na réplica de leitura do cluster de banco de dados multi-AZ.
- Ao realizar uma atualização de versão secundária em uma réplica de leitura do cluster de banco de dados multi-AZ do RDS para PostgreSQL, a instância de banco de dados de leitor não muda para a instância de banco de dados de gravador após a atualização. Portanto, o cluster de banco de dados pode passar por um tempo de inatividade enquanto o Amazon RDS atualiza a instância de gravador.
- Não é possível fazer upgrade da versão principal em uma réplica de leitura do cluster de banco de dados multi-AZ.
- Você pode realizar uma atualização da versão principal na instância de banco de dados de origem de uma réplica de leitura de cluster de banco de dados multi-AZ, mas a replicação para a réplica de leitura será interrompida e não poderá ser reiniciada.
- A réplica de leitura do cluster de banco de dados multi-AZ não comporta réplicas de leitura em cascata.
- Para o RDS para PostgreSQL, as réplicas de leitura de cluster de banco de dados multi-AZ não oferecem recurso de failover.

Criar uma réplica de leitura de instância de banco de dados de um cluster de banco de dados multi-AZ

Você pode criar uma réplica de leitura de instância de banco de dados por meio de um cluster de banco de dados multi-AZ para escalar além da capacidade computacional ou de E/S do cluster para workloads de banco de dados de uso intenso de leitura. Você pode direcionar esse tráfego de leitura excessivo a uma ou mais réplicas de leitura de instância de banco de dados. Também é possível usar réplicas de leitura para migrar de um cluster de banco de dados multi-AZ para uma instância de banco de dados.

Para criar uma réplica de leitura, especifique um cluster de banco de dados multi-AZ como origem de replicação. Uma das instâncias de leitor do cluster de banco de dados multi-AZ é sempre a origem

da replicação, não a instância do gravador. Essa condição garante que a réplica esteja sempre sincronizada com o cluster de origem, mesmo em casos de failover.

Tópicos

- Comparar instâncias de banco de dados de leitor e réplicas de leitura de instância de banco de dados
- Considerações
- Criar uma réplica de leitura de instância de banco de dados
- Promover a réplica de leitura de instância de banco de dados
- Limitações à criação de uma réplica de leitura de instância de banco de dados de um cluster de banco de dados multi-AZ

Comparar instâncias de banco de dados de leitor e réplicas de leitura de instância de banco de dados

Uma réplica de leitura de instância de banco de dados de um cluster de banco de dados multi-AZ é diferente das instâncias de banco de dados de leitor do cluster de banco de dados multi-AZ das seguintes formas:

- As instâncias de banco de dados do leitor atuam como destinos de failover automático, enquanto as réplicas de leitura de instância de banco de dados não.
- As instâncias de banco de dados do leitor devem reconhecer uma alteração da instância de banco de dados de gravador para poder confirmar a alteração. No entanto, para réplicas de leitura de instância de banco de dados, as atualizações são copiadas de forma assíncrona na réplica de leitura sem exigir confirmação.
- As instâncias de banco de dados do leitor sempre compartilham a mesma classe de instância, tipo de armazenamento e versão do mecanismo da instância de banco de dados do gravador do cluster de banco de dados multi-AZ. No entanto, as réplicas de leitura de instância de banco de dados não precisam necessariamente compartilhar as mesmas configurações do cluster de origem.
- Você pode promover uma réplica de leitura de banco de dados para uma instância de banco de dados autônoma. Não é possível promover uma instância de banco de dados do leitor de um cluster de banco de dados multi-AZ para uma instância autônoma.

- O endpoint do leitor somente encaminha as solicitações para as instâncias de banco de dados do leitor do cluster de banco de dados multi-AZ. Ele nunca encaminha solicitações para uma réplica de leitura de instância de banco de dados.

Para ter mais informações sobre instâncias de banco de dados do leitor e do gravador, consulte the section called “Arquitetura de cluster de banco de dados multi-AZ”.

Considerações

Considere o seguinte antes de criar uma réplica de leitura de instância de banco de dados de um cluster de banco de dados multi-AZ:

- Ao criar a réplica de leitura de instância de banco de dados, ela deve estar na mesma versão principal do cluster de origem e na mesma versão secundária ou superior. Depois de criá-la, você pode atualizar a réplica de leitura para uma versão secundária superior à do cluster de origem.
- Quando você cria a réplica de leitura da instância de banco de dados, o armazenamento alocado deve ser igual ao armazenamento alocado do cluster de banco de dados multi-AZ de origem. Você pode alterar o armazenamento alocado após a criação da réplica de leitura.
- Para o RDS para MySQL, o parâmetro `gtid-mode` deve ser definido como `ON` para o cluster de banco de dados multi-AZ de origem. Para ter mais informações, consulte the section called “Grupos de parâmetros de cluster de banco de dados”.
- Uma transação ativa e de longa duração pode atrasar o processo de criação da réplica de leitura. Recomendamos que você aguarde até que as transações de longa duração sejam concluídas antes de criar uma réplica de leitura.
- Se você excluir o cluster de banco de dados multi-AZ de uma réplica de leitura de instância de banco de dados, todas as réplicas de leitura nas quais ela estiver sendo gravada serão promovidas para instâncias de banco de dados autônomas.

Criar uma réplica de leitura de instância de banco de dados

Você pode criar uma réplica de leitura de instância de banco de dados de um cluster de banco de dados multi-AZ usando o AWS Management Console, a AWS CLI ou a API do RDS.

> **Note**
>
> É altamente recomendável criar todas as réplicas de leitura na mesma nuvem privada virtual (VPC) com base na mesma Amazon VPC do cluster de banco de dados multi-AZ de origem.

Se você criar uma réplica de leitura em uma VPC diferente do cluster de banco de dados multi-AZ de origem, os intervalos de Encaminhamento Entre Domínios Sem Classificação (CIDR) podem se sobrepor entre a réplica e o sistema do RDS. A sobreposição de CIDR torna a réplica instável, o que pode afetar negativamente as aplicações que se conectam a ela. Se você receber um erro ao criar a réplica de leitura, escolha um grupo de sub-redes de banco de dados de destino diferente. Para ter mais informações, consulte the section called “Trabalhar com uma instância de banco de dados em uma VPC”.

Console

Para criar uma réplica de leitura de instância de banco de dados de um cluster de banco de dados multi-AZ, execute as etapas a seguir usando o AWS Management Console.

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Bancos de dados.
3. Selecione o cluster de banco de dados multi-AZ que você deseja usar como a origem da réplica de leitura.
4. Em Actions (Ações), escolha Create read replica (Criar réplica de leitura).
5. Para Origem de réplica, o cluster de banco de dados multi-AZ correto deve estar selecionado.
6. Em Identificador de banco de dados, insira um nome para a réplica de leitura.
7. Nas seções restantes, especifique suas configurações de instância de banco de dados. Para obter informações sobre a configuração, consulte the section called “Configurações disponíveis”.

   Note

   O armazenamento alocado para a réplica de leitura de instância de banco de dados deve ser igual ao armazenamento alocado do cluster de banco de dados multi-AZ de origem.

8. Escolha Create read replica (Criar réplica de leitura).

AWS CLI

Para criar uma réplica de leitura de instância de banco de dados de um cluster de banco de dados multi-AZ, use o comando `create-db-instance-read-replica` da AWS CLI. Para `--source-db-cluster-identifier`, especifique o identificador do cluster de banco de dados multi-AZ.

Para Linux, macOS ou Unix:

```
aws rds create-db-instance-read-replica \
  --db-instance-identifier myreadreplica \
  --source-db-cluster-identifier mymultiazdbcluster
```

Para Windows:

```
aws rds create-db-instance-read-replica ^
  --db-instance-identifier myreadreplica ^
  --source-db-cluster-identifier mymultiazdbcluster
```

API do RDS

Para criar uma réplica de leitura de instância de banco de dados de um cluster de banco de dados multi-AZ, use a operação `CreateDBInstanceReadReplica`.

Promover a réplica de leitura de instância de banco de dados

Se você não precisar mais da réplica de leitura de instância de banco de dados, poderá promovê-la para uma instância de banco de dados autônoma. Ao promover uma réplica de leitura, a instância de banco de dados é reiniciada antes que ela fique disponível. Para obter instruções, consulte the section called “Promover uma réplica de leitura”.

Se você estiver usando a réplica de leitura para migrar uma implantação de cluster de banco de dados multi-AZ para uma implantação single-AZ ou implantação de instância de banco de dados multi-AZ, interrompa todas as transações que estejam sendo gravadas no cluster de banco de dados de origem. Depois, aguarde até que todas as atualizações sejam realizadas na réplica de leitura. As atualizações do banco de dados serão realizadas na réplica de leitura depois que forem feitas em uma das instâncias de banco de dados do leitor do cluster de banco de dados multi-AZ. Esse atraso na replicação pode variar significativamente. Use a métrica `ReplicaLag` para determinar quando todas as atualizações foram feitas na réplica de leitura. Para ter mais informações sobre o atraso na replicação, consulte the section called “Monitoramento da replicação de leitura”.

Depois de promover a réplica de leitura, aguarde o status da instância de banco de dados promovida ser `Available` antes de direcionar suas aplicações a usarem a instância de banco de dados promovida. Opcionalmente, exclua a implantação de cluster de banco de dados multi-AZ se não precisar mais dela. Para obter instruções, consulte the section called “Excluir um cluster de banco de dados multi-AZ”.

Limitações à criação de uma réplica de leitura de instância de banco de dados de um cluster de banco de dados multi-AZ

As limitações a seguir se aplicam à criação de uma réplica de leitura de instância de banco de dados de uma implantação de cluster de banco de dados multi-AZ.

- Não é possível criar uma réplica de leitura de instância de banco de dados em uma Conta da AWS diferente da Conta da AWS que tem o cluster de banco de dados multi-AZ de origem.
- Não é possível criar uma réplica de leitura de instância de banco de dados em uma Região da AWS diferente da do cluster de banco de dados multi-AZ de origem.
- Não é possível recuperar uma réplica de leitura de instância de banco de dados para um ponto no tempo.
- A criptografia de armazenamento deve ter as mesmas configurações no cluster de banco de dados multi-AZ de origem e na réplica de leitura de instância de dados.
- Se o cluster de banco de dados multi-AZ de origem estiver criptografado, a réplica de leitura de instância de banco de dados deverá ser criptografada usando a mesma chave do KMS.
- Para realizar uma atualização de versão secundária no cluster de banco de dados multi-AZ de origem, primeiro você deve realizar a atualização da versão secundária na réplica de leitura de instância de banco de dados.
- A réplica de leitura de instância de banco de dados não comporta réplicas de leitura em cascata.
- Para o RDS para PostgreSQL, o cluster de banco de dados multi-AZ de origem deve estar executando o PostgreSQL versão 13.11, 14.8 ou 15.2.R2 ou posterior para criar uma réplica de leitura da instância de banco de dados.
- Você pode realizar uma atualização da versão principal no cluster de banco de dados multi-AZ de origem de uma réplica de leitura da instância de banco de dados, mas a replicação para a réplica de leitura será interrompida e não poderá ser reiniciada.

Configurar a replicação externa de clusters de banco de dados multi-AZ para o Amazon RDS

É possível configurar a replicação entre um cluster de banco de dados multi-AZ e um banco de dados externo ao Amazon RDS.

A replicação externa permite que clusters de banco de dados multi-AZ repliquem dados entre uma instância de banco de dados do RDS e um banco de dados externo, seja no ambiente on-premises ou em outro ambiente de nuvem. Isso é benéfico para a recuperação de desastres, a migração de dados e a manutenção da consistência entre sistemas em locais diferentes. A seção aborda os pré-requisitos para configurar a replicação, instruções de configuração do processo e as principais considerações, como latência da replicação, largura de banda e compatibilidade com diferentes mecanismos de banco de dados.

RDS para MySQL

Para configurar a replicação externa para um cluster de banco de dados multi-AZ do RDS para MySQL, é necessário reter os arquivos de log binários nas instâncias de banco de dados dentro do cluster por tempo suficiente para garantir que as alterações sejam aplicadas à réplica antes que o Amazon RDS exclua o arquivo de log binário. Para fazer isso, configure a retenção de logs binários chamando o procedimento armazenado `mysql.rds_set_configuration` e especificando o parâmetro `binlog retention hours`. Para ter mais informações, consulte the section called “horas de retenção do log binário”.

O valor padrão para `binlog retention hours` é `NULL`, o que significa que os logs binários não são retidos (0 horas). Se quiser configurar a replicação externa para um cluster de banco de dados multi-AZ, defina o parâmetro como um valor diferente de `NULL`.

Só é possível configurar a retenção de logs binários da instância de banco de dados de gravador do cluster de banco de dados multi-AZ, e a configuração é propagada para todas as instâncias de banco de dados de leitor de forma assíncrona.

Além disso, é altamente recomendável habilitar a replicação baseada em GTID na réplica externa. Então, se uma das instâncias de banco de dados falhar, você poderá retomar a replicação de outra instância de banco de dados íntegra no cluster. Para ter mais informações, consulte Replication with Global Transaction Identifiers na documentação do MySQL.

RDS para PostgreSQL

Para configurar a replicação externa para um cluster de banco de dados multi-AZ do RDS para PostgreSQL, é necessário habilitar a replicação lógica. Para obter instruções, consulte the section called “Replicação lógica do PostgreSQL com clusters de banco de dados multi-AZ”.

Excluir um cluster de banco de dados multi-AZ para o Amazon RDS

Você pode excluir um cluster de banco de dados multi-AZ usando o AWS Management Console, a AWS CLI ou a API do RDS.

O tempo necessário para excluir um cluster de banco de dados multi-AZ pode variar dependendo dos seguintes fatores:

- O período de retenção de backup (ou seja, quantos backups serão excluídos).
- Quantos dados são excluídos.
- Se foi criado um snapshot final.

A proteção contra exclusão deve estar desabilitada no cluster de banco de dados multi-AZ para que você possa excluí-lo. Para ter mais informações, consulte the section called “Pré-requisitos para excluir uma instância de banco de dados”. É possível desabilitar a proteção contra exclusão modificando o cluster de banco de dados multi-AZ. Para ter mais informações, consulte the section called “Modificar um cluster de banco de dados multi-AZ”.

Console

Para excluir um cluster de banco de dados multi-AZ

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Bancos de dados e o cluster de banco de dados multi-AZ que você deseja excluir.
3. Em Ações, escolha Excluir.
4. Escolha Criar snapshot final? para criar um snapshot de banco de dados final para o cluster de banco de dados multi-AZ.

   Se você criar um snapshot final, insira um nome para Final snapshot name (Nome do snapshot final).
5. Escolha Retain automated backups (Reter backups automatizados) para reter backups automatizados.
6. Digite **`delete me`** na caixa.
7. Escolha Excluir.

AWS CLI

Para excluir um cluster de banco de dados multi-AZ usando a AWS CLI, chame o comando delete-db-cluster com as seguintes opções:

- `--db-cluster-identifier`
- `--final-db-snapshot-identifier` ou `--skip-final-snapshot`

Example Com um snapshot final

Para Linux, macOS ou Unix:

```
aws rds delete-db-cluster \
    --db-cluster-identifier mymultiazdbcluster \
    --final-db-snapshot-identifier mymultiazdbclusterfinalsnapshot
```

Para Windows:

```
aws rds delete-db-cluster ^
    --db-cluster-identifier mymultiazdbcluster ^
    --final-db-snapshot-identifier mymultiazdbclusterfinalsnapshot
```

Example Sem snapshot final

Para Linux, macOS ou Unix:

```
aws rds delete-db-cluster \
    --db-cluster-identifier mymultiazdbcluster \
    --skip-final-snapshot
```

Para Windows:

```
aws rds delete-db-cluster ^
    --db-cluster-identifier mymultiazdbcluster ^
    --skip-final-snapshot
```

API do RDS

Para excluir um cluster de banco de dados multi-AZ usando a API do Amazon RDS, chame a operação DeleteDBCluster com os seguintes parâmetros:

- `DBClusterIdentifier`
- `FinalDBSnapshotIdentifier` ou `SkipFinalSnapshot`

Limitações de clusters de banco de dados multi-AZ do Amazon RDS

Um cluster de banco de dados multi-AZ tem uma instância de banco de dados de gravador e duas instâncias de banco de dados de leitor em três zonas de disponibilidade separadas. Clusters de banco de dados multi-AZ oferecem alta disponibilidade, maior capacidade para workloads de leitura e menor latência quando comparados a implantações multi-AZ. Para obter mais informações sobre clusters de banco de dados multi-AZ, consulte Implantações de cluster de banco de dados multi-AZ para o Amazon RDS.

As seguintes limitações se aplicam aos clusters de banco de dados multi-AZ.

- Clusters de banco de dados multi-AZ não são compatíveis com os seguintes recursos:
  - Conexões IPv6 (modo de duas pilhas)
  - Backups automatizados entre regiões
  - Autenticação de Kerberos
  - Modifique a porta. Como alternativa, você pode restaurar um cluster de banco de dados multi-AZ para um ponto no tempo e especificar uma porta diferente.
  - Grupos de opções
  - Recuperação para um ponto no tempo (PITR) para clusters excluídos
  - Escalabilidade automática do armazenamento definindo o armazenamento máximo alocado. Como alternativa, é possível escalar o armazenamento manualmente.
  - Interromper e iniciar o cluster de banco de dados multi-AZ
  - Copiar um snapshot de um cluster de banco de dados multi-AZ
  - Criptografar um cluster de banco de dados multi-AZ não criptografado
- Os clusters de banco de dados do RDS para MySQL multi-AZ são compatíveis apenas com os seguintes procedimentos armazenados no sistema:
  - `mysql.rds_rotate_general_log`
  - `mysql.rds_rotate_slow_log`
  - `mysql.rds_show_configuration`
  - `mysql.rds_set_external_master_with_auto_position`
  - `mysql.rds_set_configuration`
- Os clusters de banco de dados multi-AZ do RDS para PostgreSQL não são compatíveis com as seguintes extensões: `aws_s3` e `pg_transport`.

- Os clusters de banco de dados multi-AZ do RDS para PostgreSQL não oferecem suporte ao uso de um servidor DNS personalizado para acesso à rede de saída.

Suporte estendido do Amazon RDS com Amazon RDS

Após a data de término do suporte padrão do RDS, o Suporte estendido do RDS permite que você continue executando um banco de dados em uma versão principal do mecanismo por um custo adicional.

Você só pode inscrever um banco de dados no Suporte estendido do RDS habilitando o Suporte estendido do RDS ao criar ou restaurar uma instância de banco de dados pela primeira vez. Não é possível atualizar o status de sua inscrição no Suporte estendido do RDS em instâncias de banco de dados existentes, a menos que as esteja restaurando.

Se você tiver habilitado o Suporte estendido do RDS durante a criação de uma instância de banco de dados, após a data de fim do suporte padrão do RDS, o Amazon RDS inscreverá automaticamente a instância de banco de dados no Suporte estendido do RDS. A inscrição automática no suporte estendido do RDS não altera o mecanismo do banco de dados e não afeta o tempo de atividade nem a performance da instância de banco de dados.

O Suporte estendido do RDS fornece as seguintes atualizações e suporte técnico:

- Atualizações de segurança para CVEs críticas e altas para a instância de banco de dados ou cluster de banco de dados, incluindo o mecanismo de banco de dados
- Correções de erros e patches para problemas críticos
- A capacidade de abrir casos de suporte e receber ajuda para solução de problemas dentro do acordo de serviço padrão do Amazon RDS

Essa oferta paga oferece a você mais tempo para atualizar para uma versão principal compatível do mecanismo. Por exemplo, a data de término do suporte padrão do RDS para MySQL versão 5.7 é 29 de fevereiro de 2024. No entanto, não é possível atualizar manualmente para o RDS para MySQL versão 8.0 antes dessa data. Nesse caso, o Amazon RDS inscreverá automaticamente os bancos de dados no Suporte estendido do RDS a partir de 29 de fevereiro de 2024. É possível continuar executando o RDS para MySQL versão 5.7. A partir de 1.º de março de 2024, o Amazon RDS cobrará você automaticamente pelo Suporte estendido do RDS.

O Suporte estendido do RDS ficará disponível por até três anos após a data de fim do suporte padrão do RDS para uma versão principal. Após esse período, se você não tiver feito upgrade para a versão principal do mecanismo para uma versão compatível, o Amazon RDS fará upgrade

automático da versão do mecanismo principal. Recomendamos que você atualize para uma versão principal compatível do mecanismo o mais rápido possível.

Para ter mais informações sobre as datas de fim do suporte padrão e do suporte estendido do RDS, consulte Versões principais do MySQL compatíveis com o Amazon RDS e Release calendar for Amazon RDS for PostgreSQL.

Tópicos

- Visão geral do Suporte estendido do Amazon RDS
- Cobranças do Suporte estendido do Amazon RDS
- Versões com Suporte estendido do Amazon RDS
- Responsabilidades do Amazon RDS e do cliente com o Suporte estendido do Amazon RDS
- Criação de uma instância de banco de dados ou um cluster de banco de dados multi-AZ com o Suporte estendido do Amazon RDS
- Visualizar a inscrição de instâncias de banco de dados ou clusters de banco de dados multi-AZ no Suporte estendido do Amazon RDS
- Visualizar datas de suporte de versões de mecanismo no Suporte estendido do Amazon RDS
- Restauração de uma instância de banco de dados ou um cluster de banco de dados multi-AZ com o Suporte estendido do Amazon RDS

Visão geral do Suporte estendido do Amazon RDS

Após a data de fim do suporte padrão do RDS, se você não tiver desabilitado o suporte estendido do RDS durante a criação ou a restauração das instâncias de banco de dados, o Amazon RDS as inscreverá automaticamente no suporte estendido do RDS. O Amazon RDS atualizará a instância de banco de dados para a versão secundária mais recente lançada antes da data de término do suporte padrão do RDS, caso você ainda não esteja executando essa versão. O Amazon RDS só atualizará a versão secundária depois da data de término do suporte padrão do RDS.

É possível criar bancos de dados com as principais versões do mecanismo que atingiram a data de término do suporte padrão do RDS. O RDS inscreve automaticamente esses novos bancos de dados no Suporte estendido do RDS e cobra por essa oferta.

Se você realizar a atualização para um mecanismo que ainda esteja sob o suporte padrão do RDS antes da data de término do suporte padrão do RDS, o Amazon RDS não inscreverá o mecanismo no Suporte estendido do RDS.

Se você tentar restaurar um snapshot de um banco de dados compatível com um mecanismo que já passou da data de término do suporte padrão do RDS, mas não está inscrito no Suporte estendido do RDS, o Amazon RDS tentará atualizar o snapshot para ser compatível com a versão mais recente do mecanismo que ainda está sob o suporte padrão do RDS. Se a restauração falhar, o Amazon RDS inscreverá automaticamente o mecanismo no Suporte estendido do RDS com uma versão compatível com o snapshot.

É possível encerrar a inscrição no Suporte estendido do RDS a qualquer momento. Para encerrar a inscrição, atualize cada mecanismo inscrito para uma versão mais nova do mecanismo que ainda esteja sob o suporte padrão do RDS. O término da inscrição no Suporte estendido do RDS entrará em vigor no dia em que você concluir a atualização para uma versão mais recente do mecanismo que ainda esteja sob o suporte padrão do RDS.

Para ter mais informações sobre as datas de fim do suporte padrão e do suporte estendido do RDS, consulte Versões principais do MySQL compatíveis com o Amazon RDS e Release calendar for Amazon RDS for PostgreSQL.

Cobranças do Suporte estendido do Amazon RDS

Você pagará por todos os mecanismos inscritos no Suporte estendido do RDS desde o dia seguinte à data de término do suporte padrão do RDS. Para saber a data de fim do suporte padrão do RDS, consulte Versões principais do MySQL compatíveis e Release calendar for Amazon RDS for PostgreSQL. As cobranças do Suporte estendido do RDS se aplicam às instâncias em espera em implantações multi-AZ.

A cobrança adicional pelo Suporte estendido do RDS é interrompida automaticamente ao realizar uma das seguintes ações:

- Atualizar para uma versão do mecanismo coberta pelo suporte padrão.
- Excluir o banco de dados que está executando uma versão principal após a data de fim do suporte padrão do RDS.

As cobranças serão reiniciadas se a versão do mecanismo de destino entrar no Suporte estendido do RDS no futuro.

Por exemplo, o RDS para PostgreSQL 11 entra no Suporte estendido em 1.º de março de 2024, mas as cobranças só começam a partir de 1.º de abril de 2024. Você atualiza o banco de dados

do RDS para PostgreSQL 11 para o RDS para PostgreSQL 12 em 30 de abril de 2024. Você só receberá cobrança por trinta dias de Suporte estendido no RDS para PostgreSQL 11. Você continua executando o RDS para PostgreSQL 12 nessa instância de banco de dados após a data de fim do suporte padrão do RDS, em 28 de fevereiro de 2025. O banco de dados vai gerar novamente cobranças do Suporte estendido do RDS desde 1.º de março de 2025.

Para obter mais informações, consulte Preço do Amazon RDS para MySQL e Preço do Amazon RDS para PostgreSQL.

Evitar cobranças do Suporte estendido do Amazon RDS

É possível evitar a cobrança do Suporte estendido do RDS impedindo o RDS de criar ou restaurar uma instância de banco de dados ou um cluster de banco de dados multi-AZ depois da data de término do suporte padrão do RDS. Para fazer isso, use a AWS CLI ou a API do RDS.

Na AWS CLI, especifique `open-source-rds-extended-support-disabled` para a opção `--engine-lifecycle-support`. Na API do RDS, especifique `open-source-rds-extended-support-disabled` para o parâmetro `LifeCycleSupport`. Para obter mais informações, consulte Criação de uma instância de banco de dados ou um cluster de banco de dados multi-AZ ou Restauração de uma instância de banco de dados ou um cluster de banco de dados multi-AZ.

Versões com Suporte estendido do Amazon RDS

O Suporte estendido do RDS só está disponível para versões principais. Ele não está disponível para versões secundárias.

O Suporte estendido do RDS está disponível para o RDS para MySQL e o RDS para PostgreSQL. Para ter mais informações, consulte Versões principais do MySQL compatíveis e Release calendar for Amazon RDS for PostgreSQL em Amazon RDS for PostgreSQL Release Notes.

Também é possível visualizar informações sobre datas de suporte para versões de mecanismo usando a AWS CLI ou a API do RDS. Para obter mais informações, consulte Visualizar datas de suporte de versões de mecanismo no Suporte estendido do Amazon RDS.

Nomenclatura da versão do Suporte estendido do Amazon RDS

O Amazon RDS lançará novas versões secundárias com correções e patches de CVE para mecanismos no Suporte estendido do RDS. Para ter mais informações, consulte Versões do Suporte

estendido do Amazon RDS para RDS para MySQL e Amazon RDS Extended Support updates for RDS for PostgreSQL em Amazon RDS for PostgreSQL Release Notes.

Os nomes dessas versões secundárias estarão no formato major.minor-RDS.YYYYMMDD.patch.YYYYMMDD, por exemplo, 5.7.44-RDS.20240208.R2.20240210 (para o RDS para MySQL) ou 11.22-RDS.20240208.R2.20240210 (para o RDS para PostgreSQL).

principal

Para o MySQL, o número da versão principal é o número inteiro e a primeira parte fracionária do número da versão, por exemplo, 8.0. Uma atualização de versão principal aumenta a parte principal do número da versão. Por exemplo, uma atualização de 5.7.44 para 8.0.33 é um upgrade de versão principal, em que 5.7 e 8.0 são os números da versão principal.

Para o PostgreSQL, o número da versão principal é o número inteiro, por exemplo, 11.

minor-RDS.YYYYMMDD

Para o MySQL, o número da versão secundária é a terceira parte do número da versão, por exemplo, 44-RDS.20240208 em 5.7.44-RDS.20240208.

Para o PostgreSQL, o número da versão secundária é a segunda parte do número da versão, por exemplo, 22-RDS.20240208 em 11.22-RDS.20240208.

A data é quando o Amazon RDS criou a versão secundária do Amazon RDS.

patch

A versão do patch é a que segue a data em que o Amazon RDS criou a versão secundária do Amazon RDS, por exemplo, o R2 em 5.7.44-RDS.20240208.R2 ou 11.22-RDS.20240208.R2.

Uma versão do patch do Amazon RDS inclui correções de bugs importantes adicionadas a uma versão secundária do Amazon RDS após o lançamento.

YYYYMMDD

A data é quando o Amazon RDS criou a versão do patch, por exemplo, 20240210 em 5.7.44-RDS.20240208.R2.20240210 ou 11.22-RDS.20240208.R2.20240210.

Uma versão datada do Amazon RDS é um patch de segurança que inclui correções de segurança importantes adicionadas a uma versão secundária após o lançamento. Ela não inclui nenhuma correção que possa mudar o comportamento de um mecanismo.

Responsabilidades do Amazon RDS e do cliente com o Suporte estendido do Amazon RDS

O conteúdo a seguir descreve as responsabilidades do Amazon RDS, bem como as suas, com o Suporte estendido do RDS.

Tópicos

- Responsabilidades do Amazon RDS
- Suas responsabilidades

Responsabilidades do Amazon RDS

Após a data de término do suporte padrão do RDS, o Amazon RDS fornecerá patches, correções de erros e atualizações para mecanismos inscritos no Suporte estendido do RDS. Isso ocorrerá por até três anos ou até que você pare de usar os mecanismos, o que ocorrer primeiro.

Os patches serão para CVEs graves e altos, conforme definido pelas classificações de gravidade CVSS do National Vulnerability Database (NVD). Para obter mais informações, consulte Vulnerability Metrics.

Suas responsabilidades

Você é responsável por aplicar os patches, correções de erros e atualizações fornecidos para instâncias de banco de dados ou clusters de banco de dados multi-AZ inscritos no Suporte estendido do RDS. O Amazon RDS reserva-se o direito de alterar, substituir ou retirar esses patches, correções de erros e atualizações a qualquer momento. Caso seja necessário um patch para resolver problemas críticos de segurança ou estabilidade, o Amazon RDS reserva-se o direito de atualizar suas instâncias de banco de dados ou clusters de banco de dados multi-AZ com o patch ou exigir que você instale o patch.

Você também é responsável por atualizar o mecanismo para uma versão mais nova antes da data de término do Suporte estendido do RDS. A data de término do Suporte estendido do RDS normalmente é três anos após a data de término do suporte padrão do RDS. Em relação à data de término do Suporte estendido do RDS para a versão principal do mecanismo do banco de dados, consulte Versões principais do MySQL compatíveis e Release calendar for Amazon RDS for PostgreSQL.

Se você não atualizar o mecanismo, após o fim do suporte estendido do RDS, o Amazon RDS tentará atualizá-lo para a versão mais recente aceita pelo suporte padrão do RDS. Caso a

atualização falhe, o Amazon RDS reserva-se o direito de excluir a instância de banco de dados ou o cluster de banco de dados multi-AZ que está executando o mecanismo após a data de término do suporte padrão do RDS. No entanto, antes de fazer isso, o Amazon RDS preservará os dados desse mecanismo.

Criação de uma instância de banco de dados ou um cluster de banco de dados multi-AZ com o Suporte estendido do Amazon RDS

Ao criar uma instância de banco de dados ou um cluster de banco de dados multi-AZ, selecione Habilitar o Suporte estendido do RDS no console, ou use a opção de Suporte estendido na AWS CLI ou o parâmetro na API do RDS. Ao inscrever uma instância de banco de dados ou um cluster de banco de dados multi-AZ no Suporte estendido do Amazon RDS, ele é inscrito permanentemente no Suporte estendido do RDS durante a vida da instância de banco de dados ou do cluster de banco de dados multi-AZ.

Se você usar o console, deverá selecionar Habilitar Suporte estendido do RDS. A configuração não está selecionada por padrão.

Se você usar a AWS CLI ou a API do RDS e não especificar a configuração do Suporte estendido do RDS, o Amazon RDS usará como padrão o Suporte estendido do RDS. Ao automatizar usando o AWS CloudFormation ou outros serviços, esse comportamento padrão mantém a disponibilidade do banco de dados após a data de fim do suporte padrão do RDS.

É possível impedir a inscrição no Suporte estendido do RDS usando a AWS CLI ou a API do RDS para criar uma instância de banco de dados ou um cluster de banco de dados multi-AZ.

Tópicos

- Comportamento do Suporte estendido do RDS
- Considerações para o Suporte estendido do RDS
- Criar uma instância de banco de dados ou um cluster de banco de dados multi-AZ com o Suporte estendido do RDS

Comportamento do Suporte estendido do RDS

A tabela a seguir resume o que acontece quando uma versão principal do mecanismo atinge o fim do suporte padrão do RDS .

Status do Suporte estendido do RDS*	Comportamento
Habilitada	O Amazon RDS cobra pelo Suporte estendido do RDS.
Desabilitado	O Amazon RDS atualiza a instância de banco de dados ou um cluster de banco de dados multi-AZ para uma versão do mecanismo compatível. Essa atualização ocorre na data ou logo após a data de fim do suporte padrão do RDS.

* No console do RDS, o status do Suporte estendido do RDS aparece como Sim ou Não. Na AWS CLI ou na API do RDS, o status do Suporte estendido do RDS é exibido como `open-source-rds-extended-support` ou `open-source-rds-extended-support-disabled`.

Considerações para o Suporte estendido do RDS

Antes de criar uma instância de banco de dados ou um cluster de banco de dados multi-AZ, considere os seguintes itens:

- Depois que a data de fim do suporte padrão do RDS tiver passado, você poderá impedir a criação de uma instância de banco de dados ou um cluster de banco de dados multi-AZ e evitar cobranças do Suporte estendido do RDS. Para fazer isso, use a AWS CLI ou a API do RDS. Na AWS CLI, especifique `open-source-rds-extended-support-disabled` para a opção `--engine-lifecycle-support`. Na API do RDS, especifique `open-source-rds-extended-support-disabled` para o parâmetro `LifeCycleSupport`. Se você especificar `open-source-rds-extended-support-disabled` e a data de fim do suporte padrão do RDS tiver passado, a criação de uma instância de banco de dados ou um cluster de banco de dados multi-AZ sempre falhará.
- O Suporte estendido do RDS é definido no nível do cluster. Os membros de um cluster sempre terão a mesma configuração para o Suporte estendido do RDS no console do RDS, em `--engine-lifecycle-support` na AWS CLI e em `EngineLifecycleSupport` na API do RDS.

Para ter mais informações, consulte Versões do MySQL e Release calendars for Amazon RDS for PostgreSQL.

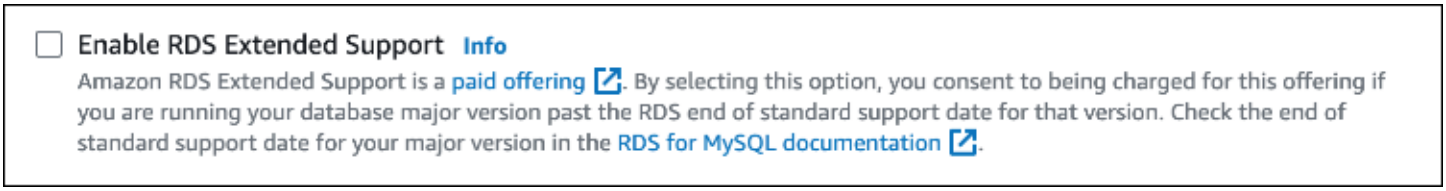
Criar uma instância de banco de dados ou um cluster de banco de dados multi-AZ com o Suporte estendido do RDS

É possível criar uma instância de banco de dados ou um cluster de banco de dados multi-AZ com uma versão do Suporte estendido do RDS usando o AWS Management Console, a AWS CLI ou a API do RDS.

Console

Ao criar uma instância de banco de dados ou um cluster de banco de dados multi-AZ, selecione Habilitar Suporte estendido do RDS na seção Opções do mecanismo. Por padrão, essa configuração não está selecionada.

A imagem a seguir mostra a configuração Habilitar Suporte estendido do RDS:



AWS CLI

Ao executar o comando create-db-instance ou create-db-cluster (cluster de banco de dados multi-AZ) da AWS CLI, selecione o Suporte estendido do RDS especificando `open-source-rds-extended-support` para a opção `--engine-lifecycle-support`. Essa opção é definida como `open-source-rds-extended-support` por padrão.

Para evitar a criação de uma instância de banco de dados ou um cluster de banco de dados multi-AZ após a data de fim do suporte padrão do RDS, especifique `open-source-rds-extended-support-disabled` para a opção `--engine-lifecycle-support`. Ao fazer isso, você evitará cobranças associadas ao Suporte estendido do RDS.

API do RDS

Ao usar a operação de API CreateDBInstance ou CreateDBCluster (cluster de banco de dados multi-AZ) do Amazon RDS, selecione o Suporte estendido do RDS definindo o parâmetro `EngineLifecycleSupport` como `open-source-rds-extended-support`. Esse parâmetro é definido como `open-source-rds-extended-support` por padrão.

Para evitar a criação de uma instância de banco de dados ou um cluster de banco de dados multi-AZ após a data de fim do suporte padrão do RDS, especifique `open-source-rds-extended-`

`support-disabled` para o parâmetro `EngineLifecycleSupport`. Ao fazer isso, você evitará cobranças associadas ao Suporte estendido do RDS.

Para obter mais informações, consulte os tópicos a seguir.

- Para criar uma instância de banco de dados, siga as instruções do seu mecanismo de banco de dados em Criar uma instância de banco de dados do Amazon RDS.
- Para criar um cluster de banco de dados multi-AZ, siga as instruções relacionadas ao seu mecanismo de banco de dados em Criar um cluster de banco de dados multi-AZ para o Amazon RDS.

Visualizar a inscrição de instâncias de banco de dados ou clusters de banco de dados multi-AZ no Suporte estendido do Amazon RDS

É possível visualizar a inscrição de instâncias de banco de dados ou clusters de banco de dados multi-AZ no Suporte estendido do RDS usando o AWS Management Console, a AWS CLI ou a API do RDS.

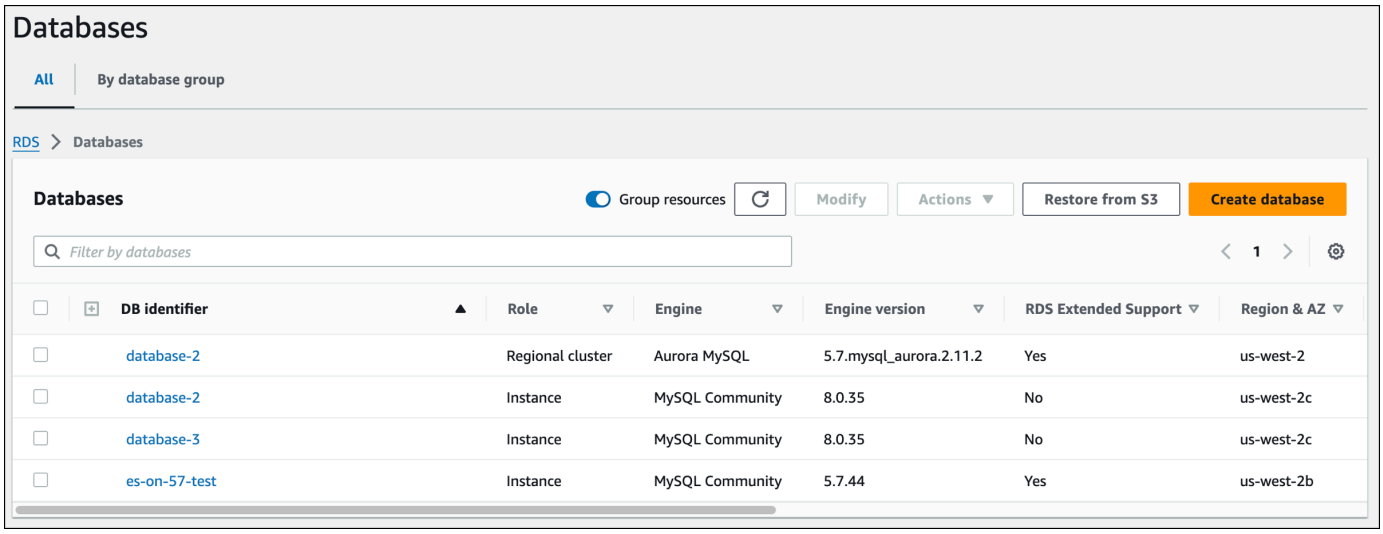
> **Note**
>
> A coluna Suporte estendido do RDS no console, a opção `--engine-lifecycle-support` na AWS CLI e o parâmetro `EngineLifecycleSupport` na API do RDS indicam somente a inscrição no Suporte estendido do RDS. As cobranças pelo Suporte estendido do RDS só começam quando sua versão do mecanismo de banco de dados atinge o final do suporte padrão do RDS. Para ter mais informações, consulte Versões principais do MySQL compatíveis e Release calendar for Amazon RDS for PostgreSQL em Amazon RDS for PostgreSQL Release Notes.
> Por exemplo, você tem um banco de dados do RDS para MySQL 5.7 que está inscrito no Suporte estendido do RDS. Em 1º de março de 2024, o Amazon RDS começou a cobrar pelo Suporte estendido do RDS para esse banco de dados. Em 31 de julho de 2024, você atualizou esse banco de dados para RDS para MySQL 8.0. O status do Suporte estendido do RDS para esse banco de dados permanece habilitado. No entanto, as cobranças do Suporte estendido do RDS para esse banco de dados foram interrompidas porque o RDS para MySQL 8.0 ainda não atingiu o fim do suporte padrão do RDS. O Amazon RDS não cobrará pelo Suporte estendido do RDS para esse banco de dados até 1º de agosto de 2026, quando o suporte padrão do RDS para o RDS para MySQL 8.0 termina.

Console

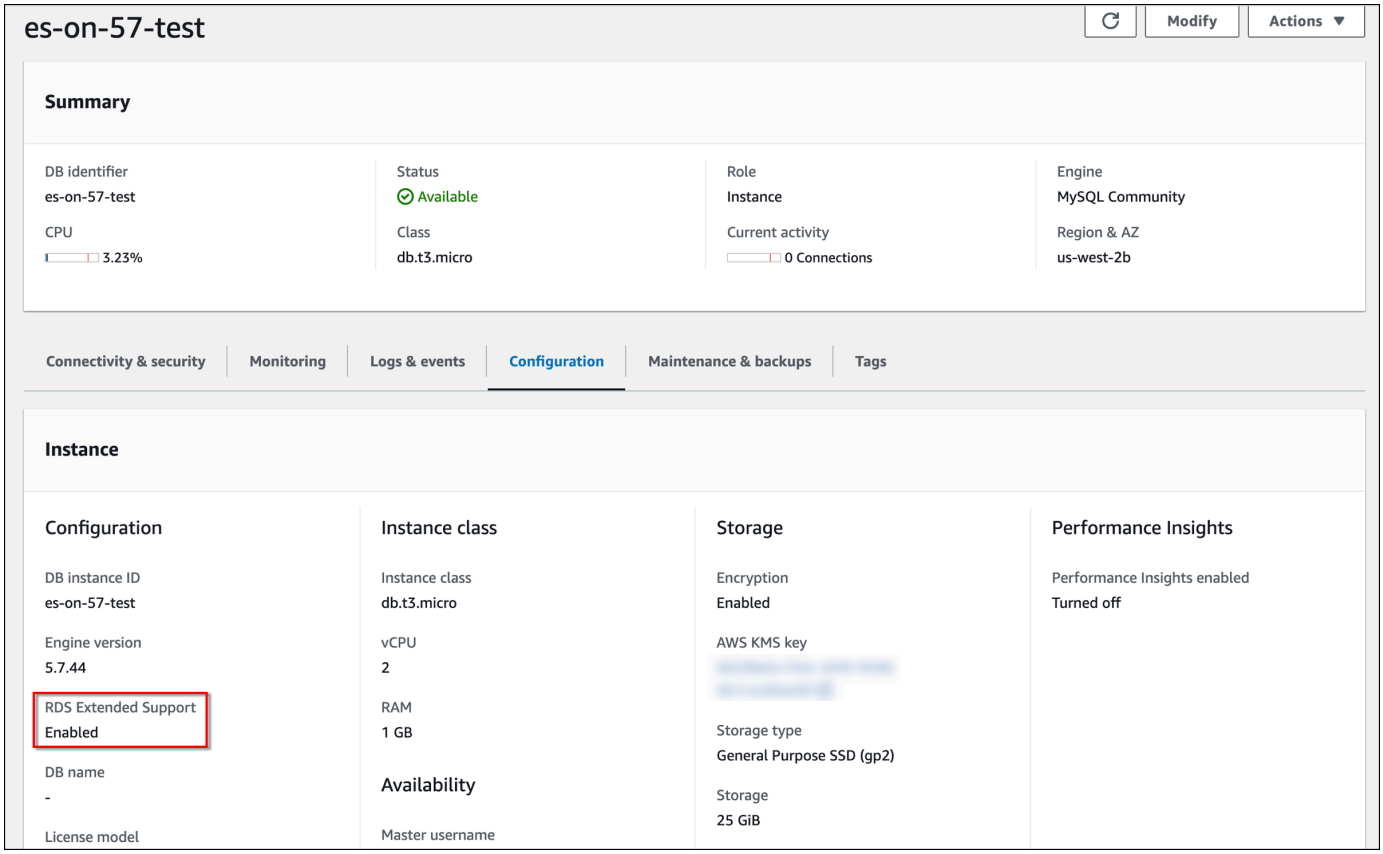
Para visualizar a inscrição de instâncias de banco de dados ou clusters de banco de dados multi-AZ no Suporte estendido do RDS

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Bancos de dados. O valor em Suporte estendido do RDS indica se foi feita a inscrição de uma instância de banco de dados ou um cluster de banco de dados multi-AZ no Suporte estendido do RDS. Se nenhum valor for exibido, o Suporte estendido do RDS não estará disponível para o banco de dados.

> **Tip**
>
> Se a coluna Suporte estendido do RDS não aparecer, selecione o ícone Preferências e ative Suporte estendido do RDS.



3. Você também pode visualizar a inscrição na guia Configuração de cada banco de dados. Escolha um banco de dados em Identificador de banco de dados. Na guia Configuração, confira em Suporte estendido se o banco de dados está inscrito ou não.



AWS CLI

Para visualizar a inscrição de seus bancos de dados no Suporte estendido do RDS usando a AWS CLI, execute o comando describe-db-instances ou describe-db-clusters (clusters de banco de dados multi-AZ).

Se o Suporte estendido do RDS estiver disponível para um banco de dados, a resposta incluirá o parâmetro `EngineLifecycleSupport`. O valor `open-source-rds-extended-support` indica que uma instância de banco de dados ou um cluster de banco de dados multi-AZ está inscrito no Suporte estendido do RDS. O valor `open-source-rds-extended-support-disabled` indica que a inscrição da instância de banco de dados ou do cluster de banco de dados multi-AZ no Suporte estendido do RDS foi desabilitada.

Exemplo

O comando a seguir retorna informações de todas as instâncias de banco de dados:

```
aws rds describe-db-instances
```

A resposta a seguir mostra que um mecanismo do PostgreSQL em execução na instância de banco de dados `database-1` está inscrito no Suporte estendido do RDS:

```
{
    "DBInstanceIdentifier": "database-1",
    "DBInstanceClass": "db.t3.large",
    "Engine": "postgres",
    ...
    "EngineLifecycleSupport": "open-source-rds-extended-support"
}
```

API do RDS

Para visualizar a inscrição dos bancos de dados no Suporte estendido do RDS utilizando a API do Amazon RDS, use a operação DescribeDBInstances ou DescribeDBClusters.

Se o Suporte estendido do RDS estiver disponível para um banco de dados, a resposta incluirá o parâmetro `EngineLifecycleSupport`. O valor `open-source-rds-extended-support` indica que uma instância de banco de dados ou um cluster de banco de dados multi-AZ está inscrito no Suporte estendido do RDS. O valor `open-source-rds-extended-support-disabled` indica que a inscrição da instância de banco de dados ou do cluster de banco de dados multi-AZ no Suporte estendido do RDS foi desabilitada.

Visualizar datas de suporte de versões de mecanismo no Suporte estendido do Amazon RDS

É possível visualizar informações sobre datas de suporte para versões de mecanismo de instâncias de banco de dados ou clusters de banco de dados multi-AZ no Suporte estendido do Amazon RDS usando a AWS CLI ou a API do RDS. Essas informações podem ajudar você a planejar as atualizações.

Os comandos da AWS CLI e as operações de API do RDS retornam as datas de início e término do suporte padrão do RDS e do Suporte estendido do RDS. Essas datas também podem ser encontradas nas tabelas de versões principais do mecanismo. Para ter mais informações, consulte Versões principais do MySQL compatíveis com o Amazon RDS e Release calendar for Amazon RDS for PostgreSQL em Amazon RDS for PostgreSQL Release Notes.

AWS CLI

Para visualizar as datas de início e término do suporte padrão do RDS e do Suporte estendido do RDS para versões principais do mecanismo usando a AWS CLI, execute o comando describe-db-major-engine-versions.

Esse comando retorna os parâmetros relevantes a seguir para mecanismos de código aberto (MariaDB, MySQL e PostgreSQL). Ele não retorna esses parâmetros para mecanismos comerciais (Db2, SQL Server e Oracle).

- `SupportedEngineLifecycles`: este parâmetro é uma matriz de objetos que inclui `LifecycleSupportName`, `LifecycleSupportStartDate` e `LifecycleSupportEndDate`.
- `LifecycleSupportName` – Esse parâmetro indica o tipo de suporte em que a versão do mecanismo está: suporte padrão do RDS (`open-source-rds-standard-support`) ou RDS Extended Support (`open-source-rds-extended-support`). Para o MariaDB, este parâmetro retorna apenas o suporte padrão do RDS (`open-source-rds-standard-support`).
- `LifecycleSupportStartDate`: este parâmetro lista a data de início do suporte padrão do RDS ou do Suporte estendido do RDS para a versão principal do mecanismo, dependendo do valor de `LifecycleSupportName`.
- `LifecycleSupportEndDate`: este parâmetro lista a data de término do suporte padrão do RDS ou do Suporte estendido do RDS para a versão principal do mecanismo, dependendo do valor de `LifecycleSupportName`.

Exemplo

O exemplo de resposta mostra as seguintes informações:

- As datas de início e término dos ciclos de vida `open-source-rds-standard-support` e `open-source-rds-extended-support` do mecanismo compatível para MySQL 5.7. O Suporte estendido do RDS está disponível para o MySQL 5.7.
- As datas de início e término do ciclo de vida `open-source-rds-standard-support` do mecanismo compatível para MariaDB 10.6. O Suporte estendido do RDS não está disponível para o MariaDB 10.6.
- Não há informações sobre os ciclos de vida compatíveis do mecanismo para o SQL Server Enterprise Edition 13 porque o SQL Server é um mecanismo comercial.

```
{
    "DBMajorEngineVersions": [
        {
            "Engine": "mysql",
            "MajorEngineVersion": "5.7",
            "SupportedEngineLifecycles": [
                {
                    "LifecycleSupportName": "open-source-rds-standard-support",
                    "LifecycleSupportStartDate": "2016-02-22T00:00:00+00:00",
                    "LifecycleSupportEndDate": "2024-02-29T23:59:59.999000+00:00"
                },
                {
                    "LifecycleSupportName": "open-source-rds-extended-support",
                    "LifecycleSupportStartDate": "2024-03-01T00:00:00+00:00",
                    "LifecycleSupportEndDate": "2027-02-28T23:59:59.999000+00:00"
                }
            ]
        },
        {
            "Engine": "mariadb",
            "MajorEngineVersion": "10.6",
            "SupportedEngineLifecycles": [
                {
                    "LifecycleSupportName": "open-source-rds-standard-support",
                    "LifecycleSupportStartDate": "2022-02-03T00:00:00+00:00",
                    "LifecycleSupportEndDate": "2026-07-31T23:59:59.999000+00:00"
                }
            ]
        },
        {
            "Engine": "sqlserver-ee",
            "MajorEngineVersion": "13.00"
        },
        ...
    ]
}
```

API do RDS

Para ver as datas de início e término do suporte padrão do RDS e do Suporte estendido do RDS para as versões principais do mecanismo usando a API do RDS, use a operação DescribeDBMajorEngineVersions.

Se o Suporte estendido do RDS estiver disponível para uma versão do mecanismo, a resposta incluirá o parâmetro `SupportedEngineLifeCycles` como uma matriz com dois objetos. O primeiro objeto inclui as datas de início e término do suporte padrão do RDS. O segundo inclui as datas de início e término do Suporte estendido do RDS.

Se o Suporte estendido do RDS não estiver disponível para uma versão de mecanismo de código aberto (MariaDB, MySQL e PostgreSQL), a resposta incluirá somente o parâmetro `SupportedEngineLifeCycles` como uma matriz com um único objeto. Esse objeto inclui as datas de início e término do suporte padrão do RDS.

Se a versão do mecanismo for para um mecanismo comercial (Db2, SQL Server e Oracle), a resposta não incluirá o parâmetro `SupportedEngineLifeCycles`.

Restauração de uma instância de banco de dados ou um cluster de banco de dados multi-AZ com o Suporte estendido do Amazon RDS

Ao restaurar uma instância de banco de dados ou um cluster de banco de dados multi-AZ, selecione Habilitar o Suporte estendido do RDS no console, ou use a opção de Suporte estendido na AWS CLI ou o parâmetro na API do RDS. Ao inscrever uma instância de banco de dados ou um cluster de banco de dados multi-AZ no Suporte estendido do RDS, ele é inscrito permanentemente no Suporte estendido do RDS durante a vida da instância de banco de dados ou do cluster de banco de dados multi-AZ.

O padrão para a configuração do Suporte estendido do RDS depende de você usar o console, a AWS CLI ou API do RDS para restaurar o banco de dados. Se você usa o console, não seleciona Habilitar Suporte estendido do RDS, e a versão principal do mecanismo que você está restaurando já passou do fim do suporte padrão do RDS . Então, o Amazon RDS atualiza automaticamente a instância de banco de dados para uma versão mais recente do mecanismo. Se você usar a AWS CLI ou a API do RDS e não especificar a configuração do Suporte estendido do RDS, o Amazon RDS usará como padrão o Suporte estendido do RDS. Ao automatizar usando o AWS CloudFormation ou outros serviços, esse comportamento padrão mantém a disponibilidade do banco de dados após a data de fim do suporte padrão do RDS. É possível desabilitar o Suporte estendido do RDS usando a AWS CLI ou a API do RDS.

Tópicos

- Comportamento do Suporte estendido do RDS

- Considerações para o Suporte estendido do RDS
- Restaurar uma instância de banco de dados ou um cluster de banco de dados multi-AZ com o Suporte estendido do RDS

Comportamento do Suporte estendido do RDS

A tabela a seguir resume o que acontece quando uma versão principal do mecanismo de uma instância de banco de dados ou um cluster de banco de dados multi-AZ que você está restaurando atinge o fim do suporte padrão do RDS.

Status do Suporte estendido do RDS*	Comportamento
Habilitada	O Amazon RDS cobra pelo Suporte estendido do RDS.
Desabilitado**	Após a conclusão da restauração, o Amazon RDS atualiza automatic amente a instância de banco de dados ou o cluster de banco de dados multi-AZ para uma versão mais recente do mecanismo (em uma futura janela de manutenção).

* No console do RDS, o status do Suporte estendido do RDS aparece como Sim ou Não. Na AWS CLI ou na API do RDS, o status do Suporte estendido do RDS é exibido como `open-source-rds-extended-support` ou `open-source-rds-extended-support-disabled`.

** Essa opção só está disponível ao restaurar uma instância de banco de dados ou um cluster de banco de dados multi-AZ executando o PostgreSQL 12 e posterior ou o MySQL 8 e posterior.

Considerações para o Suporte estendido do RDS

Antes de restaurar uma instância de banco de dados ou um cluster de banco de dados multi-AZ, considere os seguintes itens:

- Depois que a data de término do suporte padrão do RDS tiver passado, se quiser restaurar uma instância de banco de dados ou um cluster de banco de dados multi-AZ pelo Amazon S3, você só poderá fazer isso usando a AWS CLI ou a API do RDS. Use a opção `--engine-lifecycle-support` no comando restore-db-cluster-from-s3 da AWS CLI ou o parâmetro `EngineLifecycleSupport` na operação de API RestoreDBClusterFromS3 do RDS.

- Se você quiser impedir que o RDS restaure os bancos de dados para as versões do Suporte estendido do RDS, especifique `open-source-rds-extended-support-disabled` na AWS CLI ou na API do RDS. Ao fazer isso, você evitará cobranças associadas ao Suporte estendido do RDS.

  Se você especificar essa configuração, o Amazon RDS atualizará automaticamente o banco de dados restaurado para uma versão principal compatível mais recente. Se o upgrade falhar nas verificações pré-upgrade, o Amazon RDS voltará com segurança para a versão do mecanismo do Suporte estendido do RDS. Esse banco de dados permanecerá no modo de Suporte estendido do RDS e o Amazon RDS cobrará pelo Suporte estendido do RDS até que você faça upgrade manual do banco de dados.

  Por exemplo, se você restaurar um snapshot do MySQL 5.7 sem usar o Suporte estendido do RDS, o Amazon RDS tentará fazer upgrade automático do banco de dados para o MySQL 8.0. Se essa atualização falhar devido a um problema que você precisa resolver, o Amazon RDS reverterá o banco de dados para o MySQL 5.7. O Amazon RDS manterá o banco de dados no Suporte estendido do RDS até que você possa corrigir o problema. Por exemplo, uma atualização pode falhar devido ao espaço de armazenamento insuficiente. Depois de corrigir o problema, você deve iniciar o upgrade. Após a primeira tentativa de upgrade do banco de dados, o Amazon RDS não tentará fazer upgrade novamente.
- O Suporte estendido do RDS é definido no nível do cluster. Os membros de um cluster sempre terão a mesma configuração para o Suporte estendido do RDS no console do RDS, em `--engine-lifecycle-support` na AWS CLI e em `EngineLifecycleSupport` na API do RDS.

Para ter mais informações, consulte Versões do MySQL e Release calendars for Amazon RDS for PostgreSQL.

Restaurar uma instância de banco de dados ou um cluster de banco de dados multi-AZ com o Suporte estendido do RDS

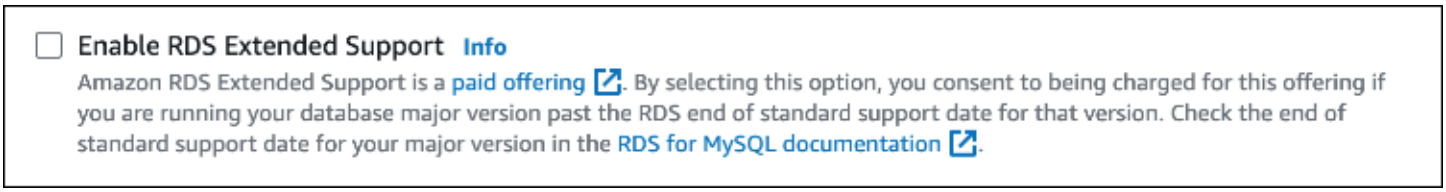
É possível restaurar uma instância de banco de dados ou um cluster de banco de dados multi-AZ com uma versão do Suporte estendido do RDS usando o AWS Management Console, a AWS CLI ou a API do RDS.

Console

Ao restaurar uma instância de banco de dados ou um cluster de banco de dados multi-AZ, selecione Habilitar Suporte estendido do RDS na seção Opções do mecanismo. Se você não selecionar essa

configuração e a versão principal do mecanismo que você está restaurando tiver passado do fim do suporte padrão do RDS, o Amazon RDS atualizará automaticamente a instância de banco de dados ou o cluster de banco de dados multi-AZ para uma versão sob o suporte padrão do RDS.

A imagem a seguir mostra a configuração Habilitar Suporte estendido do RDS:



AWS CLI

Ao executar o comando restore-db-instance-from-db-snapshot ou restore-db-cluster-from-snapshot da AWS CLI, selecione o Suporte estendido do RDS especificando `open-source-rds-extended-support` para a opção `--engine-lifecycle-support`.

Se você quiser evitar cobranças associadas ao Suporte estendido do RDS, defina a opção `--engine-lifecycle-support` como `open-source-rds-extended-support-disabled`. Essa opção é definida como `open-source-rds-extended-support` por padrão.

Também é possível especificar esse valor usando os seguintes comandos da AWS CLI:

- restore-db-cluster-from-s3
- restore-db-cluster-to-point-in-time
- restore-db-instance-from-s3
- restore-db-instance-to-point-in-time

API do RDS

Ao usar a operação de API RestoreDBInstanceFromDBSnapshot ou RestoreDBClusterFromSnapshot do Amazon RDS, selecione o Suporte estendido do RDS definindo o parâmetro `EngineLifecycleSupport` como `open-source-rds-extended-support`.

Se você quiser evitar cobranças associadas ao Suporte estendido do RDS, defina o parâmetro `EngineLifecycleSupport` como `open-source-rds-extended-support-disabled`. Esse parâmetro é definido como `open-source-rds-extended-support` por padrão.

Também é possível especificar esse valor usando as seguintes operações de API do RDS:

- RestoreDBClusterFromS3
- RestoreDBClusterToPointInTime
- RestoreDBInstanceFromS3
- RestoreDBInstanceToPointInTime

Consulte mais informações sobre a restauração de uma instância de banco de dados ou um cluster de banco de dados multi-AZ e siga as instruções do mecanismo de banco de dados em Restaurar uma instância de banco de dados.

Usar implantações azul/verde do Amazon RDS para atualizações de banco de dados

Uma implantação azul/verde copia um ambiente de banco de dados de produção em um ambiente de teste separado e sincronizado. Usando as implantações azul/verde do Amazon RDS, é possível fazer alterações no banco de dados no ambiente de teste sem prejudicar o ambiente de produção. Por exemplo, você pode atualizar a versão principal ou secundária do mecanismo de banco de dados ou alterar parâmetros do banco de dados no ambiente de preparação. Quando estiver tudo pronto, você poderá promover o ambiente de teste como o novo ambiente de banco de dados de produção, com tempo de inatividade normalmente inferior a um minuto.

> **Note**
>
> No momento, as implantações azul/verde são compatíveis apenas com o RDS para MariaDB, RDS para MySQL e RDS para PostgreSQL. Para ter informações sobre a disponibilidade do Amazon Aurora, consulte Usar implantações azul/verde do Amazon Aurora para atualizações de banco de dados no Guia do usuário do Amazon Aurora.

Tópicos

- Visão geral das implantações azul/verde do Amazon RDS
- Criar uma implantação azul/verde no Amazon RDS
- Visualizar uma implantação azul/verde no Amazon RDS
- Mudar uma implantação azul/verde no Amazon RDS
- Excluir uma implantação azul/verde no Amazon RDS

Visão geral das implantações azul/verde do Amazon RDS

Ao usar implantações azul/verde do Amazon RDS, você pode fazer e testar alterações no banco de dados antes de implementá-las em um ambiente de produção. Uma implantação azul/verde cria um ambiente de teste que copia o ambiente de produção. Em uma implantação azul/verde, o ambiente azul é o ambiente de produção atual. O ambiente verde é o ambiente de teste. O ambiente de teste permanece sincronizado com o ambiente de produção atual usando replicação lógica.

Você pode fazer alterações nas instâncias de banco de dados do RDS no ambiente verde sem afetar as workloads de produção. Por exemplo, você pode atualizar a versão principal ou secundária do mecanismo de banco de dados, atualizar a configuração do sistema de arquivos subjacente ou alterar os parâmetros do banco de dados no ambiente de preparação. Você pode testar minuciosamente as alterações no ambiente verde. Quando estiver pronto, você pode fazer a transição dos ambientes para promover o ambiente verde para o novo ambiente de produção. A transição normalmente leva menos de um minuto, sem perda de dados e sem necessidade de alterações na aplicação.

Como o ambiente verde é uma cópia da topologia do ambiente de produção, o ambiente verde inclui os recursos usados pela instância de banco de dados. Esses recursos incluem réplicas de leitura, configuração de armazenamento, snapshots de banco de dados, backups automatizados, Performance Insights e monitoramento aprimorado. Se a instância de banco de dados azul for uma implantação de instância de banco de dados multi-AZ, a instância de banco de dados verde também será uma implantação de instância de banco de dados multi-AZ.

> **Note**
>
> No momento, as implantações azul/verde são compatíveis apenas com o RDS para MariaDB, RDS para MySQL e RDS para PostgreSQL. Para ter informações sobre a disponibilidade do Amazon Aurora, consulte Usar implantações azul/verde do Amazon RDS para atualizações de banco de dados no Guia do usuário do Amazon Aurora.

Tópicos

- Disponibilidade de região e versão
- Benefícios do uso de implantações azul/verde do Amazon RDS
- Fluxo de trabalho de uma implantação azul/verde
- Autorizar o acesso às operações de implantação azul/verde do Amazon RDS

- Limitações e considerações relativas às implantações azul/verde do Amazon RDS
- Práticas recomendadas para implantações azul/verde do Amazon RDS

Disponibilidade de região e versão

A disponibilidade e a compatibilidade de recursos variam entre versões específicas de cada mecanismo de banco de dados e entre Regiões da AWS. Para ter mais informações, consulte the section called “Implantações azul/verde”.

Benefícios do uso de implantações azul/verde do Amazon RDS

Ao usar implantações azul/verde do Amazon RDS, você pode se manter atualizado sobre os patches de segurança, melhorar a performance do banco de dados e adotar novos recursos de banco de dados com um tempo de inatividade curto e previsível. As implantações azul/verde reduzem os riscos e o tempo de inatividade das atualizações do banco de dados, como atualizações principais ou secundárias de versões do mecanismo.

As implantações azul/verde oferecem os seguintes benefícios:

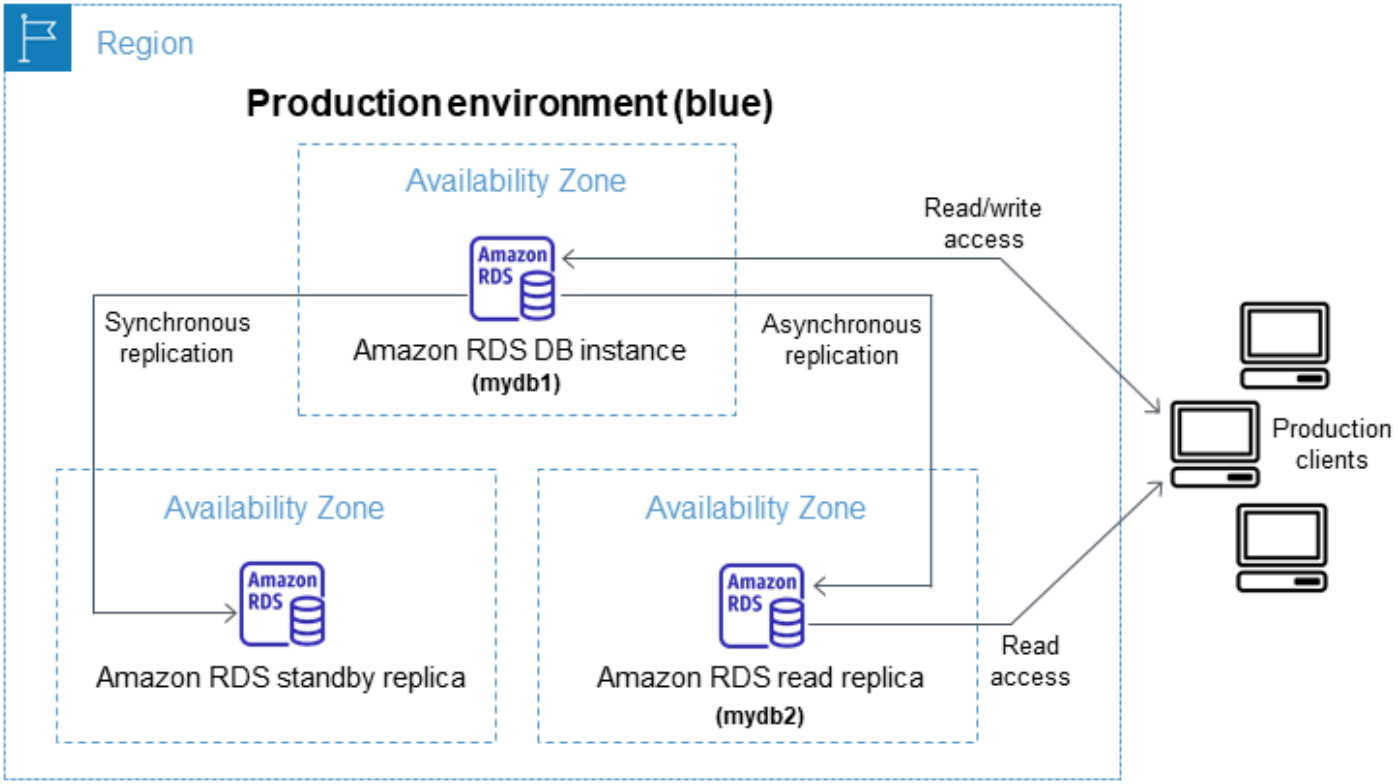
- Crie facilmente um ambiente de teste pronto para produção.
- Replique automaticamente as alterações do banco de dados do ambiente de produção para o ambiente de teste.
- Teste as alterações do banco de dados em um ambiente de teste seguro sem afetar o ambiente de produção.
- Mantenha-se atualizado com os patches do banco de dados e as atualizações do sistema.
- Implemente e teste novos recursos de banco de dados.
- Faça a transição de seu ambiente de teste para ser o novo ambiente de produção sem alterações em sua aplicação.
- Faça a transição com segurança por meio do uso de grades de proteção de transição integradas.
- Elimine a perda de dados durante a transição.
- Faça a transição rapidamente, normalmente em menos de um minuto, dependendo da sua workload.

Fluxo de trabalho de uma implantação azul/verde

Conclua as etapas principais a seguir ao usar uma implantação azul/verde para atualizações do banco de dados.

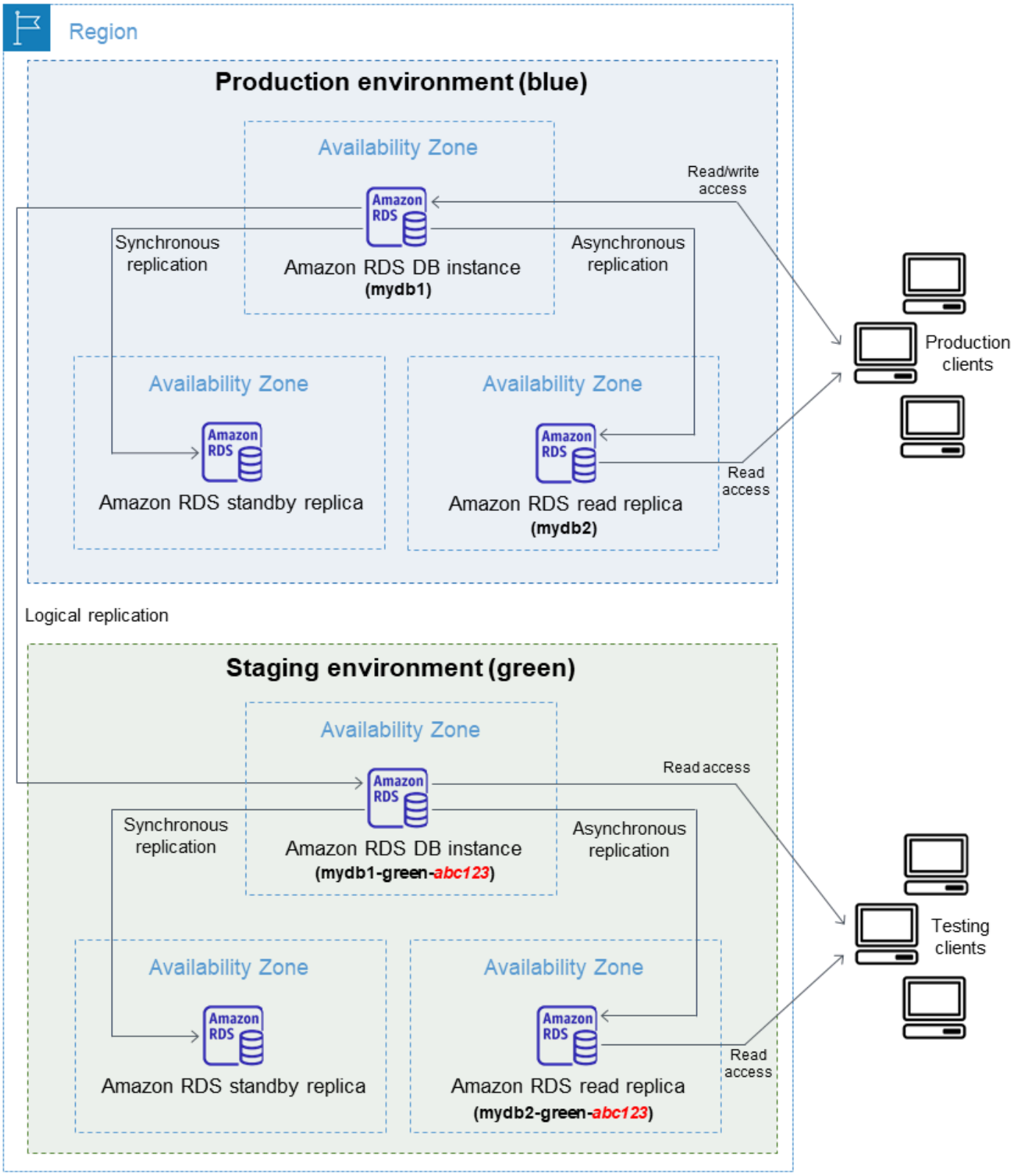
1. Identifique um ambiente de produção que exija atualizações.

   Por exemplo, o ambiente de produção nesta imagem tem uma implantação de instância de banco de dados multi-AZ (mydb1) e uma réplica de leitura (mydb2).



2. Crie a implantação azul/verde Para obter instruções, consulte Criar uma implantação azul/verde no Amazon RDS.

   A imagem a seguir mostra um exemplo de implantação azul/verde do ambiente de produção da etapa 1. Ao criar a implantação azul/verde, o RDS copia a topologia e a configuração completas da instância de banco de dados primária para criar o ambiente verde. Os nomes das instâncias de banco de dados copiadas são anexados com `-green-`*`random-characters`*. O ambiente de teste na imagem contém uma implantação de instância de banco de dados multi-AZ (mydb1-green-*`abc123`*) e uma réplica de leitura (mydb2-green-*`abc123`*).



Ao criar a implantação azul/verde, você pode atualizar sua versão do mecanismo de banco de dados e especificar um grupo de parâmetros de banco de dados diferente para as instâncias de

banco de dados no ambiente verde. O RDS também configura a replicação lógica da instância de banco de dados primária no ambiente azul para a instância de banco de dados primária no ambiente verde.

Depois de criar a implantação azul/verde, a instância de banco de dados no ambiente verde é somente para leitura por padrão.

3. Faça alterações adicionais no ambiente de teste, se necessário.

   Por exemplo, você pode fazer alterações de esquema em seu banco de dados ou alterar a classe da instância de banco de dados usada por uma ou mais instâncias de banco de dados no ambiente verde.

   Para ter mais informações sobre como modificar uma instância de banco de dados , consulte Modificar uma instância de banco de dados do Amazon RDS.

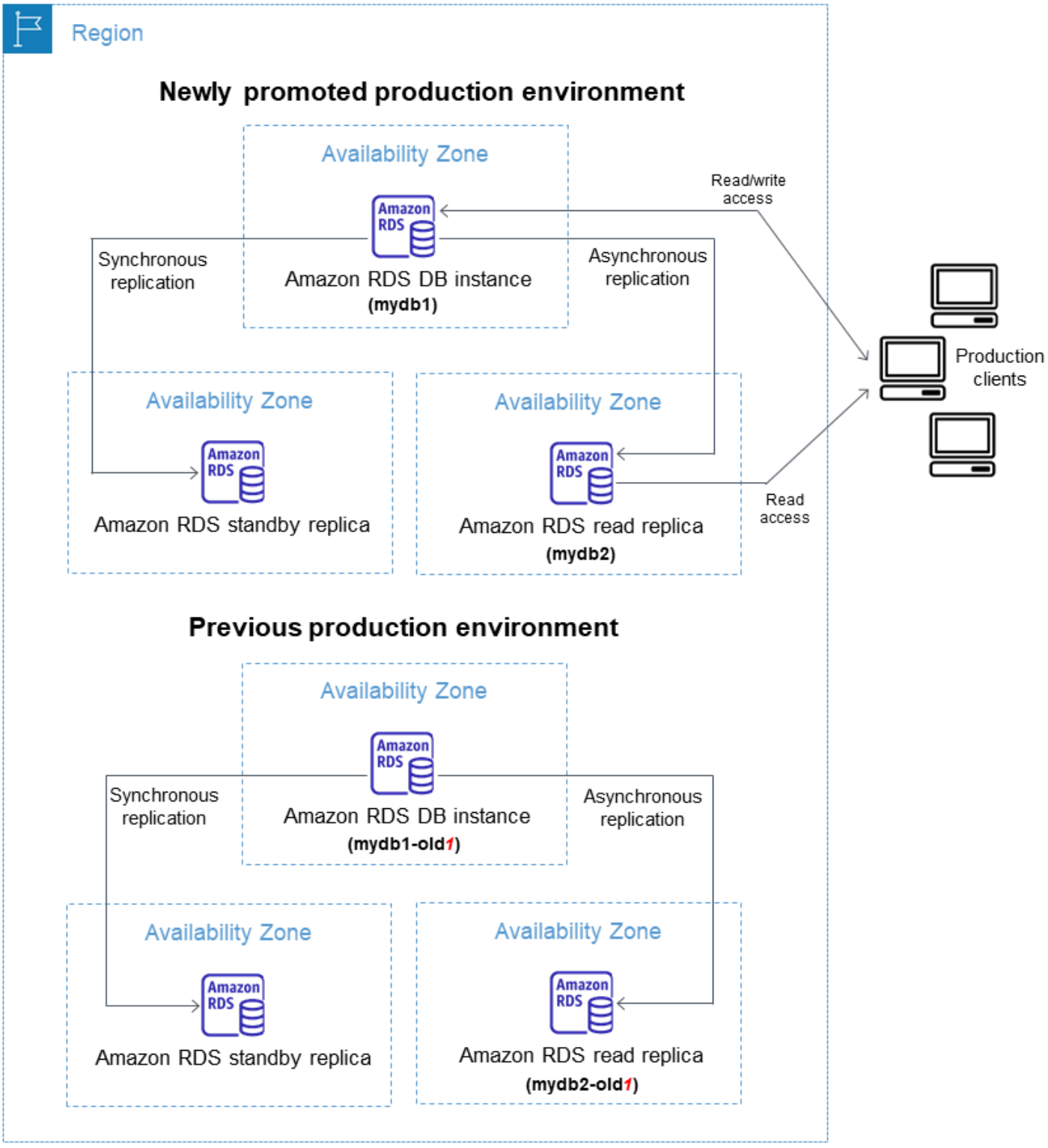
4. Teste seu ambiente de teste.

   Durante o teste, recomendamos que você mantenha seus bancos de dados no ambiente verde somente leitura. Habilite operações de gravação no ambiente verde com cuidado, pois elas podem causar conflitos de replicação. Elas também podem ocasionar dados não intencionais nos bancos de dados de produção após a transição. Para habilitar as operações de gravação para o RDS para MySQL, defina o parâmetro `read_only` como `0` e reinicialize a instância de banco de dados. Para o RDS para PostgreSQL, defina o parâmetro `default_transaction_read_only` como `off` no nível da sessão.

5. Quando estiver pronto, faça a transição para promover o ambiente de teste para o novo ambiente de produção. Para obter instruções, consulte Mudar uma implantação azul/verde no Amazon RDS.

   A transição ocasiona tempo de inatividade. O tempo de inatividade geralmente é inferior a um minuto, mas pode ser maior dependendo de sua workload.

   A imagem a seguir mostra as instâncias de banco de dados após a transição.



Após a transição, as instâncias de banco de dados que estavam no ambiente verde se tornam as novas instâncias de banco de dados de produção. Os nomes e os endpoints no ambiente de produção atual são atribuídos ao ambiente de produção recém-promovido, sem exigir alterações

em sua aplicação. Como resultado, seu tráfego de produção agora flui para o novo ambiente de produção. As instâncias de banco de dados no ambiente azul anterior são renomeadas anexando `-old`*n* ao nome atual, em que *n* é um número. Por exemplo, suponha que o nome da instância de banco de dados no ambiente azul seja `mydb1`. Após a transição, o nome da instância de banco de dados pode ser `mydb1-old1`.

No exemplo da imagem, as seguintes alterações ocorrem durante a alternância:

- A implantação de instância de banco de dados multi-AZ do ambiente verde denominada `mydb1-green-abc123` se torna a implantação de instância de banco de dados multi-AZ de produção denominada `mydb1`.
- A réplica de leitura do ambiente verde chamada `mydb2-green-abc123` se torna a réplica de leitura de produção `mydb2`.
- A implantação de instância de banco de dados multi-AZ do ambiente azul denominada `mydb1` torna-se `mydb1-old1`.
- A réplica de leitura do ambiente azul denominada `mydb2` torna-se `mydb2-old1`.

6. Caso não precise mais de uma implantação azul/verde, você pode excluí-la. Para obter instruções, consulte Excluir uma implantação azul/verde no Amazon RDS.

   Após a transição, o ambiente de produção anterior não é excluído para que você possa usá-lo para testes de regressão, se necessário.

Autorizar o acesso às operações de implantação azul/verde do Amazon RDS

Os usuários devem ter as permissões necessárias para realizar operações relacionadas às implantações azul/verde. É possível criar políticas do IAM que concedam aos usuários e perfis permissão para executar operações de API específicas nos recursos especificados de que precisam. Depois, você pode anexar essas políticas aos conjuntos de permissões do IAM ou às funções que exigem essas permissões. Para ter mais informações, consulte Gerenciamento de identidade e acesso no Amazon RDS.

O usuário que cria uma implantação azul/verde deve ter permissões para realizar as seguintes operações do RDS:

- `rds:CreateBlueGreenDeployment`
- `rds:AddTagsToResource`

- `rds:CreateDBInstanceReadReplica`

O usuário que faz a transição de uma implantação azul/verde deve ter permissões para realizar as seguintes operações do RDS:

- `rds:SwitchoverBlueGreenDeployment`
- `rds:ModifyDBInstance`
- `rds:PromoteReadReplica`

O usuário que exclui uma implantação azul/verde deve ter permissões para realizar as seguintes operaçõe do RDS:

- `rds:DeleteBlueGreenDeployment`
- `rds:DeleteDBInstance`

O Amazon RDS provisiona e modifica recursos no ambiente de preparação em seu nome. Esses recursos incluem instâncias de banco de dados que usam uma convenção de nomenclatura definida internamente. Portanto, as políticas do IAM anexadas não podem conter padrões parciais de nomes de recursos, como `my-db-prefix-*`. Somente curingas (*) são compatíveis. Em geral, recomendamos o uso de tags de recursos e outros atributos compatíveis para controlar o acesso a esses recursos, em vez do uso de curingas. Consulte mais informações em Actions, resources, and condition keys for Amazon RDS.

Limitações e considerações relativas às implantações azul/verde do Amazon RDS

Implantações azul/verde no Amazon RDS exigem uma análise cuidadosa de fatores, como slots de replicação, gerenciamento de recursos, dimensionamento de instâncias e possíveis impactos na performance do banco de dados. As seções a seguir fornecem orientação para ajudar a otimizar a estratégia de implantação para garantir o mínimo de tempo de inatividade, transições perfeitas e gerenciamento eficaz do ambiente de banco de dados.

Tópicos

- Limitações para implantações azul/verde
- Considerações sobre implantações azul/verde

Limitações para implantações azul/verde

As seguintes limitações se aplicam às implantações azul/verde:

Tópicos

- Limitações para implantações azul/verde
- Limitações do RDS para MySQL em implantações azul/verde
- Limitações do RDS para PostgreSQL para implantações azuis/verdes com replicação física
- Limitações do RDS para PostgreSQL em implantações azuis/verdes com replicação lógica

Limitações para implantações azul/verde

As seguintes limitações se aplicam às implantações azul/verde:

- As implantações azul/verde não são compatíveis com o gerenciamento de senhas de usuário principal com AWS Secrets Manager.
- Se o volume de log dedicado (DLV) estiver habilitado no banco de dados azul, ele deverá estar habilitado em todas as instâncias de banco de dados, incluindo réplicas de leitura.
- Durante a transição, os ambientes azul e verde não podem ter integrações ETL zero com o Amazon Redshift. Você deve excluir a integração primeiro, alternar e, depois, recriar a integração.
- O Agendador de Eventos (parâmetro `event_scheduler`) deve ser desativado no ambiente verde ao criar uma implantação azul/verde. Isso impede que eventos sejam gerados no ambiente verde e causem inconsistências.
- Você não pode alterar uma instância de banco de dados não criptografada em uma instância de banco de dados não criptografada. Além disso, não é possível alterar uma instância de banco de dados não criptografada em uma instância de banco de dados não criptografada.
- Não é possível alterar uma instância de banco de dados azul para uma versão de mecanismo posterior à instância de banco de dados verde correspondente.
- Os recursos no ambiente azul e no ambiente verde devem estar na mesma Conta da AWS.
- As implantações azul/verde não são compatíveis com os seguintes recursos:
  - Amazon RDS Proxy
  - Propagar réplicas de leitura
  - Réplicas de leitura entre regiões
  - AWS CloudFormation

- Implantações de clusters de banco de dados multi-AZ

  O recurso de implantação azul/verde é compatível com implantações de instâncias de banco de dados multi-AZ. Para ter mais informações sobre implantações Multi-AZ, consulte Configurar e gerenciar uma implantação multi-AZ para o Amazon RDS.

Limitações do RDS para MySQL em implantações azul/verde

As seguintes limitações se aplicam às implantações azul/verde do RDS para MySQL:

- A instância de banco de dados azul não pode ser uma réplica externa de log binário.
- Se o banco de dados de origem estiver associado a um grupo de opções personalizado, você não poderá especificar uma atualização de versão principal ao criar a implantação azul/verde.

  Nesse caso, você pode criar uma implantação azul/verde sem especificar uma atualização de versão principal. Depois, você pode atualizar o banco de dados no ambiente verde. Para obter mais informações, consulte Atualizar a versão de mecanismo de uma instância de banco de dados.
- As implantações azul/verde não são compatíveis com o driver JDBC da AWS para MySQL. Consulte mais informações em Known Limitations no GitHub.

Limitações do RDS para PostgreSQL para implantações azuis/verdes com replicação física

As seguintes limitações não se aplicam às implantações azuis/verdes do RDS para PostgreSQL que usam replicação física. Para obter uma explicação de quando as implantações azuis/verdes usam a replicação física em vez da replicação lógica, consulte ???.

- Depois que o ambiente verde for criado, não será possível fazer uma atualização manual da versão principal.
- As implantações azuis/verdes que usam replicação física não oferecem suporte a alterações de esquema no ambiente verde, pois ele é estritamente somente leitura.
- A instância de banco de dados azul não pode ser uma origem lógica (publicador) nem uma réplica (assinante).

Limitações do RDS para PostgreSQL em implantações azuis/verdes com replicação lógica

As seguintes limitações do se aplicam às implantações azuis/verdes do RDS para PostgreSQL que usam replicação lógica. Para obter uma explicação de quando as implantações azuis/verdes usam a replicação lógica em vez da replicação física, consulte ???.

- As tabelas não registradas não são replicadas no ambiente verde.
- A instância de banco de dados azul não pode ser uma origem lógica (publicador) nem uma réplica (assinante).
- Se o de banco de dados estiver configurado como o servidor externo de uma extensão de invólucro de dados externo (FDW), você deverá usar o nome do endpoint do da instância em vez dos endereços IP. Isso permite que a configuração permaneça funcional após a transição.
- Em uma implantação azul/verde, cada banco de dados exige um slot de replicação lógica. À medida que o número de bancos de dados aumenta, a sobrecarga de recursos aumenta e pode gerar um atraso na replicação, principalmente se a instância de banco de dados não estiver suficientemente escalada. O impacto depende de fatores, como a workload de bancos de dados e o número de conexões. Para atenuar isso, pense em escalar sua classe de instância de banco de dados ou reduzir o número de bancos de dados na instância de origem.
- O processo de aplicação da replicação lógica no ambiente verde é de thread único. Se o ambiente azul gerar um alto volume de tráfego de gravação, o ambiente verde talvez não consiga acompanhar o ritmo. Isso pode causar atraso ou falha na replicação, especialmente para workloads que produzem um throughput de gravação alto e contínuo. Teste minuciosamente suas workloads. Em cenários que exigem atualizações de versão principal e processamento de workloads com alto volume de gravação, considere abordagens alternativas, como usar o AWS Database Migration Service (AWS DMS).
- Não é possível criar outras partições em tabelas particionadas durante implantações azul/verde do RDS para PostgreSQL. A criação de outras partições requer operações de linguagem de definição de dados (DDL), como `CREATE TABLE`, que não são replicadas do ambiente azul para o ambiente verde. No entanto, as tabelas particionadas existentes e os respectivos dados serão replicados para o ambiente verde.
- As limitações a seguir se aplicam às extensões do PostgreSQL:
  - A extensão `pg_partman` deve ser desabilitada no ambiente azul quando uma implantação azul/verde é criada. A extensão realiza operações de DDL como `CREATE TABLE`, que interrompem a replicação lógica do ambiente azul no ambiente verde.
  - A extensão `pg_cron` deve permanecer desativada em todos os bancos de dados verdes após a criação da implantação azul/verde. A extensão tem trabalhadores em segundo plano que são

executados como superusuários e ignoram a configuração somente leitura do ambiente verde, o que pode causar conflitos de replicação.

- As extensões `pglogical` e `pgactive` devem ser desativadas no ambiente azul ao criar uma implantação azul/verde. Depois de fazer a transição do ambiente verde para o novo ambiente de produção, será possível habilitar as extensões novamente. Além disso, o banco de dados azul não pode ser um assinante lógico de uma instância externa.
- Se você estiver usando a extensão `pgAudit`, ela deverá permanecer nas bibliotecas compartilhadas (`shared_preload_libraries`) nos grupos de parâmetros de banco de dados personalizados para as instâncias de banco de dados azul e verde. Para obter mais informações, consulte the section called “Configurar a extensão pgAudit”.

Limitações específicas de replicação lógica para implantações azuis/verdes

O PostgreSQL tem certas restrições relacionadas à replicação lógica, que se traduzem em limitações ao criar implantações azul/verdes para clusters de banco de dados para instâncias de banco de dados PostgreSQL.

A tabela a seguir descreve as limitações de replicação lógica que se aplicam às implantações azul/verde do . Para obter mais informações sobre a replicação lógica do PostgreSQL, consulte a documentação do PostgreSQL.

Limitação	Explicação
Declarações de linguagem de definição de dados (DDL), como `CREATE TABLE` e`CREATE SCHEMA`, não são replicadas do ambiente azul para o ambiente verde.	Se o Amazon RDS detectar uma alteração de DDL no ambiente azul, seus bancos de dados verdes entrarão em um estado de replicação degradada . Você deve excluir a implantação azul/verde e todos os bancos de dados verdes e, em seguida, recriá-la.
As instruções de linguagem	Se o Amazon RDS para PostgreSQL detectar uma tentativa de execução de uma instrução de DCL no ambiente azul, você verá uma mensagem de

Limitação	Explicação
de controle de dados (DCL), como `GRANT` e `REVOKE`, não são replicadas do ambiente azul para o ambiente verde.	aviso. Não há configuração ou API disponível para alterar esse comportam ento, pois é uma limitação do processo de implantação azul/verde.
As operações `NEXTVAL` em objetos de sequência não são sincronizadas entre o ambiente azul e o ambiente verde.	Durante a transição, o Amazon RDS incrementa os valores da sequência no ambiente verde para corresponder aos do ambiente azul. Se você tiver milhares de sequências, isso pode atrasar a transição.
Os objetos grandes no ambiente azul não são replicada s no ambiente verde. Isso inclui objetos grandes existente s e quaisquer objetos grandes recém-criados ou modificad os durante o processo de implantação azul/ verde.	Se o Amazon RDS detectar a criação ou modificação de objetos grandes no ambiente azul que estão armazenados na tabela do **`pg_largeobject`** sistema, seus bancos de dados verdes entrarão em um estado de replicação degradada. Você deve excluir a implantação azul/verde e todos os bancos de dados verdes e, em seguida, recriá-la.

Limitação	Explicação
As visões materializadas não são atualizadas automaticamente no ambiente verde.	Atualizar visualizações materializadas no ambiente azul não as atualiza no ambiente verde. Após a transição, é possível atualizá-los manualmente usando o comando REFRESH MATERIALIZED VIEW ou agendar uma atualização.
As operações UPDATE e DELETE não são permitidas em tabelas que não têm uma chave primária.	Antes de criar uma implantação azul/verde, verifique se todas as tabelas têm uma chave primária ou use `REPLICA IDENTITY FULL`. No entanto, use somente `REPLICA IDENTITY FULL` se não houver uma chave primária ou única, pois isso afeta a performance da replicação. Para obter mais informações, consulte a Documentação do PostgreSQL.

Considerações sobre implantações azul/verde

O Amazon RDS rastreia recursos em implantações azul/verde com o `DbiResourceId` de cada recurso. Esse ID de recurso é um identificador imutável e exclusivo da Região da AWS do recurso.

O ID do recurso é diferente do ID da instância de banco de dados. Cada um está listado na configuração do banco de dados no console do RDS.

O nome (ID da instância) de um recurso muda quando você faz a transição de uma implantação azul/verde, mas cada recurso mantém o mesmo ID de recurso. Por exemplo, um identificador de instância de banco de dados pode ser `mydb` no ambiente azul. Após a transição, a mesma instância de banco de dados pode ser renomeada para `mydb-old1`. No entanto, o ID do recurso da instância de banco de dados não muda durante a transição. Portanto, na transição dos recursos verdes para os novos recursos de produção, os respectivos IDs de recurso não correspondem aos IDs de recurso azuis que estavam anteriormente em produção.

Depois que você realizar a transição de uma implantação azul/verde, pense em atualizar os IDs de recurso para aqueles dos recursos de produção cuja transição foi feita recentemente para recursos e serviços integrados que você utilizou com os recursos de produção. Especificamente, considere as seguintes atualizações:

- Se você realizar a filtragem usando a API e os IDs de recursos do RDS, ajuste os IDs de recursos usados na filtragem após a transição.
- Se você usa o CloudTrail para recursos de auditoria, ajuste os consumidores do CloudTrail para rastrear os novos IDs de recursos após a transição. Para ter mais informações, consulte Monitorar chamadas de API do Amazon RDSno AWS CloudTrail.
- Se você usar a API do Performance Insights, ajuste os IDs dos recursos nas chamadas para a API após a transição. Para ter mais informações, consulte Monitorar a carga de banco de dados com o Performance Insights no Amazon RDS.

  Você pode monitorar um banco de dados com o mesmo nome após a transição, mas ele não contém os dados de antes da transição.
- Se você usar IDs de recurso nas políticas do IAM, adicione os IDs dos recursos dos quais foi feita a transição recentemente quando necessário. Para obter mais informações, consulte Gerenciamento de identidade e acesso no Amazon RDS.
- Se você tiver perfis do IAM associados à instância de banco de dados, associe-os novamente depois da transição. Os perfis anexados não são copiados automaticamente no ambiente verde.
- Se você se autenticar na instância de banco de dados usando a autenticação do banco de dados do IAM, garanta que a política do IAM usada para acesso ao banco de dados tenha os bancos de dados azul e verde listados sob o elemento `Resource` da política. Isso é necessário para se conectar ao banco de dados verde após a transição. Para obter mais informações, consulte the section called “Criar e usar uma política do IAM para acesso do banco de dados do IAM”.
- Se você usa AWS Backup para gerenciar backups automatizados de recursos em uma implantação azul/verde, ajuste os IDs de recursos usados por AWS Backup após a transição. Para ter mais informações, consulte Usar o AWS Backup no gerenciamento de backups automatizados para o Amazon RDS.
- Se você quiser restaurar um snapshot de banco de dados manual ou automatizado para uma instância de banco de dados que fazia parte de uma implantação azul/verde, restaure o snapshot de banco de dados correto examinando a hora em que o snapshot foi obtido. Para ter mais informações, consulte Restaurar uma instância de banco de dados.
- Se você quiser descrever o backup automatizado de uma instância de banco de dados do ambiente azul anterior ou restaurá-lo para um determinado momento, use o ID do recurso para a operação.

  Como o nome da instância de banco de dados muda durante a transição, você não pode usar seu nome anterior para operações `DescribeDBInstanceAutomatedBackups` ou `RestoreDBInstanceToPointInTime`.

Para ter mais informações, consulte Restaurar uma instância de banco de dados para um momento especificado no Amazon RDS.

- Quando você adiciona uma réplica de leitura a uma instância de banco de dados no ambiente verde de uma implantação azul/verde, a nova réplica de leitura não substituirá uma réplica de leitura no ambiente azul quando você fizer a transição. No entanto, a nova réplica de leitura é mantida no novo ambiente de produção após a transição.
- Depois de mudar, as tarefas de replicação do AWS Database Migration Service (AWS DMS) não podem ser retomadas porque o ponto de verificação do ambiente azul é inválido no ambiente verde. Você deve recriar a tarefa do DMS com um novo ponto de verificação para continuar a replicação.
- Quando você exclui uma instância de banco de dados no ambiente verde de uma implantação azul/verde, não é possível criar uma instância de banco de dados para substituí-la na implantação azul/verde.

  Se você criar uma instância de banco de dados com o mesmo nome e nome do recurso da Amazon (ARN) da instância de banco de dados excluída, ela terá um `DbiResourceId` diferente, portanto, não fará parte do ambiente verde.

  Ocorrerá o comportamento a seguir se você excluir uma instância de banco de dados no ambiente verde:

  - Se existir uma instância de banco de dados no ambiente azul com o mesmo nome, não será feita a transição dela para a instância de banco de dados no ambiente verde. Essa instância de banco de dados não será renomeada adicionando `-old`*n* ao nome da instância de banco de dados.
  - Qualquer aplicação que aponte para a instância de banco de dados no ambiente azul continua usando a mesma instância de banco de dados após a transição.

  O mesmo comportamento se aplica às instâncias de banco de dados e às réplicas de leitura.

Práticas recomendadas para implantações azul/verde do Amazon RDS

Veja as práticas recomendadas para implantações azul/verde.

Tópicos

- Práticas recomendadas gerais para implantações azuis/verdes
- Práticas recomendadas do RDS para MySQL para implantações azuis/verdes

- Práticas recomendadas do RDS para PostgreSQL para implantações azuis/verdes

Práticas recomendadas gerais para implantações azuis/verdes

Considere as seguintes práticas recomendadas gerais ao criar uma implantação azul/verde.

- Teste minuciosamente as instâncias de banco de dados no ambiente verde antes da transição.
- Mantenha seus bancos de dados no ambiente verde somente leitura. Recomendamos que você habilite as operações de gravação no ambiente verde com cuidado, pois elas podem causar conflitos de replicação. Elas também podem ocasionar dados não intencionais nos bancos de dados de produção após a transição.
- Se você usar uma implantação azul/verde para implementar alterações de esquema, faça somente alterações compatíveis com a replicação.

  Por exemplo, é possível adicionar novas colunas ao final de uma tabela sem interromper a replicação da implantação azul para a implantação verde. No entanto, alterações de esquema, como renomear colunas ou renomear tabelas, transformam a replicação na implantação verde.

  Para ter mais informações sobre alterações compatíveis com replicação, consulte Replicação com diferentes definições de tabela na origem e na réplica na documentação do MySQL e Restrições na documentação de replicação lógica do PostgreSQL.

  > **Note**
  >
  > Essa limitação não se aplica às implantações azuis/verdes do RDS para PostgreSQL que usam replicação física. Para ter mais informações, consulte the section called “Limitações do RDS para PostgreSQL para implantações azuis/verdes com replicação física”.

- Depois de criar a implantação azul/verde, manipule o carregamento lento, se necessário. O carregamento de dados deve ser concluído antes da transição. Para ter mais informações, consulte Carregamento lento e inicialização de armazenamento para implantações azuis/verdes.
- Ao realizar a transição de uma implantação azul/verde, siga as práticas recomendadas de transição. Para ter mais informações, consulte the section called “Práticas recomendadas de transição”.

Práticas recomendadas do RDS para MySQL para implantações azuis/verdes

Considere as seguintes práticas recomendadas ao criar uma implantação azul/verde em uma instância de banco de dados do RDS para MySQL.

- Evite usar mecanismos de armazenamento não transacionais, como o MyISAM, que não são otimizados para replicação.
- Otimize as réplicas de leitura e o ambiente verde para replicação de logs binários. Se compatível com o mecanismo de banco de dados, habilite as replicações GTID, paralela e à prova de falhas para garantir consistência e durabilidade dos dados antes de criar a implantação azul/verde. Para ter mais informações, consulte Usar a replicação baseada em GTID.
- Se o ambiente verde apresentar atraso na réplica, pense no seguinte:
  - Defina o parâmetro `innodb_flush_log_at_trx_commit` temporariamente como 2 no grupo de parâmetros verde do banco de dados. Depois de atualizar a replicação, faça a reversão para o valor padrão de 1 antes da transição. Se ocorrer um desligamento inesperado ou uma falha com os valores dos parâmetros temporários, compile o ambiente verde novamente para evitar que nenhuma corrupção de dados passe despercebida.
  - Para reduzir a latência de gravação e melhorar o throughput da replicação, altere temporariamente as instâncias verdes de banco de dados multi-AZ para instâncias de banco de dados single-AZ. Reabilite a multi-AZ logo antes da transição.

Práticas recomendadas do RDS para PostgreSQL para implantações azuis/verdes

Considere as seguintes práticas recomendadas ao criar uma implantação azul/verde em uma instância de banco de dados do RDS para PostgreSQL.

Tópicos

- Práticas recomendadas gerais do RDS para PostgreSQL para implantações azuis/verdes
- Práticas recomendadas do RDS para PostgreSQL para implantações azuis/verdes com replicação física
- Práticas recomendadas do RDS para PostgreSQL para implantações azuis/verdes com replicação lógica

Práticas recomendadas gerais do RDS para PostgreSQL para implantações azuis/verdes

Considere as seguintes práticas recomendadas gerais ao criar uma implantação azul/verde em uma instância de banco de dados do RDS para PostgreSQL.

- Atualize todas as extensões do PostgreSQL para a versão mais recente antes de criar uma implantação azul/verde. Para ter mais informações, consulte the section called “Atualizar extensões do PostgreSQL”.
- Transações de longa duração podem causar um atraso significativo na réplica. Para reduzir o atraso na réplica, pense no seguinte:
  - Reduza as transações de longa duração que podem ser adiadas até que o ambiente verde alcance o ambiente azul.
  - Reduza as operações em massa no ambiente azul até que o ambiente verde alcance o ambiente azul.
  - Inicie uma operação manual de congelamento de vacuum em tabelas ocupadas antes de criar a implantação azul/verde.
  - Em relação ao PostgreSQL versão 12 e posterior, desabilite o parâmetro `index_cleanup` em tabelas grandes ou ocupadas para aumentar a taxa de manutenção normal em bancos de dados azuis. Para ter mais informações, consulte the section called “Aspirar uma tabela o mais rápido possível”.

    > **Note**
    >
    > Ignorar regularmente a limpeza do índice durante a aspiração pode causar inchaço no índice, o que pode prejudicar o desempenho da limpeza. Como prática recomendada, utilize essa abordagem somente ao usar uma implantação azul/verde. Depois que a implantação estiver concluída, recomendamos retomar a manutenção e a limpeza regulares do índice.

- A replicação lenta pode fazer com que remetentes e destinatários sejam reiniciados com frequência, o que atrasa a sincronização. Para garantir que eles permaneçam ativos, desabilite os tempos limite definindo o parâmetro `wal_sender_timeout` como `0` no ambiente azul e o parâmetro `wal_receiver_timeout` como `0` no ambiente verde.
- Para evitar que segmentos de log de gravação antecipada (WAL) sejam removidos do ambiente azul, defina o parâmetro `wal_keep_segments` como 15625 para o PostgreSQL versão 13 e

anterior. Para a versão 14 e posterior, defina o parâmetro `wal_keep_size` como 1 TiB, se houver espaço de armazenamento livre suficiente.

Práticas recomendadas do RDS para PostgreSQL para implantações azuis/verdes com replicação física

Com a replicação física, o Amazon RDS cria uma réplica de leitura da instância de banco de dados de origem. Para parâmetros relacionados, monitoramento, ajuste e solução de problemas, consulte the section called “Trabalhar com réplicas de leitura do RDS para PostgreSQL”.

Para obter uma explicação de quando as implantações azuis/verdes usam a replicação física em vez da replicação lógica, consulte ???.

Práticas recomendadas do RDS para PostgreSQL para implantações azuis/verdes com replicação lógica

Considere as seguintes práticas recomendadas ao criar uma implantação azul/verde que usa replicação lógica. Para obter uma explicação de quando as implantações azuis/verdes usam a replicação lógica em vez da replicação física, consulte ???.

- Se o banco de dados tiver memória livre suficiente, aumente o valor do parâmetro de banco de dados `logical_decoding_work_mem` no ambiente azul. Isso permite menos decodificação no disco e uso da memória. Para obter mais informações, consulte a Documentação do PostgreSQL.
  - É possível monitorar o estouro de transações que está sendo gravado em disco usando a métrica `ReplicationSlotDiskUsage` do CloudWatch. Essa métrica oferece insights sobre o uso do espaço em disco por slots de replicação, ajudando a identificar quando os dados da transação excedem a capacidade de memória e são armazenados em disco. Você pode monitorar a memória livre com a métrica do `FreeableMemory` do CloudWatch. Para ter mais informações, consulte the section called “Métricas específicas da instância do Amazon CloudWatch para Amazon RDS”.
  - No RDS para PostgreSQL versão 14 e posterior, é possível monitorar o tamanho dos arquivos de estouro lógico usando a visualização `pg_stat_replication_slots` do sistema.
- Se você estiver usando a extensão `aws_s3`, conceda à instância de banco de dados verde acesso ao Amazon S3 por meio de um perfil do IAM após a criação do ambiente verde. Isso permite que os comandos de importação e exportação continuem funcionando após a transição. Para instruções, consulte the section called “Configurar o acesso a um bucket do Amazon S3”.

- Analise o desempenho das instruções UPDATE e DELETE e avalie se a criação de um índice na coluna utilizada na cláusula WHERE pode otimizar essas consultas. Isso pode melhorar o desempenho quando as operações são repetidas no ambiente verde.
- Se você estiver usando gatilhos, garanta que eles não interfiram na criação, atualização e eliminação de objetos `pg_catalog.pg_publication`, `pg_catalog.pg_subscription` e `pg_catalog.pg_replication_slots` cujos nomes comecem com “rds”.
- Se você especificar uma versão posterior do mecanismo para o ambiente verde, execute a operação `ANALYZE` em todos os bancos de dados para atualizar a tabela `pg_statistic`. As estatísticas do otimizador não são transferidas durante uma atualização de versão principal, portanto, é necessário gerar novamente todas as estatísticas para evitar problemas de performance. Para conhecer práticas recomendadas adicionais durante as principais atualizações de versões, consulte the section called “Como atualizar a versão principal”.
- Evite configurar gatilhos como `ENABLE REPLICA` ou `ENABLE ALWAYS` se o gatilho for usado na origem para manipular dados. Caso contrário, o sistema de replicação propagará as alterações e executará o gatilho, o que ocasiona duplicação.

Criar uma implantação azul/verde no Amazon RDS

Ao criar uma implantação azul/verde, você especifica a instância de banco de dados a ser copiada na implantação. A instância de banco de dados selecionada é a instância de banco de dados de produção e se torna a instância de banco de dados primária no ambiente azul. Essa instância de banco de dados é copiada no ambiente verde, e o RDS configura a replicação da instância de banco de dados no ambiente azul para a instância de banco de dados no ambiente verde.

O RDS copia a topologia e os recursos do ambiente azul para uma área de preparação. Se uma instância de banco de dados azul tiver réplicas de leitura, elas serão copiadas como réplicas da instância verde. O armazenamento alocado de todas as réplicas verdes corresponde à instância primária verde, enquanto outros parâmetros de armazenamento são herdados das réplicas azuis.

Se a instância de banco de dados azul for uma implantação de instância de banco de dados multi-AZ, a instância de banco de dados verde será criada como uma implantação de instância de banco de dados multi-AZ.

Tópicos

- Preparação para uma implantação azul/verde
- Especificar as alterações ao criar uma implantação azul/verde

- Carregamento lento e inicialização de armazenamento para implantações azuis/verdes
- Criar uma implantação azul/verde
- Configurações para criar implantações azuis/verdes

Preparação para uma implantação azul/verde

Há algumas etapas que você deve seguir antes de criar uma implantação azul/verde, dependendo do mecanismo que a instância de banco de dados está executando.

Tópicos

- Preparação de uma instância de banco de dados do RDS para MySQL ou RDS para MariaDB para uma implantação azul/verde
- Preparação de uma instância de banco de dados do RDS para PostgreSQL para uma implantação azul/verde com replicação física
- Preparação de uma instância de banco de dados do RDS para PostgreSQL para uma implantação azul/verde com replicação lógica

Preparação de uma instância de banco de dados do RDS para MySQL ou RDS para MariaDB para uma implantação azul/verde

Antes de criar uma implantação azul/verde para uma instância de banco de dados do RDS para MySQL ou RDS para MariaDB, é necessário habilitar os backups automatizados. Para instruções, consulte the section called “Ativar backups automáticos”.

Preparação de uma instância de banco de dados do RDS para PostgreSQL para uma implantação azul/verde com replicação física

Antes de criar uma implantação azul/verde do RDS para PostgreSQL que usa replicação física, faça o seguinte: Para obter uma lista de versões que usam replicação física versus replicação lógica, consulte ???.

- Habilite backups automatizados na instância de banco de dados. Para instruções, consulte the section called “Ativar backups automáticos”.
- Confirme se a instância de banco de dados não é a origem ou o destino da replicação externa. Para obter mais informações, consulte the section called “Limitações gerais”.

Preparação de uma instância de banco de dados do RDS para PostgreSQL para uma implantação azul/verde com replicação lógica

Antes de criar uma implantação azul/verde do RDS para PostgreSQL que usa replicação lógica, faça o seguinte: Para obter uma lista de versões que usam replicação lógica versus replicação física, consulte ???.

- Associe a instância a um grupo de parâmetros de banco de dados personalizado com a replicação lógica (`rds.logical_replication`) ativada. A replicação lógica é necessária para a replicação do ambiente azul no ambiente verde. Para instruções, consulte the section called “Modificar parâmetros em um grupo de parâmetros de banco de dados”.

  Como as implantações azul/verde exigem pelo menos um operador em segundo plano por banco de dados, ajuste as configurações a seguir de acordo com a workload. Para ter instruções sobre como ajustar cada configuração, consulte Configuration Settings na documentação do PostgreSQL.

  - `max_replication_slots`
  - `max_wal_senders`
  - `max_logical_replication_workers`
  - `max_worker_processes`

  Depois de ativar a replicação lógica e definir todas as opções de configuração, reinicialize a instância de banco de dados para que as alterações tenham efeito. As implantações azul/verde exigem que a instância do banco de dados esteja sincronizada com o grupo de parâmetros do banco de dados; caso contrário, a criação falhará. Para obter mais informações, consulte the section called “Reinicializar uma instância de banco de dados”.
- Confirme se a instância de banco de dados não é a origem ou o destino da replicação externa. Para obter mais informações, consulte the section called “Limitações gerais”.
- Certifique-se de que todas as tabelas na instância de banco de dados tenham uma chave primária. A replicação lógica do PostgreSQL não permite operações `UPDATE` ou `DELETE` em tabelas que não têm uma chave primária.
- O RDS para PostgreSQL usa a replicação lógica nativa do PostgreSQL, armazenando segmentos do log de gravação antecipada (WAL) na instância azul até que sejam reproduzidos no ambiente verde. Antes de criar uma implantação azul/verde, verifique se a instância azul tem capacidade adequada conferindo as seguintes métricas:
  - `FreeStorageSpace`

- `TransactionLogsGeneration`
- `TransactionLogsDiskUsage`
- `OldestReplicationSlotLag`

Para estimar o armazenamento adicional necessário na instância azul, monitore a métrica `TransactionLogsGeneration` do CloudWatch durante os períodos de pico da workload. Por exemplo, se a workload gerar 100 GB de dados do WAL em 24 horas, tenha pelo menos 100 GB de armazenamento extra para acomodar um dia de segmentos do WAL. Para obter mais informações, consulte Métricas de monitoramento em uma instância de banco de dados.

Especificar as alterações ao criar uma implantação azul/verde

Você pode fazer as seguintes alterações na instância de banco de dados no ambiente verde ao criar a implantação azul/verde:

Você pode fazer outras modificações na instância de banco de dados no ambiente verde após sua implantação. Por exemplo, você pode especificar uma versão posterior do mecanismo ou um grupo de parâmetros diferente.

Para ter mais informações sobre como modificar uma instância de banco de dados , consulte Modificar uma instância de banco de dados do Amazon RDS.

Tópicos

- Especifique uma versão de mecanismo superior
- Especificar outro grupo de parâmetros de banco de dados
- Modificar as configurações de armazenamento e desempenho
- Utilizar gravações otimizadas pelo RDS
- Atualizar a configuração do armazenamento

Especifique uma versão de mecanismo superior

Você poderá especificar uma versão superior do mecanismo se quiser testar uma atualização do mecanismo de banco de dados. Após a transição, o banco de dados é atualizado para a versão principal ou secundária do mecanismo de banco de dados que você especificar.

Especificar outro grupo de parâmetros de banco de dados

É possível testar como as alterações de parâmetros afetam as instâncias de banco de dados no ambiente verde ou especificar um grupo de parâmetros para uma nova versão principal do mecanismo de banco de dados no caso de uma atualização.

Se você especificar um grupo de parâmetros de banco de dados diferente, o grupo de parâmetros de banco de dados especificado será associado a todas as instâncias de banco de dados no ambiente verde. Se você não especificar um grupo de parâmetros diferente, cada instância de banco de dados no ambiente verde será associada ao grupo de parâmetros de sua instância de banco de dados azul correspondente.

Modificar as configurações de armazenamento e desempenho

Ajuste as configurações de armazenamento e desempenho no ambiente verde para otimizar a alocação de recursos. Essas configurações incluem armazenamento alocado, IOPS provisionadas, tipo de armazenamento e throughput de armazenamento (para armazenamento gp3).

É possível alterar o tipo de armazenamento da instância de banco de dados verde para gp2, gp3, io1 ou io2. Para armazenamento gp3, você também pode ajustar o throughput de armazenamento para melhorar o desempenho da transferência de dados para workloads de alta demanda ou para reduzir os custos de aplicações menos intensivas. Para obter mais informações, consulte Armazenamento de instâncias de banco de dados do Amazon RDS.

Você também pode optar por aumentar ou diminuir o armazenamento alocado no ambiente verde. No entanto, uma redução de armazenamento só ocorrerá se o armazenamento alocado de destino for pelo menos 20% maior do que o uso atual do armazenamento. Se você diminuir o armazenamento alocado, o Amazon RDS iniciará uma atualização da configuração de armazenamento. Para obter mais informações, consulte the section called “Atualizar a configuração do armazenamento”.

Se a instância de banco de dados azul usar armazenamento magnético, você deverá alterar a instância de banco de dados verde para um tipo de armazenamento de uso geral ou de IOPS provisionadas para aumentar ou diminuir o armazenamento alocado.

Utilizar gravações otimizadas pelo RDS

Você pode usar implantação azul/verde para fazer a atualização para uma classe de instância de banco de dados compatível com gravações otimizadas para RDS. Você só pode habilitar gravações

otimizadas para RDS em um banco de dados criado com uma classe de instância de banco de dados compatível. Portanto, essa opção cria um banco de dados verde que usa uma classe de instância de banco de dados compatível, o que permite ativar gravações otimizadas para RDS na instância de banco de dados verde.

Se você estiver fazendo uma atualização de uma classe de instância de banco de dados que não seja compatível com gravações otimizadas para RDS para uma que seja, você também deverá atualizar a configuração de armazenamento da instância de banco de dados verde. Para obter mais informações, consulte the section called “Atualizar a configuração do armazenamento”.

Você só pode atualizar a classe de instância de banco de dados da instância de banco de dados verde primária. Por padrão, as réplicas de leitura no ambiente verde herdam as configurações da instância de banco de dados do ambiente azul. Depois que o ambiente verde for criado com sucesso, você deverá modificar manualmente a classe de instância de banco de dados das réplicas de leitura no ambiente verde.

Algumas atualizações de classe de instância não são suportadas, dependendo da versão do mecanismo e da classe da instância de banco de dados azul. Para ter mais informações sobre classes de instância de banco de dados, consulte the section called “Classes da instância de banco de dados”.

Atualizar a configuração do armazenamento

Se seu banco de dados azul não estiver na configuração de armazenamento mais recente, o RDS poderá migrar a instância de banco de dados verde da configuração de armazenamento mais antiga (sistema de arquivos de 32 bits) para a configuração preferencial. Você pode usar a implantação azul/verde do RDS para superar as limitações de escalabilidade no armazenamento e no tamanho do arquivo para sistemas de arquivos mais antigos de 32 bits. Além disso, essa configuração alterará a configuração de armazenamento para ser compatível com gravações otimizadas para RDS se a classe de instância de banco de dados especificada for compatível com as gravações otimizadas.

> **Note**
>
> A atualização da configuração de armazenamento é uma operação intensiva de E/S e leva a tempos de criação mais longos para implantações azul/verdes. O processo de atualização de armazenamento é mais rápido se a instância de banco de dados azul usar armazenamento SSD de IOPS provisionadas (io1 ou io2 Block Express) e se você provisionou o ambiente verde com um tamanho de instância de 4xlarge ou maior. As atualizações de armazenamento que envolvem armazenamento SSD de uso geral (gp2)

podem esgotar seu saldo de créditos de E/S, resultando em tempos de upgrade mais longos. Para obter mais informações, consulte the section called “Armazenamento da instância de banco de dados”.
Durante a atualização do armazenamento, a instância de banco de dados verde fica temporariamente indisponível, enquanto a instância de banco de dados azul permanece disponível. A replicação permanece pausada por enquanto. Monitore o armazenamento na instância azul e pense em escalar se o armazenamento atingir 90%, pois a instância verde aumenta automaticamente em 10% após a atualização.

Essa opção só estará disponível se o banco de dados azul não estiver na configuração de armazenamento mais recente ou se você estiver alterando a classe da instância de banco de dados na mesma solicitação. Só é possível atualizar a configuração de armazenamento ao criar inicialmente uma implantação azul/verde.

Carregamento lento e inicialização de armazenamento para implantações azuis/verdes

Quando você cria uma implantação azul/verde, o Amazon RDS cria a instância de banco de dados primária no ambiente verde realizando a restauração a partir de um snapshot de banco de dados. Depois de criada, a instância de banco de dados verde e as réplicas de leitura continuam carregando dados em segundo plano em um processo conhecido como carregamento lento.

O carregamento lento carrega blocos de dados apenas quando as aplicações os solicitam. Se você tentar acessar dados que ainda não foram carregados, o Amazon EBS os recuperará imediatamente do Amazon S3, enquanto os dados restantes continuarão sendo carregados em segundo plano. Para ter mais informações, consulte Snapshots do Amazon EBS.

Para acelerar o desempenho total do volume, o Amazon RDS fornece inicialização de armazenamento, que lê todos os blocos no volume do ambiente verde. O Amazon EBS baixa proativamente blocos do Amazon S3, fornecendo desempenho máximo do volume desde o primeiro uso. A inicialização do armazenamento ocorre inteiramente em segundo plano, garantindo que não haja impacto na disponibilidade da instância de banco de dados ou nas atividades contínuas, como patches ou atualizações.

A inicialização do armazenamento está disponível somente para instâncias em implantações azuis/verdes com tipos de volume `gp2`, `gp3`, `io1` e `io2`. Ela é compatível com todas as classes de instância, exceto as famílias t3 e t4. Se você modificar uma instância de banco de dados verde

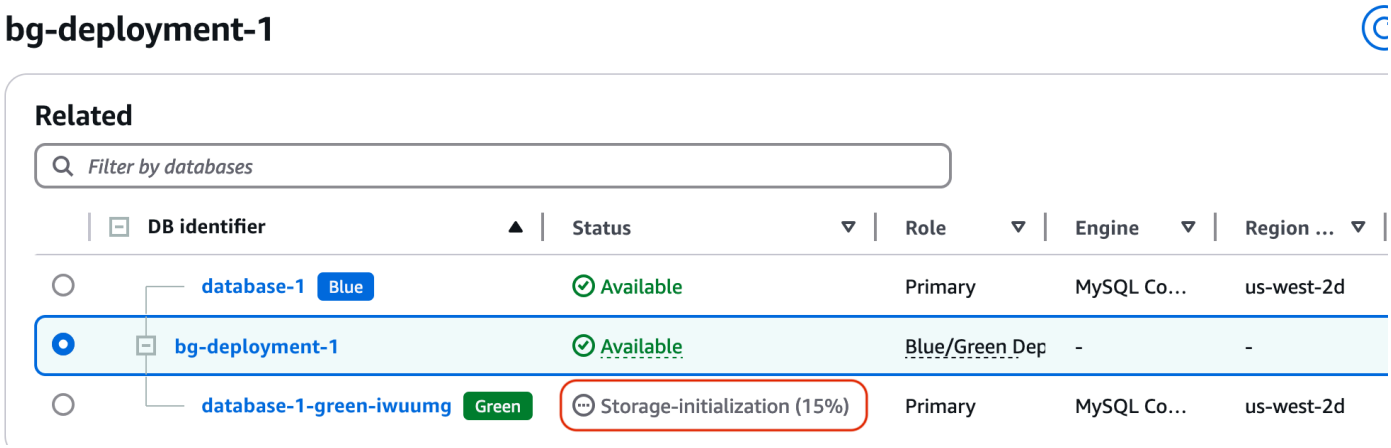
em uma implantação single-AZ para uma implantação de instância de banco de dados multi-AZ, a inicialização do armazenamento incluirá o nó secundário na configuração multi-AZ.

Durante a inicialização do armazenamento, a instância permanece totalmente disponível e utilizável para operações de banco de dados, embora o armazenamento possa não atingir o desempenho total até que a inicialização seja concluída. Enquanto a inicialização do armazenamento está em andamento, o status geral da instância muda para Storage-initialization, e o indicador de progresso reflete o nível mínimo de inicialização em todos os volumes da instância de banco de dados.

Use o console, a AWS CLI ou a API do Amazon RDS para monitorar a inicialização do armazenamento.

Console

No AWS Management Console, você vê o progresso da inicialização do armazenamento com o status da instância de banco de dados.



AWS CLI

Com a AWS CLI, você pode monitorar a inicialização do armazenamento com o comando describe-db-instances. O campo `PercentProgress` na resposta mostra qual porcentagem de dados foi recuperada do Amazon S3.

```
aws rds describe-db-instances --db-instance-identifier my-db-instance

{
    "DBInstances": [
        {
            "DBInstanceIdentifier": "my-db-instance",
            "DBInstanceClass": "db.m5.2xlarge",
            "Engine": "postgres",
```

```
            "DBInstanceStatus": "storage-initialization",
            ...
            "PercentProgress": "34"
        }
    ]
}
```

Amazon RDS API

Com a API do Amazon RDS, você recupera o status da inicialização do armazenamento chamando a ação DescribeDBInstances.

O indicador de progresso é atualizado à medida que a tarefa de inicialização em segundo plano avança, permitindo que você acompanhe a prontidão do armazenamento antes que a inicialização completa do armazenamento seja concluída. A inicialização do armazenamento permite um desempenho otimizado à medida que sua instância de banco de dados verde se torna totalmente operacional.

Criar uma implantação azul/verde

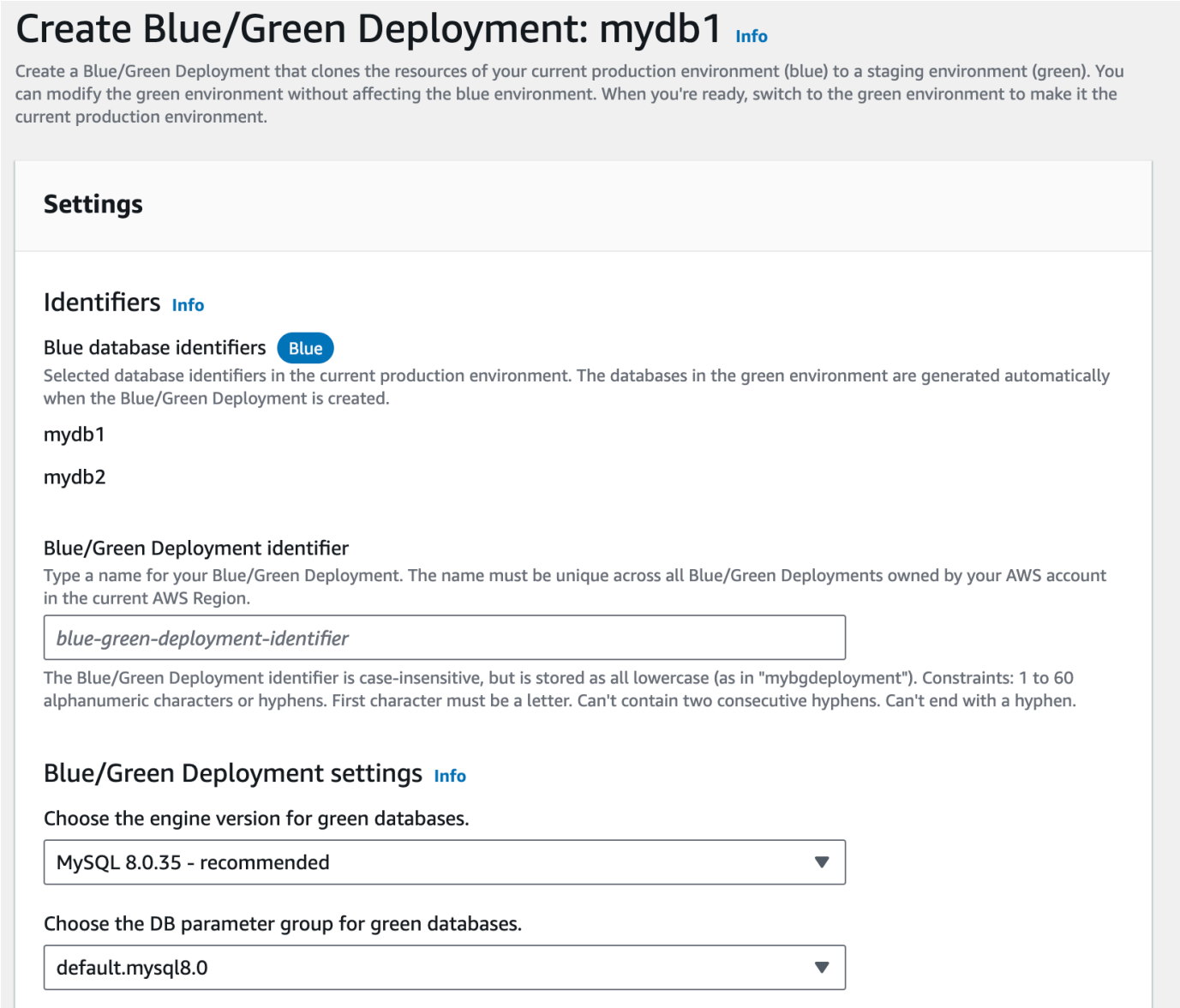
Você pode criar a implantação azul/verde usando o AWS Management Console, a AWS CLI ou a API do RDS.

Console

Como criar uma implantação azul/verde

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, selecione Databases (Bancos de dados) e, depois, a instância de banco de dados que você deseja copiar em um ambiente verde.
3. Selecione Ações, Criar implantação azul/verde.

   A página Criar implantação azul/verde é exibida.



4. Analise os identificadores azuis do banco de dados. Eles devem corresponder às instâncias de banco de dados que você espera no ambiente azul. Caso contrário, selecione Cancel (Cancelar).
5. Em Nome da implantação azul/verde, insira um nome para sua implantação azul/verde.
6. Nas seções restantes, especifique as configurações do ambiente verde. Para obter informações sobre cada configuração, consulte the section called “Configurações disponíveis”.

   Você pode fazer outras modificações nos bancos de dados no ambiente verde após sua implantação.
7. Escolha Criar.

AWS CLI

Para criar uma implantação azul/verde usando a AWS CLI, utilize o comando create-blue-green-deployment. Para obter informações sobre todas as opções disponíveis, consulte the section called "Configurações disponíveis".

Example

Para Linux, macOS ou Unix:

```
aws rds create-blue-green-deployment \
    --blue-green-deployment-name my-blue-green-deployment \
    --source arn:aws:rds:us-east-2:123456789012:db:mydb1 \
    --target-engine-version 8.0.31 \
    --target-db-parameter-group-name mydbparametergroup
```

Para Windows:

```
aws rds create-blue-green-deployment ^
    --blue-green-deployment-name my-blue-green-deployment ^
    --source arn:aws:rds:us-east-2:123456789012:db:mydb1 ^
    --target-engine-version 8.0.31 ^
    --target-db-parameter-group-name mydbparametergroup
```

API do RDS

Para criar uma implantação azul/verde usando a API do Amazon RDS, use a operação `CreateBlueGreenDeployment`. Para ter mais informações sobre cada opção, consulte the section called "Configurações disponíveis".

Configurações para criar implantações azuis/verdes

A tabela a seguir explica as configurações que você pode escolher ao criar uma implantação azul/verde. Consulte mais informações sobre as opções da AWS CLI em create-blue-green-deployment. Consulte mais informações sobre os parâmetros da API do RDS em CreateBlueGreenDeployment.

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS
Armazenamento alocado	O valor de armazenamento a ser alocado para a sua instância	Opção da CLI:

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS

	de banco de dados verde (em gigabytes). Você pode optar por aumentar ou diminuir o armazenamento alocado.
	
	Se a instância de banco de dados azul usar armazenamento magnético (`standard`), você deverá alterar a instância de banco de dados verde para um tipo de armazenamento de uso geral ou de IOPS provisionadas para modificar o armazenamento alocado no ambiente verde.
	
	Para obter mais informações, consulte Armazenamento de instâncias de banco de dados do Amazon RDS.
		`--target-allocated-storage`
		
		Parâmetro da API:
		
		`TargetAllocatedStorage`

Identificador de implantação azul/verde	Um nome para a implantação azul/verde.	Opção da CLI:
		
		`--blue-green-deployment-name`
		
		Parâmetro da API:
		
		`BlueGreenDeploymentName`

Identificador de banco de dados azul	O identificador da instância que você deseja copiar no ambiente verde. Ao usar a CLI ou a API, especifique o nome do recurso da Amazon (ARN) da instância.	Opção da CLI:
		
		`--source`
		
		Parâmetro da API:
		
		`Source`

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS

Grupo de parâmetros do de banco de dados para bancos de dados verdes	Um grupo de parâmetros para associar aos bancos de dados no ambiente verde.	Opção da CLI:
		
		`--target-db-parameter-group-name`
		
		`--target-db-cluster-parameter-group-name`
		
		Parâmetro da API:
		
		`TargetDBParameterGroupName`
		
		`TargetDBClusterParameterGroupName`

Habilitar Gravações otimizadas para banco de dados verdes	Habilite as Gravações otimizadas pelo RDS na instância de banco de dados primária verde. Para obter mais informações, consulte the section called “Utilizar gravações otimizadas pelo RDS”. Se você estiver mudando de uma classe de instância de banco de dados que não suporta gravações otimizadas para uma que oferece, você também precisa realizar uma atualização da configuração de armazenamento. Para obter mais informações, consulte the section called “Atualizar a configuração do armazenamento”.	Para a CLI e a API, quando se especific a uma classe de instância de banco de dados de destino que permita Gravações otimizadas pelo RDS, ela é habilitada automaticamente na instância de banco de dados primária verde.

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS

Versão do mecanismo para bancos de dados verdes	Faça upgrade dos bancos de dados no ambiente verde para a versão especificada do mecanismo de banco de dados.
	
	Se não for especificada, cada banco de dados no ambiente verde será criado com a mesma versão do mecanismo da instância de banco de dados correspondente no ambiente azul.
	
	Se você escolher um uma instância de banco de dados do RDS para PostgreSQL para uma instância que usa replicação lógica, analise e reconheça as limitações da replicação lógica. Para obter mais informações, consulte the section called “Limitações específicas de replicação lógica para implantações azuis/verdes”.
		Opção da CLI:
		
		`--target-engine-version`
		
		Parâmetro da API do RDS:
		
		`TargetEngineVersion`

Classe de instância de banco de dados verde	A capacidade de computação e memória de cada instância de banco de dados no ambiente verde; por exemplo, `db.m5d.xlarge` .
	
	Essa opção só fica visível quando você habilita as Gravações otimizadas pelo RDS para o banco de dados verde.
		Opção da CLI:
		
		`--target-db-instance-class`
		
		Parâmetro da API do RDS:
		
		`TargetDBInstanceClass`

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS

IOPS provisionadas	A quantidade de operações de entrada e saída por segundo (IOPS) provisionadas a serem inicialmente alocadas ao banco de dados verde.
	
	Esse valor se aplica somente à instância de banco de dados primária verde, não às réplicas verdes.
		Opção da CLI:
		
		`--target-iops`
		
		Parâmetro da API do RDS:
		
		`TargetIops`

Upgrade da configuração de armazenamento	Escolha se deseja fazer upgrade da configuração do sistema de arquivos de armazenamento. Se você habilitar essa definição, o RDS migrará o banco de dados verde do antigo sistema de arquivos de armazenamento para a configuração preferencial.
	
	Essa opção só estará disponível se o banco de dados azul não estiver na configuração de armazenamento mais recente ou se você estiver ativando gravações otimizadas para RDS na mesma solicitação. Só é possível atualizar a configuração de armazenamento ao criar inicialmente uma implantação azul/verde.
	
	Para obter mais informações, consulte the section called “Atualizando o sistema de arquivos de armazenamento”.
		Opção da CLI:
		
		`--upgrade-target-storage-config`
		
		Parâmetro da API do RDS:
		
		`UpgradeTargetStorageConfig`

Configuração do console	Descrição da configuração	Opção da CLI e parâmetro da API do RDS

Storage throughput (Throughput de armazenamento)	O valor do throughput de armazenamento para o banco de dados verde. Essa configuração ficará visível somente se você escolher o SSD de uso geral (gp3) como o tipo de armazenamento.
	
	Esse valor se aplica somente à instância de banco de dados primária verde, não às réplicas verdes.
	
	Para obter mais informações, consulte Armazenamento gp3 (recomendado).
		Opção da CLI:
		
		`--target-storage-throughput`
		
		Parâmetro da API do RDS:
		
		`TargetStorageThroughput`

Tipo de armazenamento	O tipo de armazenamento para o banco de dados verde. Somente os seguintes tipos de armazenamento são compatíveis:
	
	• SSD de uso geral (gp2)
	• SSD de uso geral (gp3)
	• IOPS provisionadas (io1)
	• SSD de IOPS provisionadas (io2)
	
	Esse valor se aplica somente à instância de banco de dados primária verde, não às réplicas verdes.
	
	Para obter mais informações, consulte Tipos de armazenamento do Amazon RDS.
		Opção da CLI:
		
		`--target-storage-type`
		
		Parâmetro da API do RDS:
		
		`TargetStorageType`

Visualizar uma implantação azul/verde no Amazon RDS

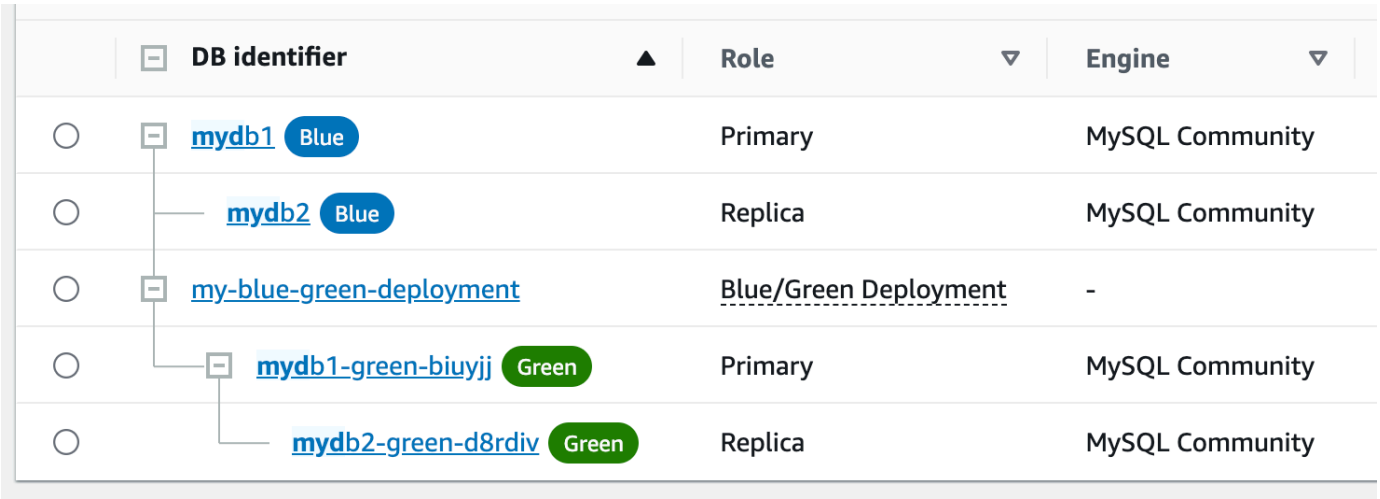
Você pode visualizar os detalhes de uma implantação azul/verde usando o AWS Management Console, a AWS CLI ou a API do RDS.

Você também pode visualizar e assinar eventos para obter informações sobre uma implantação azul/verde. Para ter mais informações, consulte Eventos de implantação azul/verde.

Console

Como visualizar os detalhes sobre uma implantação azul/verde

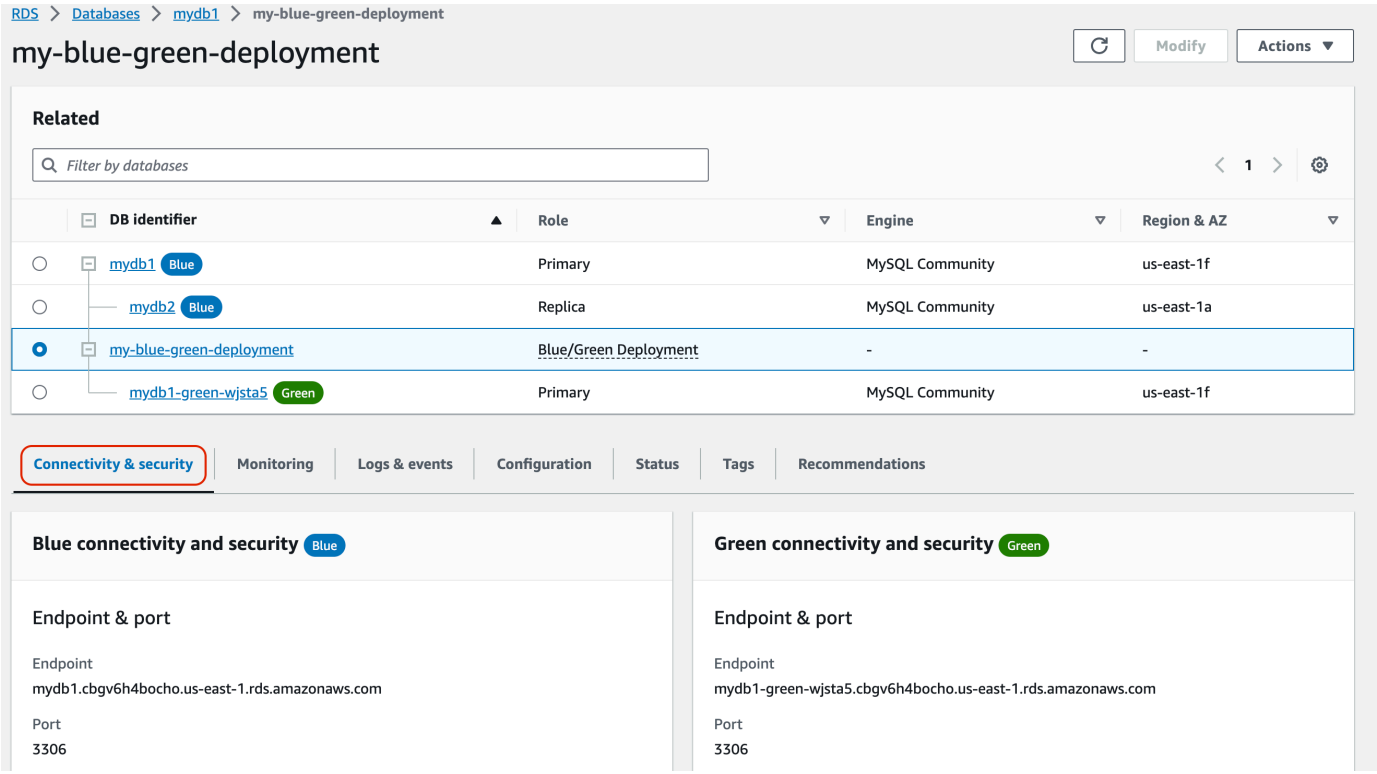
1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, selecione Databases (Bancos de dados) e, depois, localize a implantação azul/verde na lista.



   O valor de Role (Função) para a implantação azul/verde é Blue/Green Deployment (Implantação azul/verde).
3. Selecione o nome da implantação azul/verde que você deseja visualizar para exibir seus detalhes.

   Cada guia tem uma seção para a implantação azul e uma seção para a implantação verde. Por exemplo, na guia Configuração, a versão do mecanismo de banco de dados pode ser diferente no ambiente azul e no ambiente verde se você estiver atualizando a versão do mecanismo de banco de dados no ambiente verde.

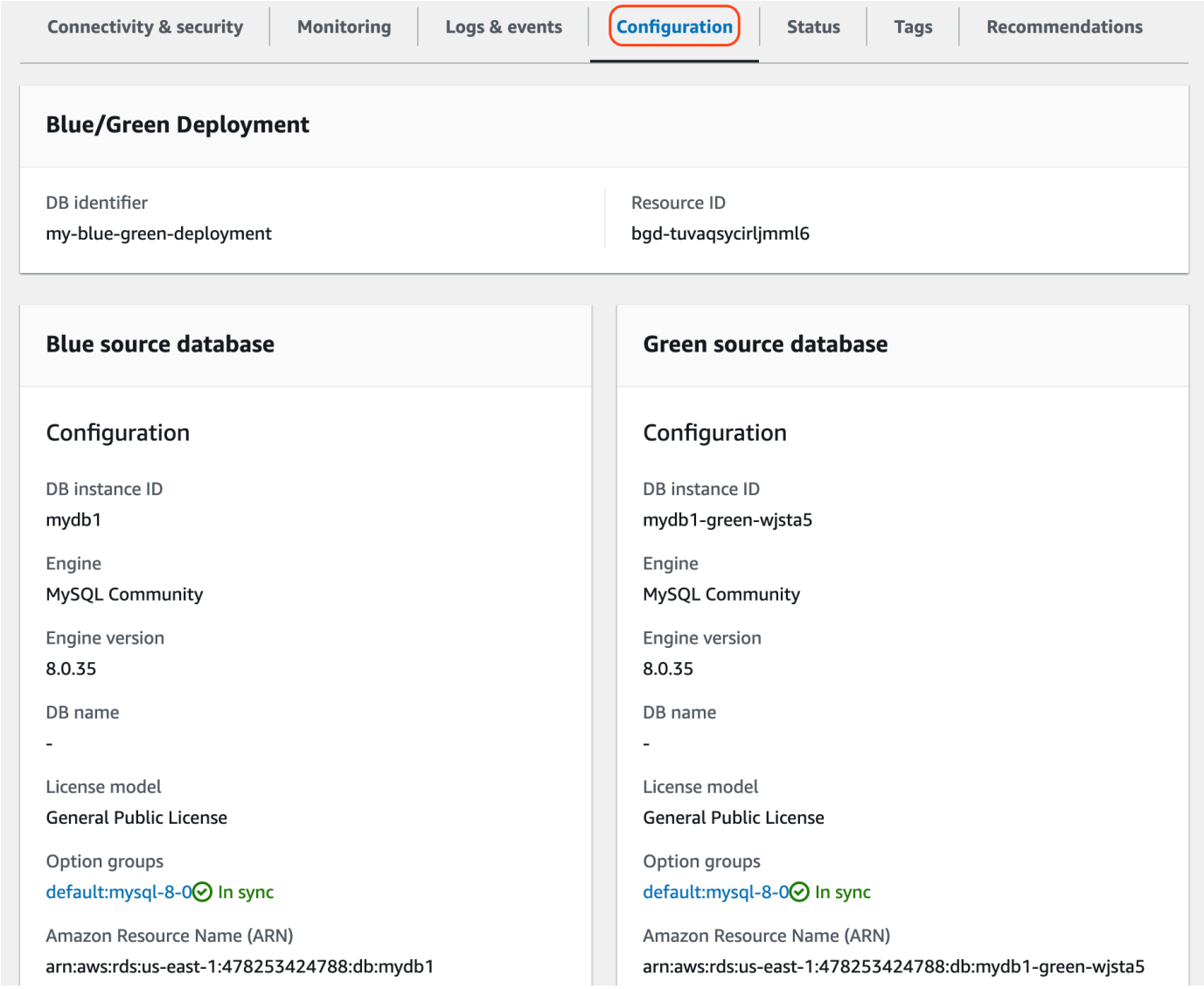
   A imagem a seguir mostra um exemplo da guia Conectividade e segurança.



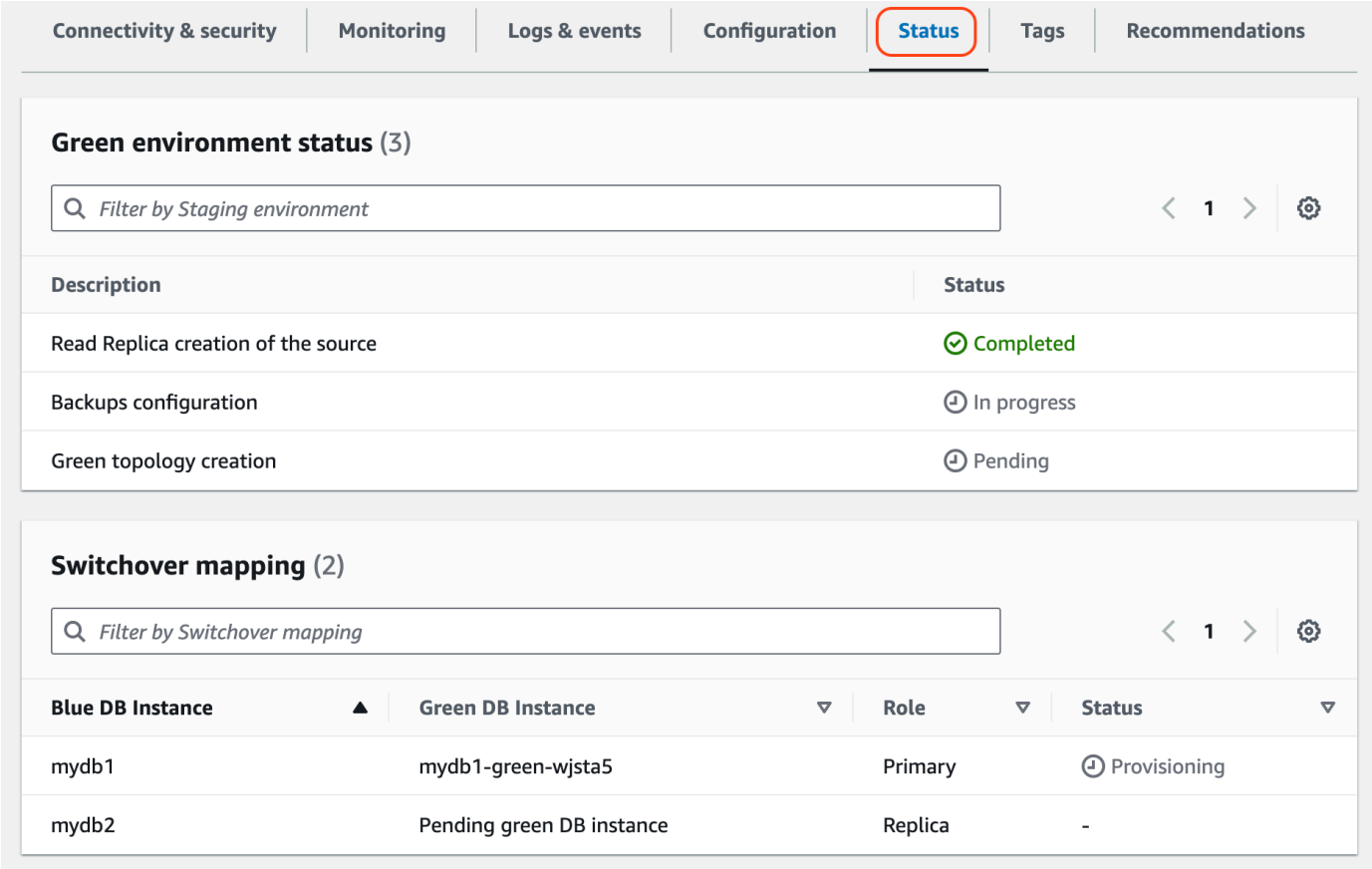
A guia Conectividade e segurança também inclui uma seção chamada Replicação, que mostra o estado atual da replicação e o atraso da réplica entre os ambientes azul e verde. Se o estado de replicação for `Replicating`, a implantação azul/verde está sendo replicada com êxito.

Para implantações azuis/verdes do RDS para PostgreSQL que usam replicação lógica, o estado da replicação pode mudar para `Replication degraded` se você fizer alterações de DDL incompatíveis ou de objetos grandes no ambiente azul. Para ter mais informações, consulte the section called “Limitações específicas de replicação lógica para implantações azuis/verdes”.

A imagem a seguir mostra um exemplo da guia Configuração.



A seguinte imagem mostra um exemplo da guia Status:



AWS CLI

Para ver os detalhes sobre uma implantação azul/verde usando a AWS CLI, use o comando describe-blue-green-deployments.

Example Veja os detalhes sobre uma implantação azul/verde filtrando por seu nome

Ao usar o comando describe-blue-green-deployments, você pode filtrar no `--blue-green-deployment-name`.

O exemplo a seguir mostra os detalhes de uma implantação azul/verde chamada *`my-blue-green-deployment`*.

```
aws rds describe-blue-green-deployments \
  --filters Name=blue-green-deployment-name,Values=my-blue-green-deployment
```

Example Visualizar os detalhes sobre uma implantação azul/verde especificando seu identificador

Ao usar o comando describe-blue-green-deployments, é possível especificar a opção `--blue-green-deployment-identifier`.

O exemplo a seguir mostra os detalhes de uma implantação azul/verde com o identificador *bgd-1234567890abcdef*.

```
aws rds describe-blue-green-deployments \
  --blue-green-deployment-identifier bgd-1234567890abcdef
```

API do RDS

Para ver os detalhes sobre uma implantação azul/verde usando a API do Amazon RDS, use a operação `DescribeBlueGreenDeployments` e especifique o `BlueGreenDeploymentIdentifier`.

Mudar uma implantação azul/verde no Amazon RDS

Uma transição transforma o ambiente verde no novo ambiente de produção. Quando a instância de banco de dados verde tem réplicas de leitura, elas também são transicionadas. Antes de você fazer a transição, o tráfego de produção é roteado para a instância de banco de dados e lê as réplicas no ambiente azul. Antes de você fazer a transição, o tráfego de produção é roteado para a instância de banco de dados e lê as réplicas no ambiente verde.

Mudar para uma implantação azul/verde não é o mesmo que promover a instância de banco de dados verde na implantação azul/verde. Se você promover manualmente a instância de banco de dados escolhendo Promover no menu Ações, a replicação entre os ambientes azul e verde será interrompida e a implantação azul/verde entrará no estado de Configuração inválida.

Tópicos

- Tempo limite de transição
- Barreiras de proteção de transição
- Ações de transição
- Práticas recomendadas de transição
- Verificar as métricas do CloudWatch antes da transição
- Monitorar o atraso da réplica antes da transição

- Realizar a transição de uma implantação azul/verde
- Após a transição

Tempo limite de transição

Você pode especificar um tempo limite de transição entre 30 segundos e 3.600 segundos (uma hora). Se a transição demorar mais do que o especificado, todas as alterações serão revertidas e nenhuma alteração será feita em nenhum dos ambientes. O limite de tempo padrão é 300 segundos (cinco minutos).

Barreiras de proteção de transição

Quando você inicia uma transição, o Amazon RDS executa algumas verificações básicas para testar a prontidão dos ambientes azul e verde para a transição. Essas verificações são conhecidas como barreiras de proteção de transição. Essas barreiras evitarão uma transição se os ambientes não estiverem prontos para isso. Portanto, elas evitam tempo de inatividade mais longo do que o esperado e evitam a perda de dados entre os ambientes azul e verde que pode ocorrer se a transição for iniciada.

O Amazon RDS executa as seguintes verificações de barreira de proteção no ambiente verde:

- Integridade da replicação: confira se o status de replicação da instância de banco de dados primária do cluster de banco de dados verde é íntegro. A instância de banco de dados primária verde é uma réplica da instância de banco de dados primária azul.
- Atraso na replicação: confira se o atraso da réplica da instância de banco de dados primária está nos limites permitidos para a transição. Os limites permitidos são baseados no tempo limite especificado. O atraso da réplica indica até que ponto a instância de banco de dados primária verde está atrás de sua instância de banco de dados primária azul. Para ter mais informações, consulte the section called “Monitorar o atraso da réplica antes da transição”.
- Gravações ativas: certifique-se de que não haja gravações ativas na instância de banco de dados primária.

O Amazon RDS executa as seguintes verificações de barreira de proteção no ambiente azul:

- Replicação externa: para o RDS for PostgreSQL, garante que o ambiente azul não seja uma fonte lógica autogerenciada (publicador) nem uma réplica (assinante). Se ele for, recomendamos que você elimine os slots de replicação autogerenciados e as assinaturas em todos os bancos de

dados no ambiente azul, continue com a transição e, depois, recrie-os para retomar a replicação. Para o RDS para MySQL e RDS para MariaDB, verifica se o banco de dados azul não é uma réplica externa de log binário. Se for, verifique se ele não está se replicando ativamente.

- Gravações ativas de longa duração: verifica se não há gravações ativas de longa duração na instância de banco de dados primária azul, pois elas podem aumentar o atraso da réplica.
- Instruções DDL de longa duração: verifica se não há instruções DDL de longa duração na instância de banco de dados primária azul, pois elas podem aumentar o atraso da réplica.
- Alterações não compatíveis do PostgreSQL: para RDS para PostgreSQL que usam replicação lógica, verifica se não há nenhuma alteração de DDL e se nenhuma adição ou modificação de objetos grandes foi realizada no ambiente azul. Para ter mais informações, consulte the section called “Limitações específicas de replicação lógica para implantações azuis/verdes”.

  Se o Amazon RDS detectar alterações não compatíveis do PostgreSQL, ele alterará o estado de replicação para `Replication degraded` e notificará você de que a transição não está disponível para a implantação azul/verde. Para continuar com a transição, recomendamos que você exclua e recrie a implantação azul/verde e todos os bancos de dados verdes. Para fazer isso, escolha Ações, Excluir com bancos de dados verdes.

Ações de transição

Quando você alterna uma implantação azul/verde, o RDS realiza as seguintes ações:

1. Executa verificações de barreira de proteção para verificar se os ambientes azul e verde estão prontos para a transição.
2. Interrompe novas operações de gravação na instância de banco de dados primária nos dois ambientes.
3. Descarta conexões com as instâncias de banco de dados em ambos os ambientes e não permite novas conexões.
4. Espera que a replicação alcance o ambiente verde para que este esteja em sincronia com o ambiente azul.
5. Renomeia as instâncias de banco de dados nos dois ambientes.

   O RDS renomeia as instâncias de banco de dados no ambiente verde para corresponder às instâncias de banco de dados no ambiente azul. Por exemplo, suponha que o nome de uma instância de banco de dados no ambiente azul seja `mydb`. Suponha também que o nome da instância de banco de dados correspondente no ambiente verde seja `mydb-green-abc123`.

Durante a transição, o nome da instância de banco de dados no ambiente verde é alterado para `mydb`.

O RDS renomeia as instâncias de banco de dados no ambiente azul anexando `-old`*n* ao nome atual, em que *n* é um número. Por exemplo, suponha que o nome de uma instância de banco de dados no ambiente azul seja `mydb`. Após a transição, o nome da instância de banco de dados pode ser `mydb-old1`.

O RDS também renomeia os endpoints no ambiente verde para corresponder aos endpoints correspondentes no ambiente azul, para que as alterações na aplicação não sejam necessárias.

6. Permite conexões com bancos de dados nos dois ambientes.
7. Permite operações de gravação na instância de banco de dados primária no novo ambiente de produção.

   Após a transição, a instância de banco de dados primária de produção anterior só permite operações de leitura até que você defina o parâmetro `read_only` (para RDS para MySQL) ou o parâmetro `default_transaction_read_only` (para RDS para PostgreSQL) como `0` e reinicialize a instância de banco de dados.

Você pode monitorar o status de uma transição usando o Amazon EventBridge. Para ter mais informações, consulte the section called “Eventos de implantação azul/verde”.

Se você tiver tags configuradas no ambiente azul, essas tags serão copiadas no novo ambiente de produção durante a transição. Para ter mais informações sobre tags, consulte Marcar recursos do Amazon RDS.

Se a transição começar e parar antes de terminar por qualquer motivo, todas as alterações serão revertidas e nenhuma alteração será feita em nenhum dos ambientes.

Práticas recomendadas de transição

Antes de fazer a transição, é altamente recomendável que você siga as práticas recomendadas concluindo as seguintes tarefas:

- Teste minuciosamente os recursos no ambiente verde. Eles devem funcionar de forma adequada e eficiente.
- Monitore as métricas relevantes do Amazon CloudWatch. Para ter mais informações, consulte the section called “Verificar as métricas do CloudWatch antes da transição”.

- Identifique o melhor momento para a transição.

  Durante a transição, as gravações são cortadas dos bancos de dados nos dois ambientes. Identifique um momento em que o tráfego é o menor em seu ambiente de produção. Transações de longa duração, como DDLs ativas, podem aumentar seu tempo de transição, ocasionando maior tempo de inatividade para suas workloads de produção.

  Se houver um grande número de conexões em suas instâncias de banco de dados, considere reduzi-las manualmente até a quantidade mínima necessária para sua aplicação antes de realizar a transição da implantação azul/verde. Uma maneira de fazer isso é criar um script que monitore o status da implantação azul/verde e comece a limpar as conexões quando detectar que o status mudou para `SWITCHOVER_IN_PROGRESS`.
- As instâncias nos dois ambientes devem estar no estado `Available`.
- A instância de banco de dados primária no ambiente verde devem estar funcionando e sendo replicada .
- Garanta que suas configurações de rede e cliente não aumentem o tempo de vida útil (TTL) do cache DNS além de cinco segundos, que é o padrão para zonas DNS do RDS.
  Caso contrário, as aplicações continuarão a enviar tráfego de gravação ao ambiente azul após transição.
- O carregamento de dados deve ser concluído antes da transição. Para ter mais informações, consulte the section called “Carregamento lento e inicialização do armazenamento”.
- Para implantações azuis/verdes do RDS para PostgreSQL que usam replicação lógica, faça o seguinte:
  - Analise as limitações de replicação lógica e realize todas as ações necessárias antes da transição. Para ter mais informações, consulte the section called “Limitações específicas de replicação lógica para implantações azuis/verdes”.
  - Execute a operação `ANALYZE` para atualizar a tabela `pg_statistics`. Isso reduz o risco de problemas de desempenho após a transição.

> **Note**
>
> Durante uma transição, você não pode modificar nenhuma instância de banco de dados incluída na transição .

Verificar as métricas do CloudWatch antes da transição

Antes de realizar a transição de uma implantação azul/verde, recomendamos que verifique os valore das métrica a seguir no Amazon CloudWatch.

- `DatabaseConnections`: use esta métrica para estimar o nível de atividade na implantação azul/verde; certifique-se de que o valor esteja em um nível aceitável para sua implantação antes da transição. Se o recurso Insights de Performance estiver ativado, `DBLoad` será uma métrica mais precisa.

Para ter mais informações, consulte the section called “Métricas do CloudWatch para RDS”.

Monitorar o atraso da réplica antes da transição

Antes de realizar a transição de uma implantação azul/verde, verifique se o atraso de réplica é próximo de zero para reduzir o tempo de inatividade.

RDS para MySQL e RDS para MariaDB

Para implantações azuis/verdes do MySQL e MariaDB, verifique a métrica `ReplicaLag` do CloudWatch no ambiente verde para identificar o atraso atual da réplica. Para ter mais informações, consulte the section called “Diagnosticar e resolver atrasos entre réplicas de leitura”.

RDS para PostgreSQL

Para implantações azuis/verdes do PostgreSQL que usam replicação física, consulte the section called “Monitoração e ajuste do processo de replicação” para obter instruções sobre como identificar o atraso atual da réplica.

Para implantações azuis/verdes do PostgreSQL que usam replicação lógica, verifique a métrica `OldestReplicationSlotLag` do CloudWatch no ambiente azul para identificar o atraso atual da réplica. Para ter mais informações, consulte the section called “Métricas específicas da instância do Amazon CloudWatch para Amazon RDS”.

Além disso, você pode executar a consulta SQL a seguir no ambiente azul:

```
SELECT slot_name,
       confirmed_flush_lsn as flushed,
```

```
       pg_current_wal_lsn(),
       (pg_current_wal_lsn() - confirmed_flush_lsn) AS lsn_distance
FROM pg_catalog.pg_replication_slots
WHERE slot_type = 'logical';

slot_name        |    flushed    | pg_current_wal_lsn | lsn_distance
-----------------+---------------+--------------------+------------
logical_replica1 | 47D97/CF32980 | 47D97/CF3BAC8      | 37192
```

O `confirmed_flush_lsn` representa o número de sequência de log (LSN) mais recente enviado à réplica. O `pg_current_wal_lsn` representa onde o banco de dados está agora. Um `lsn_distance` de 0 significa que a réplica foi capturada.

Para obter uma explicação de quando as implantações azuis/verdes usam replicação física versus replicação lógica, consulte ???.

Realizar a transição de uma implantação azul/verde

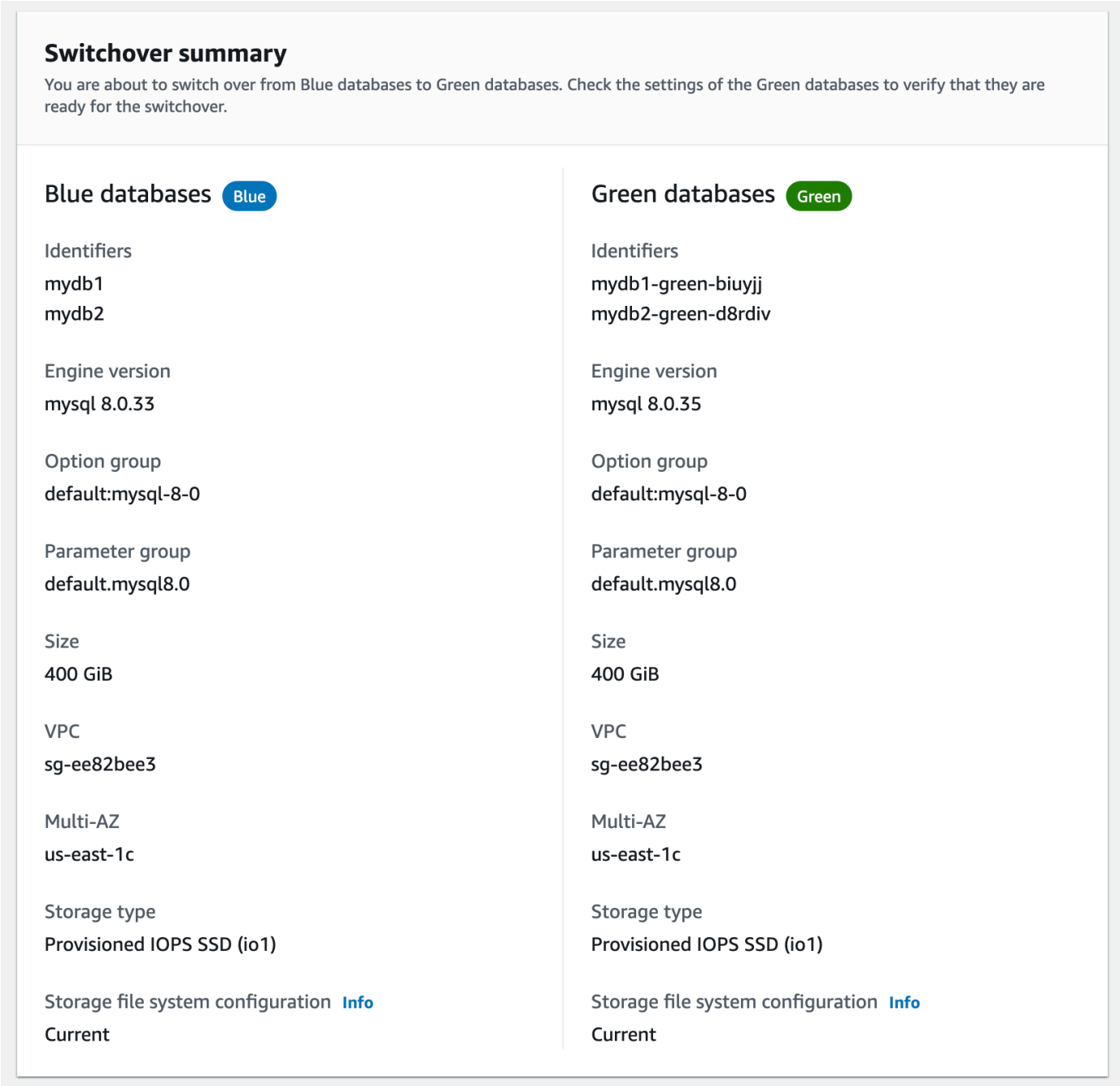
Você pode fazer a transição de uma implantação azul/verde usando o AWS Management Console, a AWS CLI ou a API do RDS.

Console

Como realizar a transição de uma implantação azul/verde

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, selecione Databases (Bancos de dados) e, depois, selecione a implantação azul/verde da qual você deseja realizar a transição.
3. Para Actions (Ações), selecione Switch over (Realizar transição).

   A página Switch over (Realizar transição) é exibida.



4. Na página Switch over (Realizar transição), revise o resumo da transição. Os recursos nos dois ambientes devem corresponder ao que você espera. Caso contrário, selecione Cancel (Cancelar).
5. Em Timeout (Tempo limite), insira o limite de tempo para a transição.
6. Se seu , analise e confirme as recomendações de pré-transição. Para ter mais informações, consulte the section called “Limitações específicas de replicação lógica para implantações azuis/verdes”.

7. Selecione Switch Role (Realizar transição).

AWS CLI

Para realizar a transição de uma implantação azul/verde usando a AWS CLI, utilize o comando switchover-blue-green-deployment com as seguintes opções:

- `--blue-green-deployment-identifier`: especifique o ID do recurso da implantação azul/verde.
- `--switchover-timeout`: especifique o limite de tempo para a transição, em segundos. O padrão é 300.

Example Fazer a transição de uma implantação azul/verde

Para Linux, macOS ou Unix:

```
aws rds switchover-blue-green-deployment \
    --blue-green-deployment-identifier bgd-1234567890abcdef \
    --switchover-timeout 600
```

Para Windows:

```
aws rds switchover-blue-green-deployment ^
    --blue-green-deployment-identifier bgd-1234567890abcdef ^
    --switchover-timeout 600
```

API do RDS

Para realizar a transição de uma implantação azul/verde usando a API do Amazon RDS, use a operação `SwitchoverBlueGreenDeployment` com os seguintes parâmetros:

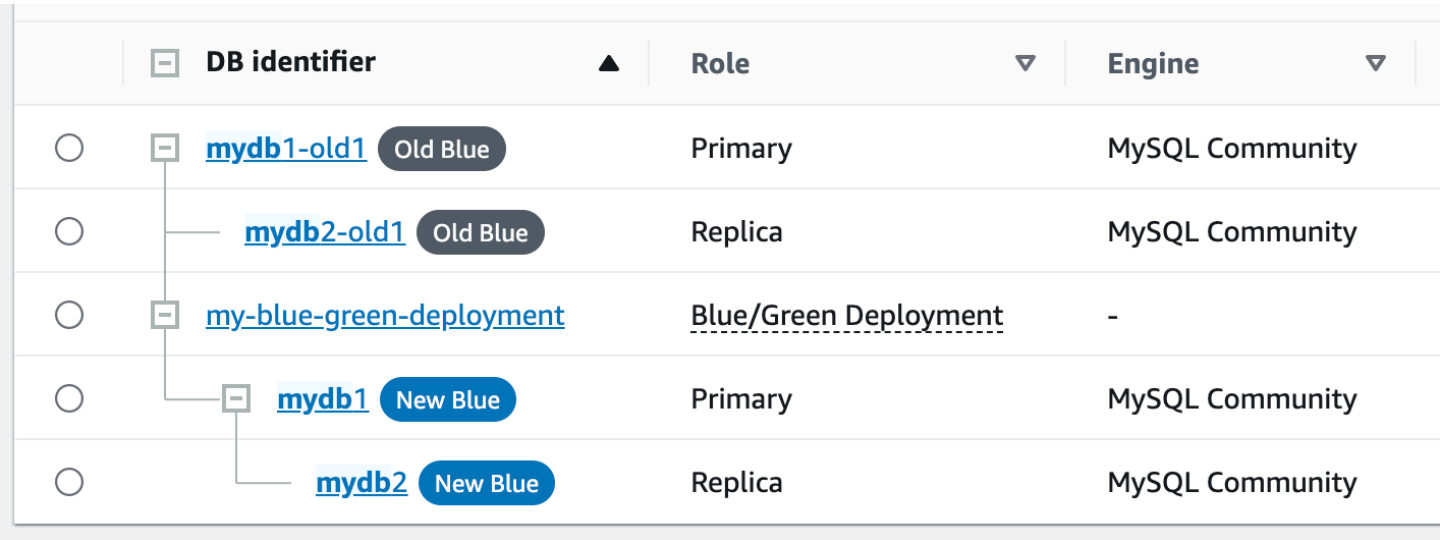
- `BlueGreenDeploymentIdentifier`: especifique o ID do recurso da implantação azul/verde.
- `SwitchoverTimeout`: especifique o limite de tempo para a transição, em segundos. O padrão é 300.

Após a transição

Depois de uma transição, as instâncias de banco de dados no ambiente azul anterior são retidas. Os custos padrão se aplicam a esses recursos. A replicação entre os ambientes azul e verde é interrompida.

O RDS renomeia as instâncias de banco de dados no ambiente azul anexando `-old`*n* ao nome de recurso atual, em que *n* é um número. As instâncias de banco de dados no antigo ambiente azul são somente leitura até que você defina o parâmetro `read_only` (para RDS para MySQL) ou o parâmetro `default_transaction_read_only` (para RDS para PostgreSQL) como `0`. O RDS nomeia as instâncias de banco de dados no ambiente verde `-new`*n*.

Se você excluir o recurso de implantação azul/verde, o RDS reterá os recursos `-old`*n* e `-new`*n*.



Atualizar o nó principal para consumidores

O RDS oferece réplicas de leitura totalmente gerenciadas. No entanto, ele também oferece a opção de configurar réplicas autogerenciadas, também conhecidas como réplicas externas. As réplicas externas possibilitam o uso de recursos de terceiros como destinos de replicação.

Depois de fazer a transição de uma implantação azul/verde do RDS para MariaDB ou do RDS para MySQL , se a instância de banco de dados tiver alguma réplica externa ou consumidores de log binário antes da transição, será necessário atualizar o nó principal após a transição para manter a continuidade da replicação.

Como atualizar o nó pai

1. Após a transição, a instância de banco de dados de que estava anteriormente no ambiente verde emite um evento que contém o nome do arquivo de log principal e a posição do log principal. Para localizar o evento, acesse o console do RDS e escolha Eventos no painel de navegação esquerdo.
2. Filtre por eventos em que a origem é o nome da antiga instância de banco de dados verde do , antes da transição.
3. Localize o evento que contém as coordenadas de logs binários. A mensagem do evento é semelhante a: `Binary log coordinates in green environment after switchover: file mysql-bin-changelog.000003 and position 40134574`.
4. Garanta que o consumidor ou a réplica tenha aplicado todos os logs binários do antigo ambiente azul. Depois, use as coordenadas de logs binários fornecidas para retomar a replicação nos consumidores. Por exemplo, se você estiver executando uma réplica do MySQL no EC2, poderá usar o seguinte comando:

   MySQL 8.0.22 e versões principais e secundárias anteriores

   ```
   CHANGE MASTER TO MASTER_HOST='{new-writer-endpoint}', MASTER_LOG_FILE='mysql-bin-changelog.000003', MASTER_LOG_POS=40134574;
   ```

   MySQL 8.0.23 e versões principais e secundárias posteriores

   ```
   CHANGE REPLICATION SOURCE TO SOURCE_HOST='{new-writer-endpoint}',
    SOURCE_LOG_FILE='mysql-bin-changelog.000003', SOURCE_LOG_POS=40134574;
   ```

Se o consumidor for outra instância de banco de dados do RDS para MariaDB ou do RDS para MariaDB, execute os seguintes procedimentos armazenados na ordem:

1. the section called “mysql.rds_stop_replication”
2. mysql.rds_reset_external_master (para a versão 8.0 e anteriores) ou mysql_rds_reset_external_source (para a versão 8.4 e posteriores)
3. mysql.rds_set_external_master (para a versão 8.0 e anteriores) ou mysql_rds_set_external_source (para a versão 8.4 e posteriores)
4. the section called “mysql.rds_start_replication”

Excluir uma implantação azul/verde no Amazon RDS

Você pode excluir uma implantação azul/verde antes ou depois de realizar a transição.

Quando você exclui uma implantação azul/verde antes de realizar a transição, o Amazon RDS exclui opcionalmente as instâncias de banco de dados no ambiente verde:

- Se você optar por excluir as instâncias de banco de dados no ambiente verde (`--delete-target`), elas deverão ter proteção contra exclusão desativada.
- Se você não excluir as instâncias de banco de dados no ambiente verde (`--no-delete-target`), elas serão retidas, mas elas não farão mais parte de uma implantação azul/verde. Para o RDS para MySQL, a replicação continua entre os ambientes. Para o RDS para PostgreSQL, o ambiente verde é promovido a um ambiente autônomo, então a replicação é interrompida.

A opção de excluir bancos de dados verdes não estará disponível no console depois da transição. Ao excluir uma implantação azul/verde usando a AWS CLI, você não poderá especificar a opção `--delete-target` se o status da implantação for `SWITCHOVER_COMPLETED`.

> **Important**
>
> A exclusão de uma implantação azul/verde não afeta o ambiente azul.

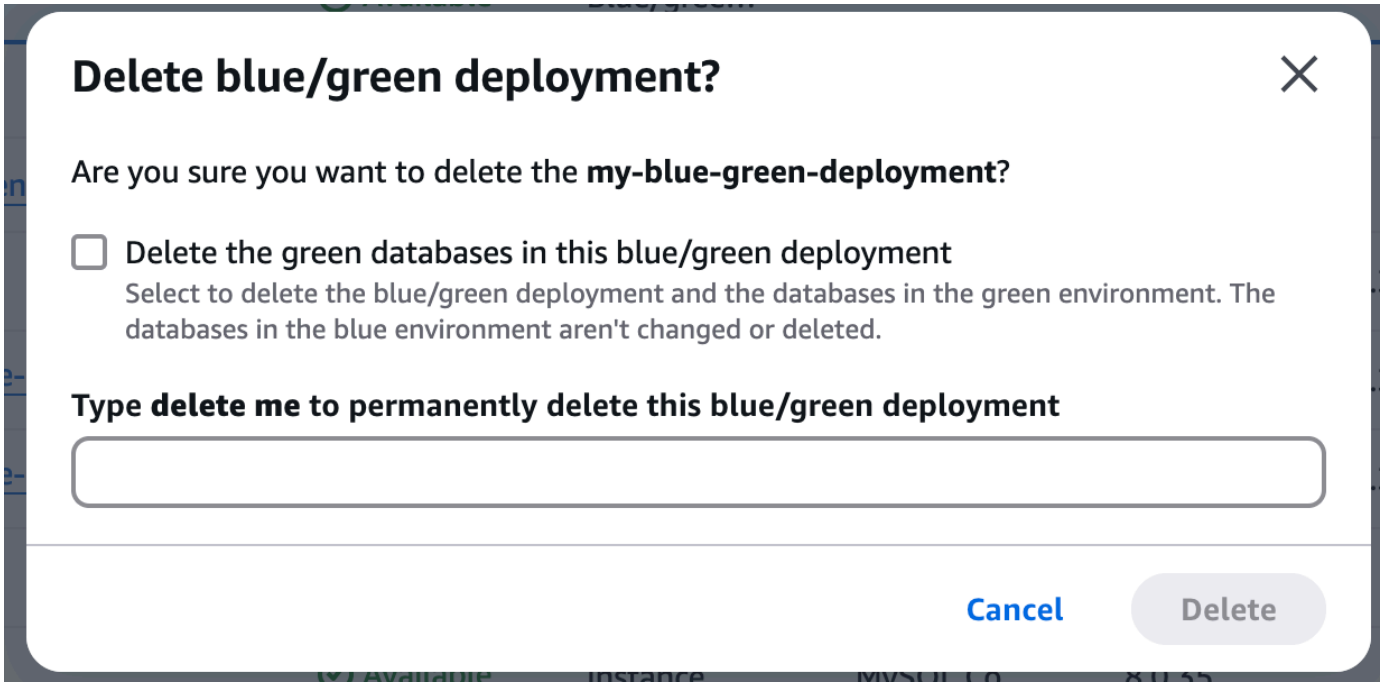
Você pode excluir a implantação azul/verde usando o AWS Management Console, a AWS CLI ou a API do RDS.

Console

Como excluir uma implantação azul/verde

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, selecione Databases (Bancos de dados) e, depois, selecione a implantação azul/verde que você deseja excluir.
3. Em Ações, escolha Excluir.

   A janela Excluir implantação azul/verde? é exibida.



Para excluir os bancos de dados verdes, selecione Excluir os bancos de dados verdes nesta implantação azul/verde.

4. Digite **`delete me`** na caixa.
5. Escolha Excluir.

AWS CLI

Para excluir uma implantação azul/verde usando a AWS CLI, use o comando delete-blue-green-deployment com as seguintes opções:

- `--blue-green-deployment-identifier`: o ID do recurso da implantação azul/verde a ser excluída.
- `--delete-target`: especifica que as instâncias de banco de dados no ambiente verde sejam excluídas . Você não poderá especificar essa opção se a implantação azul/verde tiver um status de `SWITCHOVER_COMPLETED`.
- `--no-delete-target`: especifica que as instâncias de banco de dados no ambiente verde sejam retidas .

Example Exclua uma implantação azul/verde e as instâncias de banco de dados no ambiente verde

Para Linux, macOS ou Unix:

```
aws rds delete-blue-green-deployment \
    --blue-green-deployment-identifier bgd-1234567890abcdef \
    --delete-target
```

Para Windows:

```
aws rds delete-blue-green-deployment ^
    --blue-green-deployment-identifier bgd-1234567890abcdef ^
    --delete-target
```

Example Exclua uma implantação azul/verde, mas mantenha as instâncias de banco de dados no ambiente verde

Para Linux, macOS ou Unix:

```
aws rds delete-blue-green-deployment \
    --blue-green-deployment-identifier bgd-1234567890abcdef \
    --no-delete-target
```

Para Windows:

```
aws rds delete-blue-green-deployment ^
    --blue-green-deployment-identifier bgd-1234567890abcdef ^
    --no-delete-target
```

API do RDS

Para excluir uma implantação azul/verde usando a API do Amazon RDS, use a operação `DeleteBlueGreenDeployment` com os seguintes parâmetros:

- `BlueGreenDeploymentIdentifier`: o ID do recurso da implantação azul/verde a ser excluída.
- `DeleteTarget`: especifique `TRUE` se deseja excluir as instâncias de banco de dados no ambiente verde ou `FALSE` para mantê-las. Não poderá ser `TRUE` se a implantação azul/verde tiver um status de `SWITCHOVER_COMPLETED`.

Backup, restauração e exportação de dados

Essa seção mostra como fazer backup, restaurar e exportar dados de uma instância de banco de dados ou um cluster de banco de dados multi-AZ do Amazon RDS.

Consulte informações sobre como fazer backup da instância de banco de dados ou um cluster de banco de dados multi-AZ do Amazon RDS nos tópicos a seguir.

- Introdução aos backups
- Gerenciar backups automatizados
- Gerenciar backups manuais

Consulte informações sobre como restaurar a instância de banco de dados ou um cluster de banco de dados multi-AZ do Amazon RDS em Restaurar uma instância de banco de dados.

Consulte informações sobre como copiar, compartilhar ou exportar snapshots de banco de dados nos tópicos a seguir.

- Copiar um snapshot de banco de dados para o Amazon RDS
- Compartilhar um snapshot de banco de dados para o Amazon RDS
- Exportação de dados de snapshot de banco de dados para o Amazon S3 para Amazon RDS

Consulte informações sobre como visualizar backups automáticos gerenciados pelo AWS Backup em Usar o AWS Backup no gerenciamento de backups automatizados para o Amazon RDS.

Introdução aos backups

O Amazon RDS cria e salva backups automáticos da instância de banco de dados ou do cluster de banco de dados multi-AZ durante a janela de backup da instância de banco de dados. O RDS cria um snapshot do volume de armazenamento da instância de banco de dados, fazendo o backup de toda a instância de banco de dados e não apenas dos bancos de dados individuais. O RDS salva os backups automatizados da instância de banco de dados de acordo com o período de retenção de backup especificado. Se necessário, você poderá recuperar a instância de banco de dados para qualquer ponto no tempo durante o período de retenção de backup.

Os backups automatizados seguem estas regras:

- Sua instância de banco de dados deve estar no estado `available` para fazer backups automáticos. Os backups automáticos não ocorrerão enquanto a instância de banco de dados estiver em um estado diferente de `available`; por exemplo, `storage_full`.
- Os backups automatizados não ocorrem enquanto uma cópia de snapshot do banco de dados está em execução na mesma Região da AWS para o mesmo banco de dados.

Também é possível fazer backup da instância de banco de dados manualmente criando um DB snapshot manualmente. Para ter mais informações sobre como criar um snapshot de banco de dados manualmente, consulte Criar um snapshot de banco de dados para uma instância de banco de dados single-AZ para o Amazon RDS.

O primeiro snapshot de uma instância de banco de dados contém dados do banco de dados completo. Os snapshots subsequentes do mesmo banco de dados são incrementais, o que significa que somente os dados que foram alterados depois do snapshot mais recente serão salvos.

Você pode copiar snapshot de banco de dadoss automáticos e manuais, e compartilhar snapshot de banco de dadoss manuais. Para ter mais informações sobre como copiar um snapshot de banco de dados, consulte Copiar um snapshot de banco de dados para o Amazon RDS. Para ter mais informações sobre como compartilhar um snapshot de banco de dados, consulte Compartilhar um snapshot de banco de dados para o Amazon RDS.

Armazenamento de backup

Seu armazenamento de backup do Amazon RDS para cada Região da AWS é composto por backups automáticos e snapshots de banco de dados manuais para essa região. O espaço total do armazenamento de backup é igual à soma do armazenamento de todos os backups nessa região. A

mudança de um DB snapshot para outra região aumenta o armazenamento de backup na região de destino. Os backups são armazenados no Amazon S3.

Para ter mais informações sobre os custos de armazenamento de backup, consulte Definição de preço do Amazon RDS.

Se você optar por reter backups automáticos ao excluir uma instância de banco de dados, os backups automáticos serão salvos para o período de retenção total. Se você não escolher Retain automated backups (Reter backups automatizados) ao excluir uma instância de banco de dados, todos os backups automatizados serão excluídos com a instância de banco de dados. Depois que forem excluídos, os backups automatizados não poderão ser recuperados. Se você preferir que o Amazon RDS crie um snapshot de banco de dados final antes de excluir sua instância de banco de dados, poderá usá-lo para recuperar sua instância de banco de dados. Opcionalmente, você pode usar um snapshot manual criado anteriormente. Os snapshots manuais não são excluídos. É possível ter até 100 snapshots manuais por região.

Gerenciar backups automatizados

Esta seção mostra como fazer gerenciar backups automáticos de instâncias de banco de dados e de clusters de banco de dados multi-AZ.

Tópicos

- Janela de backup
- Backup retention period (Período de retenção de backup)
- Ativar backups automáticos
- Reter backups automatizados
- Excluir backups automatizados retidos
- Backups automáticos com mecanismos de armazenamento MySQL sem suporte
- Backups automáticos com mecanismos de armazenamento MariaDB sem suporte
- Replicar backups automatizados para outra Região da AWS

Janela de backup

Os backups automáticos são feitos diariamente durante a janela de backup escolhida. Se o backup exigir mais tempo do que o da janela de backup, ele continuará após a janela encerrar, até que esteja concluído. A janela de backup não pode se sobrepor à janela de manutenção semanal da instância de banco de dados ou do cluster de banco de dados multi-AZ.

Durante a janela de backup automático, as E/S de armazenamento podem ser suspensas brevemente enquanto o processo de backup é inicializado (geralmente durante alguns segundos). Você pode perceber latências elevadas por alguns minutos durante os backups de implantações Multi-AZ. Para MariaDB, MySQL, Oracle e PostgreSQL, a atividade de E/S não é suspensa na primária durante o backup para implantações multi-AZ, porque o backup é retomado do modo de espera. Para o SQL Server, a atividade de E/S é suspensa brevemente durante o backup para implantações single-AZ e multi-AZ, pois o backup é obtido da primária. Para o Db2, a atividade de E/S também é suspensa brevemente durante o backup, mesmo que o backup seja obtido do modo de espera.

Os backups automatizados podem ser ignorados ocasionalmente se a instância ou o cluster de banco de dados tiver uma workload pesada no momento em que um backup deveria ser iniciado. Se um backup for ignorado, ainda será possível fazer uma recuperação point-in-time (PITR), e ainda se tentará fazer backup durante a próxima janela de backup. Para ter mais informações sobre o PITR,

consulte Restaurar uma instância de banco de dados para um momento especificado no Amazon RDS.

Se você não especificar uma janela de backup preferida ao criar a instância de banco de dados ou o cluster de banco de dados multi-AZ, o Amazon RDS atribuirá uma janela de backup de 30 minutos padrão. Essa janela é selecionada aleatoriamente em um bloco de tempo de 8 horas para cada Região da AWS. A tabela a seguir lista os blocos de tempo de cada Região da AWS a partir dos quais as janelas de backups padrão são atribuídas.

Nome da região	Região	Bloco de hora
Leste dos EUA (Norte da Virgínia)	us-east-1	Das 3h às 11h (UTC)
Leste dos EUA (Ohio)	us-east-2	De 03:00 a 11:00 UTC
US West (N. California)	us-west-1	De 06:00 a 14:00 UTC
US West (Oregon)	us-west-2	De 06:00 a 14:00 UTC
Africa (Cape Town)	af-south-1	De 03:00 a 11:00 UTC
Asia Pacific (Hong Kong)	ap-east-1	Das 6h às 14h (UTC)
Ásia-Pacífico (Hyderabad)	ap-south-2	06h30 a 14h30 UTC
Ásia-Pacífico (Jacarta)	ap-southeast-3	Das 08h às 16h UTC
Ásia-Pacífico (Malásia)	ap-southeast-5	Das 9h às 17h (UTC)
Ásia-Pacífico (Melbourne)	ap-southeast-4	Das 11h às 19h UTC
Ásia-Pacífico (Mumbai)	ap-south-1	De 16:30 a 00:30 UTC

Nome da região	Região	Bloco de hora
Asia Pacific (Osaka)	ap-northeast-3	De 00:00 a 08:00 UTC
Asia Pacific (Seoul)	ap-northeast-2	De 13:00 a 21:00 UTC
Ásia-Pacífico (Singapura)	ap-southeast-1	De 14:00 a 22:00 UTC
Asia Pacific (Sydney)	ap-southeast-2	De 12:00 a 20:00 UTC
Asia Pacific (Tokyo)	ap-northeast-1	De 13:00 a 21:00 UTC
Canada (Central)	ca-central-1	Das 3h às 11h (UTC)
Oeste do Canadá (Calgary)	ca-west-1	Das 18h às 2h (UTC)
China (Pequim)	cn-north-1	De 06:00 a 14:00 UTC
China (Ningxia)	cn-northwest-1	De 06:00 a 14:00 UTC
Europe (Frankfurt)	eu-central-1	De 20:00 a 04:00 UTC
Europe (Ireland)	eu-west-1	De 22:00 a 06:00 UTC
Europe (London)	eu-west-2	Das 22:00 às 06:00 (UTC)
Europa (Milão)	eu-south-1	De 02:00 a 10:00 UTC
Europa (Paris)	eu-west-3	De 07:29 a 14:29 UTC
Europa (Espanha)	eu-south-2	De 02:00 a 10:00 UTC
Europe (Stockholm)	eu-north-1	De 23:00 a 07:00 UTC
Europa (Zurique)	eu-central-2	De 02:00 a 10:00 UTC
Israel (Tel Aviv)	il-central-1	Das 3h às 11h (UTC)
Oriente Médio (Barém)	me-south-1	Das 6h às 14h (UTC)

Nome da região	Região	Bloco de hora
Oriente Médio (Emirados Árabes Unidos)	me-central-1	Das 5h às 13h UTC
América do Sul (São Paulo)	sa-east-1	De 23:00 a 07:00 UTC
AWS GovCloud (Leste dos EUA)	us-gov-east-1	De 17:00 a 01:00 UTC
AWS GovCloud (Oeste dos EUA)	us-gov-west-1	Das 6h às 14h (UTC)

Backup retention period (Período de retenção de backup)

Você pode definir o período de retenção de backup quando ao criar uma instância de banco de dados ou um cluster de banco de dados multi-AZ. Se você criar uma instância de banco de dados usando a API do Amazon RDS ou a AWS CLI e se não definir o período de retenção de backup, o período de retenção de backup padrão será um dia. Se você criar uma instância de banco de dados usando o console, o período de retenção de backup será sete dias.

Após criar uma instância ou um cluster de banco de dados, você poderá modificar o período de retenção de backup. Você pode definir o período de retenção de backup de uma instância de banco de dados entre 0 e 35 dias. Definir o período de retenção de backup como 0 desabilita os backups automatizados. Em relação a um cluster de banco de dados multi-AZ, é possível definir o período de retenção de backup entre 1 e 35 dias. Os limites de snapshot manual (100 por região) não se aplicam a backups automatizados.

> **Important**
>
> Ocorrerá uma interrupção se você alterar o período de retenção de backup de uma instância de banco de dados de zero para um valor diferente de zero ou vice-versa.

O RDS não inclui o tempo gasto no estado `stopped` quando o período de retenção de backup é calculado. Os backups automatizados não serão criados enquanto uma instância ou um cluster

de banco de dados estiver interrompido. Os backups podem ser retidos por mais tempo do que o período de retenção de backup se uma instância de banco de dados tiver sido interrompida.

Ativar backups automáticos

Se sua instância de banco de dados não tiver backups automáticos ativados, você poderá habilitá-los a qualquer momento. Habilite backups automáticos configurando o período de retenção de backup para um valor positivo diferente de zero. Quando os backups automáticos estão ativados, a instância de banco de dados é desconectada e um backup é criado imediatamente.

> **Note**
>
> Se você gerenciar seus backups no AWS Backup, não poderá habilitar os backups automatizados. Para ter mais informações, consulte Usar o AWS Backup no gerenciamento de backups automatizados para o Amazon RDS.

Console

Para habilitar imediatamente backups automáticos

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Bancos de dados e selecione a instância de banco de dados ou o cluster de banco de dados multi-AZ que você deseja modificar.
3. Escolha Modificar.
4. Em Backup Retention Period (Período de retenção de backup), escolha um valor positivo diferente de zero; por exemplo, 3 dias.
5. Escolha Continue.
6. Escolha Apply immediately (Aplicar imediatamente).
7. Escolha Modificar instância de banco de dados ou Modificar cluster para salvar as alterações e ativar os backups automatizados.

AWS CLI

Para habilitar backups automatizados, use o comando AWS CLI modify-db-instance ou modify-db-cluster.

Inclua os seguintes parâmetros:

- `--db-instance-identifier` (ou `--db-cluster-identifier` para um cluster de banco de dados multi-AZ)
- `--backup-retention-period`
- `--apply-immediately` ou `--no-apply-immediately`

No exemplo a seguir, habilitamos backups automatizados definindo o período de retenção de backup como três dias. As alterações são aplicadas imediatamente.

Example

Para Linux, macOS ou Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier my_db_instance  \
    --backup-retention-period 3 \
    --apply-immediately
```

Para Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier my_db_instance  ^
    --backup-retention-period 3 ^
    --apply-immediately
```

API do RDS

Para habilitar backups automatizados, use a operação ModifyDBInstance ou ModifyDBCluster da API do RDS com os seguintes parâmetros obrigatórios:

- `DBInstanceIdentifier` ou `DBClusterIdentifier`
- `BackupRetentionPeriod`

Visualizar backups automatizados

Para visualizar backups automatizados, escolha Automated backups (Backups automatizados) no painel de navegação. Para visualizar snapshots individuais associados a um backup automatizado, escolha Snapshots no painel de navegação. Como alternativa, você pode descrever snapshots

individuais associados a um backup automatizado. Lá, restaure diretamente uma instância de banco de dados de um desses snapshots.

Os nomes de snapshots automáticos seguem o padrão `rds:<database-name>-yyyy-mm-dd-hh-mm`, com `yyyy-mm-dd-hh-mm` representando a data e a hora em que o snapshot foi criado.

Para descrever os backups automatizados para suas instâncias de banco de dados existentes com a AWS CLI, utilize um dos seguintes comandos:

```
aws rds describe-db-instance-automated-backups --db-instance-
identifier DBInstanceIdentifier
```

or

```
aws rds describe-db-instance-automated-backups --dbi-resource-id DbiResourceId
```

Para descrever os backups automatizados retidos para suas instâncias de banco de dados existentes utilizando a API do RDS, chame a ação `DescribeDBInstanceAutomatedBackups` com um dos seguintes parâmetros:

- `DBInstanceIdentifier`
- `DbiResourceId`

Reter backups automatizados

> **Note**
>
> Você só pode reter backups automatizados de instância de banco de dados, não de clusters de banco de dados multi-AZ.

Ao excluir uma instância de banco de dados, você pode optar por reter os backups automatizados. Os backups automatizados podem ser retidos por um número de dias igual ao período de retenção de backup configurado para a instância de banco de dados no momento em que você a exclui.

Os backups automatizados retidos contêm snapshots de sistema e logs de transação de uma instância de banco de dados. Eles também incluem as propriedades da instância de banco de dados, como o armazenamento alocado e a classe da instância do banco de dados, obrigatórios para restaurar uma instância ativa.

Os backups automatizados retidos e os snapshots manuais incorrerão em cobranças até serem excluídos. Para ter mais informações, consulte Custos de retenção.

É possível reter backups automatizados para instâncias do RDS que estejam executando mecanismos Db2, MariaDB, MySQL, PostgreSQL, Oracle e Microsoft SQL Server.

Restaure ou remova backups automatizados retidos usando o AWS Management Console, a API do RDS e a AWS CLI.

Tópicos

- Período de retenção
- Visualização de backups retidos
- Restauração
- Custos de retenção
- Limitações

Período de retenção

Os snapshots de sistema e os logs de transação em um backup automatizado retido expiram da mesma maneira que expiram para a instância de banco de dados de origem. Como não há novos snapshots ou logs criados para essa instância, os backups automatizados retidos acabam expirando por completo. Efetivamente, eles duram o mesmo que o snapshot do sistema mais recente, com base nas configurações do período de retenção que a instância de origem quando você a excluiu. Os backups automatizados retidos serão removidos pelo sistema depois que o último snapshot do sistema expirar.

Remova um backup automatizado retido da mesma maneira que exclui uma instância de banco de dados. Remova backups automatizados retidos usando o console ou a operação da API do RDS `DeleteDBInstanceAutomatedBackup`.

Os snapshots finais independem dos backups automatizados retidos. É altamente recomendável gerar um snapshot final, mesmo que você retenha backups automatizados, porque eles acabarão expirando. O snapshot final não expira.

Visualização de backups retidos

Para visualizar os backups automatizados retidos, escolha Automated backups (Backups automatizados) no painel de navegação e, em seguida, escolhaRetained (Retido). Para visualizar

instantâneos individuais associadas a um backup automatizado retido, escolha Snapshots no painel de navegação. Ou descreva snapshots individuais associados a um backup automatizado retido. Lá, restaure diretamente uma instância de banco de dados de um desses snapshots.

Para descrever seus backups automatizados retidos com a AWS CLI, utilize o seguinte comando:

```
aws rds describe-db-instance-automated-backups --dbi-resource-id DbiResourceId
```

Para descrever seus backups automatizados retidos utilizando a API do RDS, chame a ação `DescribeDBInstanceAutomatedBackups` ação com o parâmetro `DbiResourceId`.

Restauração

Para obter informações sobre como restaurar instâncias de banco de dados a partir de backups automatizados, consulte Restaurar uma instância de banco de dados para um momento especificado no Amazon RDS.

Custos de retenção

O custo de um backup automatizado retido é o custo de armazenamento total dos snapshots de sistema associados. Não há cobrança adicional de logs de transação ou metadados de instância. Todas as outras regras da definição de preço se aplicam a instâncias restauráveis.

Por exemplo, suponhamos que o armazenamento alocado total das instâncias em execução seja 100 GB. Suponhamos também que você tenha 50 GB de snapshots manuais mais 75 GB de snapshots de sistema associados a um backup automatizado retido. Nesse caso, você só é cobrado pelos 25 GB adicionais do armazenamento de backup, como: (50 GB + 75 GB) – 100 GB = 25 GB.

Limitações

As seguintes limitações se aplicam a backups automatizados retidos:

- O número máximo de backups automáticos retidos em uma região da AWS é 40. Ele não está incluído na cota de instâncias de banco de dados. É possível ter 40 instâncias de banco de dados em execução e mais 40 backups automatizados retidos simultaneamente.
- Os backups automatizados retidos não contêm informações sobre grupos de parâmetros ou opções.
- É possível restaurar uma instância excluída para um ponto no tempo que esteja dentro do período de retenção no momento da exclusão.

- Não é possível modificar um backup automatizado retido. Isso ocorre porque ele consiste em backups de sistema, logs de transação e propriedades de instância de banco de dados existentes no momento em que você excluiu a instância de origem.

Excluir backups automatizados retidos

Você pode excluir backups automatizados retidos quando eles não são mais necessários.

Console

Como excluir um backup automatizado retido

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Automated backups (Backups automatizados).
3. Na guia Retained (Retido), escolha o backup automatizado retido que você quer excluir.
4. Em Actions, selecione Delete.
5. Na página de confirmação, insira **`delete me`** e escolha Delete (Excluir).

AWS CLI

Você pode excluir um backup automatizado retido usando o comando da AWS CLI delete-db-instance-automated-backup com a seguinte opção:

- `--dbi-resource-id` – o identificador de recurso da instância de banco de dados de origem.

  Você pode encontrar o identificador de recurso da instância de banco de dados de origem de um backup automatizado retido executando o comando da AWS CLI describe-db-instance-automated-backups.

Example

O exemplo a seguir exclui o backup automatizado retido com o identificador de recurso da instância de banco de dados de origem `db-123ABCEXAMPLE`.

Para Linux, macOS ou Unix:

```
aws rds delete-db-instance-automated-backup \
```

```
    --dbi-resource-id db-123ABCEXAMPLE
```

Para Windows:

```
aws rds delete-db-instance-automated-backup ^
    --dbi-resource-id db-123ABCEXAMPLE
```

API do RDS

Você pode excluir um backup automatizado retido usando a operação da API do Amazon RDS DeleteDBInstanceAutomatedBackup com o seguinte parâmetro:

- `DbiResourceId` – o identificador de recurso da instância de banco de dados de origem.

  Você pode encontrar o identificador de recurso da instância de banco de dados de origem de um backup automatizado retido usando a operação da API do Amazon RDS DescribeDBInstanceAutomatedBackups.

Desativar backups automáticos

Convém desabilitar temporariamente os backups automatizados em determinadas situações; por exemplo, durante o carregamento de grandes quantidades de dados.

> **Important**
>
> É altamente desaconselhável desativar os backups automáticos, pois isso desabilita a recuperação point-in-time. Desativar backups automatizados para uma instância de banco de dados ou um cluster de banco de dados multi-AZ excluirá todos os backups automatizados existentes do banco de dados. Se você desativar e, em seguida, reativar os backups automatizados, só será possível restaurar a partir do momento em que os backups automatizados foram reativados.

Console

Para desabilitar imediatamente backups automáticos

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.

2. No painel de navegação, escolha Bancos de dados e selecione a instância de banco de dados ou o cluster de banco de dados multi-AZ que você deseja modificar.
3. Escolha Modificar.
4. Em Backup Retention Period (Período de retenção de backup), selecione 0 days (0 dias).
5. Escolha Continue.
6. Escolha Apply immediately (Aplicar imediatamente).
7. Escolha Modificar instância de banco de dados ou Modificar cluster para salvar as alterações e desativar os backups automatizados.

AWS CLI

Para desativar os backups automatizados imediatamente, use o comando modify-db-instance ou modify-db-cluster e defina o período de retenção de backup como 0 com `--apply-immediately`.

Example

O exemplo a seguir desativa imediatamente backups automatizados em um cluster de banco de dados multi-AZ.

Para Linux, macOS ou Unix:

```
aws rds modify-db-cluster \
    --db-cluster-identifier mydbcluster \
    --backup-retention-period 0 \
    --apply-immediately
```

Para Windows:

```
aws rds modify-db-cluster ^
    --db-cluster-identifier mydbcluster ^
    --backup-retention-period 0 ^
    --apply-immediately
```

Para saber quando a modificação está em vigor, chame `describe-db-instances` para a instância de banco de dados (ou `describe-db-clusters` para um cluster de banco de dados multi-AZ) até que o valor para o período de retenção de backup seja 0 e o status `mydbcluster` esteja disponível.

```
aws rds describe-db-clusters --db-cluster-identifier mydcluster
```

API do RDS

Para desativar os backups automatizados imediatamente, chame a operação ModifyDBInstance ou ModifyDBCluster com os seguintes parâmetros:

- `DBInstanceIdentifier = mydbinstance` (ou `DBClusterIdentifier = mydbcluster`)
- `BackupRetentionPeriod = 0`

Example

```
https://rds.amazonaws.com/
    ?Action=ModifyDBInstance
    &DBInstanceIdentifier=mydbinstance
    &BackupRetentionPeriod=0
    &SignatureVersion=2
    &SignatureMethod=HmacSHA256
    &Timestamp=2009-10-14T17%3A48%3A21.746Z
    &AWSAccessKeyId=<&AWS; Access Key ID>
    &Signature=<Signature>
```

Backups automáticos com mecanismos de armazenamento MySQL sem suporte

Para o mecanismo de banco de dados MySQL, os backups automatizados só são compatíveis com o mecanismo de armazenamento InnoDB. O uso desses recursos com outros mecanismos de armazenamento do MySQL, incluindo MyISAM, pode levar a um comportamento não confiável durante a restauração de backups. Mais especificamente, como os mecanismos de armazenamento como o MyISAM não são compatíveis com a recuperação de falhas de maneira confiável, as tabelas podem ser corrompidas em caso de falha. Por esse motivo, aconselhamos você a usar o mecanismo de armazenamento InnoDB.

- Para converter tabelas do MyISAM existentes em tabelas do InnoDB, use o comando `ALTER TABLE`; por exemplo: `ALTER TABLE table_name ENGINE=innodb, ALGORITHM=COPY;`
- Se optar por usar o MyISAM, você poderá tentar reparar manualmente as tabelas que forem danificadas após uma falha usando o comando `REPAIR`. Para ter mais informações, consulte REPAIR TABLE Statement na documentação do MySQL. No entanto, conforme observado na documentação do MySQL, há grandes chances de que você não consiga recuperar todos os dados.

- Se você quiser fazer um snapshot das tabelas do MyISAM antes da restauração, siga estas etapas:

  1. Interrompa todas as atividades nas tabelas do MyISAM (isto é, feche todas as sessões).

     Você pode fechar todas as sessões chamando o comando mysql.rds_kill para cada processo que é retornado pelo comando `SHOW FULL PROCESSLIST`.

  2. Bloqueie e limpe cada uma de suas tabelas do MyISAM. Por exemplo, os comandos a seguir bloqueiam e limpam duas tabelas chamadas `myisam_table1` e `myisam_table2`:

     ```
     mysql> FLUSH TABLES myisam_table, myisam_table2 WITH READ LOCK;
     ```

  3. Crie um snapshot de cluster de sua instância de banco de dados ou cluster de banco de dados multi-AZ. Quando o snapshot for concluído, liberte os bloqueios e retome a atividade nas tabelas do MyISAM. Você pode liberar os bloqueios em suas tabelas usando o seguinte comando:

     ```
     mysql> UNLOCK TABLES;
     ```

  Essas etapas forçam o MyISAM a limpar os dados armazenados na memória em disco, o que garante uma inicialização limpa quando você restaura de um snapshot de banco de dados. Para ter mais informações sobre como criar um snapshot de banco de dados, consulte Criar um snapshot de banco de dados para uma instância de banco de dados single-AZ para o Amazon RDS.

Backups automáticos com mecanismos de armazenamento MariaDB sem suporte

Para o mecanismo de banco de dados MariaDB, os backups automatizados só são compatíveis com o mecanismo de armazenamento InnoDB. O uso desses recursos com outros mecanismos de armazenamento do MariaDB, incluindo Aria, pode levar a um comportamento não confiável durante a restauração de backups. Mesmo que o Aria seja uma alternativa resistente a falhas em relação ao MyISAM, suas tabelas ainda podem ficar corrompidas em caso de falha. Por esse motivo, aconselhamos você a usar o mecanismo de armazenamento InnoDB.

- Para converter tabelas do Aria existentes em tabelas do InnoDB, você pode usar o comando `ALTER TABLE`. Por exemplo: `ALTER TABLE` *`table_name`* `ENGINE=innodb, ALGORITHM=COPY;`

- Se você optar por usar o Aria, pode tentar reparar manualmente as tabelas que forem danificadas após uma falha usando o comando `REPAIR TABLE`. Para ter mais informações, consulte http://mariadb.com/kb/en/mariadb/repair-table/.
- Se você quiser fazer um snapshot das tabelas do Aria antes da restauração, siga estas etapas:
  1. Interrompa todas as atividades nas tabelas do Aria (isto é, feche todas as sessões).
  2. Bloqueie e limpe cada uma de suas tabelas do Aria.
  3. Crie um snapshot de cluster de sua instância de banco de dados ou cluster de banco de dados multi-AZ. Quando o snapshot for concluído, liberte os bloqueios e retome a atividade nas tabelas do Aria. Essas etapas obrigam o Aria a limpar os dados armazenados na memória movendo-os para o disco, garantindo assim uma inicialização limpa quando você restaurar a partir de um snapshot de banco de dados.

Replicar backups automatizados para outra Região da AWS

Para maior capacidade de recuperação de desastres, você pode configurar a instância de banco de dados do Amazon RDS para replicar snapshots e logs de transações para uma Região da AWS de destino de sua escolha. Quando a replicação de backup é configurada para uma instância de banco de dados, o RDS inicia uma cópia entre regiões de todos os snapshots e logs de transações assim que estiverem prontos na instância de banco de dados.

As cobranças de cópia do DB snapshot aplicam-se à transferência de dados. Depois que o DB snapshot for copiado, as cobranças padrão se aplicam ao armazenamento na Região de destino. Para obter mais detalhes, consulte Definição de preço do RDS.

Para obter um exemplo de uso da replicação de backup, consulte a conversa técnica online da AWS Managed Disaster Recovery with Amazon RDS for Oracle Cross-Region Automated Backups.

> **Note**
>
> A replicação automatizada de backup não é compatível com clusters de banco de dados multi-AZ.

Tópicos

- Disponibilidade de região e versão
- Suporte a Região da AWS na origem e no destino
- Habilitar backups automatizados entre regiões
- Encontrar informações sobre backups replicados
- Restaurar para um horário especificado a partir de um backup replicado
- Interromper a replicação automatizada de backup
- Excluir backups replicados

Disponibilidade de região e versão

A disponibilidade e a compatibilidade de recursos varia entre versões específicas de cada mecanismo de banco de dados e entre Regiões da AWS. Para obter mais informações sobre a disponibilidade de versões e regiões com os backups automatizados entre regiões, consulte Regiões e mecanismos de banco de dados compatíveis com backups automatizados entre regiões no Amazon RDS.

Suporte a Região da AWS na origem e no destino

A replicação de backup é compatível entre as seguintes Regiões da AWS:

Região de origem	Regiões de destino disponíveis
Asia Pacific (Mumbai)	Asia Pacific (Singapore) Leste dos EUA (Norte da Virgínia), Leste dos EUA (Ohio), Oeste dos EUA (Oregon)
Asia Pacific (Osaka)	Asia Pacific (Tokyo)
Asia Pacific (Seoul)	Ásia-Pacífico (Singapura), Ásia-Pacífico (Tóquio) Leste dos EUA (Norte da Virgínia), Leste dos EUA (Ohio), Oeste dos EUA (Oregon)
Asia Pacific (Singapore)	Ásia-Pacífico (Mumbai), Ásia-Pacífico (Seul), Ásia-Pacífico (Sydney), Ásia-Pacífico (Tóquio) Leste dos EUA (Norte da Virgínia), Leste dos EUA (Ohio), Oeste dos EUA (Oregon)
Asia Pacific (Sydney)	Asia Pacific (Singapore) Leste dos EUA (Norte da Virgínia), Oeste dos EUA (Norte da Califórnia), Oeste dos EUA (Oregon)
Asia Pacific (Tokyo)	Ásia-Pacífico (Osaka), Ásia-Pacífico (Seul), Ásia-Pacífico (Singapura) Leste dos EUA (Norte da Virgínia), Leste dos EUA (Ohio), Oeste dos EUA (Oregon)
Canada (Central)	Europe (Ireland) Leste dos EUA (Norte da Virgínia), Leste dos EUA (Ohio), Oeste dos EUA (Norte da Califórnia), Oeste dos EUA (Oregon)
China (Pequim)	China (Ningxia)

Região de origem	Regiões de destino disponíveis
China (Ningxia)	China (Pequim)
Europa (Frankfurt)	Europa (Irlanda), Europa (Londres), Europa (Paris), Europa (Estocolmo) Leste dos EUA (Norte da Virgínia), Leste dos EUA (Ohio), Oeste dos EUA (Oregon)
Europe (Ireland)	Canada (Central) Europa (Frankfurt), Europa (Londres), Europa (Paris), Europa (Estocolmo) Leste dos EUA (Norte da Virgínia), Leste dos EUA (Ohio), Oeste dos EUA (Norte da Califórnia), Oeste dos EUA (Oregon)
Europe (London)	Europa (Frankfurt), Europa (Irlanda), Europa (Paris), Europa (Estocolmo) US East (N. Virginia)
Europe (Paris)	Europa (Frankfurt), Europa (Irlanda), Europa (Londres), Europa (Estocolmo) US East (N. Virginia)
Europe (Stockholm)	Europa (Frankfurt), Europa (Irlanda), Europa (Londres), Europa (Paris) US East (N. Virginia)
South America (São Paulo)	Leste dos EUA (Norte da Virgínia), Leste dos EUA (Ohio)
AWS GovCloud (Leste dos EUA)	AWS GovCloud (Oeste dos EUA)
AWS GovCloud (Oeste dos EUA)	AWS GovCloud (Leste dos EUA)

Região de origem	Regiões de destino disponíveis
US East (N. Virginia)	Ásia-Pacífico (Mumbai), Ásia-Pacífico (Seul), Ásia-Pacífico (Singapura), Ásia-Pacífico (Sydney), Ásia-Pacífico (Tóquio) Canada (Central) Europa (Frankfurt), Europa (Irlanda), Europa (Londres), Europa (Paris), Europa (Estocolmo) South America (São Paulo) Leste dos EUA (Ohio), Oeste dos EUA (Norte da Califórnia), Oeste dos EUA (Oregon)
US East (Ohio)	Ásia-Pacífico (Mumbai), Ásia-Pacífico (Seul), Ásia-Pacífico (Singapura), Ásia-Pacífico (Tóquio) Canada (Central) Europa (Frankfurt), Europa (Irlanda) South America (São Paulo) Leste dos EUA (Norte da Virgínia), Oeste dos EUA (Norte da Califórnia), Oeste dos EUA (Oregon)
US West (N. California)	Asia Pacific (Sydney) Canada (Central) Europe (Ireland) Leste dos EUA (Norte da Virgínia), Leste dos EUA (Ohio), Oeste dos EUA (Oregon)

Região de origem	Regiões de destino disponíveis
US West (Oregon)	Ásia-Pacífico (Mumbai), Ásia-Pacífico (Seul), Ásia-Pacífico (Singapura), Ásia-Pacífico (Sydney), Ásia-Pacífico (Tóquio) Canada (Central) Europa (Frankfurt), Europa (Irlanda) Leste dos EUA (Norte da Virgínia), Leste dos EUA (Ohio), Oeste dos EUA (Norte da Califórnia)

Você também pode usar o comando `describe-source-regions` da AWS CLI para descobrir quais Regiões da AWS podem replicar entre si. Para obter mais informações, consulte Encontrar informações sobre backups replicados.

Habilitar backups automatizados entre regiões

Você pode habilitar a replicação de backup em instâncias de banco de dados novas ou existentes usando o console do Amazon RDS. Você também pode usar o comando `start-db-instance-automated-backups-replication` AWS CLI ou a operação `StartDBInstanceAutomatedBackupsReplication` da API do RDS. Você pode replicar até 20 backups para cada Região da AWS de destino de cada Conta da AWS.

> **Note**
>
> Para poder replicar backups automatizados, certifique-se de habilitá-los. Para obter mais informações, consulte Ativar backups automáticos.

Console

Você pode habilitar a replicação de backup para uma instância de banco de dados nova ou existente:

- Para uma nova instância de banco de dados, associe-o ao executar a instância. Para obter mais informações, consulte Configurações para instâncias de banco de dados.
- Para uma instância de banco de dados existente, use o procedimento a seguir.

Para habilitar a replicação de backup para uma instância de banco de dados existente

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Automated backups (Backups automatizados).
3. Na guia Current Region (Região atual), escolha a instância de banco de dados para a qual você deseja habilitar a replicação de backup.
4. Em Actions (Ações), escolha Manage cross-Region replication (Gerenciar replicação entre regiões).
5. Em Replicação de backup, selecione Habilitar replicação para outra Região da AWS.
6. Escolha a Destination Region (Região de destino).
7. Escolha o Replicated backup retention period (Período de retenção de backup replicado).
8. Se você habilitou a criptografia na instância de banco de dados de origem, escolha a AWS KMS key para criptografar os backups ou insira o ARN da chave.
9. Escolha Salvar.

Na região de origem, os backups replicados são listados na guia Current Region (Região atual) da página Automated backups (Backups automatizados). Na região de destino, os backups replicados são listados na guia Replicated backups (Backups replicados) da página Automated backups (Backups automatizados).

AWS CLI

Habilite a replicação de backup usando o comando da AWS CLI `start-db-instance-automated-backups-replication`.

O exemplo de CLI a seguir replica backups automatizados de uma instância de banco de dados na Região Oeste dos EUA (Oregon) para Leste dos EUA (Norte da Virgínia) Região. Ele também criptografa os backups replicados usando uma AWS KMS key na região de destino.

Para habilitar a replicação de backup

- Execute um dos seguintes comandos:

  Para Linux, macOS ou Unix:

```
aws rds start-db-instance-automated-backups-replication \
```

```
--region us-east-1 \
--source-db-instance-arn "arn:aws:rds:us-west-2:123456789012:db:mydatabase" \
--kms-key-id "arn:aws:kms:us-east-1:123456789012:key/AKIAIOSFODNN7EXAMPLE" \
--backup-retention-period 7
```

Para Windows:

```
aws rds start-db-instance-automated-backups-replication ^
--region us-east-1 ^
--source-db-instance-arn "arn:aws:rds:us-west-2:123456789012:db:mydatabase" ^
--kms-key-id "arn:aws:kms:us-east-1:123456789012:key/AKIAIOSFODNN7EXAMPLE" ^
--backup-retention-period 7
```

A opção `--source-region` é necessária quando você criptografa backups entre as regiões GovCloud (Leste dos EUA) da AWS e GovCloud (Oeste dos EUA) da AWS. Em `--source-region`, especifique a Região da AWS da instância de banco de dados de origem.

Se a `--source-region` não estiver especificada, especifique um valor de `--pre-signed-url`. Uma pre-signed URL é aquela que contém uma solicitação assinada do Signature Versão 4 para o comando `start-db-instance-automated-backups-replication` chamado na Região da AWS de origem. Para saber mais a respeito da opção `pre-signed-url`, consulte pystart-db-instance-automated-backups-replication na Referência de comandos da AWS CLI.

API do RDS

Habilite a replicação de backup usando a operação `StartDBInstanceAutomatedBackupsReplication` da API do RDS com os seguintes parâmetros:

- `Region` (se você não estiver chamando a operação da API da região de destino).
- `SourceDBInstanceArn`
- `BackupRetentionPeriod`
- `KmsKeyId` (opcional)
- `PreSignedUrl` (necessário se você usar o `KmsKeyId`)

> **Note**
>
> Se você criptografar os backups, também deverá incluir uma URL pré-assinada. Para obter mais informações sobre URLs pré-assinadas, consulte Authenticating Requests: Using Query Parameters (AWS Signature Version 4) (Solicitações de autenticação: uso de parâmetros de consulta) na referência da API do Amazon Simple Storage Service e Signature Version 4 signing process (Processo de assinatura da versão 4) no Guia de referência geral da AWS).

Encontrar informações sobre backups replicados

Você pode usar os seguintes comandos CLI para encontrar informações sobre backups replicados:

- `describe-source-regions`
- `describe-db-instances`
- `describe-db-instance-automated-backups`

O exemplo de `describe-source-regions` a seguir lista as Regiões da AWS de origem das quais os backups automatizados podem ser replicados para a região Oeste dos EUA (Oregon) de destino.

Para mostrar informações sobre as regiões de origem

- Execute o seguinte comando.

```
aws rds describe-source-regions --region us-west-2
```

A saída mostra que os backups podem ser replicados de US East (N. Virginia), mas não de Leste dos EUA (Ohio) ou Oeste dos EUA (Norte da Califórnia) para Oeste dos EUA (Oregon).

```
{
    "SourceRegions": [
        ...
        {
            "RegionName": "us-east-1",
            "Endpoint": "https://rds.us-east-1.amazonaws.com",
            "Status": "available",
            "SupportsDBInstanceAutomatedBackupsReplication": true
        },
```

```
        {
            "RegionName": "us-east-2",
            "Endpoint": "https://rds.us-east-2.amazonaws.com",
            "Status": "available",
            "SupportsDBInstanceAutomatedBackupsReplication": false
        },
            "RegionName": "us-west-1",
            "Endpoint": "https://rds.us-west-1.amazonaws.com",
            "Status": "available",
            "SupportsDBInstanceAutomatedBackupsReplication": false
        }
    ]
}
```

O `describe-db-instances` de exemplo a seguir mostra os backups automatizados de uma instância de banco de dados.

Para mostrar os backups replicados de uma instância de banco de dados

- Execute um dos seguintes comandos:

  Para Linux, macOS ou Unix:

  ```
  aws rds describe-db-instances \
  --db-instance-identifier mydatabase
  ```

  Para Windows:

  ```
  aws rds describe-db-instances ^
  --db-instance-identifier mydatabase
  ```

A saída inclui os backups replicados.

```
{
    "DBInstances": [
        {
            "StorageEncrypted": false,
            "Endpoint": {
                "HostedZoneId": "Z1PVIF0B656C1W",
                "Port": 1521,
            ...
```

```
            "BackupRetentionPeriod": 7,
            "DBInstanceAutomatedBackupsReplications":
 [{"DBInstanceAutomatedBackupsArn": "arn:aws:rds:us-east-1:123456789012:auto-backup:ab-
L2IJCEXJP7XQ7HOJ4SIEXAMPLE"}]
        }
    ]
}
```

O `describe-db-instance-automated-backups` de exemplo a seguir mostra os backups automatizados de uma instância de banco de dados.

Para mostrar backups automatizados para uma instância de banco de dados

- Execute um dos seguintes comandos:

  Para Linux, macOS ou Unix:

  ```
  aws rds describe-db-instance-automated-backups \
  --db-instance-identifier mydatabase
  ```

  Para Windows:

  ```
  aws rds describe-db-instance-automated-backups ^
  --db-instance-identifier mydatabase
  ```

A saída mostra a instância de banco de dados de origem e backups automatizados em Oeste dos EUA (Oregon), com backups replicados para US East (N. Virginia).

```
{
    "DBInstanceAutomatedBackups": [
        {
            "DBInstanceArn": "arn:aws:rds:us-west-2:868710585169:db:mydatabase",
            "DbiResourceId": "db-L2IJCEXJP7XQ7HOJ4SIEXAMPLE",
            "DBInstanceAutomatedBackupsArn": "arn:aws:rds:us-west-2:123456789012:auto-
backup:ab-L2IJCEXJP7XQ7HOJ4SIEXAMPLE",
            "BackupRetentionPeriod": 7,
            "DBInstanceAutomatedBackupsReplications":
 [{"DBInstanceAutomatedBackupsArn": "arn:aws:rds:us-east-1:123456789012:auto-backup:ab-
L2IJCEXJP7XQ7HOJ4SIEXAMPLE"}]
            "Region": "us-west-2",
```

```
            "DBInstanceIdentifier": "mydatabase",
            "RestoreWindow": {
                "EarliestTime": "2020-10-26T01:09:07Z",
                "LatestTime": "2020-10-31T19:09:53Z",
            }
            ...
        }
    ]
}
```

O `describe-db-instance-automated-backups` de exemplo a seguir usa a opção `--db-instance-automated-backups-arn` para mostrar os backups replicados na região de destino.

Para mostrar backups replicados

- Execute um dos seguintes comandos:

  Para Linux, macOS ou Unix:

  ```
  aws rds describe-db-instance-automated-backups \
  --db-instance-automated-backups-arn "arn:aws:rds:us-east-1:123456789012:auto-backup:ab-L2IJCEXJP7XQ7HOJ4SIEXAMPLE"
  ```

  Para Windows:

  ```
  aws rds describe-db-instance-automated-backups ^
  --db-instance-automated-backups-arn "arn:aws:rds:us-east-1:123456789012:auto-backup:ab-L2IJCEXJP7XQ7HOJ4SIEXAMPLE"
  ```

A saída mostra a instância de banco de dados de origem em Oeste dos EUA (Oregon), com backups replicados em US East (N. Virginia).

```
{
    "DBInstanceAutomatedBackups": [
        {
            "DBInstanceArn": "arn:aws:rds:us-west-2:868710585169:db:mydatabase",
            "DbiResourceId": "db-L2IJCEXJP7XQ7HOJ4SIEXAMPLE",
            "DBInstanceAutomatedBackupsArn": "arn:aws:rds:us-east-1:123456789012:auto-backup:ab-L2IJCEXJP7XQ7HOJ4SIEXAMPLE",
            "Region": "us-west-2",
            "DBInstanceIdentifier": "mydatabase",
```

```
            "RestoreWindow": {
                "EarliestTime": "2020-10-26T01:09:07Z",
                "LatestTime": "2020-10-31T19:01:23Z"
            },
            "AllocatedStorage": 50,
            "BackupRetentionPeriod": 7,
            "Status": "replicating",
            "Port": 1521,
            ...
        }
    ]
}
```

Restaurar para um horário especificado a partir de um backup replicado

Você pode restaurar uma instância de banco de dados para um determinado ponto no tempo a partir de um backup replicado usando o console do Amazon RDS. Você também pode usar o comando `restore-db-instance-to-point-in-time` AWS CLI ou a operação `RestoreDBInstanceToPointInTime` da API do RDS.

Para obter informações gerais sobre recuperação point-in-time (PITR), consulte Restaurar uma instância de banco de dados para um momento especificado no Amazon RDS.

> **Note**
>
> No RDS para SQL Server, os grupos de opções não são copiados entre Regiões da AWS quando os backups automatizados são replicados. Se você associou um grupo de opções personalizadas à sua instância de banco de dados do RDS for SQL Server, poderá recriar esse grupo de opções na região de destino. Em seguida, restaure a instância de banco de dados na região de destino e associe o grupo de opções personalizadas a ela. Para obter mais informações, consulte Trabalhar com grupos de opções.

Console

Para restaurar uma instância de banco de dados em um horário especificado a partir de um backup replicado

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.

2. Escolha a região de destino (para a qual os backups são replicados) no seletor de regiões.
3. No painel de navegação, escolha Automated backups (Backups automatizados).
4. Na guia Replicated backups (Backups replicados), escolha a instância de banco de dados que você deseja restaurar.
5. Em Actions (Ações), escolha Restore to point in time (Restaurar para point-in-time).
6. Escolha Latest restorable time (Hora da última restauração) para restaurar no último horário possível ou escolha Custom (Personalizar) para escolher um horário.

   Se você escolher Custom (Personalizar), insira a data e a hora para as quais deseja restaurar a instância.

   > **Note**
   >
   > Os horários são mostrados no fuso horário local, que é indicado por um deslocamento do Tempo Universal Coordenado (UTC). Por exemplo, UTC-5 é a Hora Padrão do Leste dos EUA/Horário de Verão Central.

7. Em Identificador de instância de banco de dados, digite o nome da instância de banco de dados restaurada de destino.
8. (Opcional) Escolha outras opções conforme necessário, como ativar o dimensionamento automático.
9. Escolha Restore to point in time (Restaurar para point-in-time).

AWS CLI

Use o comando `restore-db-instance-to-point-in-time` da AWS CLI para criar uma nova instância de banco de dados.

Para restaurar uma instância de banco de dados em um horário especificado a partir de um backup replicado

- Execute um dos seguintes comandos:

  Para Linux, macOS ou Unix:

```
aws rds restore-db-instance-to-point-in-time \
    --source-db-instance-automated-backups-arn "arn:aws:rds:us-
east-1:123456789012:auto-backup:ab-L2IJCEXJP7XQ7HOJ4SIEXAMPLE" \
```

```
    --target-db-instance-identifier mytargetdbinstance \
    --restore-time 2020-10-14T23:45:00.000Z
```

Para Windows:

```
aws rds restore-db-instance-to-point-in-time ^
    --source-db-instance-automated-backups-arn "arn:aws:rds:us-
east-1:123456789012:auto-backup:ab-L2IJCEXJP7XQ7HOJ4SIEXAMPLE" ^
    --target-db-instance-identifier mytargetdbinstance ^
    --restore-time 2020-10-14T23:45:00.000Z
```

API do RDS

Para restaurar uma instância de banco de dados em um horário específico, chame a operação `RestoreDBInstanceToPointInTime` da API do Amazon RDS com os seguintes parâmetros:

- `SourceDBInstanceAutomatedBackupsArn`
- `TargetDBInstanceIdentifier`
- `RestoreTime`

Interromper a replicação automatizada de backup

Você pode interromper a replicação de backup para instâncias de banco de dados usando o console do Amazon RDS. Você também pode usar o comando `stop-db-instance-automated-backups-replication` AWS CLI ou a operação `StopDBInstanceAutomatedBackupsReplication` da API do RDS.

Os backups replicados são mantidos, sujeitos ao período de retenção de backup definido quando foram criados.

Console

Interrompa a replicação de backup na página Automated backups (Backups automatizados) na região de origem.

Como interromper a replicação de backup em uma Região da AWS

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.

2. Escolha a região de origem no Region selector (Seletor de regiões).
3. No painel de navegação, escolha Automated backups (Backups automatizados).
4. Na guia Current Region (Região atual), escolha a instância de banco de dados para a qual você deseja interromper a replicação de backup.
5. Em Actions (Ações), escolha Manage cross-Region replication (Gerenciar replicação entre regiões).
6. Em Replicação de backup, desmarque a caixa de seleção Habilitar replicação para outra Região da AWS.
7. Escolha Salvar.

Os backups replicados são listados na guia Retained (Retidos) da página Automated backups (Backups automatizados) na região de destino.

AWS CLI

Interrompa a replicação de backup usando o comando `stop-db-instance-automated-backups-replication` da AWS CLI.

O exemplo de CLI a seguir impede que os backups automatizados de uma instância de banco de dados sejam replicados na região Oeste dos EUA (Oregon).

Para interromper a replicação de backup

- Execute um dos seguintes comandos:

  Para Linux, macOS ou Unix:

```
aws rds stop-db-instance-automated-backups-replication \
--region us-east-1 \
--source-db-instance-arn "arn:aws:rds:us-west-2:123456789012:db:mydatabase"
```

  Para Windows:

```
aws rds stop-db-instance-automated-backups-replication ^
--region us-east-1 ^
--source-db-instance-arn "arn:aws:rds:us-west-2:123456789012:db:mydatabase"
```

API do RDS

Interrompa a replicação de backup usando a operação `StopDBInstanceAutomatedBackupsReplication` da API do RDS com os seguintes parâmetros:

- `Region`
- `SourceDBInstanceArn`

Excluir backups replicados

Você pode excluir backups replicados para instâncias de banco de dados usando o console do Amazon RDS. Você também pode usar o comando `delete-db-instance-automated-backups` AWS CLI ou a operação `DeleteDBInstanceAutomatedBackup` da API do RDS.

Console

Exclua backups replicados na região de destino pela página Automated backups (Backups automatizados).

Para excluir backups replicados

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. Escolha a região de destino no Region selector (Seletor de regiões).
3. No painel de navegação, escolha Automated backups (Backups automatizados).
4. Na guia Replicated backups (Backups replicados), escolha a instância de banco de dados para a qual você deseja excluir os backups replicados.
5. Em Actions, selecione Delete.
6. Na página de confirmação, insira **`delete me`** e escolha Delete (Excluir).

AWS CLI

Exclua backups replicados usando o comando `delete-db-instance-automated-backup` da AWS CLI.

Você pode usar o comando `describe-db-instances` da CLI para localizar os nomes de recursos da Amazon (ARNs) dos backups replicados. Para obter mais informações, consulte Encontrar informações sobre backups replicados.

Para excluir backups replicados

- Execute um dos seguintes comandos:

  Para Linux, macOS ou Unix:

```
aws rds delete-db-instance-automated-backup \
--db-instance-automated-backups-arn "arn:aws:rds:us-east-1:123456789012:auto-
backup:ab-L2IJCEXJP7XQ7HOJ4SIEXAMPLE"
```

  Para Windows:

```
aws rds delete-db-instance-automated-backup ^
--db-instance-automated-backups-arn "arn:aws:rds:us-east-1:123456789012:auto-
backup:ab-L2IJCEXJP7XQ7HOJ4SIEXAMPLE"
```

API do RDS

Exclua backups replicados usando a operação `DeleteDBInstanceAutomatedBackup` da API do RDS com o parâmetro `DBInstanceAutomatedBackupsArn`.

Gerenciar backups manuais

Essa seção mostra como fazer o gerenciamento de backups manuais de instâncias e clusters de banco de dados.

Consulte informações sobre como criar um snapshot de banco de dados para uma instância de banco de dados Single-AZ em Criar um snapshot de banco de dados para uma instância de banco de dados single-AZ para o Amazon RDS.

Consulte informações sobre como criar um snapshot de banco de dados para um cluster de banco de dados multi-AZ em Criar um snapshot do cluster de banco de dados multi-AZ para o Amazon RDS.

Consulte informações sobre como excluir um snapshot de banco de dados em Excluir um snapshot de banco de dados para o Amazon RDS.

Criar um snapshot de banco de dados para uma instância de banco de dados single-AZ para o Amazon RDS

O Amazon RDS cria um snapshot do volume de armazenamento de sua instância de banco de dados, fazendo o backup de toda a instância de banco de dados, não apenas dos bancos de dados individuais. A criação deste snapshot de banco de dados em uma instância de banco de dados Single-AZ resulta em uma breve suspensão da E/S, que pode durar de alguns segundos a alguns minutos, dependendo do tamanho e da classe de sua instância de banco de dados. Para MariaDB, MySQL, Oracle e PostgreSQL, a atividade de E/S não é suspensa no principal durante o backup para implantações Multi-AZ, porque o backup é retomado do modo de espera. Para o SQL Server, a atividade de E/S é suspensa brevemente durante o backup para implantações Multi-AZ.

Ao criar um snapshot de banco de dados, você precisa identificar de qual instância de banco de dados deseja fazer backup e, em seguida, dar um nome para a sua instância de banco de dados para que você possa restaurar a partir dela depois. O tempo necessário para criar um snapshot varia com o tamanho dos bancos de dados. Como o snapshot inclui todo o volume de armazenamento, o tamanho de arquivos, como arquivos temporários, também afeta o tempo necessário para criar o snapshot.

> **Note**
>
> Sua instância de banco de dados deve estar no estado `available` para obter um snapshot de banco de dados.
> Para instâncias de banco de dados PostgreSQL, os dados em tabelas não registradas em log podem não ser restaurados de snapshots. Para obter mais informações, consulte Práticas recomendadas para trabalhar com PostgreSQL.

Diferentemente dos backups automatizados, os snapshots manuais não estão sujeitos ao período de retenção de backup. Os snapshots não expiram.

Para backups de muito longo prazo de dados MariaDB, MySQL e PostgreSQL, recomendamos exportar dados de snapshot para o Amazon S3. Se a versão principal do mecanismo de banco de dados não for mais compatível, você não poderá restaurar para essa versão a partir de um snapshot. Para obter mais informações, consulte Exportação de dados de snapshot de banco de dados para o Amazon S3 para Amazon RDS.

Você pode criar um snapshot de banco de dados usando o AWS Management Console, a AWS CLI ou a API do RDS.

Console

Para criar um snapshot de banco de dados

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Snapshots.

   A lista de snapshots manuais aparece.
3. Selecione Take Snapshot (Fazer snapshot).

   A janela Take snapshot de banco de dados (Fazer snapshot de banco de dados) é exibida.
4. Escolha a Instância de banco de dados para a qual você deseja criar um snapshot.
5. Insira o Nome do snapshot.
6. Selecione Take Snapshot (Fazer snapshot).

A lista Snapshots manuais é exibida com o status do novo snapshot do banco de dados mostrado como `Creating`. Com o status é `Available`, você pode ver a hora de criação.

AWS CLI

Ao criar um snapshot de banco de dados usando a AWS CLI, você precisa identificar de qual instância de banco de dados deseja fazer backup e, em seguida, dar um nome para a sua instância de banco de dados para que você possa restaurar a partir dela depois. Você pode fazer isso usando o comando AWS CLI da `create-db-snapshot` com os seguintes parâmetros:

- `--db-instance-identifier`
- `--db-snapshot-identifier`

Neste exemplo, você cria um snapshot de banco de dados chamado *`mydbsnapshot`* para uma instância de banco de dados chamada *`mydbinstance`*.

Example

Para Linux, macOS ou Unix:

```
aws rds create-db-snapshot \
    --db-instance-identifier mydbinstance \
    --db-snapshot-identifier mydbsnapshot
```

Para Windows:

```
aws rds create-db-snapshot ^
    --db-instance-identifier mydbinstance ^
    --db-snapshot-identifier mydbsnapshot
```

API do RDS

Ao criar um snapshot de banco de dados usando a API do Amazon RDS, você precisa identificar de qual instância de banco de dados deseja fazer backup e, em seguida, dar um nome para a sua instância de banco de dados para que você possa restaurar a partir dela depois. Você pode fazer isso usando o comando `CreateDBSnapshot` da API do Amazon RDS com os seguintes parâmetros:

- `DBInstanceIdentifier`
- `DBSnapshotIdentifier`

Criar um snapshot do cluster de banco de dados multi-AZ para o Amazon RDS

Ao criar um snapshot de cluster de banco de dados Multi-AZ, certifique-se de identificar de qual cluster de banco de dados Multi-AZ você fará backup e, em seguida, dê um nome ao seu snapshot para que você possa restaurá-lo mais tarde. Você também pode compartilhar um snapshot de cluster de banco de dados multi-AZ. Para obter instruções, consulte the section called “Compartilhar um snapshot do de banco de dados”.

Você pode criar um snapshot de cluster de banco de dados Multi-AZ usando o AWS Management Console, a AWS CLI ou a API do RDS.

Para backups de muito longo prazo, recomendamos exportar dados de snapshot para o Amazon S3. Se a versão principal do mecanismo de banco de dados não for mais compatível, você não poderá restaurar para essa versão a partir de um snapshot. Para ter mais informações, consulte Exportação de dados de snapshot de banco de dados para o Amazon S3 para Amazon RDS.

Console

Para criar um snapshot de cluster de banco de dados

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Bancos de dados.
3. Na lista, escolha o cluster de banco de dados Multi-AZ do qual você deseja fazer um snapshot.
4. Em Actions (Ações), escolha Take snapshot (Fazer snapshot).

   A janela Take snapshot de banco de dados (Fazer snapshot de banco de dados) é exibida.
5. Para Snapshot name (Nome do snapshot), insira o nome do snapshot.
6. Selecione Take Snapshot (Fazer snapshot).

A página Snapshots é exibida com o status do novo snapshot de cluster de banco de dados Multi-AZ mostrado como `Creating`. Com o status é `Available`, você pode ver a hora de criação.

AWS CLI

É possível criar um snapshot de cluster de banco de dados Multi-AZ utilizando o comando da AWS CLI create-db-cluster-snapshot com as seguintes opções:

- `--db-cluster-identifier`
- `--db-cluster-snapshot-identifier`

Nesse exemplo, você cria um snapshot de cluster de banco de dados multi-AZ chamado *`mymultiazdbclustersnapshot`* para um cluster de banco de dados chamado *`mymultiazdbcluster`*.

Example

Para Linux, macOS ou Unix:

```
aws rds create-db-cluster-snapshot \
    --db-cluster-identifier mymultiazdbcluster \
    --db-cluster-snapshot-identifier mymultiazdbclustersnapshot
```

Para Windows:

```
aws rds create-db-cluster-snapshot ^
    --db-cluster-identifier mymultiazdbcluster ^
    --db-cluster snapshot-identifier mymultiazdbclustersnapshot
```

API do RDS

Você pode criar um snapshot de cluster de banco de dados Multi-AZ utilizando a operação CreateDBClusterSnapshot da API do Amazon RDS com os seguintes parâmetros:

- `DBClusterIdentifier`
- `DBClusterSnapshotIdentifier`

Excluir um snapshot de cluster de banco de dados multi-AZ

É possível excluir snapshots de banco de dados multi-AZ gerenciados pelo Amazon RDS quando eles não são mais necessários. Para obter instruções, consulte the section called “Excluir um snapshot de banco de dados”.

Excluir um snapshot de banco de dados para o Amazon RDS

É possível excluir snapshots de banco de dados gerenciados pelo Amazon RDS quando eles não são mais necessários.

> **Note**
>
> Para excluir backups gerenciados pelo AWS Backup, use o console do AWS Backup. Para obter mais informações sobre o AWS Backup, consulte o Guia do desenvolvedor do AWS Backup.

Excluir um snapshot de banco de dados

É possível excluir um snapshot de banco de dados manual, compartilhado ou público usando o AWS Management Console, a AWS CLI ou a API do RDS.

Para excluir um snapshot compartilhado ou público, você deve fazer login na conta da AWS que tem esse snapshot.

Se você tiver snapshots de banco de dados automatizados que deseja excluir sem excluir a instância de banco de dados, altere o período de retenção de backup dessa instância para 0. Os snapshots automatizados serão excluídos quando a alteração for aplicada. Você poderá aplicar a alteração imediatamente se não quiser esperar até o próximo período de manutenção. Após a conclusão da alteração, você poderá reabilitar os backups automáticos, configurando o período de retenção de backup como um número maior que 0. Para mais informações sobre a modificação de uma instância de banco de dados , consulte Modificar uma instância de banco de dados do Amazon RDS.

Os backups automatizados retidos e os snapshots manuais incorrerão em cobranças até serem excluídos. Para ter mais informações, consulte Custos de retenção.

Se você tiver excluído uma instância de banco de dados, poderá excluir seus snapshot de banco de dadoss automatizados removendo os backups automatizados dessa instância. Para obter informações sobre backups automatizados, consulte Introdução aos backups.

Console

Para excluir um snapshot de banco de dados

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.

2. No painel de navegação, escolha Snapshots.

   A lista de snapshots manuais aparece.
3. Escolha o snapshot de banco de dados que você deseja excluir.
4. Em Actions (Ações), selecione Delete Snapshot (Excluir snapshot).
5. Escolha Delete (Excluir) na página de confirmação.

AWS CLI

É possível excluir um snapshot de banco de dados usando o comando da AWS CLI delete-db-snapshot.

As seguintes opções são usadas para excluir um snapshot de banco de dados.

- `--db-snapshot-identifier` – O identificador do snapshot de banco de dados.

Example

O código a seguir exclui o snapshot de banco de dados `mydbsnapshot`.

Para Linux, macOS ou Unix:

```
aws rds delete-db-snapshot \
    --db-snapshot-identifier mydbsnapshot
```

Para Windows:

```
aws rds delete-db-snapshot ^
    --db-snapshot-identifier mydbsnapshot
```

API do RDS

É possível excluir um snapshot de banco de dados usando a operação DeleteDBSnapshot da API do Amazon RDS.

Os seguintes parâmetros são usados para excluir um snapshot de banco de dados.

- `DBSnapshotIdentifier` – O identificador do snapshot de banco de dados.

Restaurar uma instância de banco de dados

Esta seção mostra como restaurar uma instância de banco de dados. Essa página mostra como restaurar uma instância de banco de dados do Amazon RDS de um snapshot do banco de dados.

O Amazon RDS cria um snapshot do volume de armazenamento de sua instância de banco de dados, fazendo o backup de toda a instância de banco de dados, não apenas dos bancos de dados individuais. É possível criar uma instância de banco de dados com uma restauração de um snapshot de banco de dados. Forneça o nome do snapshot de banco de dados do qual restaurar e um nome para a nova instância de banco de dados que é criada na restauração. Não é possível restaurar de um snapshot de banco de dados para uma instância de banco de dados existente. Uma nova instância de banco de dados é criada na restauração.

É possível usar a instância de banco de dados restaurada assim que o status for `available`. A instância de banco de dados continua a carregar dados em segundo plano. Isso é conhecido como carregamento lento.

Se você acessar dados que ainda não foram carregados, a instância de banco de dados baixará imediatamente os dados solicitados do Amazon S3 e continuará carregando o restante dos dados do volume em segundo plano. Para ter mais informações, consulte Snapshots do Amazon EBS.

Para ajudar a atenuar os efeitos do carregamento lento em tabelas às quais necessita de acesso rápido, você pode executar operações que envolvem verificações de tabela completas, como `SELECT *`. Isso permite que o Amazon RDS baixe todos os dados da tabela de backup do S3.

Você pode restaurar uma instância de banco de dados e usar um tipo de armazenamento diferente que o snapshot de banco de dados de origem. Nesse caso, o processo de restauração é mais lento devido ao trabalho adicional necessário para migrar os dados para o novo tipo de armazenamento. Se você restaurar para ou de um armazenamento magnético, o processo de migração será o mais lento. Isso ocorre porque o armazenamento magnético não possui o recurso de armazenamento de IOPS provisionadas ou de Uso geral (SSD).

É possível usar o AWS CloudFormation para restaurar uma instância de banco de dados de um snapshot de instância de banco de dados. Para obter mais informações, consulte AWS::RDS::DBInstance, no Guia do usuário do AWS CloudFormation.

> **Note**
>
> Não é possível restaurar uma instância de banco de dados a partir de um DB snapshot que seja compartilhado e criptografado. Em vez disso, você pode fazer uma cópia do snapshot

de banco de dados e restaurar a instância de banco de dados a partir da cópia. Para ter mais informações, consulte Copiar um snapshot de banco de dados para o Amazon RDS.

Consulte informações sobre a restauração de uma instância de banco de dados com uma versão do Suporte estendido do RDS em Restauração de uma instância de banco de dados ou um cluster de banco de dados multi-AZ com o Suporte estendido do Amazon RDS.

Restauração a partir de um snapshot

É possível restaurar uma instância de banco de dados usando o AWS Management Console. a AWS CLI ou a API do RDS.

> **Note**
>
> Não é possível reduzir a quantidade de armazenamento ao restaurar uma instância de banco de dados. Ao aumentar o armazenamento alocado, o aumento deve ser de pelo menos 10%. Ao tentar aumentar o valor em menos de 10%, você obtém um erro. Não é possível aumentar o armazenamento alocado ao restaurar instâncias de banco de dados do RDS para SQL Server.

Console

Para restaurar uma instância de banco de dados de um DB snapshot

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Snapshots.
3. Escolha o DB snapshot do qual você deseja restaurar.
4. Em Actions (Ações), escolha Restore snapshot (Restaurar snapshot).
5. Na página Restore snapshot (Restaurar snapshot), em DB Instance Identifier (Identificador da instância do banco de dados), digite o nome da instância de bancos de dados restaurada.
6. Especifique outras configurações, como o tamanho de armazenamento alocado.

   Para obter informações sobre cada configuração, consulte Configurações para instâncias de banco de dados.
7. Escolha Restore DB Instance.

AWS CLI

Para restaurar uma instância de banco de dados a AWS CLIpartir de um snapshot de banco de dados, use o comando restore-db-instance-from-db-snapshot da .

Neste exemplo, restaure a partir de um snapshot de banco de dados chamado `mydbsnapshot`. Restaure em uma nova instância de banco de dados chamada `mynewdbinstance`. Esse exemplo também define o tamanho de armazenamento alocado.

Você pode especificar outras configurações. Para obter informações sobre cada configuração, consulte Configurações para instâncias de banco de dados.

Example

Para Linux, macOS ou Unix:

```
aws rds restore-db-instance-from-db-snapshot \
    --db-instance-identifier mynewdbinstance \
    --db-snapshot-identifier mydbsnapshot \
    --allocated-storage 100
```

Para Windows:

```
aws rds restore-db-instance-from-db-snapshot ^
    --db-instance-identifier mynewdbinstance ^
    --db-snapshot-identifier mydbsnapshot ^
    --allocated-storage 100
```

Esse comando retorna uma saída semelhante à seguinte:

```
DBINSTANCE  mynewdbinstance  db.t3.small  MySQL     50       sa             creating
 3  n  8.0.28  general-public-license
```

API do RDS

Para restaurar uma instância de banco de dados a partir de um snapshot de banco de dados, chame a função RestoreDBInstanceFromDBSnapshot da API do Amazon RDS com os seguintes parâmetros:

- `DBInstanceIdentifier`

- `DBSnapshotIdentifier`

Considerações

Consulte considerações de como restaurar uma instância de banco de dados de um snapshot de banco de dados nos tópicos a seguir.

Tópicos

- Considerações de grupos de parâmetros
- Considerações sobre os grupos de segurança
- Considerações de grupos de opções
- Considerações sobre marcação de recursos
- Considerações sobre o Db2
- Considerações sobre o Microsoft SQL Server
- Considerações sobre SQL
- Considerações sobre o Oracle Database

Considerações de grupos de parâmetros

Recomendamos que você retenha o grupo de parâmetros de banco de dados para quaisquer snapshots do banco de dados criados, para que você possa associar uma instância de banco de dados restaurada ao grupo de parâmetros correto.

O grupo de parâmetros de banco de dados padrão é associado à instância restaurada, a menos que você escolha um diferente. Não há configurações de parâmetros personalizados disponíveis no grupo de parâmetros padrão.

Você pode especificar o grupo de parâmetros ao restaurar a instância de banco de dados.

Para obter mais informações sobre os grupos de parâmetros de banco de dados, consulte Grupos de parâmetros para Amazon RDS.

Considerações sobre os grupos de segurança

Ao restaurar uma instância de banco de dados, a nuvem privada virtual (VPC) padrão, o grupo de sub-rede de banco de dados e o grupo de segurança da VPC são associados à instância restaurada, a menos que você escolha outros.

- Se estiver usando o console do Amazon RDS, você poderá especificar um grupo de segurança da VPC personalizado para associar à instância ou criar um novo grupo de segurança da VPC.
- Se estiver usando a AWS CLI, você poderá especificar um grupo de segurança da VPC personalizado para associar à instância incluindo a opção `--vpc-security-group-ids` no comando `restore-db-instance-from-db-snapshot`.
- Se você estiver usando a API do Amazon RDS, será possível incluir o parâmetro `VpcSecurityGroupIds.VpcSecurityGroupId.N` na ação `RestoreDBInstanceFromDBSnapshot`.

Assim que a restauração for concluída e anova instância de banco de dados estiver disponível, você também poderá alterar as configurações da VPC modificando a instância de banco de dados. Para obter mais informações, consulteModificar uma instância de banco de dados do Amazon RDS

Considerações de grupos de opções

Quando você restaura uma instância de banco de dados, o grupo de opções de banco de dados padrão é associado à instância restaurada.

A exceção é quando a fonte da instância de banco de dados é associada a um grupo de opções que contém uma opção persistente ou permanente. Por exemplo, se a fonte da instância de banco de dados usar o Oracle Transparent Data Encryption (TDE), a instância de banco de dados restaurada deverá usar um grupo de opções que tenha a opção TDE.

Se você restaurar uma instância de banco de dados em uma VPC diferente, deverá executar um dos seguintes procedimentos para atribuir um grupo de opções de banco de dados:

- Atribuir o grupo de opções padrão deste grupo da VPC à instância.
- Atribuir outro grupo de opções vinculado a essa VPC.
- Criar um novo grupo de opções e atribui-lo à instância de banco de dados. Com opções persistentes ou permanentes, como o Oracle TDE, você deve criar um novo grupo de opções que inclua a opção persistente ou permanente.

Para obter mais informações sobre grupos de opções de banco de dados, consulte Trabalhar com grupos de opções.

Considerações sobre marcação de recursos

Quando você restaura uma instância de banco de dados de um snapshot do banco de dados, o RDS verifica se você especifica novas tags. Se sim, as novas tags serão adicionadas à instância de banco de dados restaurada. Se não houver novas tags, o RDS adicionará as tags da instância de banco de dados de origem no momento da criação do snapshot à instância de banco de dados restaurada.

Para ter mais informações, consulte Copiar tags para snapshots de banco de dados.

Considerações sobre o Db2

Com o modelo BYOL, as instâncias de banco de dados do Amazon RDS para Db2 devem estar associadas a um grupo de parâmetros personalizado que contenha o IBM Site ID e o IBM Customer ID. Caso contrário, as tentativas de restaurar uma instância de banco de dados de um snapshot falharão. As instâncias de banco de dados do Amazon RDS para Db2 também devem estar associadas a uma licença autogerenciada do AWS License Manager. Para ter mais informações, consulte Traga a sua própria licença para Db2.

Com o modelo de licença do Db2 por meio do AWS Marketplace, você precisa de uma assinatura do AWS Marketplace ativa para a edição específica do IBM Db2 que deseja usar. Se ainda não tiver uma, assine o Db2 no AWS Marketplace para essa edição do IBM Db2. Para ter mais informações, consulte Licença do Db2 por meio do AWS Marketplace.

Considerações sobre o Microsoft SQL Server

Ao restaurar um snapshot de banco de dados do RDS for Microsoft SQL Server para uma nova instância, você sempre pode restaurar para a mesma edição do seu snapshot. Em alguns casos, você também pode alterar a edição da instância de banco de dados. As seguintes limitações são aplicáveis às alteração de edições:

- O snapshot de banco de dados deve ter armazenamento suficiente alocado para a nova edição.
- Somente há suporte para as seguintes alterações da edição:
  - Da Standard Edition para a Enterprise Edition
  - Da Web Edition para a Standard Edition ou a Enterprise Edition
  - Da Express Edition para Web Edition, Standard Edition ou Enterprise Edition

Para alterar de uma edição para uma nova que não é compatível via restauração de um snapshot, você poderá tentar usar o recurso de backup e restauração nativo. O SQL Server verifica se o seu

banco de dados é compatível com a nova edição com base em quais recursos do SQL Server você habilitou no banco de dados. Para ter mais informações, consulte Importar e exportar bancos de dados do SQL Server usando backup e restauração nativos.

Considerações sobre SQL

Para fazer uma restauração por meio de um snapshot de banco de dados do RDS para MySQL com uma versão de mecanismo incompatível, talvez seja necessário fazer a atualização do snapshot de banco de dados mais de uma vez. Consulte mais informações sobre as opções de atualização em Opções de atualização para snapshots de banco de dados com versões de mecanismo incompatíveis para o RDS para MySQL.

Consulte mais informações sobre como atualizar a versão de mecanismo de um snapshot de banco de dados do RDS para MySQL em Atualizar uma versão do mecanismo de snapshot de banco de dados do MySQL.

Considerações sobre o Oracle Database

Ao restaurar um banco de dados Oracle a partir de um DB snapshot, considere o seguinte:

- Antes de restaurar um snapshot de banco de dados, você pode atualizá-lo para uma versão posterior. Para ter mais informações, consulte Como atualizar para um snapshot de banco de dados Oracle.
- Se você restaurar um snapshot de uma instância CDB, poderá alterar o nome do PDB. Você não pode alterar os nomes do PDB quando sua instância do CDB usa a configuração multilocatário. Para ter mais informações, consulte Fazer backup e restaurar um CDB.
- Não é possível alterar o nome do CDB, que é sempre `RDSCDB`. Esse nome CDB é o mesmo para todas as instâncias CDB.
- Você não pode interagir diretamente com os bancos de dados de locatários em um DB snapshot. Se você restaurar um snapshot de uma instância CDB que usa a configuração multilocatário, restaurará todos os bancos de dados de locatários. Você pode usar describe-db-snapshot-tenant-databases para inspecionar os bancos de dados de locatários em um DB snapshot antes de restaurá-lo.
- Se você usar o Oracle GoldenGate, mantenha sempre o parameter group com o parâmetro `compatible`. Ao restaurar uma instância de banco de dados de um snapshot de banco de dados, você deve especificar o grupo de parâmetros que tem um valor de parâmetro correspondente a ou maior que `compatible`.

- É possível optar por renomear o banco de dados ao restaurar um snapshot de banco de dados. Se o tamanho total do redo log on-line for maior que 20 GB, o RDS poderá redefinir o tamanho do redo log on-line como as configurações padrão de 512 MB (4 x 128 MB). O tamanho menor permite que a operação de restauração seja concluída em um tempo razoável. É possível recriar os redo logs on-line posteriormente e alterar o tamanho.
- Você pode gerenciar sua senha de usuário principal no AWS Secrets Manager. Para ter mais informações, consulte Visão geral do gerenciamento de senhas do usuário principal com AWS Secrets Manager.

Restaurar uma instância de banco de dados para um momento especificado no Amazon RDS

É possível restaurar uma instância de banco de dados para um ponto no tempo específico criando uma instância de banco de dados sem modificar a instância de banco de dados de origem.

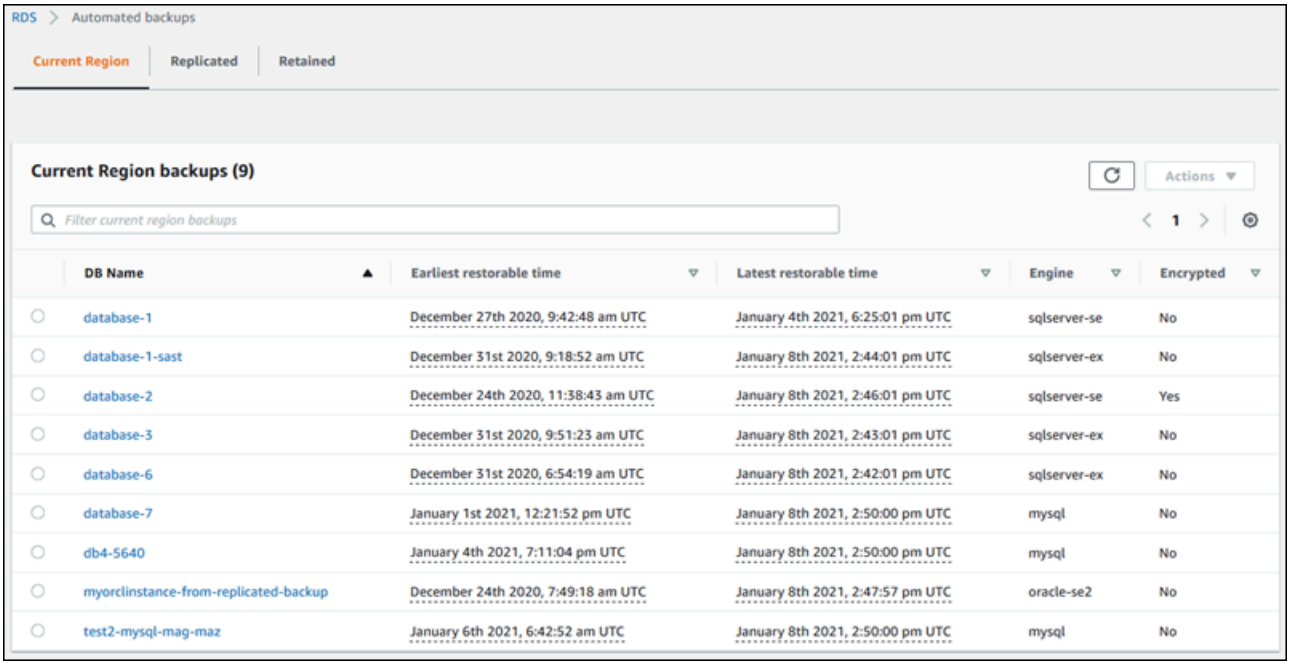
Ao restaurar uma instância de banco de dados para um momento específico, você pode escolher o grupo de segurança padrão da nuvem privada virtual (VPC). Ou você pode aplicar um grupo de segurança personalizado da VPC à sua instância de banco de dados.

As instâncias de banco de dados restauradas são associadas automaticamente aos grupos de parâmetros e de opções de banco de dados padrão. Porém, você pode aplicar um grupo de parâmetros personalizado e um grupo de opções personalizado especificando-os durante uma restauração.

Se a instância de banco de dados de origem tiver tags de recursos, o RDS adicionará as tags mais recentes à instância de banco de dados restaurada.

O RDS carrega os logs de transações de instâncias de banco de dados no Simple Storage Service (Amazon S3) a cada cinco minutos. Para visualizar o tempo restaurável mais recente para uma instância de banco de dado, use o comando AWS CLI describe-db-instances e confira o valor retornado no campo `LatestRestorableTime` para a instância de banco de dados. Para ver o tempo de restauração mais recente para cada instância de banco de dados no console Amazon RDS, selecione Backups automatizados.

É possível fazer a restauração para qualquer momento dentro do período de retenção de backup. Para ver o tempo de restauração mais antigo para cada instância de banco de dados, selecione Backups automatizados no console do Amazon RDS.



> **Note**
>
> Recomendamos restaurar para o mesmo tamanho de instância de banco de dados ou um tamanho semelhante (e IOPS se estiver usando o armazenamento de IOPS provisionadas) como a instância de banco de dados de origem. É possível receber um erro se, por exemplo, você escolher um tamanho de instância de banco de dados com um valor de IOPS incompatível.

Consulte informações sobre a restauração de uma instância de banco de dados com uma versão do Suporte estendido do RDS em Restauração de uma instância de banco de dados ou um cluster de banco de dados multi-AZ com o Suporte estendido do Amazon RDS.

Alguns dos mecanismos de banco de dados do Amazon RDS apresentam considerações especiais quando se utiliza a restauração pontual:

- Se você usar a autenticação por senha com uma instância de banco de dados do Amazon RDS para Db2, as ações de gerenciamento de usuários, inclusive `rdsadmin.add_user`, não serão capturadas nos logs. Essas ações exigem um backup de snapshot completo.

  Com o modelo BYOL, as instâncias de banco de dados do RDS para Db2 devem estar associadas a um grupo de parâmetros personalizado que contenha o IBM Site ID e o IBM Customer ID. Caso contrário, as tentativas de restaurar uma instância de banco de dados para um ponto específico no

tempo falharão. As instâncias de banco de dados do Amazon RDS para Db2 também devem estar associadas a uma licença autogerenciada do AWS License Manager. Para ter mais informações, consulte Traga a sua própria licença para Db2.

Com o modelo de licença do Db2 por meio do AWS Marketplace, você precisa de uma assinatura do AWS Marketplace ativa para a edição específica do IBM Db2 que deseja usar. Se ainda não tiver uma, assine o Db2 no AWS Marketplace para essa edição do IBM Db2. Para ter mais informações, consulte Licença do Db2 por meio do AWS Marketplace.

- Ao fazer uma restauração pontual em uma instância de banco de dados do RDS para Oracle, você pode especificar um mecanismo de banco de dados, modelo de licença e DBName (SID) diferentes para a instância de banco de dados restaurada. Também é possível especificar que o RDS deve armazenar e gerenciar as senhas de usuário principal no AWS Secrets Manager. Para ter mais informações, consulte Visão geral do gerenciamento de senhas do usuário principal com AWS Secrets Manager.
- Quando você restaura uma instância de banco de dados do SQL Server em um ponto no tempo, cada banco de dados nessa instância é restaurado a um ponto no tempo dentro de 1 segundo de cada outro banco de dados dentro da instância. As transações que abrangem vários bancos de dados dentro da instância podem ser restauradas de forma inconsistente.
- Para uma instância de banco de dados SQL Server, os modos `OFFLINE`, `EMERGENCY` e `SINGLE_USER` não são compatíveis. Configurar qualquer banco de dados em um desses modos faz com que o tempo restaurável mais recente pare de avançar para toda a instância.
- Algumas ações, como altera o modelo de recuperação de um banco de dados do SQL Server, podem interromper a sequência de logs que são usados para a recuperação em um ponto anterior no tempo. Em alguns casos, o Amazon RDS pode detectar esse problema e o mais recente tempo restaurável é impedido de avançar. Em outros casos, como quando um banco de dados do SQL Server usa o modelo de recuperação `BULK_LOGGED`, a interrupção na sequência de log não é detectada. Pode não ser possível restaurar uma instância de banco de dados do SQL Server a um ponto no tempo, se houver uma interrupção na sequência de log. Por esses motivos, o Amazon RDS não oferece suporte à alteração do modelo de recuperação dos bancos de dados do SQL Server.

Você também pode usar o AWS Backup para gerenciar backups de instâncias de banco de dados do Amazon RDS. Se sua instância de banco de dados estiver associada a um plano de backup no AWS Backup, esse plano de backup será usado para recuperação em um ponto anterior no tempo. Backups criados com o AWS Backup têm nomes que terminam com `awsbackup:`*`AWS-Backup-`*

job-number. Para obter mais informações sobre o AWS Backup, consulte o Guia do desenvolvedor do AWS Backup.

> **Note**
>
> As informações neste tópico se aplicam ao Amazon RDS. Para obter informações sobre como restaurar um cluster de bancos de dados Amazon Aurora consulte Restaurar um cluster de banco de dados em um horário especificado.

Você pode restaurar uma instância de banco de dados em um momento específico usando o AWS Management Console, a AWS CLI ou a API do RDS.

> **Note**
>
> Não é possível reduzir a quantidade de armazenamento ao restaurar uma instância de banco de dados. Ao aumentar o armazenamento alocado, o aumento deve ser de pelo menos 10%. Ao tentar aumentar o valor em menos de 10%, você obtém um erro. Não é possível aumentar o armazenamento alocado ao restaurar instâncias de banco de dados do RDS para SQL Server.

Console

Para restaurar uma instância de banco de dados em um horário específico

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Automated backups (Backups automatizados).

   Os backups automatizados são exibidos na guia Current Region (região atual).
3. Escolha a instância de banco de dados que você quer restaurar.
4. Em Actions (Ações), escolha Restore to point in time (Restaurar para point-in-time).

   A janela Restore to point in time (Restaurar para point-in-time) é exibida.
5. Escolha Latest restorable time (Hora da última restauração) para restaurar no último horário possível ou escolha Custom (Personalizar) para escolher um horário.

Se você escolher Custom (Personalizado), insira a data e a hora para a qual deseja restaurar a instância.

> **Note**
>
> Os horários são mostrados no fuso horário local, que é indicado por um deslocamento do Tempo Universal Coordenado (UTC). Por exemplo, UTC-5 é a Hora Padrão do Leste dos EUA/Horário de Verão Central.

6. Em Identificador de instância de banco de dados, digite o nome da instância de banco de dados restaurada de destino. O nome deve ser exclusivo.
7. Escolha outras opções conforme necessário, como classe de instância de banco de dados, armazenamento e se você quer usar a escalabilidade automática de armazenamento.

   Para obter informações sobre cada configuração, consulte Configurações para instâncias de banco de dados.
8. Escolha Restore to point in time (Restaurar para point-in-time).

AWS CLI

Para restaurar uma instância de banco de dados de volta para um horário especificado, use o comando restore-db-instance-to-point-in-time da AWS CLI para criar uma instância de banco de dados. Esse exemplo também define o tamanho de armazenamento alocado e habilita a autoescalabilidade do armazenamento.

A marcação de recursos é compatível com esta operação. Quando você usa a opção `--tags`, as tags da instância de banco de dados de origem são ignoradas e as fornecidas são utilizadas. Caso contrário, as tags mais recentes da instância de origem serão utilizadas.

Você pode especificar outras configurações. Para obter informações sobre cada configuração, consulte Configurações para instâncias de banco de dados.

Example

Para Linux, macOS ou Unix:

```
aws rds restore-db-instance-to-point-in-time \
    --source-db-instance-identifier mysourcedbinstance \
```

```
    --target-db-instance-identifier mytargetdbinstance \
    --restore-time 2017-10-14T23:45:00.000Z \
    --allocated-storage 100 \
    --max-allocated-storage 1000
```

Para Windows:

```
aws rds restore-db-instance-to-point-in-time ^
    --source-db-instance-identifier mysourcedbinstance ^
    --target-db-instance-identifier mytargetdbinstance ^
    --restore-time 2017-10-14T23:45:00.000Z ^
    --allocated-storage 100 ^
    --max-allocated-storage 1000
```

API do RDS

Para restaurar uma instância de banco de dados em um horário específico, chame a operação `RestoreDBInstanceToPointInTime` da API do Amazon RDS com os seguintes parâmetros:

- `SourceDBInstanceIdentifier`
- `TargetDBInstanceIdentifier`
- `RestoreTime`

Restaurar um cluster de banco de dados multi-AZ para um horário especificado

É possível restaurar seu cluster de banco de dados multi-AZ para um momento específico criando um cluster de banco de dados multi-AZ.

O RDS carrega logs de transações dos clusters de banco de dados multi-AZ no Amazon S3 continuamente. É possível fazer a restauração para qualquer momento dentro do período de retenção de backup. Para ver o horário restaurável mais antigo para um cluster de banco de dados multi-AZ, use o comando da AWS CLI describe-db-clusters. Veja o valor retornado no campo `EarliestRestorableTime` para o cluster de banco de dados. Para visualizar o tempo restaurável mais recente de um cluster de banco de dados multi-AZ, confira o valor retornado no campo `LatestRestorableTime` do cluster de banco de dados.

Ao restaurar um cluster de banco de dados multi-AZ em um ponto no tempo, você pode escolher o grupo de segurança da VPC padrão para o cluster de banco de dados multi-AZ ou aplicar um grupo de segurança da VPC personalizado ao cluster de banco de dados multi-AZ.

Os clusters de banco de dados multi-AZ restaurados são automaticamente associados ao grupo de parâmetros de cluster de banco de dados padrão. No entanto, você pode aplicar um grupo de parâmetros de cluster de banco de dados personalizado especificando-o durante uma restauração.

Se o cluster de banco de dados de origem tiver tags de recursos, o RDS adicionará as tags mais recentes ao cluster de banco de dados restaurado.

> **Note**
>
> Convém restaurar para o mesmo tamanho de cluster de banco de dados multi-AZ ou similar que o cluster de banco de dados de origem. Também recomendamos que você restaure com o mesmo valor de IOPS ou similar se estiver usando o armazenamento de IOPS provisionadas. É possível receber um erro se, por exemplo, você escolher um tamanho de cluster de banco de dados com um valor de IOPS incompatível.
>
> Se o cluster de banco de dados multi-AZ de origem usar o armazenamento SSD de uso geral (gp3) e tiver menos de 400 GiB de armazenamento alocado, você não poderá modificar as IOPS provisionadas para o cluster de banco de dados restaurado.

Consulte informações sobre a restauração de cluster de banco de dados multi-AZ com uma versão do Suporte estendido do RDS em Restauração de uma instância de banco de dados ou um cluster de banco de dados multi-AZ com o Suporte estendido do Amazon RDS.

Você pode restaurar um cluster de banco de dados multi-AZ para um ponto no tempo usando 0 AWS Management Console, a AWS CLI ou a API do RDS.

Console

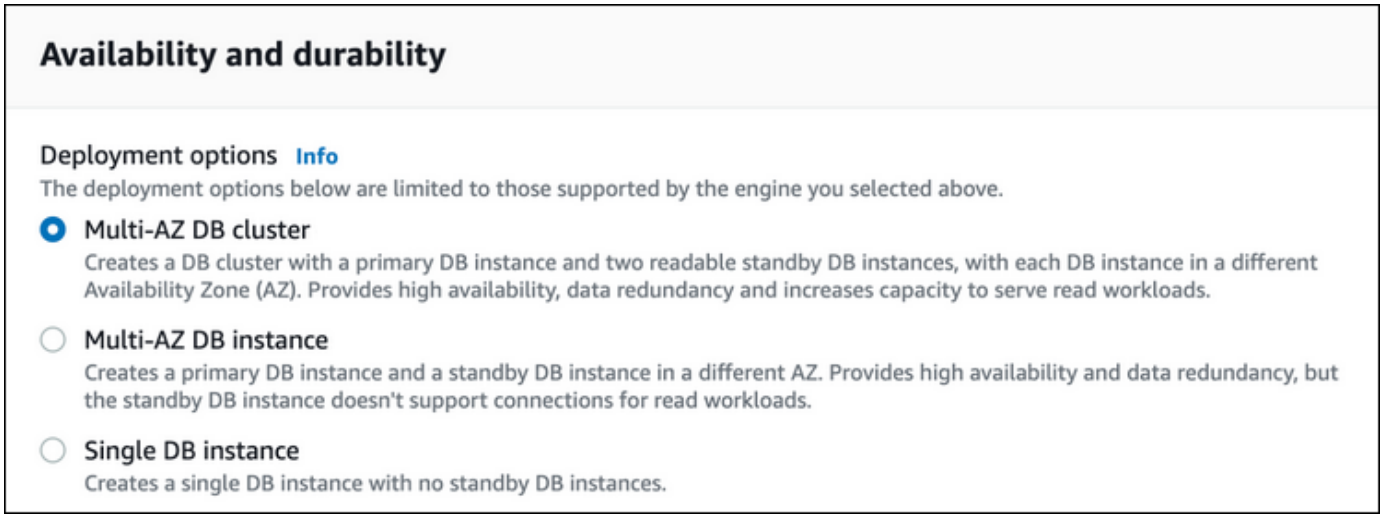
Para restaurar um cluster de banco de dados multi-AZ para um horário especificado

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Bancos de dados.
3. Escolha o cluster de banco de dados multi-AZ que você deseja restaurar.
4. Em Actions (Ações), escolha Restore to point in time (Restaurar para point-in-time).

   A janela Restore to point in time (Restaurar para point-in-time) é exibida.
5. Escolha Latest restorable time (Hora da última restauração) para restaurar no último horário possível ou escolha Custom (Personalizar) para escolher um horário.

   Se você escolher Personalizar, insira a data e a hora para a qual deseja restaurar o cluster de banco de dados multi-AZ.

   > **Note**
   >
   > Os horários são mostrados no fuso horário local, que é indicado por um deslocamento do Tempo Universal Coordenado (UTC). Por exemplo, UTC-5 é a Hora Padrão do Leste dos EUA/Horário de Verão Central.
6. Para identificador de cluster de banco de dados, insira o nome do cluster de banco de dados multi-AZ recém-restaurado.
7. Em Availability and durability (Disponibilidade e durabilidade), escolha Multi-AZ DB cluster (Cluster de banco de dados multi-AZ).



8. Em DB instance class (Classe de instância do banco de dados), escolha uma classe de instância de banco de dados.

   Atualmente, os clusters de banco de dados multi-AZ são compatíveis apenas com classes de instância de banco de dados db.m6gd e db.r6gd. Para ter mais informações sobre classes de instância de banco de dados, consulte Classes de instâncias de banco de dados .
9. Nas seções restantes, especifique suas configurações de cluster de banco de dados. Para obter informações sobre cada configuração, consulte Configurações para criar clusters de banco de dados multi-AZ.
10. Escolha Restore to point in time (Restaurar para point-in-time).

AWS CLI

Para restaurar um cluster de banco de dados multi-AZ para um horário especificado, use o comando da AWS CLI restore-db-cluster-to-point-in-time para criar um novo cluster de banco de dados multi-AZ.

Atualmente, os clusters de banco de dados multi-AZ são compatíveis apenas com classes de instância de banco de dados db.m6gd e db.r6gd. Para ter mais informações sobre classes de instância de banco de dados, consulte Classes de instâncias de banco de dados .

Example

Para Linux, macOS ou Unix:

```
aws rds restore-db-cluster-to-point-in-time \
    --source-db-cluster-identifier mysourcemultiazdbcluster \
```

```
    --db-cluster-identifier mytargetmultiazdbcluster \
    --restore-to-time 2021-08-14T23:45:00.000Z \
    --db-cluster-instance-class db.r6gd.xlarge
```

Para Windows:

```
aws rds restore-db-cluster-to-point-in-time ^
    --source-db-cluster-identifier mysourcemultiazdbcluster ^
    --db-cluster-identifier mytargetmultiazdbcluster ^
    --restore-to-time 2021-08-14T23:45:00.000Z ^
    --db-cluster-instance-class db.r6gd.xlarge
```

API do RDS

Para restaurar um cluster de bancos de dados em um horário específico, chame a operação RestoreDBClusterToPointInTime da API do Amazon RDS com os seguintes parâmetros:

- `SourceDBClusterIdentifier`
- `DBClusterIdentifier`
- `RestoreToTime`

Restaurar a partir de um snapshot para cluster de banco de dados multi-AZ

É possível restaurar um snapshot para um cluster de banco de dados multi-AZ utilizando o AWS Management Console, a AWS CLI ou a API do RDS. Você pode restaurar cada um desses tipos de snapshots para um cluster de banco de dados multi-AZ:

- Um snapshot de uma implantação single-AZ
- Um snapshot de uma implantação de cluster de banco de dados multi-AZ com uma única instância de banco de dados
- Um snapshot de um cluster de banco de dados multi-AZ

Para obter mais informações sobre implantações multi-AZ, consulte Configurar e gerenciar uma implantação multi-AZ para o Amazon RDS.

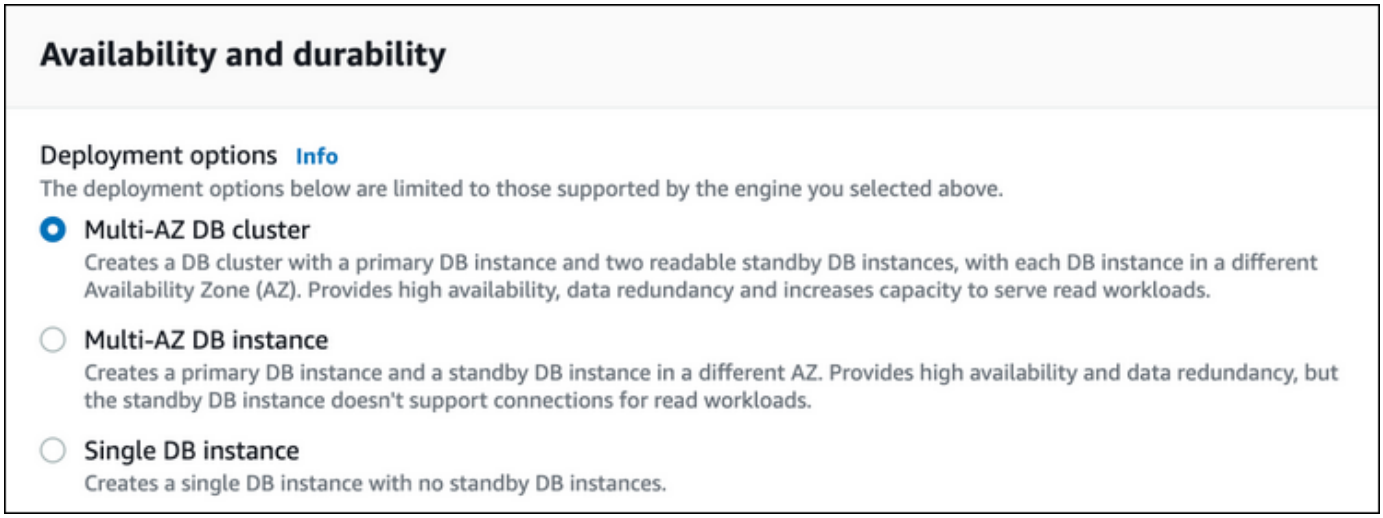
> **Tip**
>
> Você pode migrar uma implantação de uma única zona de disponibilidade ou uma implantação de instância de banco de dados multi-AZ para uma implantação de cluster de banco de dados multi-AZ restaurando um snapshot.

Consulte informações sobre a restauração de um cluster de banco de dados multi-AZ com uma versão do Suporte estendido do RDS em Restauração de uma instância de banco de dados ou um cluster de banco de dados multi-AZ com o Suporte estendido do Amazon RDS.

Console

Para restaurar um snapshot para um cluster de banco de dados multi-AZ

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Snapshots.
3. Escolha o snapshot do qual você deseja restaurar.
4. Em Actions (Ações), escolha Restore snapshot (Restaurar snapshot).
5. Na página Restore snapshot (Restaurar snapshot), em Availability and durability (Disponibilidade e durabilidade), escolha Multi-AZ DB cluster (Cluster de banco de dados multi-AZ).



6. Para identificador de cluster de banco de dados, insira o nome do cluster de banco de dados multi-AZ recém-restaurado.
7. Nas seções restantes, especifique suas configurações de cluster de banco de dados. Para obter informações sobre cada configuração, consulte Configurações para criar clusters de banco de dados multi-AZ.
8. Escolha Restore DB Instance.

AWS CLI

Para restaurar um snapshot para um cluster de banco de dados multi-AZ, use o comando da AWS CLI restore-db-cluster-from-snapshot.

No exemplo a seguir, restaure a partir de um instantâneo já criado, chamado `mysnapshot`. Restaure em um novo cluster de banco de dados multi-AZ chamado `mynewmultiazdbcluster`. Você também especifica a classe da instância de banco de dados utilizada pelas instâncias de banco de dados no cluster de banco de dados multi-AZ. Especifique um `mysql` ou `postgres` para o mecanismo de banco de dados.

Para a opção `--snapshot-identifier`, você pode usar o nome ou o nome do recurso da Amazon (ARN) para especificar um snapshot do cluster de banco de dados. No entanto, é possível usar apenas o ARN para especificar um DB snapshot.

Para a opção `--db-cluster-instance-class`, especifique a classe da instância de banco de dados para o novo cluster de banco de dados multi-AZ. Os clusters de banco de dados multi-AZ são compatíveis apenas com algumas classes de instância de banco de dados específicas,

como db.m6gd e db.r6gd. Para ter mais informações sobre classes de instância de banco de dados, consulte Classes de instâncias de banco de dados .

Você também pode especificar outras opções.

Example

Para Linux, macOS ou Unix:

```
aws rds restore-db-cluster-from-snapshot \
    --db-cluster-identifier mynewmultiazdbcluster \
    --snapshot-identifier mysnapshot \
    --engine mysql|postgres \
    --db-cluster-instance-class db.r6gd.xlarge
```

Para Windows:

```
aws rds restore-db-cluster-from-snapshot ^
    --db-cluster-identifier mynewmultiazdbcluster ^
    --snapshot-identifier mysnapshot ^
    --engine mysql|postgres ^
    --db-cluster-instance-class db.r6gd.xlarge
```

Depois de restaurar o cluster de banco de dados, você poderá adicionar o cluster de banco de dados multi-AZ ao grupo de segurança associado ao cluster de banco de dados ou à instância de banco de dados utilizada para criar o snapshot, se aplicável. A conclusão desta ação fornece as mesmas funções do cluster ou da instância de banco de dados anterior.

API do RDS

Para restaurar um snapshot em um cluster de banco de dados multi-AZ, chame a operação de API do RDS RestoreDBClusterFromSnapshot com os seguintes parâmetros:

- `DBClusterIdentifier`
- `SnapshotIdentifier`
- `Engine`

Você também pode especificar outros parâmetros opcionais.

Depois de restaurar o cluster de banco de dados, você poderá adicionar o cluster de banco de dados multi-AZ ao grupo de segurança associado ao cluster de banco de dados ou à instância de banco

de dados utilizada para criar o snapshot, se aplicável. A conclusão desta ação fornece as mesmas funções do cluster ou da instância de banco de dados anterior.

Restauração de um snapshot de cluster de banco de dados multi-AZ para uma instância de banco de dados

Um snapshot de cluster de banco de dados multi-AZ é um snapshot de volume de armazenamento do seu cluster de banco de dados que faz backup de todo o cluster de banco de dados, e não apenas dos bancos de dados individuais. Você pode restaurar um snapshot de cluster de banco de dados multi-AZ para uma implantação single-AZ ou uma implantação de instância de banco de dados multi-AZ. Para obter mais informações sobre implantações multi-AZ, consulte Configurar e gerenciar uma implantação multi-AZ para o Amazon RDS.

> **Note**
>
> Também é possível restaurar um snapshot de cluster de banco de dados multi-AZ para um novo cluster de banco de dados multi-AZ. Para obter instruções, consulte Restaurar a partir de um snapshot para cluster de banco de dados multi-AZ.

Consulte informações sobre a restauração de cluster de banco de dados multi-AZ com uma versão do Suporte estendido do RDS em Restauração de uma instância de banco de dados ou um cluster de banco de dados multi-AZ com o Suporte estendido do Amazon RDS.

Use o AWS Management Console, a AWS CLI ou a API do RDS para restaurar um snapshot de cluster de banco de dados multi-AZ para uma implantação single-AZ ou uma implantação de instância de banco de dados multi-AZ.

Console

Como restaurar um snapshot de cluster de banco de dados multi-AZ para uma implantação single-AZ ou uma implantação de instância de banco de dados multi-AZ

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Snapshots.
3. Escolha o snapshot de cluster de banco de dados multi-AZ que deseja restaurar.
4. Em Actions (Ações), escolha Restore snapshot (Restaurar snapshot).
5. Na página Restore snapshot (Restaurar snapshot), em Availability and durability (Disponibilidade e durabilidade), escolha uma das seguintes opções:

- Single DB instance (Instância de banco de dados única): restaura o snapshot em uma instância de banco de dados sem instância de banco de dados em espera.
- Multi-AZ DB instance (Instância de banco de dados multi-AZ): restaura o snapshot para uma implantação de instância de banco de dados multi-AZ com uma instância de banco de dados primária e uma instância de banco de dados em espera.

6. Em DB instance identifier (Identificador da instância de banco de dados), insira o nome da instância de banco de dados restaurada.
7. Nas seções restantes, especifique suas configurações de instância de banco de dados. Para obter informações sobre cada configuração, consulte Configurações para instâncias de banco de dados.
8. Escolha Restore DB Instance.

AWS CLI

Para restaurar um snapshot de cluster de banco de dados para uma implantação de instância de banco de dados, use o comando restore-db-instance-from-db-snapshot da AWS CLI.

O exemplo a seguir restaura um snapshot de cluster de banco de dados multi-AZ já criado chamado `myclustersnapshot`. A restauração ocorre para uma nova implantação de instância de banco de dados multi-AZ com uma instância de banco de dados primária chamada `mynewdbinstance`. Para a opção `--db-cluster-snapshot-identifier`, especifique o nome do snapshot do cluster de banco de dados multi-AZ.

Para a opção `--db-instance-class`, especifique a classe da instância de banco de dados para a implantação da nova instância de banco de dados. Para ter mais informações sobre classes de instância de banco de dados, consulte Classes de instâncias de banco de dados .

Você também pode especificar outras opções.

Example

Para Linux, macOS ou Unix:

```
aws rds restore-db-instance-from-db-snapshot \
    --db-instance-identifier mynewdbinstance \
    --db-cluster-snapshot-identifier myclustersnapshot \
    --engine mysql \
    --multi-az \
```

```
    --db-instance-class db.r6g.xlarge
```

Para Windows:

```
aws rds restore-db-instance-from-db-snapshot ^
    --db-instance-identifier mynewdbinstance ^
    --db-cluster-snapshot-identifier myclustersnapshot ^
    --engine mysql ^
    --multi-az ^
    --db-instance-class db.r6g.xlarge
```

Depois de restaurar a instância de banco de dados, você poderá adicioná-la ao grupo de segurança associado ao cluster de banco de dados multi-AZ usado para criar o snapshot, se aplicável. A conclusão desta ação fornece as mesmas funções do cluster de banco de dados multi-AZ anterior.

API do RDS

Para restaurar um snapshot de cluster de banco de dados multi-AZ para uma implantação de instância de banco de dados, chame a operação de API do RDS RestoreDBInstanceFromDBSnapshot com os seguintes parâmetros:

- `DBInstanceIdentifier`
- `DBClusterSnapshotIdentifier`
- `Engine`

Você também pode especificar outros parâmetros opcionais.

Depois de restaurar a instância de banco de dados, você poderá adicioná-la ao grupo de segurança associado ao cluster de banco de dados multi-AZ usado para criar o snapshot, se aplicável. A conclusão desta ação fornece as mesmas funções do cluster de banco de dados multi-AZ anterior.

Tutorial: restaurar uma instância de banco de dados do Amazon RDS de um snapshot do banco de dados

Com frequência, ao trabalhar com o Amazon RDS, você pode ter uma instância de banco de dados com a qual trabalha ocasionalmente, mas que não precisa usar em tempo integral. Por exemplo, suponha que você tenha uma pesquisa trimestral de clientes que usa uma instância do Amazon EC2 para hospedar um site de pesquisa de clientes. Você também tem uma instância de banco de dados utilizada para armazenar os resultados da pesquisa. Uma maneira de economizar dinheiro nesse cenário é obter um snapshot da instância de banco de dados após a conclusão da pesquisa. Depois, você exclui a instância de banco de dados e a restaura quando precisa realizar a pesquisa novamente.

Ao restaurar a instância de banco de dados, você fornece o nome do snapshot de banco de dados do qual deseja restaurar. Depois, você fornece um nome para a nova instância de banco de dados criada na operação de restauração.

Para ter mais informações sobre a restauração de uma instância de banco de dados a partir de snapshots, consulte Restaurar uma instância de banco de dados.

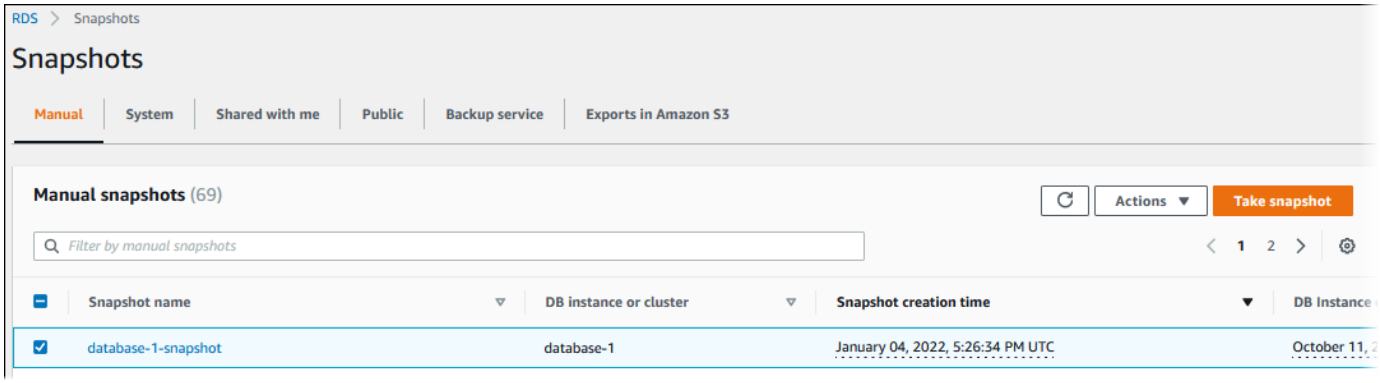
Para ter informações sobre o gerenciamento de chaves do AWS KMS para o Amazon RDS, consulte Gerenciamento de AWS KMS key.

Restaurar uma instância de banco de dados de um snapshot de banco de dados

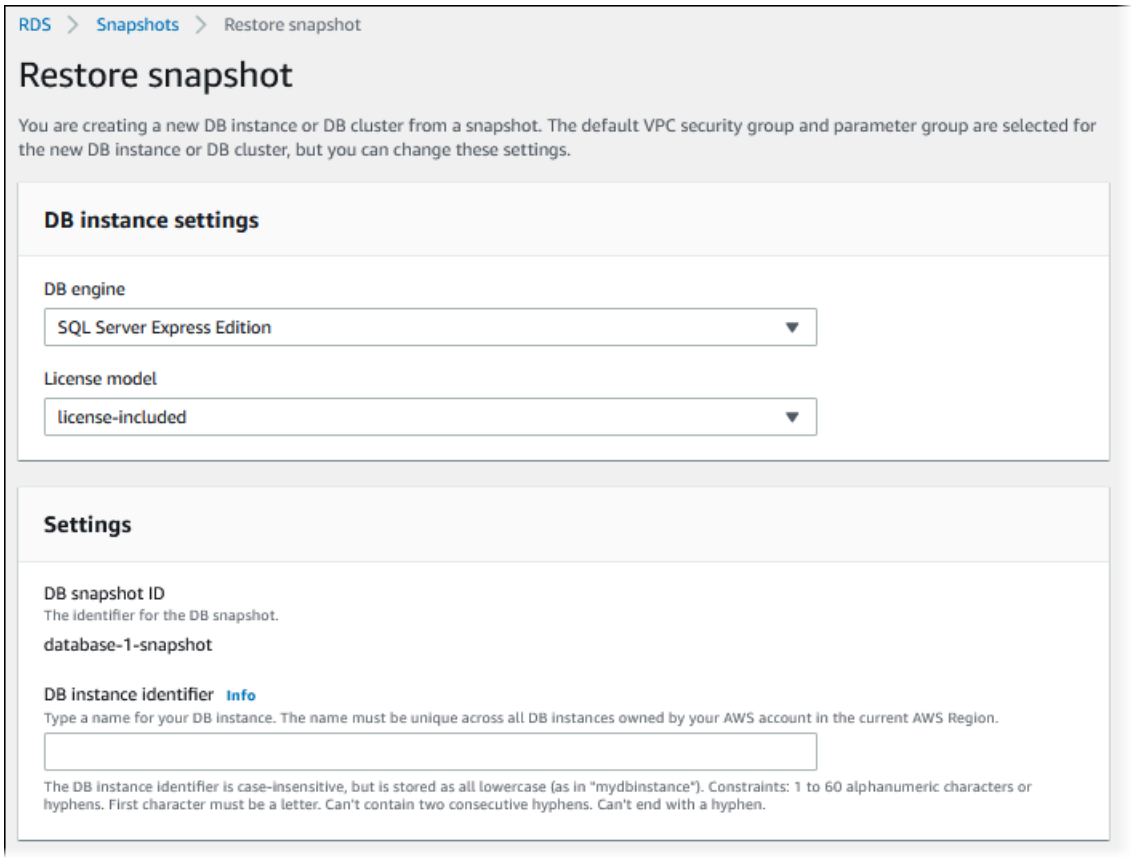
Use o procedimento a seguir para restaurar de um snapshot no AWS Management Console.

Para restaurar uma instância de banco de dados de um DB snapshot

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Snapshots.
3. Escolha o DB snapshot do qual você deseja restaurar.
4. Em Actions (Ações), escolha Restore snapshot (Restaurar snapshot).



A página Restore snapshot (Restaurar snapshot) é exibida.



5. Em DB instance settings (Configurações da instância de banco de dados), use as configurações padrão para DB engine (Mecanismo de banco de dados) e License model (Modelo de licença) (para Oracle ou Microsoft SQL Server).
6. Em Settings (Configurações), para DB instance identifier (Identificador de instância de banco de dados), insira o nome exclusivo que você quer usar na instância de banco de dados restaurada, por exemplo, **`mynewdbinstance`**.

Se você estiver restaurando de uma instância de banco de dados excluída após fazer o snapshot de banco de dados, será possível usar o nome dessa instância de banco de dados.

7. Em Disponibilidade e durabilidade, escolha se quer criar uma instância em espera em outra zona de disponibilidade.

   Para este tutorial, não crie uma instância em espera.

8. Em Connectivity (Conectividade, use as configurações padrão para o seguinte:

   - Nuvem privada virtual (VPC)
   - DB subnet group (Grupo de subredes do banco de dados)
   - Acesso público
   - VPC security group (firewall) [Grupo de segurança da VPC (firewall)]

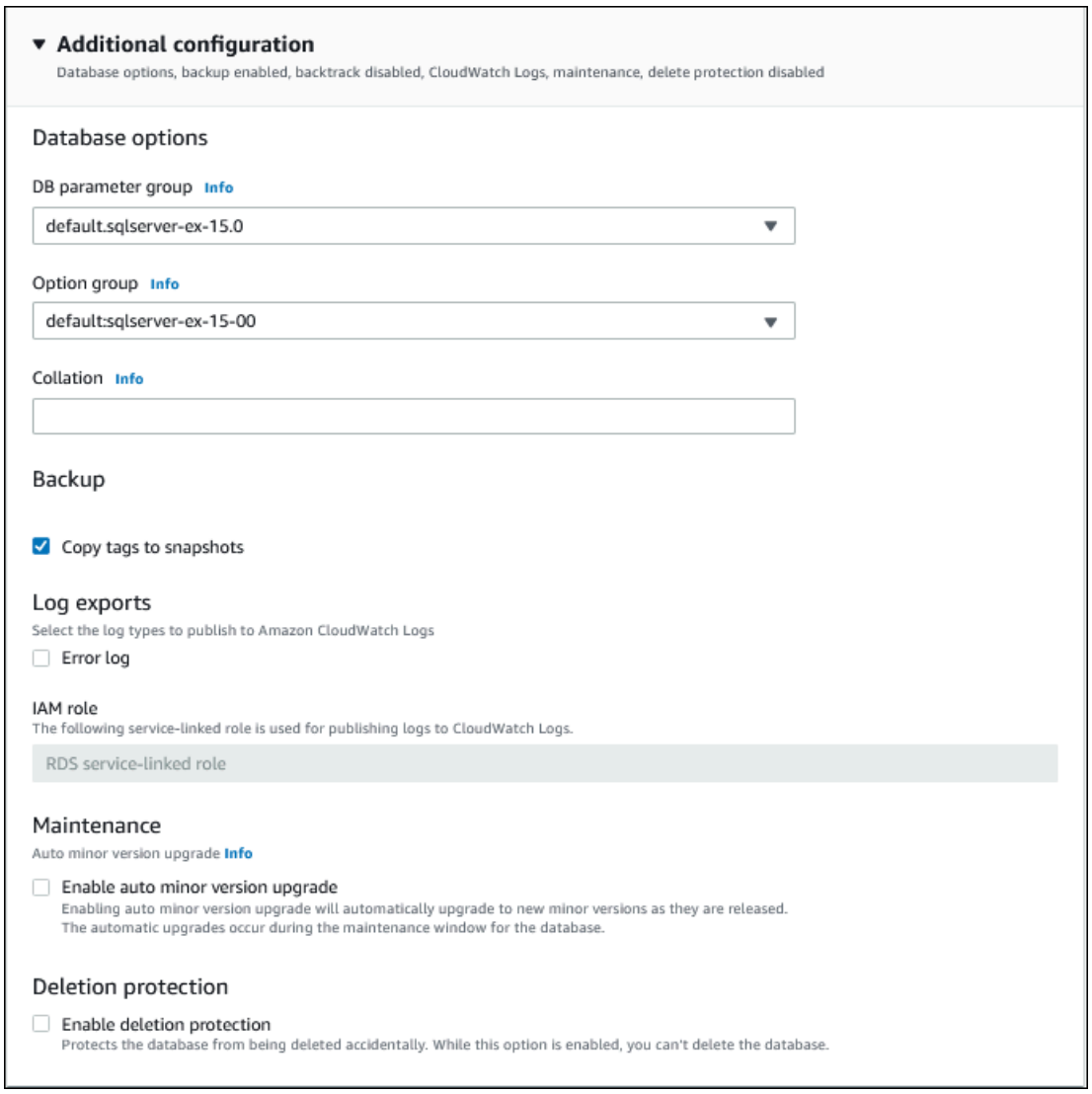
9. Escolha a DB instance class (Classe da instância de banco de dados)

   Para este tutorial, escolha Burstable classes (includes t classes) (Classes com capacidade de intermitência (inclui classes t)) e, depois, escolha db.t3.small.

10. Para Encryption (Criptografia), use as configurações padrão.

    Se a fonte da instância de banco de dados do snapshot tiver sido criptografada, a instância de banco de dados restaurada também será criptografada. Não é possível fazer isso sem criptografia.

11. Expanda Additional configuration (Configuração adicional) na parte inferior da página.



12. Faça o seguinte em Database options (Opções do banco de dados):

    a. Escolha o DB parameter group (Grupo de parâmetros de banco de dados).

       Para este tutorial, use o grupo de parâmetros padrão.

    b. Escolha o Option group (Grupo de opções).

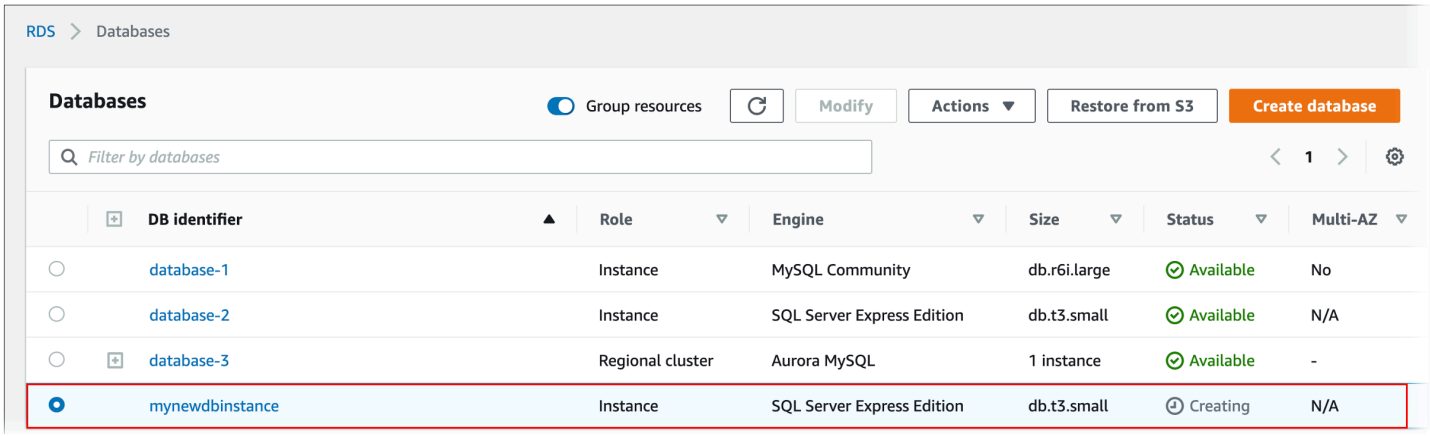
       Para este tutorial, use o grupo de opções padrão.

> **Important**
>
> Em alguns casos, você pode restaurar a partir de um snapshot de banco de dados que use uma opção persistente ou permanente. Em caso afirmativo, selecione um grupo de opções que use a mesma opção.

c. Para Deletion protection (Proteção contra exclusão), escolha Enable deletion protection (Habilitar proteção contra exclusão).

13. Escolha Restore DB Instance.

A página Databases (Banco de dados) exibe a instância de banco de dados restaurada, com um status de `Creating`.



Copiar um snapshot de banco de dados para o Amazon RDS

Com o Amazon RDS, é possível copiar snapshots de banco de dados manuais ou backups automatizados. Depois de copiar um snapshot, a cópia é um snapshot manual. É possível fazer várias cópias de um backup automatizado ou snapshot manual, mas cada um deve ter um identificador exclusivo.

É possível copiar um snapshot dentro da mesma Região da AWS ou entre Regiões da AWS e também copiar snapshots compartilhados.

Copiar um snapshot de banco de dados

Para cada Conta da AWS, é possível copiar até vinte snapshots de banco de dados por vez de uma Região da AWS em outra. Se copiar um snapshot de banco de dados para outra Região da AWS, você criará um snapshot de banco de dados manual que será retido nessa Região da AWS. Copiar um snapshot de banco de dados fora da Região da AWS de origem resultará em cobranças de transferência de dados do Amazon RDS.

Para obter mais informações sobre a definição de preço da transferência de dados, consulte Definição de preço do Amazon RDS.

Depois que a cópia do snapshot de banco de dados tiver sido criada na nova Região da AWS, ela se comportará da mesma forma como todos os outros snapshots de banco de dados nessa Região da AWS.

Você pode copiar um snapshot de banco de dados usando o AWS Management Console, a AWS CLI ou a API do RDS.

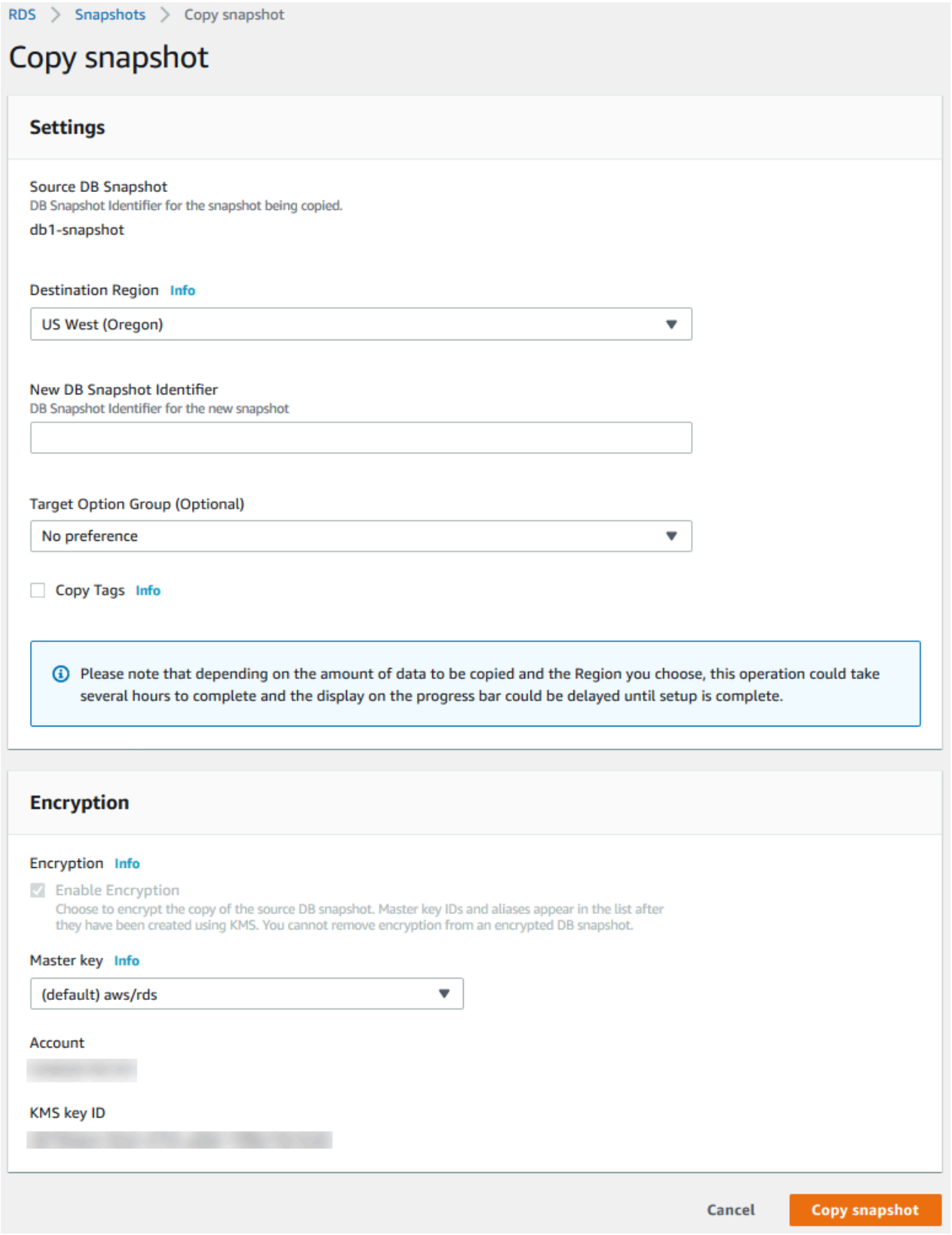
Console

O procedimento a seguir copia um snapshot de banco de dados criptografado ou não criptografado na mesma Região da AWS, ou entre regiões, usando o AWS Management Console.

Para copiar um snapshot de banco de dados

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Snapshots.
3. Selecione o snapshot de banco de dados que você deseja copiar.
4. Para Actions (Ações), escolha Copy Snapshot (Copiar snapshot).

A página Copy snapshot (Copiar instantâneo) aparece.



5. Em Target option group (Grupo de opções de destino), selecione um novo grupo de opções.

Especifique essa opção se estiver copiando um snapshot de uma Região da AWS para outra, e sua instância de banco de dados usar um grupo de opções diferente do padrão.

Se a sua instância de banco de dados de origem usar a Transparent Data Encryption for Oracle ou o Microsoft SQL Server, você deverá especificar essa opção ao fazer cópias entre regiões. Para obter mais informações, consulte Considerações sobre grupos de opções.

6. (Opcional) Para copiar o snapshot de banco de dados para uma Região da AWS diferente, em Destination Region (Região de destino), escolha a nova Região da AWS.

> **Note**
>
> A Região da AWS de destino deve ter a mesma versão de mecanismo de banco de dados disponível que a Região da AWS de origem.

7. Em New DB snapshot identifier (Novo identificador de snapshot de banco de dados), digite o nome da cópia do snapshot de banco de dados.

   É possível fazer várias cópias de um backup automatizado ou snapshot manual, mas cada cópia deve ter um identificador exclusivo.

8. (Opcional) Selecione Copy Tags (Copiar tags) para copiar tags e valores do snapshot para a cópia do snapshot.

9. (Opcional) Para Encryption (Criptografia), faça o seguinte:

   a. Escolha Enable encryption (Habilitar criptografia), se o snapshot de banco de dados não estiver criptografado, mas você quiser criptografar a cópia.

   > **Note**
   >
   > Se o snapshot de banco de dados estiver criptografado, você deverá criptografar a cópia, portanto a caixa de seleção já estará marcada.

   b. Em AWS KMS key, especifique o identificador da chave do KMS a ser usado para criptografar a cópia do snapshot do banco de dados.

10. Escolha Copy snapshot (Copiar snapshot).

AWS CLI

Você pode copiar um snapshot de banco de dados usando o comando copy-db-snapshot da AWS CLI. Se você estiver copiando o snapshot para uma nova Região da AWS, execute o comando na nova Região da AWS.

As seguintes opções são usadas para copiar um snapshot de banco de dados. Nem todas as opções são necessárias para todos os cenários. Use as descrições e os exemplos a seguir para determinar quais opções usar.

- `--source-db-snapshot-identifier` – O identificador do snapshot de banco de dados de origem.
  - Se o snapshot de origem estiver na mesma Região da AWS que a cópia, especifique um identificador de snapshot de banco de dados válido. Por exemplo, .`rds:mysql-instance1-snapshot-20130805`
  - Se o snapshot de origem estiver na mesma Região da AWS que a cópia e tiver sido compartilhado com sua Conta da AWS, especifique um ARN de snapshot de banco de dados válido. Por exemplo, .`arn:aws:rds:us-west-2:123456789012:snapshot:mysql-instance1-snapshot-20130805`
  - Se o snapshot de origem estiver em uma Região da AWS diferente da cópia, especifique um ARN válido do snapshot de banco de dados. Por exemplo, `arn:aws:rds:us-west-2:123456789012:snapshot:mysql-instance1-snapshot-20130805`.
  - Se você estiver copiando a partir de um snapshot de banco de dados manual compartilhado, esse parâmetro deverá ser o nome de recurso da Amazon (ARN) desse snapshot.
  - Se você estiver copiando um snapshot criptografado, esse parâmetro deverá estar no formato ARN para a Região da AWS de origem e deverá corresponder a `SourceDBSnapshotIdentifier` no parâmetro `PreSignedUrl`.
- `--target-db-snapshot-identifier` – O identificador da nova cópia do snapshot de banco de dados criptografado.
- `--copy-option-group`: copie o grupo de opções de um snapshot que foi compartilhado com sua Conta da AWS.
- `--copy-tags` – Incluir a opção de copiar tags para copiar tags e valores do snapshot para a cópia do snapshot.
- `--option-group-name` – O grupo de opções a ser associado à cópia do snapshot.

Especifique essa opção se estiver copiando um snapshot de uma Região da AWS para outra, e sua instância de Bancos de Dados usar um grupo de opções não padrão.

Se a sua instância de banco de dados de origem usar a Transparent Data Encryption for Oracle ou o Microsoft SQL Server, você deverá especificar essa opção ao fazer cópias entre regiões. Para obter mais informações, consulte[Considerações sobre grupos de opções]

- `--kms-key-id`: o identificador da chave do KMS para um snapshot do banco de dados criptografado. O identificador da chave do KMS é o nome do recurso da Amazon (ARN), o identificador de chave ou o alias da chave da chave do KMS.
  - Se você copiar um snapshot do banco de dados criptografado da Conta da AWS, poderá especificar um valor para esse parâmetro para criptografar a cópia com uma nova chave do KMS. Se você não especificar um valor para esse parâmetro, a cópia do snapshot de banco de dados será criptografada com a mesma chave do KMS que o snapshot de banco de dados de origem.
  - Se você copiar um snapshot do banco de dados criptografado que foi compartilhado de outra Conta da AWS, deverá especificar um valor para esse parâmetro.
  - Se você especificar esse parâmetro ao copiar um snapshot não criptografado, a cópia será criptografada.
  - Ao copiar um snapshot criptografado para outra Região da AWS, você deve especificar uma chave do KMS para a Região da AWS de destino. As chaves do KMS são específicas da Região da AWS em que são criadas. Não é possível usar chaves de criptografia de uma Região da AWS em outra Região da AWS.

Example de não criptografado, para a mesma região

O código a seguir cria uma cópia de um snapshot, com o novo nome `mydbsnapshotcopy`, na mesma Região da AWS que o snapshot de origem. Quando a cópia é feita, o grupo de opções de banco de dados e as tags no snapshot original são copiados na cópia do snapshot.

Para Linux, macOS ou Unix:

```
aws rds copy-db-snapshot \
    --source-db-snapshot-identifier arn:aws:rds:us-west-2:123456789012:snapshot:mysql-instance1-snapshot-20130805 \
    --target-db-snapshot-identifier mydbsnapshotcopy \
    --copy-option-group \
```

```
    --copy-tags
```

Para Windows:

```
aws rds copy-db-snapshot ^
    --source-db-snapshot-identifier arn:aws:rds:us-west-2:123456789012:snapshot:mysql-instance1-snapshot-20130805 ^
    --target-db-snapshot-identifier mydbsnapshotcopy ^
    --copy-option-group ^
    --copy-tags
```

Example de não criptografado, entre regiões

O código a seguir cria uma cópia de um snapshot, com o novo nome `mydbsnapshotcopy`, na Região da AWS em que o comando é executado.

Para Linux, macOS ou Unix:

```
aws rds copy-db-snapshot \
    --source-db-snapshot-identifier arn:aws:rds:us-east-1:123456789012:snapshot:mysql-instance1-snapshot-20130805 \
    --target-db-snapshot-identifier mydbsnapshotcopy
```

Para Windows:

```
aws rds copy-db-snapshot ^
    --source-db-snapshot-identifier arn:aws:rds:us-east-1:123456789012:snapshot:mysql-instance1-snapshot-20130805 ^
    --target-db-snapshot-identifier mydbsnapshotcopy
```

Example de criptografado, entre regiões

O exemplo de código a seguir copia um snapshot de banco de dados criptografado da região Oeste dos EUA (Oregon) para a região Leste dos EUA (N. da Virgínia). Execute o comando na região de destino (us-east-1).

Para Linux, macOS ou Unix:

```
aws rds copy-db-snapshot \
    --source-db-snapshot-identifier arn:aws:rds:us-west-2:123456789012:snapshot:mysql-instance1-snapshot-20161115 \
    --target-db-snapshot-identifier mydbsnapshotcopy \
```

```
    --kms-key-id my-us-east-1-key \
    --option-group-name custom-option-group-name
```

Para Windows:

```
aws rds copy-db-snapshot ^
    --source-db-snapshot-identifier arn:aws:rds:us-west-2:123456789012:snapshot:mysql-instance1-snapshot-20161115 ^
    --target-db-snapshot-identifier mydbsnapshotcopy ^
    --kms-key-id my-us-east-1-key ^
    --option-group-name custom-option-group-name
```

O parâmetro `--source-region` é necessário quando você está copiando um snapshot criptografado entre as regiões GovCloud (Leste dos EUA) da AWS e GovCloud (Oeste dos EUA) da AWS. Em `--source-region`, especifique a Região da AWS da instância de banco de dados de origem.

Se a `--source-region` não estiver especificada, especifique um valor de `--pre-signed-url`. Uma presigned URL é um URL que contém uma solicitação assinada do Signature Versão 4 para o comando `copy-db-snapshot` chamado na Região da AWS de origem. Para saber mais sobre a opção `pre-signed-url`, consulte copy-db-snapshot na Referência de comandos da AWS CLI.

API do RDS

É possível copiar um snapshot de banco de dados usando a operação CopyDBSnapshot da API do Amazon RDS. Se estiver copiando o snapshot para uma nova Região da AWS, execute a ação na nova Região da AWS.

As seguintes opções são usadas para copiar um snapshot de banco de dados. Nem todos os parâmetros são necessários para todos os cenários. Use as descrições e os exemplos a seguir para determinar quais parâmetros usar.

- `SourceDBSnapshotIdentifier` – O identificador do snapshot de banco de dados de origem.
  - Se o snapshot de origem estiver na mesma Região da AWS que a cópia, especifique um identificador de snapshot de banco de dados válido. Por exemplo, `.rds:mysql-instance1-snapshot-20130805`
  - Se o snapshot de origem estiver na mesma Região da AWS que a cópia e tiver sido compartilhado com sua Conta da AWS, especifique um ARN de snapshot de banco de dados válido. Por exemplo, `.arn:aws:rds:us-west-2:123456789012:snapshot:mysql-instance1-snapshot-20130805`

  - Se o snapshot de origem estiver em uma Região da AWS diferente da cópia, especifique um ARN válido do snapshot de banco de dados. Por exemplo, `arn:aws:rds:us-west-2:123456789012:snapshot:mysql-instance1-snapshot-20130805`.
  - Se você estiver copiando a partir de um snapshot de banco de dados manual compartilhado, esse parâmetro deverá ser o nome de recurso da Amazon (ARN) desse snapshot.
  - Se você estiver copiando um snapshot criptografado, esse parâmetro deverá estar no formato ARN para a Região da AWS de origem e deverá corresponder a `SourceDBSnapshotIdentifier` no parâmetro `PreSignedUrl`.
- `TargetDBSnapshotIdentifier` – O identificador da nova cópia do snapshot de banco de dados criptografado.
- `CopyOptionGroup`: defina esse parâmetro como `true` para copiar o grupo de opções de um snapshot compartilhado na cópia do snapshot. O padrão é `false`.
- `CopyTags` – Defina esse parâmetro como `true` para copiar tags e valores do snapshot para a cópia do snapshot. O padrão é `false`.
- `OptionGroupName` – O grupo de opções a ser associado à cópia do snapshot.

  Especifique esse parâmetro se estiver copiando um snapshot de uma Região da AWS para outra, e sua instância de banco de dados usar um grupo de opções não padrão.

  Se a sua instância de banco de dados de origem usar a Transparent Data Encryption for Oracle ou o Microsoft SQL Server, você deverá especificar esse parâmetro ao fazer cópias entre regiões. Para obter mais informações, consulteConsiderações sobre grupos de opções
- `KmsKeyId`: o identificador da chave do KMS para um snapshot do banco de dados criptografado. O identificador da chave do KMS é o nome do recurso da Amazon (ARN), o identificador de chave ou o alias da chave da chave do KMS.
  - Se você copiar um snapshot do banco de dados criptografado da Conta da AWS, poderá especificar um valor para esse parâmetro para criptografar a cópia com uma nova chave do KMS. Se você não especificar um valor para esse parâmetro, a cópia do snapshot de banco de dados será criptografada com a mesma chave do KMS que o snapshot de banco de dados de origem.
  - Se você copiar um snapshot do banco de dados criptografado que foi compartilhado de outra Conta da AWS, deverá especificar um valor para esse parâmetro.
  - Se você especificar esse parâmetro ao copiar um snapshot não criptografado, a cópia será criptografada.

- Ao copiar um snapshot criptografado para outra Região da AWS, você deve especificar uma chave do KMS para a Região da AWS de destino. As chaves do KMS são específicas da Região da AWS em que são criadas. Não é possível usar chaves de criptografia de uma Região da AWS em outra Região da AWS.

- `PreSignedUrl`: o URL que contém uma solicitação assinada do Signature versão 4 para a operação da API `CopyDBSnapshot` na Região da AWS de origem que contém o snapshot de banco de dados a ser copiado.

  Especifique esse parâmetro ao copiar um snapshot de banco de dados criptografado de outra Região da AWS usando a API do Amazon RDS. É possível especificar a opção da região de origem em vez desse parâmetro ao copiar um snapshot de banco de dados criptografado de outra Região da AWS usando a AWS CLI.

  O URL pré-assinado deve ser uma solicitação válida para a operação da API `CopyDBSnapshot`, que pode ser executada na Região da AWS de origem que contém o snapshot do banco de dados criptografado a ser copiado. A solicitação de URL pré-assinada deve conter os seguintes valores de parâmetros:

  - `DestinationRegion`: a Região da AWS para a qual o snapshot de banco de dados criptografado será copiado. Essa Região da AWS é a mesma em que a operação `CopyDBSnapshot` é chamada e que contém esse URL pré-assinado.

    Por exemplo, suponha que você copie um snapshot de banco de dados criptografado da região us-west-2 para a região us-east-1. Você, então, chama a operação `CopyDBSnapshot` na região us-east-1 e fornece um URL pré-assinado que contenha uma chamada para a operação `CopyDBSnapshot` na região us-west-2. Para esse exemplo, o `DestinationRegion` no URL pré-assinado deve ser definido para a região us-east-1.

  - `KmsKeyId`: o identificador da chave do KMS a ser usada para criptografar a cópia do snapshot do banco de dados na Região da AWS de destino. Esse é o mesmo identificador da operação `CopyDBSnapshot`, que é chamada na Região da AWS de destino, e da operação contida no URL pré-assinado.

  - `SourceDBSnapshotIdentifier`: o identificador de snapshot de banco de dados para o snapshot criptografado a ser copiado. Esse identificador deve estar no formato de nome do recurso da Amazon (ARN) da Região da AWS de origem Por exemplo, se você estiver copiando um snapshot de banco de dados criptografado da região us-west-2, o `SourceDBSnapshotIdentifier` se parecerá com o seguinte exemplo: `arn:aws:rds:us-west-2:123456789012:snapshot:mysql-instance1-snapshot-20161115`.

Para obter mais informações sobre as solicitações assinadas do Signature versão 4, consulte o seguinte:

- Autenticar solicitações: usar parâmetros de consulta (AWS Signature versão 4) na Referência da API do Amazon Simple Storage Service
- Processo de assinatura do Signature versão 4 no Referência geral da AWS

Example de não criptografado, para a mesma região

O código a seguir cria uma cópia de um snapshot, com o novo nome `mydbsnapshotcopy`, na mesma Região da AWS que o snapshot de origem. Quando a cópia é feita, todas as tags do snapshot original são copiadas para a cópia do snapshot.

```
https://rds.us-west-1.amazonaws.com/
 ?Action=CopyDBSnapshot
 &CopyTags=true
 &SignatureMethod=HmacSHA256
 &SignatureVersion=4
 &SourceDBSnapshotIdentifier=mysql-instance1-snapshot-20130805
 &TargetDBSnapshotIdentifier=mydbsnapshotcopy
 &Version=2013-09-09
 &X-Amz-Algorithm=AWS4-HMAC-SHA256
 &X-Amz-Credential=AKIADQKE4SARGYLE/20140429/us-west-1/rds/aws4_request
 &X-Amz-Date=20140429T175351Z
 &X-Amz-SignedHeaders=content-type;host;user-agent;x-amz-content-sha256;x-amz-date
 &X-Amz-Signature=9164337efa99caf850e874a1cb7ef62f3cea29d0b448b9e0e7c53b288ddffed2
```

Example de não criptografado, entre regiões

O código a seguir cria uma cópia de um snapshot, com o novo nome `mydbsnapshotcopy`, na região Oeste dos EUA (Norte da Califórnia).

```
https://rds.us-west-1.amazonaws.com/
 ?Action=CopyDBSnapshot
 &SignatureMethod=HmacSHA256
 &SignatureVersion=4
 &SourceDBSnapshotIdentifier=arn%3Aaws%3Ards%3Aus-east-1%3A123456789012%3Asnapshot
%3Amysql-instance1-snapshot-20130805
 &TargetDBSnapshotIdentifier=mydbsnapshotcopy
 &Version=2013-09-09
```

```
 &X-Amz-Algorithm=AWS4-HMAC-SHA256
 &X-Amz-Credential=AKIADQKE4SARGYLE/20140429/us-west-1/rds/aws4_request
 &X-Amz-Date=20140429T175351Z
 &X-Amz-SignedHeaders=content-type;host;user-agent;x-amz-content-sha256;x-amz-date
 &X-Amz-Signature=9164337efa99caf850e874a1cb7ef62f3cea29d0b448b9e0e7c53b288ddffed2
```

Example de criptografado, entre regiões

O código a seguir cria uma cópia de um snapshot, com o novo nome `mydbsnapshotcopy`, na região Leste dos EUA (N. da Virgínia).

```
https://rds.us-east-1.amazonaws.com/
    ?Action=CopyDBSnapshot
    &KmsKeyId=my-us-east-1-key
    &OptionGroupName=custom-option-group-name
    &PreSignedUrl=https%253A%252F%252Frds.us-west-2.amazonaws.com%252F
         %253FAction%253DCopyDBSnapshot
         %2526DestinationRegion%253Dus-east-1
         %2526KmsKeyId%253Dmy-us-east-1-key
         %2526SourceDBSnapshotIdentifier%253Darn%25253Aaws%25253Ards%25253Aus-
west-2%25253A123456789012%25253Asnapshot%25253Amysql-instance1-snapshot-20161115
         %2526SignatureMethod%253DHmacSHA256
         %2526SignatureVersion%253D4
         %2526Version%253D2014-10-31
         %2526X-Amz-Algorithm%253DAWS4-HMAC-SHA256
         %2526X-Amz-Credential%253DAKIADQKE4SARGYLE%252F20161117%252Fus-west-2%252Frds
%252Faws4_request
         %2526X-Amz-Date%253D20161117T215409Z
         %2526X-Amz-Expires%253D3600
         %2526X-Amz-SignedHeaders%253Dcontent-type%253Bhost%253Buser-agent%253Bx-amz-
content-sha256%253Bx-amz-date
         %2526X-Amz-Signature
%253D255a0f17b4e717d3b67fad163c3ec26573b882c03a65523522cf890a67fca613
    &SignatureMethod=HmacSHA256
    &SignatureVersion=4
    &SourceDBSnapshotIdentifier=arn%3Aaws%3Ards%3Aus-west-2%3A123456789012%3Asnapshot
%3Amysql-instance1-snapshot-20161115
    &TargetDBSnapshotIdentifier=mydbsnapshotcopy
    &Version=2014-10-31
    &X-Amz-Algorithm=AWS4-HMAC-SHA256
    &X-Amz-Credential=AKIADQKE4SARGYLE/20161117/us-east-1/rds/aws4_request
    &X-Amz-Date=20161117T221704Z
    &X-Amz-SignedHeaders=content-type;host;user-agent;x-amz-content-sha256;x-amz-date
```

```
&X-Amz-Signature=da4f2da66739d2e722c85fcfd225dc27bba7e2b8dbea8d8612434378e52adccf
```

Limitações

Algumas limitações ao copiar snapshots:

- Você não pode copiar um snapshot de ou para as regiões China (Pequim) ou China (Ningxia).
- Você pode copiar um snapshot entre a AWS GovCloud (Leste dos EUA) e AWS GovCloud (US-West). No entanto, você não pode copiar um snapshot entre essas regiões GovCloud (EUA) e regiões que não são GovCloud (EUA).
- Se você excluir um snapshot de origem antes que o snapshot de destino fique disponível, a cópia do snapshot poderá falhar. Verifique se o snapshot de destino possui um status `AVAILABLE` antes de excluir um snapshot de origem.
- Você pode ter até 20 solicitações de cópia de snapshot em andamento para uma única região de destino por conta.
- Quando você solicita várias cópias de snapshot para a mesma instância de banco de dados de origem, elas são colocadas em fila internamente. As cópias solicitadas posteriormente não serão iniciadas enquanto as cópias de snapshots anteriores não forem concluídas. Para obter mais informações, consulte Por que a criação de snapshots do EC2 AMI ou do EBS está lenta? na Central de Conhecimento da AWS.
- Dependendo das Regiões da AWS envolvidas e da quantidade de dados a serem copiados, uma cópia de snapshot entre regiões pode levar horas para ser concluída. Em alguns casos, pode haver um grande número de solicitações de cópia de snapshot entre regiões a partir de uma determinada região de origem da . Nesses casos, o Amazon RDS pode colocar novas solicitações de cópia entre regiões vindas daquela região da de origem em uma fila até que algumas cópias em andamento sejam concluídas. Nenhuma informação de progresso é exibida sobre solicitações de cópia enquanto elas estão na fila. As informações sobre o andamento são exibidas quando a cópia é iniciada.
- Se uma cópia ainda estiver pendente quando outra cópia for iniciada, esta será iniciada somente após a primeira ser concluída.
- Não é possível copiar um snapshot de um cluster de banco de dados multi-AZ.
- Só é possível copiar snapshots de instâncias de banco de dados que usam volumes io2 para Regiões da AWS onde os volumes io2 Block Express estão disponíveis. Para obter mais informações, consulte Armazenamento de instâncias de banco de dados do Amazon RDS.

Considerações

Com relação a considerações sobre a cópia de um snapshot de banco de dados, consulte os tópicos a seguir.

Tópicos

- Retenção de snapshots
- Considerações sobre a cópia de snapshots compartilhados
- Considerações sobre a cópia de snapshots de criptografia
- Considerações sobre a cópia de snapshots incrementais
- Considerações sobre a cópia de snapshots entre regiões
- Considerações sobre grupos de opções
- Considerações sobre o grupo de parâmetros

Retenção de snapshots

O Amazon RDS exclui snapshots automatizados em várias situações:

- Ao final do período de retenção.
- Quando você desabilita os backups automatizados de uma instância de banco de dados.
- Quando você exclui uma instância de banco de dados.

Se quiser manter um backup automatizado por um período mais longo, copie-o para criar um snapshot manual, que é retido até você excluí-lo. Os custos de armazenamento do Amazon RDS podem se aplicar a snapshots manuais, caso excedam seu espaço de armazenamento padrão.

Para obter mais informações sobre os custos de armazenamento de backup, consulte Definição de preço do Amazon RDS.

Considerações sobre a cópia de snapshots compartilhados

É possível copiar snapshots compartilhados com você por outras Contas da AWS. Em alguns casos, é possível copiar um snapshot criptografado que tenha sido compartilhado de outra Conta da AWS. Nesses casos, você deve ter acesso à AWS KMS key usada para criptografar o snapshot.

> **Note**
>
> Os custos de armazenamento do Amazon RDS aplicam-se aos snapshots compartilhados copiados. O Amazon RDS pode anexar o ARN da instância de banco de dados de origem ao snapshot copiado.

Será possível copiar um snapshot de banco de dados compartilhado entre Regiões da AWS, se o snapshot não estiver criptografado. No entanto, se o snapshot de banco de dados compartilhado estiver criptografado, você poderá copiá-lo apenas na mesma região.

> **Note**
>
> Copiar snapshots incrementais compartilhados na mesma Região da AWS é uma ação aceita quando estes não estão criptografados ou quando estão criptografados com a mesma chave do KMS do snapshot completo inicial. Se você usar uma chave do KMS diferente para criptografar snapshots subsequentes ao copiá-los, esses snapshots compartilhados serão completos. Para obter mais informações, consulte Considerações sobre a cópia de snapshots incrementais.

Considerações sobre a cópia de snapshots de criptografia

É possível copiar um snapshot que tenha sido criptografado usando uma chave do KMS. Se você copiar um snapshot criptografado, a cópia desse snapshot também deverá ser criptografada. Se você copiar um snapshot criptografado dentro da mesma Região da AWS, poderá criptografar a cópia com a mesma chave do KMS que o snapshot original. Ou você pode especificar uma chave do KMS diferente.

Se você copiar um snapshot criptografado entre regiões, deverá especificar uma chave do KMS válida na Região da AWS de destino. Pode ser uma chave do KMS específica da região ou uma chave de várias regiões. Para obter mais informações sobre chaves do KMS de várias regiões, consulte Usar chaves de várias regiões no AWS KMS.

O snapshot de origem permanece criptografado ao longo do processo de cópia. Para obter mais informações, consulte Limitações das instâncias de banco de dados criptografados do Amazon RDS.

Você também pode criptografar uma cópia de um snapshot não criptografado. Dessa maneira, você pode adicionar rapidamente a criptografia a uma instância de banco de dados não criptografada

anteriormente. Para fazer isso, você pode criar um snapshot da instância de banco de dados quando estiver tudo pronto para criptografá-lo. Em seguida, você cria uma cópia desse snapshot e especifica uma chave do KMS para criptografar essa cópia de snapshot. Assim, você pode restaurar uma instância de banco de dados criptografado do snapshot criptografado.

> **Important**
>
> Ao usar instruções de negação explícitas para todos os recursos (*) nas políticas de chave do AWS KMS com serviços gerenciados como o Amazon RDS, você deve especificar uma condição para dar permissão à conta proprietária do recurso. A operação de cópia pode falhar sem essa condição, mesmo que a regra de negação inclua exceções para seu usuário do IAM.

Para ter mais informações sobre o gerenciamento de chaves AWS KMS para o Amazon RDS, consulte Gerenciamento de AWS KMS key.

Considerações sobre a cópia de snapshots incrementais

Um snapshot incremental contém somente os dados que foram alterados após o snapshot mais recente da mesma instância de banco de dados. A cópia incremental de snapshot é mais rápida e resulta em custos menores de armazenamento do que a cópia completa de snapshot.

O fato de uma cópia do snapshot ser ou não incremental é determinado pela cópia do snapshot feita recentemente e o snapshot de origem. Se a cópia mais recente do snapshot tiver sido excluída, a próxima cópia será um cópia completa, não uma cópia incremental. Uma cópia do snapshot será do mesmo tipo que o snapshot de origem. Se o snapshot de origem for incremental, a respectiva cópia será um snapshot incremental.

Ao copiar um snapshot entre Contas da AWS, a cópia será incremental apenas se todas as seguintes condições forem atendidas:

- A cópia mais recente do snapshot é da mesma instância de banco de dados de origem e ainda existe na conta de destino.
- Todas as cópias do snapshot na conta de destino foram feitas sem criptografia ou foram criptografadas usando a mesma chave do KMS.
- Se a instância de banco de dados de origem for uma instância multi-AZ, ela não fez failover para outra AZ desde que o último snapshot dela foi criado.

Os exemplos a seguir ilustram a diferença entre snapshots completos e incrementais. Aplicam-se a snapshots compartilhados e não compartilhados.

Snapshot	Chave de criptografia	Completo ou incremental
S1	K1	Completo
S2	K1	Incremental de S1
S3	K1	Incremental de S2
S4	K1	Incremental de S3
Cópia de S1 (S1C)	K2	Completo
Cópia de S2 (S2C)	K3	Completo
Cópia de S3 (S3C)	K3	Incremental de S2C
Cópia de S4 (S4C)	K3	Incremental de S3C
Cópia 2 de S4 (S4C2)	K4	Completo

> **Note**
>
> Nesses exemplos, os snapshots S2, S3 e S4 só serão incrementais se o snapshot anterior ainda existir.
> O mesmo se aplica às cópias. As cópias dos snapshots S3C e S4C só serão incrementais se a cópia anterior ainda existir.

Para obter informações sobre como copiar snapshots incrementais entre Regiões da AWS, consulte Cópias completas e incrementais.

Considerações sobre a cópia de snapshots entre regiões

É possível copiar snapshots de banco de dados entre Regiões da AWS. No entanto, existem certas restrições e considerações para a cópia de snapshot entre regiões.

Solicitações de cópia de snapshot de banco de dados entre regiões

Para se comunicar com a região de origem para solicitar uma cópia de snapshot de banco de dados entre regiões, o solicitante (função do IAM ou usuário do IAM) deve ter acesso ao snapshot do banco de dados de origem e à região de origem.

Certas condições na política do IAM do solicitante podem fazer com que a solicitação falhe. Os exemplos a seguir supõem que você está copiando o snapshot de banco de dados de Leste dos EUA (Ohio) para Leste dos EUA (N. da Virgínia). Estes exemplos mostram condições na política do IAM do solicitante que fazem com que a solicitação falhe:

- A política do solicitante tem uma condição para `aws:RequestedRegion`.

```
...
"Effect": "Allow",
"Action": "rds:CopyDBSnapshot",
"Resource": "*",
"Condition": {
    "StringEquals": {
        "aws:RequestedRegion": "us-east-1"
    }
}
```

  A solicitação falha porque a política não permite acesso à região de origem. Para uma solicitação bem-sucedida, especifique as regiões de origem e de destino.

```
...
"Effect": "Allow",
"Action": "rds:CopyDBSnapshot",
"Resource": "*",
"Condition": {
    "StringEquals": {
        "aws:RequestedRegion": [
            "us-east-1",
            "us-east-2"
        ]
    }
}
```

- A política do solicitante não permite acesso ao snapshot de banco de dados de origem.

```
...
```

```
"Effect": "Allow",
"Action": "rds:CopyDBSnapshot",
"Resource": "arn:aws:rds:us-east-1:123456789012:snapshot:target-snapshot"
...
```

Para uma solicitação bem-sucedida, especifique os snapshots de origem e de destino.

```
...
"Effect": "Allow",
"Action": "rds:CopyDBSnapshot",
"Resource": [
    "arn:aws:rds:us-east-1:123456789012:snapshot:target-snapshot",
    "arn:aws:rds:us-east-2:123456789012:snapshot:source-snapshot"
]
...
```

- A política do solicitante nega `aws:ViaAWSService`.

```
...
"Effect": "Allow",
"Action": "rds:CopyDBSnapshot",
"Resource": "*",
"Condition": {
    "Bool": {"aws:ViaAWSService": "false"}
}
```

A comunicação com a região de origem é feita pelo RDS em nome do solicitante. Para uma solicitação bem-sucedida, não negue as chamadas feitas pelos serviços da AWS.

- A política do solicitante tem uma condição para `aws:SourceVpc` ou `aws:SourceVpce`.

Essas solicitações podem falhar, pois quando o RDS faz a chamada para a região remota, ela não é feita a partir da VPC ou VPC endpoint especificados.

Se você precisar usar uma das condições anteriores que causariam uma falha de uma solicitação, é possível incluir uma segunda instrução com `aws:CalledVia` em sua política, para fazer com que a solicitação seja bem-sucedida. Por exemplo, você pode usar `aws:CalledVia` com `aws:SourceVpce` como mostrado aqui:

```
...
"Effect": "Allow",
```

```
"Action": "rds:CopyDBSnapshot",
"Resource": "*",
"Condition": {
    "Condition" : {
        "ForAnyValue:StringEquals" : {
          "aws:SourceVpce": "vpce-1a2b3c4d"
        }
     }
},
{
    "Effect": "Allow",
    "Action": [
        "rds:CopyDBSnapshot"
    ],
    "Resource": "*",
    "Condition": {
        "ForAnyValue:StringEquals": {
            "aws:CalledVia": [
                "rds.amazonaws.com"
            ]
        }
    }
}
```

Para obter mais informações, consulte Políticas e permissões no IAM, no Guia do usuário do IAM.

Autorização da cópia do snapshot

Depois que uma solicitação de cópia de snapshot de banco de dados entre regiões retorna `success`, o RDS inicia a cópia em segundo plano. Uma autorização para o RDS acessar o snapshot de origem é criada. Essa autorização vincula o snapshot de banco de dados de origem ao snapshot de banco de dados de destino e permite que o RDS copie apenas para o snapshot de destino especificado.

A autorização é verificada pelo RDS usando a permissão `rds:CrossRegionCommunication` na função do IAM vinculada ao serviço. Se a cópia for autorizada, o RDS se comunica com a região de origem e conclui a cópia.

O RDS não tem acesso a snapshots de banco de dados que não foram autorizados anteriormente por uma solicitação de `CopyDBSnapshot`. A autorização é revogada quando a cópia é concluída.

O RDS usa a função vinculada ao serviço para verificar a autorização na região de origem. Se você excluir a função vinculada ao serviço durante o processo de cópia, ela falhará.

Para ter mais informações, consulte Usar funções vinculadas a serviço no Guia do usuário do IAM.

Uso de credenciais do AWS Security Token Service

Os tokens de sessão do endpoint global do AWS Security Token Service (AWS STS) são válidos apenas nas Regiões da AWS habilitadas por padrão (regiões comerciais). Se você usar credenciais da operação da API `assumeRole` no AWS STS, use o endpoint regional se a região de origem for uma região de aceitação. Caso contrário, haverá falha na solicitação. Isso acontece porque suas credenciais devem ser válidas em ambas as regiões, o que é verdadeiro para regiões opcionais somente quando o endpoint regional do AWS STS é usado.

Para usar o endpoint global, certifique-se de que ele esteja ativado para ambas as regiões nas operações. Defina o endpoint global como `Valid in all Regiões da AWS` nas configurações da conta do AWS STS.

A mesma regra se aplica às credenciais no parâmetro pre-signed URL.

Para obter mais informações, consulte Gerenciar o AWS STS em uma Região da AWS no Guia do usuário do IAM.

Latência e múltiplas solicitações de cópia

Dependendo das Regiões da AWS envolvidas e da quantidade de dados a serem copiados, uma cópia de snapshot entre regiões pode levar horas para ser concluída.

Em alguns casos, pode haver um grande número de solicitações de cópia de snapshot entre regiões de determinada Região da AWS de origem. Nesses casos, o Amazon RDS pode colocar novas solicitações de cópia entre regiões da Região da AWS de origem em uma fila até que algumas cópias em andamento sejam concluídas. Nenhuma informação sobre o andamento é exibida sobre as solicitações de cópia enquanto elas estão na fila. As informações de progresso são exibidas quando a cópia é iniciada.

Cópias completas e incrementais

Ao copiar um snapshot para uma Região da AWS diferente do snapshot de origem, a primeira cópia é uma cópia completa do snapshot, mesmo que você copie um snapshot incremental. Uma cópia completa de snapshot contém todos os dados e metadados necessários para restaurar a instância de banco de dados. Após a primeira cópia do snapshot, será possível copiar snapshots incrementais da mesma instância de bancos de dados para a mesma região de destino na mesma Conta da AWS. Para obter mais informações sobre snapshots incrementais, consulte Considerações sobre a cópia de snapshots incrementais.

A cópia incremental de snapshot entre Regiões da AWS é compatível com snapshots criptografados e não criptografados.

Ao copiar um snapshot entre Regiões da AWS, a cópia será uma cópia incremental se as seguintes condições forem atendidas:

- O snapshot foi copiado anteriormente na região de destino.
- A cópia mais recente do snapshot ainda existe na região de destino.
- Todas as cópias do snapshot na região de destino foram feitas sem criptografia ou foram criptografadas usando a mesma chave do KMS.

Considerações sobre grupos de opções

Grupos de opções de banco de dados são específicos da Região da AWS em que são criados, e você não pode usar um grupo de opções de uma Região da AWS em outra Região da AWS.

Para bancos de dados Oracle, você pode usar a AWS CLI ou a API do RDS para copiar o grupo de opções de banco de dados personalizado de um snapshot que foi compartilhado com sua Conta da AWS. Você só pode copiar grupos de opções dentro da mesma Região da AWS. O grupo de opções não será copiado se já tiver sido copiado na conta de destino e nenhuma alteração tiver sido feita nele desde que foi copiado. Se o grupo de opções de origem tiver sido copiado antes, mas tiver sido alterado desde que foi copiado, o RDS copiará a nova versão na conta de destino. Os grupos de opções padrão não são copiados.

Quando você copia um snapshot entre regiões, pode especificar um novo grupo de opções para o snapshot. Recomendamos que você prepare o novo grupo de opções antes de copiar o snapshot. Na Região da AWS de destino, crie um grupo de opções com as mesmas configurações que a instância de banco de dados original. É possível usar um grupo de parâmetros já existente na nova Região da AWS.

Em alguns casos, você pode copiar um snapshot e não especificar um novo grupo de opções para o snapshot. Nesses casos, ao restaurar o snapshot, a instância de banco de dados obtém o grupo de opções padrão. Para dar à nova instância de banco de dados as mesmas opções que a original, faça o seguinte:

1. Na Região da AWS de destino, crie um grupo de opções com as mesmas configurações que a instância de banco de dados original. É possível usar um grupo de parâmetros já existente na nova Região da AWS.

2. Depois de restaurar o snapshot na Região da AWS de destino, modifique a nova instância de banco de dados e adicione o grupo de opções novo ou existente da etapa anterior.

Considerações sobre o grupo de parâmetros

Ao copiar um snapshot entre regiões, a cópia não inclui o grupo de parâmetros usado pela instância de banco de dados original. Ao restaurar um snapshot para criar uma nova instancia de banco de dados, essa instância de banco de dados obtém o grupo de parâmetros padrão da Região da AWS na qual ela foi criada. Para dar à nova instância de banco de dados os mesmos parâmetros da original, faça o seguinte:

1. Na Região da AWS de destino, crie um grupo de parâmetros de banco de dados com as mesmas configurações que a instância de banco de dados original. É possível usar um grupo de parâmetros já existente na nova Região da AWS.
2. Depois de restaurar o snapshot na Região da AWS de destino, modifique a nova instância de banco de dados e adicione o grupo de parâmetros novo ou existente da etapa anterior.

Compartilhar um snapshot de banco de dados para o Amazon RDS

Com o Amazon RDS, é possível compartilhar um snapshot de banco de dados manual das seguintes formas:

- Compartilhar um snapshot de banco de dados manual, seja criptografado ou não, permite que as Contas da AWS autorizadas copiem o snapshot.
- Compartilhar um snapshot de banco de dados manual não criptografado permite que as Contas da AWS autorizadas restaurem diretamente uma instância de bancos de dados do snapshot em vez de fazer uma cópia dela e restaurar por meio dela. No entanto, não é possível restaurar uma instância de banco de dados a partir de um snapshot de banco de dados que seja compartilhado e criptografado. Em vez disso, você pode fazer uma cópia do snapshot de banco de dados e restaurar a instância de banco de dados a partir da cópia.

> **Note**
>
> Para compartilhar um snapshot de banco de dados automatizado, crie um snapshot de banco de dados manual copiando o snapshot automatizado e compartilhe essa cópia. Esse processo também se aplica aos recursos gerados pelo AWS Backup.

Para obter mais informações sobre a cópia de um snapshot, consulte Copiar um snapshot de banco de dados para o Amazon RDS. Para obter mais informações sobre como restaurar uma instância de banco de dados de um snapshot de banco de dados, consulte Restaurar uma instância de banco de dados.

É possível compartilhar um snapshot manual com até 20 outras Contas da AWS.

As limitações a seguir se aplicam ao compartilhar snapshots manuais com outras Contas da AWS:

- Ao restaurar uma instância de banco de dados de um snapshot compartilhado usando a AWS Command Line Interface (AWS CLI) ou a API do Amazon RDS, especifique o nome do recurso da Amazon (ARN) do snapshot compartilhado como o identificador do snapshot.
- Não é possível compartilhar um snapshot de banco de dados que usa um grupo de opções com opções permanentes ou persistentes, exceto para instâncias de banco de dados Oracle que têm a opção `Timezone`, `OLS` ou ambas.

Não é possível remover a opção permanente de um grupo de opções. Não é possível remover os grupos de opções contendo opções persistentes de uma instância de banco de dados depois que o grupo de opções já tiver sido atribuído à instância de banco de dados.

A tabela a seguir enumera as opções permanentes e persistentes e os mecanismos de banco de dados relacionados a elas.

Nome da opção	Persistente	Permanente	Mecanismo de banco de dados
TDE	Sim	Não	Microsoft SQL Server Enterprise Edition
TDE	Sim	Sim	Oracle Enterprise Edition
Fuso horário	Sim	Sim	Oracle Enterprise Edition Oracle Standard Edition Oracle Standard Edition One Oracle Standard Edition 2

Para instâncias de banco de dados Oracle, é possível copiar snapshots de banco de dados compartilhados que tenham a opção `Timezone` ou `OLS` (ou ambas). Para fazer isso, especifique um grupo de opções de destino que inclua essas opções ao copiar o snapshot de banco de dados. A opção OLS é permanente e persistente apenas para instâncias de banco de dados Oracle executando o Oracle versão 12.2 ou superior. Para obter mais informações sobre essas opções, consulte Fuso horário da Oracle e Oracle Label Security.

- Não é possível compartilhar um snapshot de um cluster de banco de dados multi-AZ.

Consulte nos tópicos a seguir informações sobre como compartilhar os snapshots públicos e os criptografados e sobre como interromper o compartilhamento de snapshots.

Tópicos

- Compartilhar snapshots públicos para o Amazon RDS

- Compartilhar snapshots criptografados para o Amazon RDS
- Interromper o compartilhamento de snapshots no Amazon RDS

Compartilhar um snapshot

Você pode compartilhar um snapshot de banco de dados usando o AWS Management Console, a ou a API do RDS.

Console

Usando o console do Amazon RDS, é possível compartilhar um snapshot de banco de dados manual com até vinte Contas da AWS. Você também pode usar o console para interromper o compartilhamento de um snapshot manual com uma ou mais contas.

Para compartilhar um snapshot manual do de banco de dados usando o console do Amazon RDS

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Snapshots.
3. Selecione o snapshot manual que você deseja compartilhar.
4. Em Actions (Ações), selecione Share Snapshot (Compartilhar snapshot).
5. Escolha uma das seguintes opções para DB snapshot visibility (Visibilidade do snapshot de banco de dados).

    - Se a fonte não estiver criptografada, selecione Público para permitir que todas as contas da AWS restaurem uma instância de banco de dados do snapshot de banco de dados manual, ou selecione Privado para permitir que apenas as Contas da AWS que você especificar restaurem uma instância de banco de dados do snapshot de banco de dados manual.

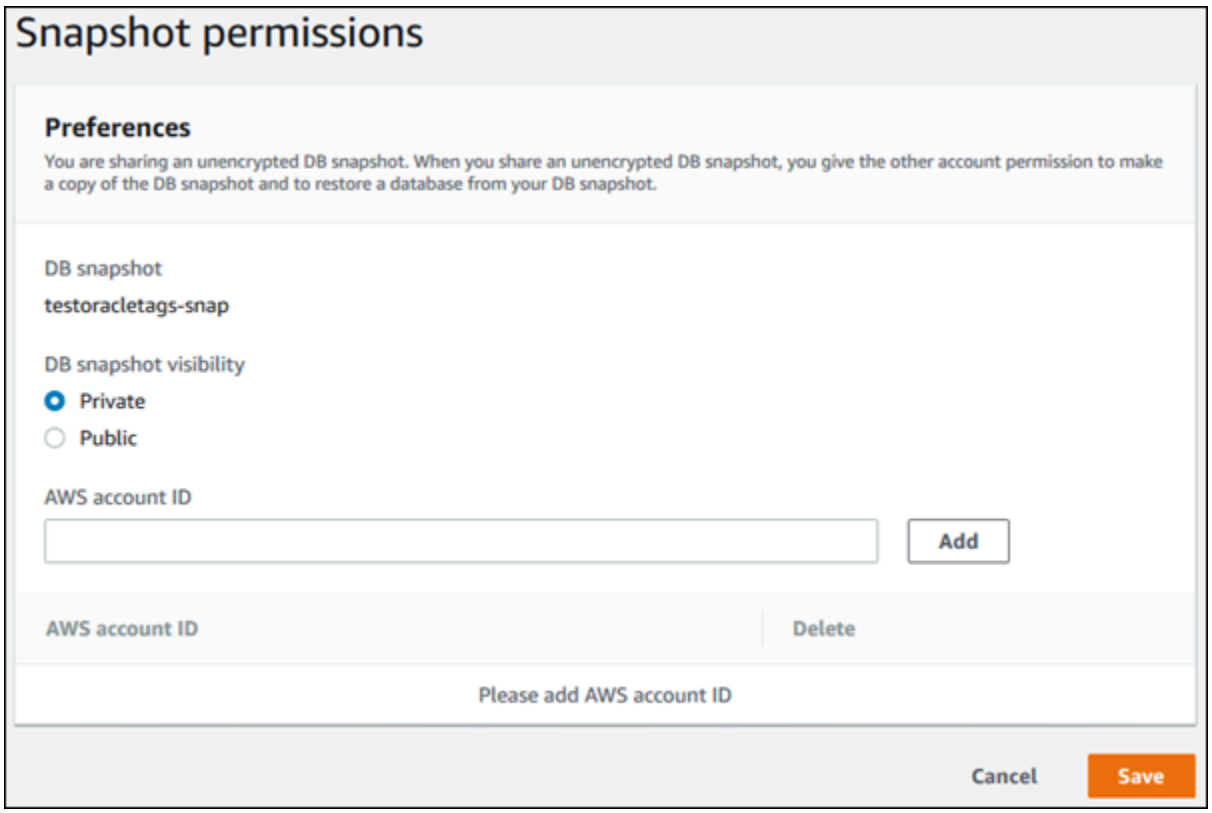
    > **Warning**
    >
    > Se você definir Visibilidade de snapshot de banco de dados como Público, todas as Contas da AWS poderão restaurar uma instância de banco de dados do snapshot de banco de dados manual, além de terem acesso aos dados. Não compartilhe nenhum DB snapshot manual que contenha informações privadas, como Public.
    > Para ter mais informações, consulte Compartilhar snapshots públicos para o Amazon RDS.

- Se a origem estiver criptografada, DB snapshot visibility (Visibilidade do snapshot de banco de dados) será definida como Private (Privada) porque os snapshots criptografados não podem ser compartilhados como públicos.

> **Note**
>
> Snapshots criptografados com a AWS KMS key padrão não podem ser compartilhados. Para ter informações sobre como contornar esse problema, consulte Compartilhar snapshots criptografados para o Amazon RDS.

6. Em ID da conta da AWS, digite o identificador da Conta da AWS à qual você deseja conceder a permissão para restaurar uma instância de banco de dados do snapshot manual e selecione Adicionar. Repita a operação para incluir outros identificadores da Conta da AWS, até vinte Contas da AWS.

   Se você errar ao adicionar o identificador da Conta da AWS à lista de contas permitidas, saiba que é possível excluí-lo da lista escolhendo Excluir à direita do identificador incorreto da Conta da AWS.



7. Depois de adicionar os identificadores de todas as Contas da AWS às quais você deseja conceder a permissão para restaurar o snapshot manual, selecione Salvar para salvar as alterações.

AWS CLI

Para compartilhar um DB snapshot, use o comando `aws rds modify-db-snapshot-attribute`. Use o parâmetro `--values-to-add` para adicionar uma lista dos IDs das Contas da AWS autorizadas a restaurar o snapshot manual.

Example compartilhamento de um snapshot com uma única conta

O exemplo a seguir permite que o identificador Conta da AWS, `123456789012`, restaure o snapshot do banco de dados chamado `db7-snapshot`.

Para Linux, macOS ou Unix:

```
aws rds modify-db-snapshot-attribute \
--db-snapshot-identifier db7-snapshot \
--attribute-name restore \
--values-to-add 123456789012
```

Para Windows:

```
aws rds modify-db-snapshot-attribute ^
--db-snapshot-identifier db7-snapshot ^
--attribute-name restore ^
--values-to-add 123456789012
```

Example compartilhamento de um snapshot com múltiplas contas

O exemplo a seguir permite que dois identificadores Conta da AWS, `111122223333` e `444455556666`, restaurem o snapshot do banco de dados chamado `manual-snapshot1`.

Para Linux, macOS ou Unix:

```
aws rds modify-db-snapshot-attribute \
--db-snapshot-identifier manual-snapshot1 \
--attribute-name restore \
--values-to-add {"111122223333","444455556666"}
```

Para Windows:

```
aws rds modify-db-snapshot-attribute ^
--db-snapshot-identifier manual-snapshot1 ^
--attribute-name restore ^
--values-to-add "[\"111122223333\",\"444455556666\"]"
```

> **Note**
>
> Ao usar o prompt de comando do Windows, você deve fazer o escape das aspas duplas (") no código JSON, prefixando-as com uma barra invertida (\).

Para listar as Contas da AWS habilitadas para restaurar um snapshot, utilize o comando `describe-db-snapshot-attributes` da AWS CLI.

API do RDS

Também é possível compartilhar um snapshot de banco de dados manual com outras Contas da AWS usando a API do Amazon RDS. Para fazer isso, chame a operação `ModifyDBSnapshotAttribute`. Especifique `restore` para `AttributeName` e use o parâmetro `ValuesToAdd` para adicionar uma lista dos IDs das Contas da AWS que têm autorização para restaurar o snapshot manual.

Para tornar um snapshot manual público e restaurável por todas as Contas da AWS, use o valor `all`. No entanto, tome cuidado para não adicionar o valor `all` a nenhum snapshot manual com informações privadas que não deseja disponibilizar para todas as Contas da AWS. Além disso, não especifique `all` para snapshots criptografados, pois não é possível torná-los públicos.

Para listar todas as Contas da AWS que têm permissão para restaurar um snapshot, utilize a operação `DescribeDBSnapshotAttributes` da API.

Compartilhar snapshots públicos para o Amazon RDS

É possível compartilhar um snapshot manual não criptografado como público, disponibilizando-o para todas as Contas da AWS. Ao compartilhar um snapshot como público, verifique se suas informações privadas não estão incluídas nos snapshots públicos.

Quando um snapshot é compartilhado publicamente, ele fornece a todas as Contas da AWS permissão para copiar o snapshot e para criar instâncias de banco de dados dele.

Você não é cobrado pelo armazenamento de backup de snapshots públicos pertencentes a outras contas. Você é cobrado apenas pelos snapshots pertencentes a você.

Se você copiar um snapshot público, será o proprietário da cópia. Você será cobrado pelo armazenamento de backup da sua cópia do snapshot. Se criar uma instância de banco de dados de um snapshot público, você será cobrado por essa instância de banco de dados. Para obter informações sobre a definição de preço do Amazon RDS, consulte a página de produto do Amazon RDS.

Você pode excluir somente os snapshots públicos pertencentes a você. Para excluir um snapshot compartilhado ou público, você deve fazer login na Conta da AWS à qual pertence esse snapshot.

Visualizar snapshots públicos pertencentes a outras Contas da AWS

Você pode visualizar snapshots públicos pertencentes a outras contas em uma região da AWS específica na guia Public (Público) na página Snapshots do console do Amazon RDS. Seus snapshots (aqueles pertencentes à sua conta) não aparecem nesta guia.

Para visualizar snapshots públicos

1. Abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Snapshots.
3. Selecione a guia Public (Público).

   Os snapshots públicos são exibidos. Você pode ver qual conta possui um snapshot público na coluna Owner (Proprietário).

   > **Note**
   >
   > Talvez seja necessário modificar as preferências da página selecionando o ícone de engrenagem no canto superior direito da lista Public snapshots (Instantâneos públicos) para ver esta coluna.

Visualização dos seus próprios snapshots públicos

É possível usar o seguinte comando da AWS CLI (somente Unix) para visualizar os snapshots públicos pertencentes à Conta da AWS em uma determinada região da AWS.

```
aws rds describe-db-snapshots --snapshot-type public --include-public |
 grep account_number
```

A saída retornada será semelhante ao exemplo a seguir se você tiver instantâneos públicos.

```
"DBSnapshotArn": "arn:aws:rds:us-east-1:123456789012:snapshot:mysnapshot1",
"DBSnapshotArn": "arn:aws:rds:us-east-1:123456789012:snapshot:mysnapshot2",
```

> **Note**
>
> Você pode ver entradas duplicadas para `DBSnapshotIdentifier` ou `SourceDBSnapshotIdentifier`.

Compartilhamento de snapshots públicos de versões obsoletas do mecanismo de banco de dados

Não é permitido restaurar nem copiar snapshots públicos de versões obsoletas do mecanismo de banco de dados.

Os mecanismos de banco de dados do RDS para Oracle e do RDS para PostgreSQL permitem o upgrade direto das versões do mecanismo do snapshot de banco de dados. É possível fazer upgrade dos snapshots e, depois, compartilhá-los novamente de forma pública. Para obter mais informações, consulte as informações a seguir.

- Como atualizar para um snapshot de banco de dados Oracle
- Atualizar uma versão do mecanismo de snapshot de banco de dados PostgreSQL

Para outros mecanismos de banco de dados, execute as etapas a seguir para disponibilizar o snapshot público incompatível existente para restauração ou cópia:

1. Marque o snapshot como privado.
2. Restaure o snapshot.
3. Faça upgrade da instância de banco de dados restaurada para uma versão de mecanismo compatível.
4. Crie um snapshot.
5. Compartilhe novamente o snapshot publicamente.

Compartilhar snapshots criptografados para o Amazon RDS

É possível compartilhar snapshots de banco de dados que foram criptografados "em repouso" usando o algoritmo de criptografia AES-256, conforme descrito em Criptografar recursos do Amazon RDS.

As seguintes restrições se aplicam ao compartilhamento de snapshots criptografados:

- Não é possível compartilhar snapshots criptografados como públicos.
- Não é possível compartilhar snapshots do Oracle ou do Microsoft SQL Server criptografados usando Transparent Data Encryption (TDE).
- Não é possível compartilhar um snapshot criptografado usando a chave do KMS padrão da Conta da AWS que compartilhou o snapshot.

  Para ter mais informações sobre o gerenciamento de chaves AWS KMS para o Amazon RDS, consulte Gerenciamento de AWS KMS key.

Para contornar o problema da chave do KMS padrão, realize as seguintes tarefas:

1. Criar uma chave gerenciada pelo cliente e conceder acesso a ela.
2. Copiar e compartilhar o snapshot da conta de origem.
3. Copiar o snapshot compartilhado na conta de destino.

Criar uma chave gerenciada pelo cliente e conceder acesso a ela

Primeiro, você deve criar uma chave do KMS personalizada na mesma Região da AWS do snapshot de banco de dados criptografado. Ao criar a chave gerenciada pelo cliente, conceda acesso a ela a outra Conta da AWS.

> **Note**
>
> Você também pode usar uma chave KMS de outra conta da AWS quando a política de chave concede acesso às contas de origem e de destino.

Como criar uma chave gerenciada pelo cliente e conceder acesso a ela

1. Faça login no AWS Management Console pela Conta da AWS de origem.

2. Abra o console do AWS KMS em https://console.aws.amazon.com/kms.
3. Para alterar o Região da AWS, use o seletor de Região no canto superior direito da página.
4. No painel de navegação, escolha Chaves gerenciadas pelo cliente.
5. Escolha Criar chave.
6. Na página Configurar chave:
    a. Em Tipo de chave, selecione Simétrico.
    b. Em Uso da chave, selecione Criptografar e descriptografar.
    c. Expanda Advanced options (Opções avançadas).
    d. Em Origem do material de chaves, selecione KMS.
    e. Em Regionalidade, selecione Chave de região única.
    f. Escolha Próximo.
7. Na página Adicionar rótulos:
    a. Para Alias, insira um nome de exibição para a chave do KMS, por exemplo, **`share-snapshot`**.
    b. (Opcional) Insira uma descrição para a chave do KMS.
    c. (Opcional) Adicione tags à chave do KMS.
    d. Escolha Próximo.
8. Na página Definir permissões administrativas da chave, escolha Próximo.
9. Na página Definir permissões de uso da chave:
    a. Em Outras Contas da AWS, selecione Adicionar outra Conta da AWS.
    b. Insira o ID da Conta da AWS à qual você deseja conceder acesso.

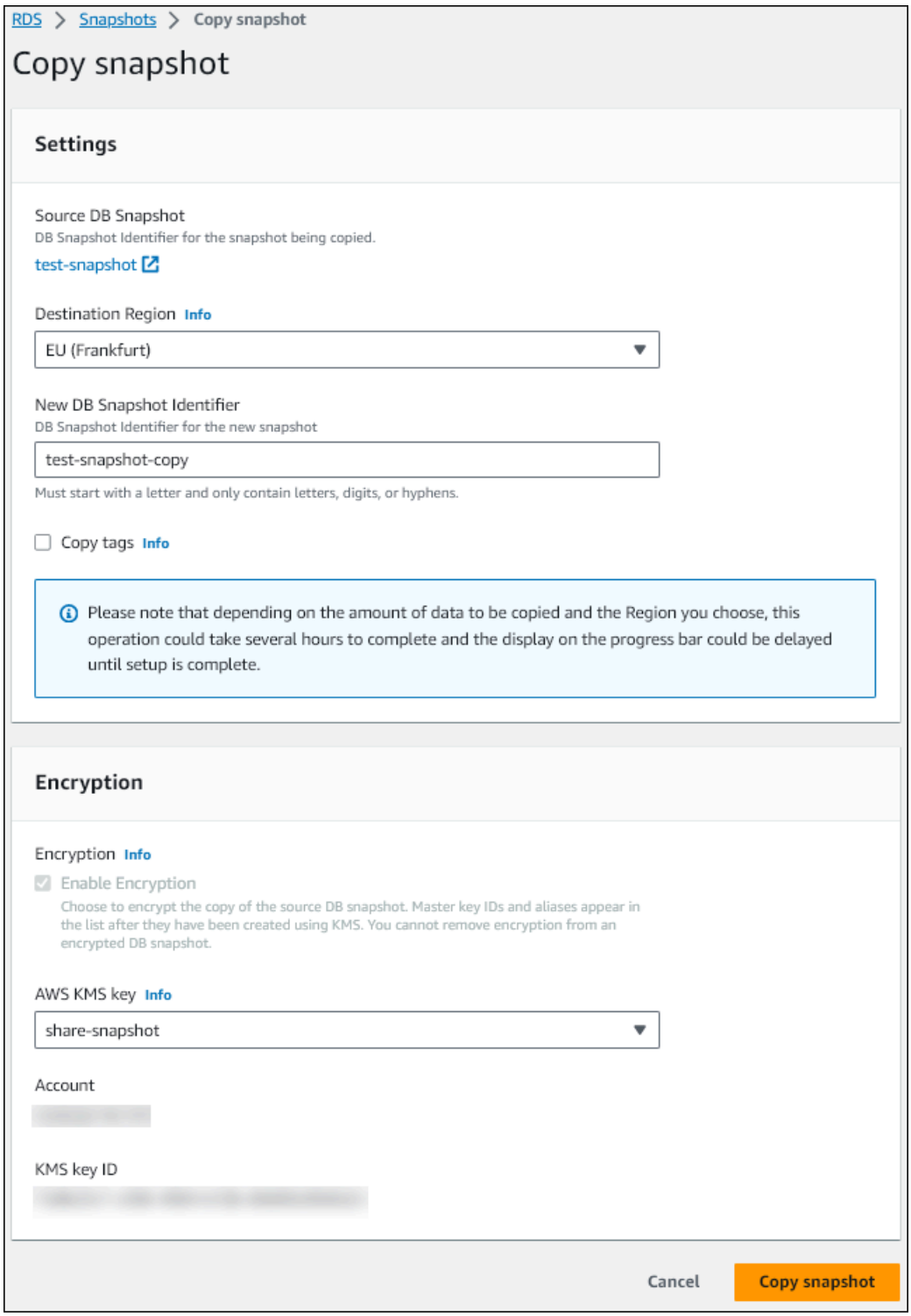
        É possível conceder acesso a várias Contas da AWS.
    c. Escolha Próximo.
10. Revise a chave do KMS e escolha Concluir.

Copiar e compartilhar o snapshot da conta de origem

Depois, você deve copiar o snapshot de banco de dados de origem para um novo snapshot usando a chave gerenciada pelo cliente. Depois, você vai compartilhá-lo com a Conta da AWS de destino.

Como copiar e compartilhar o snapshot

1. Faça login no AWS Management Console pela Conta da AWS de origem.
2. Abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
3. No painel de navegação, escolha Snapshots.
4. Selecione o snapshot de banco de dados que você deseja copiar.
5. Para Actions (Ações), escolha Copy Snapshot (Copiar snapshot).
6. Na página Copiar snapshot:
    a. Em Região de destino, selecione a Região da AWS onde você criou a chave gerenciada pelo cliente no procedimento anterior.
    b. Digite o nome da cópia do snapshot de banco de dados em Novo identificador de snapshot de banco de dados.
    c. Para AWS KMS key, selecione a chave gerenciada pelo cliente que você criou.



d. Escolha Copy snapshot (Copiar snapshot).

7. Quando a cópia do snapshot estiver disponível, selecione-a.
8. Em Actions (Ações), selecione Share Snapshot (Compartilhar snapshot).
9. Na página Permissões de snapshot:

a. Insira o ID da Conta da AWS com a qual você está compartilhando a cópia do snapshot e selecione Adicionar.
b. Escolha Salvar.

O snapshot é compartilhado.

Copiar o snapshot compartilhado na conta de destino

Agora você pode copiar o snapshot compartilhado na Conta da AWS de destino.

Como copiar o snapshot compartilhado

1. Faça login no AWS Management Console pela Conta da AWS de destino.
2. Abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
3. No painel de navegação, escolha Snapshots.
4. Selecione a guia Compartilhado comigo.
5. Selecione o snapshot compartilhado.
6. Para Actions (Ações), escolha Copy Snapshot (Copiar snapshot).
7. Escolha as configurações para copiar o snapshot como no procedimento anterior, mas use uma AWS KMS key que pertença à conta de destino.

   Escolha Copy snapshot (Copiar snapshot).

Interromper o compartilhamento de snapshots no Amazon RDS

Para parar de compartilhar um snapshot de banco de dados, é necessário remover a permissão da Conta da AWS de destino.

Console

Como interromper o compartilhamento de um snapshot de banco de dados manual com uma Conta da AWS

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Snapshots.

3. Selecione o snapshot manual que você deseja interromper o compartilhamento.
4. Selecione Actions (Ações) e, depois, Share Snapshot (Compartilhar snapshot).
5. Para remover a permissão de uma Conta da AWS, selecione Excluir para o identificador da conta da AWS na lista de contas autorizadas.
6. Escolha Salvar para salvar as alterações.

CLI

Para remover um identificador Conta da AWS na lista, use o parâmetro `--values-to-remove`.

Example interrupção do compartilhamento do snapshot

O exemplo a seguir impede que o ID 444455556666 da Conta da AWS restaure o snapshot.

Para Linux, macOS ou Unix:

```
aws rds modify-db-snapshot-attribute \
--db-snapshot-identifier manual-snapshot1 \
--attribute-name restore \
--values-to-remove 444455556666
```

Para Windows:

```
aws rds modify-db-snapshot-attribute ^
--db-snapshot-identifier manual-snapshot1 ^
--attribute-name restore ^
--values-to-remove 444455556666
```

API do RDS

Para remover a permissão de compartilhamento de uma Conta da AWS, use a operação `ModifyDBSnapshotAttribute` com `AttributeName` definido como `restore` e o parâmetro `ValuesToRemove`. Para marcar um snapshot manual como privado, remova o valor `all` na lista de valores do atributo `restore`.

Exportação de dados de snapshot de banco de dados para o Amazon S3 para Amazon RDS

Você pode exportar dados de snapshots de banco de dados para um bucket do Amazon S3. O processo de exportação é executado em segundo plano e não afeta a performance do banco de dados ativo.

Quando você exporta um snapshot de banco de dados, o Amazon RDS extrai os dados do snapshot e os armazena em um bucket do Amazon S3. Os dados são armazenados em um formato Apache Parquet que é compactado e consistente.

Você pode exportar todos os tipos de snapshots de banco de dados, inclusive snapshots manuais, snapshots automatizados do sistema e snapshots criados pelo serviço AWS Backup. Por padrão, todos os dados no snapshot são exportados. No entanto, você pode optar por exportar conjuntos específicos de bancos de dados, esquemas ou tabelas.

Depois que os dados são exportados, você pode analisar os dados exportados diretamente por meio de ferramentas, como Amazon Athena ou Amazon Redshift Spectrum. Para ter mais informações sobre como usar o Athena para ler os dados do Parquet, consulte Parquet SerDe no Guia do usuário do Amazon Athena. Para ter mais informações sobre como usar o Redshift Spectrum para ler os dados do Parquet, consulte COPY de formatos de dados colunares no Guia do desenvolvedor de banco de dados do Amazon Redshift.

> **Warning**
>
> Não é possível restaurar dados de snapshots exportados do S3 para uma nova instância de banco de dados nem importar dados de snapshots do S3 para uma instância de banco de dados existente. Entretanto, você pode processar os dados usando o Amazon Athena ou o Redshift Spectrum para análise. Além disso, você pode usar o AWS Glue para transformar os dados e depois importá-los no Amazon RDS usando ferramentas como o AWS DMS ou scripts personalizados.

Consulte mais informações sobre como exportar snapshots de banco de dados para o Amazon S3 nos tópicos a seguir.

Tópicos

- Monitorar exportações de snapshots para o Amazon RDS

- Cancelar uma tarefa de exportação de snapshot para o Amazon RDS
- Mensagens de falha de tarefas de exportação do Amazon S3 para o Amazon RDS
- Solucionar problemas de erros de permissões do RDS para PostgreSQL
- Convenções de nomenclatura de arquivos para exportações para o Amazon S3 para o Amazon RDS
- Conversão de dados ao exportar para um bucket do Amazon S3 para RDS

Visão geral da exportação de dados de snapshot

Use o seguinte processo para exportar dados de um snapshot de banco de dados para um bucket do Amazon S3. Para obter mais detalhes, consulte as seções a seguir.

1. Identifique o snapshot a ser exportado.

   Use um snapshot existente, manual ou automatizado, ou crie um snapshot manual de uma instância de banco de dados ou um cluster de banco de dados multi-AZ.
2. Configure o acesso ao bucket do Amazon S3.

   Um bucket é um contêiner de objetos ou arquivos do Amazon S3. Para fornecer informações para acesso a um bucket, execute as seguintes etapas:

   a. Identifique o bucket do S3 para o qual o snapshot deve ser exportado. O bucket do S3 deve estar na mesma região da AWS que o snapshot. Para ter mais informações, consulte Identificar o bucket do Amazon S3 para exportar.

   b. Crie uma função do AWS Identity and Access Management (IAM) que conceda à tarefa de exportação de snapshot acesso ao bucket do S3. Para ter mais informações, consulte Fornecer acesso a um bucket do Amazon S3 usando um perfil do IAM.
3. Crie uma criptografia simétrica do AWS KMS key para a criptografia do lado do servidor. A chave do KMS é usada pela tarefa de exportação de snapshot para configurar a criptografia do lado do servidor do AWS KMS ao gravar dados de exportação no S3.

   A política de chave do KMS deve incluir as permissões `kms:CreateGrant` e `kms:DescribeKey`. Para ter mais informações sobre o uso de chaves do KMS no Amazon RDS, consulte Gerenciamento de AWS KMS key.

   Além disso, se você tiver uma instrução de negação na política de chaves do KMS, exclua explicitamente a entidade principal de serviço da AWS `export.rds.amazonaws.com`.

Você pode utilizar uma chave do KMS na sua conta do AWS ou pode utilizar uma chave do KMS entre contas. Para ter mais informações, consulte Uso de um AWS KMS key entre contas para criptografar exportações do Amazon S3.

4. Exporte o snapshot para o Amazon S3 usando o console ou o comando `start-export-task` da CLI. Para ter mais informações, consulte Exportar um snapshot de banco de dados para um bucket do Amazon S3.
5. Para acessar os seus dados exportados no bucket do Amazon S3, consulte Como carregar, baixar e gerenciar objetos no Guia do usuário do Amazon Simple Storage Service.

Configurar o acesso a um bucket do Amazon S3

Para exportar dados de snapshot de banco de dados para um arquivo do Amazon S3, primeiro conceda permissão ao snapshot para acessar o bucket do Amazon S3. Crie um perfil do IAM para permitir que o serviço Amazon RDS grave no bucket do Amazon S3.

Tópicos

- Identificar o bucket do Amazon S3 para exportar
- Fornecer acesso a um bucket do Amazon S3 usando um perfil do IAM
- Utilizar um bucket do Amazon S3 entre contas
- Uso de um AWS KMS key entre contas para criptografar exportações do Amazon S3

Identificar o bucket do Amazon S3 para exportar

Identifique o bucket do Amazon S3 para o qual exportar o snapshot de banco de dados. Use um bucket do S3 existente ou crie um novo bucket do S3.

> **Note**
>
> O bucket do S3 para o qual exportar deve estar na mesma região da AWS que o snapshot.

Para ter mais informações sobre como trabalhar com buckets do Amazon S3, consulte o seguinte no Guia do usuário do Amazon Simple Storage Service:

- Como visualizar as propriedades de um bucket do S3?

- Como habilitar a criptografia padrão em um bucket do Amazon S3?
- Como criar um bucket do S3?

Fornecer acesso a um bucket do Amazon S3 usando um perfil do IAM

Antes de exportar dados de snapshot de banco de dados para o Amazon S3, forneça permissões de acesso de gravação ao bucket do Amazon S3 às tarefas de exportação.

Para conceder essa permissão, crie uma política do IAM que forneça acesso ao bucket e, depois, crie um perfil do IAM e anexe a política ao perfil. Posteriormente, você atribui o perfil do IAM à tarefa de exportação de snapshot.

Para obter mais informações sobre outras ferramentas de gerenciamento de acesso do Amazon S3, consulte Controle de acesso no Amazon S3 no Guia do usuário do Amazon S3.

> **Important**
>
> Se você planeja usar o AWS Management Console para exportar o snapshot, poderá optar por criar a política do IAM e a função automaticamente ao exportar o snapshot. Para obter instruções, consulte Exportar um snapshot de banco de dados para um bucket do Amazon S3.

Como fornecer acesso ao Amazon S3 às tarefas de snapshot de banco de dados

1. Crie uma política do IAM. Essa política fornece as permissões de bucket e objeto que permitem que sua tarefa de exportação de snapshot acesse o Amazon S3.

   Nessa política, inclua as seguintes ações necessárias para permitir a transferência de arquivos do Amazon RDS para um bucket do S3:

   - `s3:PutObject*`
   - `s3:GetObject*`
   - `s3:ListBucket`
   - `s3:DeleteObject*`
   - `s3:GetBucketLocation`

Na política, inclua os recursos a seguir para identificar o bucket do S3 e os objetos no bucket. A lista de recursos a seguir mostra o formato do nome de recurso da Amazon (ARN) para acessar o Amazon S3.

- `arn:aws:s3:::amzn-s3-demo-bucket`
- `arn:aws:s3:::amzn-s3-demo-bucket/*`

Para ter mais informações sobre como criar uma política do IAM para o Amazon RDS, consulte Criar e usar uma política do IAM para acesso do banco de dados do IAM. Consulte também Tutorial: Criar e anexar sua primeira política gerenciada pelo cliente no Guia do usuário do IAM.

O comando da AWS CLI a seguir cria uma política do IAM denominada `ExportPolicy` com essas opções. Ele concede acesso a um bucket denominado *amzn-s3-demo-bucket*.

> **Note**
>
> Depois de criar a política, anote o ARN da política. O ARN será necessário para uma etapa posterior, quando você anexar a política a um perfil do IAM.

```
aws iam create-policy  --policy-name ExportPolicy --policy-document '{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ExportPolicy",
            "Effect": "Allow",
            "Action": [
                "s3:PutObject*",
                "s3:ListBucket",
                "s3:GetObject*",
                "s3:DeleteObject*",
                "s3:GetBucketLocation"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket",
                "arn:aws:s3:::amzn-s3-demo-bucket/*"
            ]
        }
```

```
    ]
}'
```

2. Crie um perfil do IAM para que o Amazon RDS possa assumir esse perfil do IAM em seu nome a fim de acessar os buckets do Amazon S3. Para ter mais informações, consulte Criar um perfil para delegar permissões a um usuário do IAM no Guia do usuário do IAM.

   O exemplo a seguir mostra como usar o comando da AWS CLI para criar uma função chamada `rds-s3-export-role`.

```
aws iam create-role  --role-name rds-s3-export-role  --assume-role-policy-document
 '{
    "Version": "2012-10-17",
    "Statement": [
      {
        "Effect": "Allow",
        "Principal": {
           "Service": "export.rds.amazonaws.com"
         },
        "Action": "sts:AssumeRole"
      }
    ]
  }'
```

3. Anexe a política do IAM que você criou ao perfil do IAM que você criou.

   O seguinte comando da AWS CLI anexa a política criada anteriormente à função chamada `rds-s3-export-role`. Substitua *`your-policy-arn`* pelo ARN da política que você anotou em uma etapa anterior.

```
aws iam attach-role-policy  --policy-arn your-policy-arn  --role-name rds-s3-
export-role
```

Utilizar um bucket do Amazon S3 entre contas

É possível utilizar buckets do Amazon S3 entre contas da AWS. Para utilizar um bucket entre contas, adicione uma política de bucket para permitir o acesso ao perfil do IAM que você está utilizando para as exportações do S3. Para ter mais informações, consulte Exemplo 2: proprietário do bucket concedendo permissões para o bucket entre contas.

Vincule uma política de bucket ao bucket, como mostra o exemplo a seguir.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::123456789012:role/Admin"
            },
            "Action": [
                "s3:PutObject*",
                "s3:ListBucket",
                "s3:GetObject*",
                "s3:DeleteObject*",
                "s3:GetBucketLocation"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-destination-bucket",
                "arn:aws:s3:::amzn-s3-demo-destination-bucket/*"
            ]
        }
    ]
}
```

Uso de um AWS KMS key entre contas para criptografar exportações do Amazon S3

Você pode utilizar um AWS KMS key entre contas para criptografar exportações do Amazon S3. Primeiro, você adiciona uma política de chaves à conta local e depois adiciona políticas do IAM na conta externa. Para ter mais informações, consulte o tópico sobre como Permitir que usuários de outras contas utilizem uma chave do KMS.

Para utilizar uma chave do KMS entre contas

1. Adicione uma política de chaves à conta local.

   O exemplo a seguir concede a `ExampleRole` e `ExampleUser` na conta externa 444455556666 permissões na conta local 123456789012.

```
{
    "Sid": "Allow an external account to use this KMS key",
    "Effect": "Allow",
    "Principal": {
        "AWS": [
            "arn:aws:iam::444455556666:role/ExampleRole",
            "arn:aws:iam::444455556666:user/ExampleUser"
        ]
    },
    "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:CreateGrant",
        "kms:DescribeKey",
        "kms:RetireGrant"
    ],
    "Resource": "*"
}
```

2. Adicione políticas do IAM à conta externa.

   O seguinte exemplo de política do IAM permite que a entidade principal use a chave do KMS na conta 123456789012 para operações criptográficas. Para conceder essa permissão a `ExampleRole` e `ExampleUser` na conta 444455556666, vincule a política a eles nessa conta.

```
{
    "Sid": "Allow use of KMS key in account 123456789012",
    "Effect": "Allow",
    "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:CreateGrant",
        "kms:DescribeKey",
        "kms:RetireGrant"
    ],
    "Resource": "arn:aws:kms:us-
west-2:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab"
```

```
}
```

Exportar um snapshot de banco de dados para um bucket do Amazon S3

Você pode ter até cinco tarefas de exportação de snapshot de banco de dados simultâneas em andamento por Conta da AWS.

> **Note**
>
> A exportação de snapshots do RDS pode levar um tempo, dependendo do tipo e do tamanho do banco de dados. A tarefa de exportação primeiro restaura e escala todo o banco de dados antes de extrair os dados para o Amazon S3. O andamento da tarefa durante essa fase é exibido como Starting (Iniciando). Quando a tarefa muda para a exportação de dados para o S3, o andamento é exibido como In progress (Em andamento).
>
> O tempo necessário para que a exportação seja concluída depende dos dados armazenados no banco de dados. Por exemplo, tabelas com chaves primárias numéricas bem distribuída ou colunas de índice serão exportadas de maneira mais rápida. Tabelas sem uma coluna adequada para particionamento e tabelas com somente um índice em uma coluna baseada em string demorarão mais. Esse tempo de exportação mais demorado ocorre porque a exportação utiliza um processo de thread único mais lento.

Você pode exportar um snapshot de banco de dados para o Amazon S3 usando o AWS Management Console, a AWS CLI ou a API do RDS. Para exportar um snapshot de banco de dados para um bucket do Amazon S3 entre contas, use a AWS CLI ou a API do RDS.

Se você usar uma função do Lambda para exportar um snapshot, adicione a ação `kms:DescribeKey` à política da função do Lambda. Para ter mais informações, consulte Permissões doAWS Lambda.

Console

A opção de console Export to Amazon S3 (Exportar para o Amazon S3) é exibido somente para snapshots que podem ser exportados para o Amazon S3. Um snapshot pode não estar disponível para exportação devido aos seguintes motivos:

- O mecanismo de banco de dados não tem suporte para exportação do S3.
- A versão do mecanismo de banco de dados não é aceita para exportação do S3.

- A exportação do S3 não é aceita na região da AWS onde o snapshot foi criado.

Para exportar um snapshot de banco de dados

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Snapshots.
3. Nas guias, escolha o tipo de snapshot que deseja exportar.
4. Na lista de snapshots, escolha o snapshot que deseja exportar.
5. Em Actions (Ações), escolha Export to Amazon S3 (Exportar para o Amazon S3).

   A janela Export to Amazon S3 (Exportar para o Amazon S3) é exibida.
6. Em Export identifier (Identificador de exportação), insira um nome para identificar a tarefa de exportação. Esse valor também é usado para o nome do arquivo criado no bucket do S3.
7. Escolha os dados a serem exportados:

   - Escolha All (Tudo) para exportar todos os dados do snapshot.
   - Escolha Partial (Parcial) para exportar partes específicas do snapshot. Para identificar quais partes do snapshot exportar, insira um ou mais bancos de dados, esquemas ou tabelas para Identifiers (Identificadores), separados por espaços.

     Use o formato a seguir:

     ```
     database[.schema][.table] database2[.schema2][.table2] ... databasen[.scheman]
     [.tablen]
     ```

     Por exemplo:

     ```
     mydatabase mydatabase2.myschema1 mydatabase2.myschema2.mytable1
      mydatabase2.myschema2.mytable2
     ```
8. Em S3 bucket (Bucket do S3), escolha o bucket para o qual exportar.

   Para atribuir os dados exportados a um caminho de pasta no bucket do S3, insira o caminho opcional para o S3 prefix (Prefixo do S3).
9. Em IAM role (Perfil do IAM), escolha um perfil que conceda a você acesso de gravação ao bucket do S3 escolhido ou crie um novo perfil.

- Se você criou um perfil seguindo as etapas em Fornecer acesso a um bucket do Amazon S3 usando um perfil do IAM, escolha esse perfil.
- Se você não criou um perfil que conceda acesso de gravação ao bucket do S3 escolhido, selecione Create a new role (Criar um perfil) para criar o perfil automaticamente. Depois, insira um nome para o perfil em Nome do perfil do IAM.

10. Em AWS KMS key, insira o ARN da chave a ser usada para criptografar os dados exportados.
11. Escolha Export to Amazon S3 (Exportar para o Amazon S3).

AWS CLI

Para exportar um snapshot de banco de dados para o Amazon S3 usando a AWS CLI, use o comando start-export-task com as seguintes opções necessárias:

- `--export-task-identifier`
- `--source-arn`
- `--s3-bucket-name`
- `--iam-role-arn`
- `--kms-key-id`

Nos exemplos a seguir, a tarefa de exportação do snapshot é denominada *my_snapshot_export*, que exporta um snapshot para um bucket do S3 chamado *amzn-s3-demo-bucket*.

Example

Para Linux, macOS ou Unix:

```
aws rds start-export-task \
    --export-task-identifier my-snapshot-export \
    --source-arn arn:aws:rds:AWS_Region:123456789012:snapshot:snapshot-name \
    --s3-bucket-name amzn-s3-demo-bucket \
    --iam-role-arn iam-role \
    --kms-key-id my-key
```

Para Windows:

```
aws rds start-export-task ^
    --export-task-identifier my-snapshot-export ^
```

```
    --source-arn arn:aws:rds:AWS_Region:123456789012:snapshot:snapshot-name ^
    --s3-bucket-name amzn-s3-demo-bucket ^
    --iam-role-arn iam-role ^
    --kms-key-id my-key
```

Segue um exemplo de saída.

```
{
    "Status": "STARTING",
    "IamRoleArn": "iam-role",
    "ExportTime": "2019-08-12T01:23:53.109Z",
    "S3Bucket": "my-export-bucket",
    "PercentProgress": 0,
    "KmsKeyId": "my-key",
    "ExportTaskIdentifier": "my-snapshot-export",
    "TotalExtractedDataInGB": 0,
    "TaskStartTime": "2019-11-13T19:46:00.173Z",
    "SourceArn": "arn:aws:rds:AWS_Region:123456789012:snapshot:snapshot-name"
}
```

Para fornecer um caminho de pasta no bucket do S3 para a exportação do snapshot, inclua a opção `--s3-prefix` no comando start-export-task.

API do RDS

Para exportar um snapshot de banco de dados para o Amazon S3, usando a API do Amazon RDS, use a operação StartExportTask com os seguintes parâmetros necessários:

- `ExportTaskIdentifier`
- `SourceArn`
- `S3BucketName`
- `IamRoleArn`
- `KmsKeyId`

Disponibilidade de regiões e versões

A disponibilidade e a compatibilidade de recursos variam entre versões específicas de cada mecanismo de banco de dados e entre Regiões da AWS. Para ter mais informações sobre a disponibilidade de versões e regiões para a exportação de snapshots para o S3, consulte Regiões e

mecanismos de banco de dados compatíveis com a exportação de snapshots para o S3 no Amazon RDS.

Limitações

A exportação de dados de snapshots de banco de dados para o Amazon S3 apresenta as seguintes limitações:

- Você não pode executar várias tarefas de exportação para o mesmo snapshot de banco de dados simultaneamente. Isso se aplica a exportações totais e parciais.
- Não há suporte para a exportação de snapshots de bancos de dados que usam armazenamento magnético.
- As exportações para o S3 não são compatíveis com prefixos do S3 contendo dois pontos (:).
- Os seguintes caracteres no caminho do arquivo do S3 são convertidos em sublinhados (_) durante a exportação:

```
\ ` " (space)
```

- Se um banco de dados, esquema ou tabela tiver caracteres em seu nome diferentes dos a seguir, a exportação parcial não será aceita. No entanto, você pode exportar o snapshot de banco de dados inteiro.
  - Letras latinas (A–Z)
  - Dígitos (0–9)
  - Símbolo do dólar ($)
  - Sublinhado (_)
- Espaços ( ) e alguns caracteres não são compatíveis em nomes de colunas de tabelas de banco de dados. As tabelas com os seguintes caracteres em nomes de colunas são ignoradas durante a exportação:

```
, ; { } ( ) \n \t = (space)
```

- As tabelas com barras (/) em seus nomes são ignoradas durante a exportação.
- As tabelas temporárias e não registradas do RDS para PostgreSQL são ignoradas durante a exportação.
- Se os dados contiverem um objeto grande, como um BLOB ou um CLOB, com tamanho próximo ou superior a 500 MB, ocorrerá uma falha na exportação.

- Se uma tabela contiver uma linha grande próxima ou maior que 2 GB, a tabela será ignorada durante a exportação.
- Em relação a exportações parciais, a lista `ExportOnly` tem um tamanho máximo de 200 KB.
- É altamente recomendável que você use um nome exclusivo para cada tarefa de exportação. Se você não usar um nome de tarefa exclusivo, poderá receber a seguinte mensagem de erro:

  ExportTaskAlreadyExistsFault: An error occurred (ExportTaskAlreadyExists) when calling the StartExportTask operation: The export task with the ID *xxxxx* already exists [ExportTaskAlreadyExistsFault: ocorreu um erro (ExportTaskAlreadyExists) ao chamar a operação StartExportTask: a tarefa de exportação com o ID xxxxx já existe].
- Você pode excluir um snapshot enquanto exporta seus dados para o S3, mas ainda é cobrado pelo armazenamento desse snapshot até que a tarefa de exportação seja concluída.
- Não é possível restaurar dados de snapshots exportados do S3 para uma nova instância de banco de dados nem importar dados de snapshots do S3 para uma instância de banco de dados existente.
- Você pode ter até cinco tarefas de exportação de snapshot de banco de dados simultâneas em andamento por Conta da AWS.
- Para exportar um snapshot de banco de dados para um bucket do Amazon S3 entre contas, você deve usar a AWS CLI ou a API do RDS.
- Depois que o Amazon RDS concluir uma tarefa de exportação, talvez você precise esperar um curto período para iniciar outra tarefa de exportação a partir do mesmo snapshot de banco de dados.
- Não é possível exportar visualizações ou visões materializadas.
- O RDS Export to S3 não oferece suporte ao controle de acesso baseado em tags para o GuardDuty Malware Protection for S3.

Monitorar exportações de snapshots para o Amazon RDS

Você pode monitorar as exportações de snapshots de banco de dados usando o AWS Management Console, a AWS CLI e a API do RDS.

Console

Como monitorar exportações de snapshots de banco de dados

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Snapshots.
3. Para visualizar a lista de exportações de snapshots, escolha a guia Exportações no Amazon S3.
4. Para visualizar informações sobre uma exportação de snapshot específica, escolha a tarefa de exportação.

AWS CLI

Para monitorar exportações de snapshots de banco de dados usando o AWS CLI, use o comando describe-export-tasks.

O exemplo a seguir mostra como exibir informações atuais sobre todas as exportações de snapshots.

Example

```
aws rds describe-export-tasks

{
    "ExportTasks": [
        {
            "Status": "CANCELED",
            "TaskEndTime": "2019-11-01T17:36:46.961Z",
            "S3Prefix": "something",
            "ExportTime": "2019-10-24T20:23:48.364Z",
            "S3Bucket": "amzn-s3-demo-bucket",
            "PercentProgress": 0,
            "KmsKeyId": "arn:aws:kms:AWS_Region:123456789012:key/K7MDENG/
bPxRfiCYEXAMPLEKEY",
            "ExportTaskIdentifier": "anewtest",
            "IamRoleArn": "arn:aws:iam::123456789012:role/export-to-s3",
            "TotalExtractedDataInGB": 0,
            "TaskStartTime": "2019-10-25T19:10:58.885Z",
            "SourceArn": "arn:aws:rds:AWS_Region:123456789012:snapshot:parameter-
groups-test"
        },
```

```
{
            "Status": "COMPLETE",
            "TaskEndTime": "2019-10-31T21:37:28.312Z",
            "WarningMessage": "{\"skippedTables\":[],\"skippedObjectives\":[],\"general
\":[{\"reason\":\"FAILED_TO_EXTRACT_TABLES_LIST_FOR_DATABASE\"}]}",
            "S3Prefix": "",
            "ExportTime": "2019-10-31T06:44:53.452Z",
            "S3Bucket": "amzn-s3-demo-bucket1",
            "PercentProgress": 100,
            "KmsKeyId": "arn:aws:kms:AWS_Region:123456789012:key/2Zp9Utk/
h3yCo8nvbEXAMPLEKEY",
            "ExportTaskIdentifier": "thursday-events-test",
            "IamRoleArn": "arn:aws:iam::123456789012:role/export-to-s3",
            "TotalExtractedDataInGB": 263,
            "TaskStartTime": "2019-10-31T20:58:06.998Z",
            "SourceArn":
 "arn:aws:rds:AWS_Region:123456789012:snapshot:rds:example-1-2019-10-31-06-44"
        },
        {
            "Status": "FAILED",
            "TaskEndTime": "2019-10-31T02:12:36.409Z",
            "FailureCause": "The S3 bucket edgcuc-export isn't located in the current
 AWS Region. Please, review your S3 bucket name and retry the export.",
            "S3Prefix": "",
            "ExportTime": "2019-10-30T06:45:04.526Z",
            "S3Bucket": "amzn-s3-demo-bucket2",
            "PercentProgress": 0,
            "KmsKeyId": "arn:aws:kms:AWS_Region:123456789012:key/2Zp9Utk/
h3yCo8nvbEXAMPLEKEY",
            "ExportTaskIdentifier": "wednesday-afternoon-test",
            "IamRoleArn": "arn:aws:iam::123456789012:role/export-to-s3",
            "TotalExtractedDataInGB": 0,
            "TaskStartTime": "2019-10-30T22:43:40.034Z",
            "SourceArn":
 "arn:aws:rds:AWS_Region:123456789012:snapshot:rds:example-1-2019-10-30-06-45"
        }
    ]
}
```

Para exibir informações sobre uma exportação de snapshot específica, inclua a opção `--export-task-identifier` com o comando `describe-export-tasks`. Para filtrar a saída, inclua a opção `--Filters`. Para obter mais opções, consulte o comando describe-export-tasks.

API do RDS

Para exibir informações sobre exportações de snapshots de banco de dados usando a API do Amazon RDS, use a operação DescribeExportTasks .

Para rastrear a conclusão do fluxo de trabalho de exportação ou acionar outro fluxo de trabalho, você pode assinar tópicos do Amazon Simple Notification Service. Para ter mais informações sobre Amazon SNS, consulte Trabalhar com a notificação de eventos do Amazon RDS.

Cancelar uma tarefa de exportação de snapshot para o Amazon RDS

Você pode cancelar uma tarefa de exportação de snapshot de banco de dados usando o AWS Management Console, a AWS CLI ou a API do RDS.

> **Note**
>
> O cancelamento de uma tarefa de exportação de snapshot não remove os dados que foram exportados para o Amazon S3. Para obter informações sobre como excluir os dados usando o console, consulte Como excluir objetos de um bucket do S3? Para excluir os dados usando a CLI, use o comando delete-object.

Console

Como cancelar uma tarefa de exportação de snapshot

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Snapshots.
3. Escolha a guia Exports in Amazon S3 (Exportações no Amazon S3).
4. Escolha a tarefa de exportação do snapshot que você deseja cancelar.
5. Escolha Cancelar.
6. Escolha Cancel export task (Cancelar tarefa de exportação) na página de confirmação.

AWS CLI

Para cancelar uma tarefa de exportação de snapshot usando a AWS CLI, use o comando cancel-export-task. O comando requer a opção `--export-task-identifier`.

Example

```
aws rds cancel-export-task --export-task-identifier my_export
{
    "Status": "CANCELING",
    "S3Prefix": "",
    "ExportTime": "2019-08-12T01:23:53.109Z",
    "S3Bucket": "amzn-s3-demo-bucket",
    "PercentProgress": 0,
    "KmsKeyId": "arn:aws:kms:AWS_Region:123456789012:key/K7MDENG/bPxRfiCYEXAMPLEKEY",
    "ExportTaskIdentifier": "my_export",
    "IamRoleArn": "arn:aws:iam::123456789012:role/export-to-s3",
    "TotalExtractedDataInGB": 0,
    "TaskStartTime": "2019-11-13T19:46:00.173Z",
    "SourceArn": "arn:aws:rds:AWS_Region:123456789012:snapshot:export-example-1"
}
```

API do RDS

Para cancelar uma tarefa de exportação de snapshot usando a API do Amazon RDS, use a operação CancelExportTask com o parâmetro `ExportTaskIdentifier`.

Mensagens de falha de tarefas de exportação do Amazon S3 para o Amazon RDS

A tabela a seguir descreve as mensagens que são retornadas quando ocorrem falhas nas tarefas de exportação do Amazon S3.

Mensagem de falha	Descrição
Ocorreu um erro interno desconhecido.	O processamento da falha devido a um erro, uma exceção ou uma falha desconhecida.
Ocorreu um erro interno desconhecido ao gravar os metadados da tarefa de exportação no bucket do S3 [nome do bucket].	O processamento da falha devido a um erro, uma exceção ou uma falha desconhecida.
A exportação do RDS falhou ao gravar os metadados da tarefa de exportação	A tarefa de exportação assume seu perfil do IAM para validar se pode gravar metadados no seu bucket do

Mensagem de falha	Descrição
porque ela não pode assumir o perfil do IAM [ARN do perfil].	S3. Se a tarefa não puder assumir seu perfil do IAM, ela falhará.
A exportação do RDS falhou ao gravar os metadados da tarefa de exportaçã o no bucket do S3 [nome do bucket] utilizando o perfil do IAM [ARN do perfil] com a chave do KMS [ID da chave]. Código de erro: [código de erro]	Uma ou mais permissões estão ausentes e, portanto, a tarefa de exportação não consegue acessar o bucket do S3. Essa mensagem de falha é gerada quando você recebe um dos seguintes códigos de erro: • `AWSSecurityTokenServiceException` com o código de erro `AccessDenied` • `AmazonS3Exception` com o código de erro `NoSuchBucket` , `AccessDenied` , `KMS.KMSInvalidStateException` , `403 Forbidden` ou `KMS.DisabledException` Esses códigos de erro indicam que as configurações estão definidas incorretamente para o perfil do IAM, o bucket do S3 ou a chave do KMS.
O perfil do IAM [ARN do perfil] não está autorizado a chamar [ação do S3] no bucket do S3 [nome do bucket]. Revise suas permissões e tente novamente a exportação.	A política do IAM está incorretamente configurada. A permissão para a ação específica do S3 no bucket do S3 está ausente, o que causa falha na tarefa de exportação.
A verificação da chave do KMS falhou. Verifique as credenciais na sua chave do KMS e tente novamente.	A verificação de credenciais da chave KMS do falhou.
Falha na verificação de credenciais do S3. Verifique as permissões no bucket do S3 e a política do IAM.	A verificação de credenciais do S3 falhou.

Mensagem de falha	Descrição
O bucket do S3 [nome do bucket] não é válido. Ele não está localizado na região da AWS atual ou não existe. Reveja o nome do bucket do S3 e tente exportar novamente.	O bucket do S3 não é válido.
O bucket do S3 [nome do bucket] não está localizado na região da AWS atual. Reveja o nome do bucket do S3 e tente exportar novamente.	O bucket da S3 está na região da AWS incorreta.

Solucionar problemas de erros de permissões do RDS para PostgreSQL

Ao exportar bancos de dados PostgreSQL para o Amazon S3, é possível ver um erro `PERMISSIONS_DO_NOT_EXIST` informando que determinadas tabelas foram ignoradas. Esse erro geralmente ocorre quando o superusuário, especificado ao criar o banco de dados, não tem permissões para acessar essas tabelas.

Para corrigir esse erro, execute o seguinte comando:

```
GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA schema_name TO superuser_name
```

Para ter mais informações sobre privilégios de superusuário, consulte Privilégios da conta de usuário mestre.

Convenções de nomenclatura de arquivos para exportações para o Amazon S3 para o Amazon RDS

Os dados exportados para tabelas específicas são armazenados no formato *base_prefix*/*files*, onde o prefixo base é o seguinte:

```
export_identifier/database_name/schema_name.table_name/
```

Por exemplo:

```
export-1234567890123-459/rdststdb/rdststdb.DataInsert_7ADB5D19965123A2/
```

Existem duas convenções de nomenclatura para arquivos.

- Convenção atual:

  ```
  batch_index/part-partition_index-random_uuid.format-based_extension
  ```

  O índice do lote é um número de sequência que representa um lote de dados lidos da tabela. Se não conseguirmos particionar a tabela em pequenos blocos para serem exportados paralelamente, haverá vários índices de lote. O mesmo acontecerá se a tabela for particionada em várias tabelas. Haverá vários índices de lote, um para cada partição da tabela principal.

  Se conseguirmos particionar a tabela em pequenos blocos para serem lidos paralelamente, haverá apenas a pasta 1 de índices de lote.

  Na pasta de índices de lote, há um ou mais arquivos Parquet que contêm os dados da sua tabela. O prefixo do nome do arquivo Parquet é `part-`*`partition_index`*. Se a tabela estiver particionada, haverá vários arquivos que começam com o índice de partição `00000`.

  Pode haver lacunas na sequência do índice de partição. Isso acontece porque cada partição é obtida de uma consulta por intervalos na tabela. Se não houver dados no intervalo dessa partição, esse número de sequência será ignorado.

  Por exemplo, suponha que a coluna `id` seja a chave primária da tabela e os valores mínimo e máximo sejam `100` e `1000`. Quando tentamos exportar essa tabela com nove partições, a lemos com consultas paralelas, como as seguintes:

  ```
  SELECT * FROM table WHERE id <= 100 AND id < 200
  SELECT * FROM table WHERE id <= 200 AND id < 300
  ```

  Isso deve gerar nove arquivos, de `part-00000-`*`random_uuid`*`.gz.parquet` a `part-00008-`*`random_uuid`*`.gz.parquet`. No entanto, se não houver linhas com IDs entre `200` e `350`, uma das partições concluídas estará vazia e nenhum arquivo será criado para ela. No exemplo anterior, `part-00001-`*`random_uuid`*`.gz.parquet` não foi criado.
- Convenção mais antiga:

```
part-partition_index-random_uuid.format-based_extension
```

É igual à convenção atual, mas sem o prefixo *batch_index*, por exemplo:

```
part-00000-c5a881bb-58ff-4ee6-1111-b41ecff340a3-c000.gz.parquet
part-00001-d7a881cc-88cc-5ab7-2222-c41ecab340a4-c000.gz.parquet
part-00002-f5a991ab-59aa-7fa6-3333-d41eccd340a7-c000.gz.parquet
```

A convenção de nomenclatura de arquivos está sujeita a alterações. Portanto, ao ler as tabelas de destino, recomendamos que você leia tudo dentro do prefixo base para a tabela.

Conversão de dados ao exportar para um bucket do Amazon S3 para RDS

Quando você exporta um snapshot de banco de dados para um bucket do Amazon S3, o Amazon RDS converte os dados para o formato Parquet e exporta e armazena os dados nesse formato. Para ter mais informações sobre o Parquet, consulte o site Apache Parquet.

O Parquet armazena todos os dados como um dos seguintes tipos primitivos:

- BOOLEAN
- INT32
- INT64
- INT96
- FLOAT
- DOUBLE
- BYTE_ARRAY: uma matriz de bytes de comprimento variável, também conhecida como binário
- FIXED_LEN_BYTE_ARRAY: uma matriz de bytes de comprimento fixo usada quando os valores têm um tamanho constante

Os tipos de dados Parquet são poucos para reduzir a complexidade de leitura e gravação do formato. O Parquet fornece tipos lógicos para estender os tipos primitivos. Um tipo lógico é implementado como uma anotação com os dados em um campo de metadados `LogicalType`. A anotação de tipo lógico explica como interpretar o tipo primitivo.

Quando o tipo lógico `STRING` anota um tipo `BYTE_ARRAY`, ele indica que a matriz de bytes deve ser interpretada como uma string de caracteres codificada em UTF-8. Depois que uma tarefa de exportação é concluída, o Amazon RDS notifica você no caso de ocorrência de uma conversão de string. Os dados subjacentes exportados são sempre os mesmos que os dados da origem. No entanto, devido à diferença de codificação em UTF-8, alguns caracteres podem aparecer diferentes da fonte quando lidos em ferramentas como a Athena.

Para ter mais informações, consulte Definições de tipos lógicos do Parquet na documentação do Parquet.

Tópicos

- Mapeamento de tipos de dados MySQL e MariaDB para o Parquet
- Mapeamento de tipo de dados PostgreSQL para Parquet

Mapeamento de tipos de dados MySQL e MariaDB para o Parquet

A tabela a seguir mostra o mapeamento dos tipos de dados MySQL e MariaDB para tipos de dados Parquet quando os dados são convertidos e exportados para o Amazon S3.

Tipo de dados de origem	Tipo primitivo do Parquet	Anotação de tipo lógico	Notas de conversão
Tipos de dados numéricos			
BIGINT	INT64		
BIGINT UNSIGNED	FIXED_LEN _BYTE_ARRAY(9)	DECIMAL(20,0)	O Parquet é compatível apenas com tipos assinados , portanto, o mapeamento requer um byte adicional (8 mais 1) para armazenar o tipo BIGINT_UNSIGNED.
BIT	BYTE_ARRAY		

Tipo de dados de origem	Tipo primitivo do Parquet	Anotação de tipo lógico	Notas de conversão
DECIMAL	INT32	DECIMAL (p,s)	Se o valor de origem for menor que 2^{31}, ele será armazenado como INT32.
	INT64	DECIMAL (p,s)	Se o valor de origem for 2^{31} ou maior, mas inferior a 2^{63}, ele será armazenado como INT64.
	FIXED_LEN _BYTE_ARRAY(N)	DECIMAL (p,s)	Se o valor de origem for 2^{63} ou superior, ele será armazenad o como FIXED_LEN _BYTE_ARRAY(N).
	BYTE_ARRAY	STRING	O Parquet não é compatível com precisão decimal maior que 38. O valor decimal é convertid o em uma string em um tipo BYTE_ARRA Y e codificado como UTF8.
DOUBLE	DOUBLE		
FLOAT	DOUBLE		
INT	INT32		
INT UNSIGNED	INT64		
MEDIUMINT	INT32		

Tipo de dados de origem	Tipo primitivo do Parquet	Anotação de tipo lógico	Notas de conversão
MEDIUMINT UNSIGNED	INT64		
NUMERIC	INT32	DECIMAL (p,s)	Se o valor de origem for menor que 2^{31}, ele será armazenado como INT32.
	INT64	DECIMAL (p,s)	Se o valor de origem for 2^{31} ou maior, mas inferior a 2^{63}, ele será armazenado como INT64.
	FIXED_LEN _ARRAY(N)	DECIMAL (p,s)	Se o valor de origem for 2^{63} ou superior, ele será armazenad o como FIXED_LEN _BYTE_ARRAY(N).
	BYTE_ARRAY	STRING	O Parquet não é compatível com precisão numérica maior que 38. Esse valor numérico é convertido em uma string em um tipo BYTE_ARRAY e codificado como UTF8.
SMALLINT	INT32		

Tipo de dados de origem	Tipo primitivo do Parquet	Anotação de tipo lógico	Notas de conversão
SMALLINT UNSIGNED	INT32		
TINYINT	INT32		
TINYINT UNSIGNED	INT32	INT(16, true)	
Tipos de dados de string			
BINARY	BYTE_ARRAY		
BLOB	BYTE_ARRAY		
CHAR	BYTE_ARRAY		
ENUM	BYTE_ARRAY	STRING	
LINESTRING	BYTE_ARRAY		
LONGBLOB	BYTE_ARRAY		
LONGTEXT	BYTE_ARRAY	STRING	
MEDIUMBLOB	BYTE_ARRAY		
MEDIUMTEXT	BYTE_ARRAY	STRING	
MULTILINESTRING	BYTE_ARRAY		
SET	BYTE_ARRAY	STRING	
TEXT	BYTE_ARRAY	STRING	
TINYBLOB	BYTE_ARRAY		
TINYTEXT	BYTE_ARRAY	STRING	
VARBINARY	BYTE_ARRAY		

Tipo de dados de origem	Tipo primitivo do Parquet	Anotação de tipo lógico	Notas de conversão
VARCHAR	BYTE_ARRAY	STRING	
Tipos de dados de data e hora			
DATE	BYTE_ARRAY	STRING	Uma data é convertida em uma string em um tipo BYTE_ARRAY e codificada como UTF8.
DATETIME	INT64	TIMESTAMP_MICROS	
TIME	BYTE_ARRAY	STRING	Um tipo TIME é convertido em uma string em um BYTE_ARRAY e codificado como UTF8.
TIMESTAMP	INT64	TIMESTAMP_MICROS	
YEAR	INT32		
Tipos de dados geométricos			
GEOMETRY	BYTE_ARRAY		
GEOMETRYCOLLECTION	BYTE_ARRAY		
MULTIPOINT	BYTE_ARRAY		
MULTIPOLYGON	BYTE_ARRAY		
POINT	BYTE_ARRAY		

Tipo de dados de origem	Tipo primitivo do Parquet	Anotação de tipo lógico	Notas de conversão
POLYGON	BYTE_ARRAY		
Tipo de dados do JSON			
JSON	BYTE_ARRAY	STRING	

Mapeamento de tipo de dados PostgreSQL para Parquet

A tabela a seguir mostra o mapeamento de tipos de dados PostgreSQL para tipos de dados Parquet quando os dados são convertidos e exportados para o Amazon S3.

Tipo de dados do PostgreSQL	Tipo primitivo do Parquet	Anotação de tipo lógico	Notas de mapeamento
Tipos de dados numéricos			
BIGINT	INT64		
BIGSERIAL	INT64		
DECIMAL	BYTE_ARRAY	STRING	Um tipo DECIMAL é convertido em uma string em um tipo BYTE_ARRAY e codificado como UTF8. Essa conversão é para evitar complicações devido à precisão dos dados e valores de dados que não são um número (NaN).
DOUBLE PRECISION	DOUBLE		

Tipo de dados do PostgreSQL	Tipo primitivo do Parquet	Anotação de tipo lógico	Notas de mapeamento
INTEGER	INT32		
MONEY	BYTE_ARRAY	STRING	
REAL	FLOAT		
SERIAL	INT32		
SMALLINT	INT32	INT(16, true)	
SMALLSERIAL	INT32	INT(16, true)	
String e tipos de dados relacionados			
ARRAY	BYTE_ARRAY	STRING	Uma matriz é convertida em uma string e codificada como BINARY (UTF8). Essa conversão é para evitar complicações devido à precisão dos dados, os valores dos dados que não são um número (NaN) e os valores de dados de tempo.
BIT	BYTE_ARRAY	STRING	
BIT VARYING	BYTE_ARRAY	STRING	
BYTEA	BINARY		
CHAR	BYTE_ARRAY	STRING	

Tipo de dados do PostgreSQL	Tipo primitivo do Parquet	Anotação de tipo lógico	Notas de mapeamento
CHAR(N)	BYTE_ARRAY	STRING	
ENUM	BYTE_ARRAY	STRING	
NAME	BYTE_ARRAY	STRING	
TEXT	BYTE_ARRAY	STRING	
TEXT SEARCH	BYTE_ARRAY	STRING	
VARCHAR(N)	BYTE_ARRAY	STRING	
XML	BYTE_ARRAY	STRING	
Tipos de dados de data e hora			
DATE	BYTE_ARRAY	STRING	
INTERVAL	BYTE_ARRAY	STRING	
TIME	BYTE_ARRAY	STRING	
TIME WITH TIME ZONE	BYTE_ARRAY	STRING	
TIMESTAMP	BYTE_ARRAY	STRING	
TIMESTAMP WITH TIME ZONE	BYTE_ARRAY	STRING	
Tipos de dados geométricos			
BOX	BYTE_ARRAY	STRING	
CIRCLE	BYTE_ARRAY	STRING	
LINE	BYTE_ARRAY	STRING	
LINESEGMENT	BYTE_ARRAY	STRING	

Tipo de dados do PostgreSQL	Tipo primitivo do Parquet	Anotação de tipo lógico	Notas de mapeamento
PATH	BYTE_ARRAY	STRING	
POINT	BYTE_ARRAY	STRING	
POLYGON	BYTE_ARRAY	STRING	
Tipos de dados JSON			
JSON	BYTE_ARRAY	STRING	
JSONB	BYTE_ARRAY	STRING	
Outros tipos de dados			
BOOLEAN	BOOLEAN		
CIDR	BYTE_ARRAY	STRING	Tipo de dados de rede
COMPOSITE	BYTE_ARRAY	STRING	
DOMAIN	BYTE_ARRAY	STRING	
INET	BYTE_ARRAY	STRING	Tipo de dados de rede
MACADDR	BYTE_ARRAY	STRING	
OBJECT IDENTIFIER	N/D		
PG_LSN	BYTE_ARRAY	STRING	
RANGE	BYTE_ARRAY	STRING	
UUID	BYTE_ARRAY	STRING	

Usar o AWS Backup no gerenciamento de backups automatizados para o Amazon RDS

O AWS Backup é um serviço de backup totalmente gerenciado que facilita a centralização e a automação do backup de dados entre todos os serviços da AWS na nuvem e no local. Você pode gerenciar backups de seus bancos de dados do Amazon RDS no AWS Backup.

> **Note**
>
> Os backups gerenciados pelo AWS Backup são considerados snapshots manuais de banco de dados, mas não são contabilizados para a cota de snapshots de banco de dados para o RDS. Backups criados com o AWS Backup têm nomes que terminam com `awsbackup:backup-job-number`.

Para ter mais informações sobre o AWS Backup, consulte o Guia do desenvolvedor do AWS Backup.

Para visualizar backups gerenciados por AWS Backup

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, selecione Snapshots.
3. Escolha a guia Backup service (Serviço de backup).

   Seus backups do AWS Backup estarão listados em Backup service snapshots (Snapshots do serviço de backup).

Métricas de monitoramento em uma instância do Amazon RDS

Nas seções a seguir, você encontra uma visão geral do monitoramento do Amazon RDS e uma explicação sobre como acessar métricas. Para saber como monitorar eventos, logs e transmissões de atividades de banco de dados, consulte Monitorar eventos, logs e transmissões em uma instância de banco de dados Amazon RDS.

Tópicos

- Plano de monitoramento
- Linha de base de performance
- Orientações de performance
- Ferramentas de monitoramento do Amazon RDS
- Visualizar o status do cluster
- Recomendações do Amazon RDS
- Visualizar métricas no console do Amazon RDS
- Visualizar métricas combinadas com o painel do Insights de Performance
- Monitorar métricas do Amazon RDS com o Amazon CloudWatch
- Monitoramento de bancos de dados do Amazon RDS com o CloudWatch Database Insights
- Monitorar a carga de banco de dados com o Performance Insights no Amazon RDS
- Analisar anomalias de performance com o DevOps Guru para Amazon RDS
- Monitorar métricas do SO com o monitoramento avançado
- Referência de métricas do Amazon RDS

Plano de monitoramento

Antes de iniciar o monitoramento Amazon RDS, crie um plano de monitoramento. Esse plano deve responder às seguintes perguntas:

- Quais são seus objetivos de monitoramento?
- Quais recursos você vai monitorar?

- Com que frequência você vai monitorar esses recursos?
- Quais ferramentas de monitoramento você usará?
- Quem realizará o monitoramento das tarefas?
- Quem deve ser notificado quando algo der errado?

Linha de base de performance

Para atingir suas metas de monitoramento, é necessário estabelecer uma linha de base. Para fazer isso, meça a performance em diferentes condições de carga em vários momentos em seu ambiente do Amazon RDS. É possível monitorar métricas como as seguintes:

- Taxa de transferência na rede
- Conexões de cliente
- E/S para operações de leitura, gravação ou metadados
- Saldos de crédito intermitentes para suas instâncias de banco de dados

Recomendamos armazenar dados históricos de performance para o Amazon RDS. Usando os dados armazenados, é possível comparar a performance atual com as tendências anteriores. Também é possível distinguir padrões de performance normais de anomalias e criar técnicas para resolver problemas.

Orientações de performance

Em geral, os valores aceitáveis para as métricas de performance dependem do que a aplicação está fazendo em relação à sua linha de base. Investigue variações consistentes ou tendenciais de sua linha de base. Muitas vezes, as métricas a seguir são a origem dos problemas de performance:

- Alto consumo de CPU ou RAM – valores altos para o consumo de CPU ou RAM podem ser adequados, se estiverem de acordo com seus objetivos em relação ao aplicativo (como taxa de transferência ou concorrência).
- Consumo de espaço em disco – inspecione o consumo de espaço em disco caso o espaço usado seja consistentemente igual ou superior a 85% do espaço total no disco. Veja se é possível excluir dados da instância ou arquivar dados em um sistema diferente para liberar mais espaço.
- Tráfego de rede – em relação ao tráfego de rede, fale com o administrador do sistema para entender qual taxa de transferência é esperada para sua rede de domínio e conexão com a

Internet. Inspecione o tráfego de rede caso a taxa de transferência seja consistentemente menor do que a esperada.

- Conexões do banco de dados: se você observar números elevados de conexões de usuários e também diminuições na performance da instância e no tempo de resposta, considere restringir as conexões do banco de dados. O melhor número de conexões de usuários para sua instância de banco de dados varia conforme a classe da instância e a complexidade das operações em execução. Para determinar o número de conexões de banco de dados, associe sua instância de banco de dados a um grupo de parâmetros cujo parâmetro `User Connections` esteja definido como um valor diferente de 0 (ilimitado). Você pode usar um parameter group existente ou criar um novo. Para obter mais informações, consulte Grupos de parâmetros para Amazon RDS.
- Métricas de IOPS – os valores esperados para as métricas de IOPS dependem da especificação do disco e da configuração do servidor, por isso, use sua linha de base para saber os valores típicos. Inspecione caso os valores sejam consistentemente diferentes da sua linha de base. Para obter a melhor performance de IOPS, confira se o seu conjunto de trabalho típico se adéqua à memória para minimizar as operações de leitura e gravação.

Quando a performance estiver fora da linha de base estabelecida, talvez seja necessário fazer alterações para otimizar a disponibilidade do banco de dados para sua workload. Por exemplo, talvez você precise alterar a classe de sua instância de banco de dados. Ou talvez seja necessário alterar o número de instâncias de banco de dados e réplicas de leitura disponíveis para clientes.

Ferramentas de monitoramento do Amazon RDS

O monitoramento é uma parte importante para manter a confiabilidade, a disponibilidade e a performance do Amazon RDS e suas outras soluções da AWS. A AWS fornece várias ferramentas de monitoramento para observar o Amazon RDS, gerar relatórios quando algo estiver errado e executar ações automáticas quando for apropriado.

Tópicos

- Ferramentas de monitoramento automatizadas
- Ferramentas de monitoramento manual

Ferramentas de monitoramento automatizadas

Recomendamos que as tarefas de monitoramento sejam automatizadas ao máximo possível.

Tópicos

- Status e recomendações de instância do Amazon RDS
- Métricas do Amazon CloudWatch para Amazon RDS
- Insights de Performance do Amazon RDS e monitoramento do sistema operacional
- Serviços integrados

Status e recomendações de instância do Amazon RDS

É possível usar as seguintes ferramentas automatizadas para supervisionar o Amazon RDS e gerar relatórios quando algo estiver errado:

- Status da instância do Amazon RDS: exibe detalhes sobre o status atual da instância usando o console do Amazon RDS, a AWS CLI ou a API do RDS.
- Amazon RDS recomendações — responda a recomendações automatizadas para recursos de banco de dados, como instâncias de banco de dados, réplicas de leitura e grupo de parâmetros de banco de dados. Para ter mais informações, consulte Recomendações do Amazon RDS.

Métricas do Amazon CloudWatch para Amazon RDS

O Amazon RDS integra-se ao Amazon CloudWatch para oferecer recursos adicionais de monitoramento.

- Amazon CloudWatch: esse serviço monitora seus recursos da AWS e as e aplicações executadas na AWS em tempo real. É possível usar os seguintes recursos do Amazon CloudWatch com o Amazon RDS:
  - Métricas do Amazon CloudWatch: o Amazon RDS envia métricas automaticamente para o CloudWatch a cada minuto, para cada banco de dados ativo. Você não recebe cobranças adicionais para métricas do Amazon RDS no CloudWatch. Para obter mais informações, consulte Monitorar métricas do Amazon RDS com o Amazon CloudWatch.
  - Alarmes do Amazon CloudWatch– Você pode assistir a uma única métrica do Amazon RDS em um período de tempo específico. Depois, você pode realizar uma ou mais ações com base no valor da métrica em relação a um limite definido. Para obter mais informações, consulte Monitorar métricas do Amazon RDS com o Amazon CloudWatch.

Insights de Performance do Amazon RDS e monitoramento do sistema operacional

É possível usar as seguintes ferramentas automatizadas para monitorar a performance do Amazon RDS:

- Amazon RDS Performance Insights: avalie a carga no banco de dados e determine quando e onde tomar medidas. Para ter mais informações, consulte Monitorar a carga de banco de dados com o Performance Insights no Amazon RDS.
- Monitoramento avançado do Amazon RDS: observe as métricas em tempo real para o sistema operacional. Para ter mais informações, consulte Monitorar métricas do SO com o monitoramento avançado.

Serviços integrados

Os seguintes serviços da AWS estão integrados ao Amazon RDS:

- O Amazon EventBridge é um serviço de barramento de eventos sem servidor que facilita a conexão de aplicações a dados de diversas origens. Para ter mais informações, consulte Monitorar eventos do Amazon RDS.
- O Amazon CloudWatch Logs permite monitorar, armazenar e acessar os arquivos de log de instâncias do Amazon RDS, do CloudTrail e de outras fontes. Para ter mais informações, consulte Monitorar arquivos de log do Amazon RDS.
- O AWS CloudTrail captura chamadas de API e eventos relacionados feitos por sua conta da Conta da AWS ou em nome dela e entrega os arquivos de log a um bucket do Amazon S3 que você

especificar. Para ter mais informações, consulte Monitorar chamadas de API do Amazon RDSno AWS CloudTrail.

- O Database Activity Streams é um recurso do Amazon RDS que fornece uma transmissão quase em tempo real da atividade em sua instância de banco de dados Oracle. Para ter mais informações, consulte Monitorar o Amazon RDS com o recurso Database Activity Streams.

Ferramentas de monitoramento manual

É necessário monitorar manualmente os itens que os alarmes do CloudWatch não cobrem. Os painéis do console do Amazon RDS, do CloudWatch do AWS Trusted Advisor e outros painéis do console AWS apresentam uma visão rápida do estado do ambiente da AWS. Recomendamos que você também verifique os arquivos de log de sua instância de banco de dados.

- Na console do Amazon RDS, é possível monitorar os seguintes itens de seus recursos:
  - O número de conexões a uma instância de banco de dados
  - A quantidade de operações de leitura e gravação em uma instância de banco de dados
  - A quantidade de armazenamento que uma instância de banco de dados está utilizando
  - A quantidade de memória e CPU em uso para uma instância de banco de dados
  - A quantidade de tráfego de rede de e para uma instância de banco de dados
- No painel do Trusted Advisor, você pode rever as seguintes verificações de otimização de custos, segurança, tolerância a falhas e melhoria de desempenho:
  - Amazon RDS Idle DB Instances
  - Amazon RDS Security Group Access Risk
  - Amazon RDS Backups
  - Amazon RDS Multi-AZ

  Para obter mais informações sobre essas verificações, consulte Melhores práticas do Trusted Advisor (verificações).
- A página inicial do CloudWatch mostra:
  - Alertas e status atual
  - Gráficos de alertas e recursos
  - Estado de integridade do serviço

  Além disso, é possível usar o CloudWatch para fazer o seguinte:

- Criar painéis personalizados para monitorar os serviços de seu interesse.
- Colocar em gráfico dados de métrica para solucionar problemas e descobrir tendências.
- Pesquise e procure todas as métricas de recursos da AWS.
- Criar e editar alertas para ser notificado sobre problemas.

Visualizar o status do cluster

Usando o console do Amazon RDS, é possível acessar rapidamente o status da instância do banco de dados.

Tópicos

- Visualizar o status de uma instância de banco de dados Amazon RDS em um

Visualizar o status de uma instância de banco de dados Amazon RDS em um

O status de uma instância de banco de dados indica a integridade da instância de banco de dados. É possível usar os procedimentos a seguir para visualizar o status de uma instância de banco de dados no console do Amazon RDS, o comando AWS CLI ou a operação de API.

> **Note**
>
> O Amazon RDS também usa outro status chamado status de manutenção, que é mostrado na coluna Maintenance (Manutenção) do console do Amazon RDS. Este valor indica o status de todos os patches de manutenção que precisarem ser aplicados a uma instância de banco de dados. O status de manutenção é independente do status da instância do banco de dados. Para ter mais informações sobre o status de manutenção, consulte Aplicar atualizações para uma instância de banco de dados.

Encontre os valores possíveis do status de instâncias de banco de dados na tabela a seguir. Essa tabela mostra se você será cobrado pela instância de banco de dados e pelo armazenamento, apenas pelo armazenamento ou se não será cobrado. Para todos os status de instância de banco de dados, você sempre será cobrado pelo uso de backup.

Status da instância de banco de dados	Faturado	Descrição
available	Faturado	A instância de banco de dados é íntegra e está disponível.
backing-up	Faturado	No momento, está sendo feito o backup da instância de banco de dados.
configuring-enhanced-monitoring	Faturado	O monitoramento avançado está sendo habilitado ou desabilitado para essa instância de banco de dados.
configuring-iam-database-auth	Faturado	A autenticação do banco de dados do AWS Identity and Access Management (IAM) está sendo habilitada ou desabilitada para essa instância de banco de dados.

Status da instância de banco de dados	Faturado	Descrição
configuring-log-exports	Faturado	A publicação dos arquivos de log no Amazon CloudWatch Logs está sendo habilitada ou desabilitada para essa instância de banco de dados.
converting-to-vpc	Faturado	A instância de banco de dados está sendo convertida de uma instância de banco de dados que não esteja em uma Amazon Virtual Private Cloud (Amazon VPC) em uma instância de banco de dados que esteja em uma Amazon VPC.
creating	Não faturado (sem PITR) Faturado (somente PITR)	A instância de banco de dados está sendo criada. A instância de banco de dados permanece inacessível enquanto é criada. Se você restaurar um banco de dados durante a recuperação para um ponto no tempo (PITR), a cobrança será feita enquanto o banco de dados estiver no estado criando. Esse é o único cenário em que o estado criando incorre em cobranças.
delete-precheck	Não faturado	O Amazon RDS está validando que as réplicas de leitura estão íntegras e podem ser excluídas com segurança.
deleting	Não faturado	A instância de banco de dados está sendo excluída.
com falha	Não faturado	A instância de banco de dados falhou e o Amazon RDS não pode recuperá-la. Execute uma restauração point-in-time no último momento restaurável da instância de banco de dados para recuperar os dados.
inaccessible-encryption-credentials	Não faturado	A AWS KMS key usada para criptografar ou descriptografar a instância de banco de dados não pode ser acessada nem recuperada.

Status da instância de banco de dados	Faturado	Descrição
inaccessible-encryption-credentials-recoverable	Faturado para armazenamento	A chave do KMS usada para criptografar ou descriptografar a instância de banco de dados não pode ser acessada. No entanto, se a chave do KMS estiver ativa, ela pode ser recuperada reiniciando a instância de banco de dados. Para obter mais informações, consulte Criptografar uma instância de banco de dados.
incompatible-create	Não faturado	O Amazon RDS está tentando criar uma instância de banco de dados, mas não pode fazer isso porque os recursos são incompatíveis com sua instância de banco de dados. Esse status poderá ocorrer se, por exemplo, o perfil de instância do seu banco de dados não tiver as permissões corretas.
incompatible-network	Não faturado	O Amazon RDS está tentando realizar uma ação de recuperação em uma instância de banco de dados, mas não pode fazer isso porque a VPC está em um estado que impede a conclusão da ação. Este status pode ocorrer se, por exemplo, todos os endereços IP disponíveis em uma sub-rede estiverem em uso e o Amazon RDS não puder obter um endereço IP para a instância de banco de dados.
incompatible-option-group	Faturado	O Amazon RDS tentou aplicar uma alteração de grupo de opções, mas não pôde fazê-lo. O Amazon RDS não poderá reverter para o estado anterior do grupo de opções. Para ter mais informações, confira a lista Recent Events (Eventos recentes) da instância de banco de dados. Este status pode ocorrer se, por exemplo, o grupo de opções contém uma opção como TDE e a instância de banco de dados não contém informações criptografadas.

Status da instância de banco de dados	Faturado	Descrição
incompatible-parameters	Faturado	O Amazon RDS não pode iniciar a instância de banco de dados porque os parâmetros especificados no parameter group de banco de dados da instância de banco de dados não são compatíveis com a instância. Reverta as alterações de parâmetro ou as torne compatíveis com a instância de banco de dados para retomar o acesso à instância de banco de dados. Para ter mais informações sobre parâmetros incompatíveis, confira a lista Recent Events (Eventos recentes) da instância de banco de dados.
incompatible-restore	Não faturado	O Amazon RDS não pode fazer uma restauração point-in-time. As causas comuns para esse status incluem o uso de tabelas temporárias, MyISAM com MySQL ou tabelas Aria com MariaDB.
insuficiência de capacidade	Não faturado	O Amazon RDS não consegue criar a instância porque não há capacidade suficiente disponível no momento. Para criar sua instância de banco de dados na mesma AZ com o mesmo tipo de instância, exclua a instância de banco de dados, aguarde algumas horas e tente criá-la novamente. Como alternativa, crie uma nova instância usando uma classe de instância diferente ou AZ.
manutenção	Faturado	O Amazon RDS está aplicando uma atualização de manutenção na instância de banco de dados. Este status é usado para a manutenção de nível de instância que o RDS agenda com antecedência.
modifying	Faturado	A instância de banco de dados está sendo modificada por causa de uma solicitação do cliente.

Status da instância de banco de dados	Faturado	Descrição
moving-to-vpc	Faturado	A instância de banco de dados está sendo movida para uma nova Amazon Virtual Private Cloud (Amazon VPC).
reinicialização	Faturado	A instância de banco de dados está sendo reinicial izada por causa de uma solicitação do cliente ou de um processo do Amazon RDS que exige a reinicial ização da instância.
resetting-master-c redentials	Faturado	As credenciais principais da instância de banco de dados estão sendo redefinidas por causa de uma solicitação do cliente.
renaming	Faturado	A instância de banco de dados está sendo renomeada por causa de uma solicitação do cliente.
restore-error	Faturado	A instância de banco de dados encontrou um erro ao tentar restaurar para um determinado point-in-time ou de um snapshot.
starting	Faturado para armazenam ento	A instância do banco de dados está iniciando.
interrompido	Faturado para armazenam ento	A instância do banco de dados está interrompida.
interromper	Faturado para armazenam ento	A instância do banco de dados está sendo interromp ida.

Status da instância de banco de dados	Faturado	Descrição
storage-config-upgrade	Faturado	A configuração do sistema de arquivos de armazenamento da instância de banco de dados está sendo atualizada. Esse status só se aplica a bancos de dados verdes em uma implantação azul/verde ou a réplicas de leitura de instâncias de banco de dados.
storage-full	Faturado	A instância de banco de dados alcançou a alocação da capacidade de armazenamento. Esse é um status crítico e recomendamos que você corrija esse problema imediatamente. Para fazer isso, aumente seu armazenamento modificando a instância de banco de dados. Para evitar essa situação, configure os alarmes do Amazon CloudWatch para adverti-lo quando o espaço de armazenamento estiver ficando baixo.
storage-initialization	Faturado	A instância de banco de dados está carregando blocos de dados do Amazon S3 para otimizar o desempenho do volume após a restauração de um snapshot. Ela permanece disponível para operações, mas o desempenho pode não estar na capacidade máxima até que a inicialização seja concluída.
storage-optimization	Faturado	O Amazon RDS está otimizando o armazenamento de sua instância de banco de dados. O processo de otimização de armazenamento geralmente é curto, mas, às vezes, pode levar até 24 horas ou mais. Durante a otimização do armazenamento, a instância de banco de dados permanece disponível. A otimização do armazenamento é um processo em segundo plano que não afeta a disponibilidade da instância.

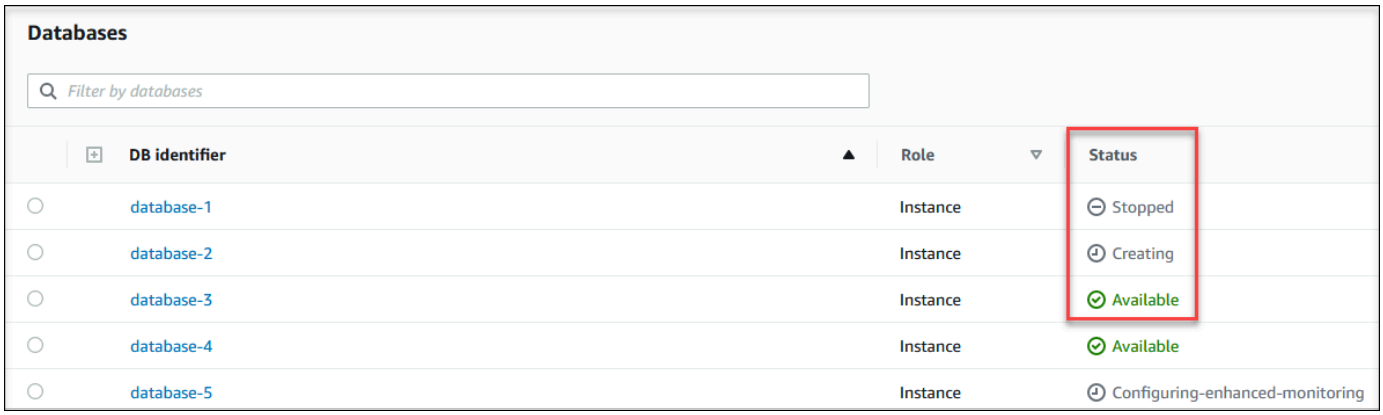
Status da instância de banco de dados	Faturado	Descrição
upgrading	Faturado	A versão do mecanismo do banco de dados ou sistema operacional está sendo atualizada.

Console

Como visualizar o status de uma instância de banco de dados

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Bancos de dados.

   A página Databases (Bancos de dados) aparece com a lista de instâncias de banco de dados. Para cada instância de banco de dados , o valor do status é exibido.



CLI

Para visualizar a instância de banco de dados e suas informações de status usando a AWS CLI, utilize o comando describe-db-instances. Por exemplo, o comando AWS CLI a seguir lista todas as informações de instâncias de banco de dados.

```
aws rds describe-db-instances
```

Para visualizar uma instância de banco de dados específica e seu status, chame o comando describe-db-instances com a seguinte opção:

- `DBInstanceIdentifier`: o nome da instância de banco de dados.

```
aws rds describe-db-instances --db-instance-identifier mydbinstance
```

Para visualizar apenas o status de todas as instâncias de banco de dados, utilize a consulta a seguir na AWS CLI.

```
aws rds describe-db-instances --query 'DBInstances[*].
[DBInstanceIdentifier,DBInstanceStatus]' --output table
```

API

Para visualizar o status da instância de banco de dados usando a API do Amazon RDS, chame a operação DescribeDBInstances.

Recomendações do Amazon RDS

O Amazon RDS fornece recomendações automatizadas para recursos de banco de dados, como instâncias de banco de dados, réplicas de leitura e grupo de parâmetros de cluster de banco de dados. Essas recomendações fornecem orientações de práticas recomendadas, analisando a configuração, o uso e os dados de performance da instância de banco de dados.

O Insights de Performance do Amazon RDS monitora métricas específicas e cria limites automaticamente analisando quais níveis são considerados possivelmente problemáticos para um recurso específico. Quando novos valores de métricas ultrapassam um limite predefinido em um período específico, o Performance Insights gera uma recomendação proativa. Essa recomendação ajuda a evitar um impacto futuro na performance do banco de dados. Por exemplo, a recomendação “Ocioso na transação” é gerada para instâncias do RDS para PostgreSQL quando as sessões conectadas ao banco de dados não estão realizando trabalho ativo, mas podem manter os recursos do banco de dados bloqueados. Para receber recomendações proativas, é necessário ativar o Performance Insights com um período de retenção de nível pago. Para ter informações sobre como ativar o Performance Insights, consulte Ativar e desativar o Insights de Performance do Amazon RDS. Para ter informações sobre preços e retenção de dados para o Performance Insights, consulte Preços e retenção de dados para o Performance Insights.

O DevOps Guru para RDS monitora determinadas métricas para detectar quando o comportamento de uma métrica se torna altamente incomum ou anômalo. Essas anomalias são relatadas como insights reativos com recomendações. Por exemplo, o DevOps Guru para RDS pode recomendar que você aumente a capacidade da CPU ou investigue eventos de espera que estão contribuindo para a carga do banco de dados. O DevOps Guru para RDS também fornece recomendações proativas baseadas em limites. Para essas recomendações, é necessário ativar o DevOps Guru para RDS. Para ter informações sobre como ativar o DevOps Guru para RDS, consulte Ativar o DevOps Guru e especificar a cobertura de recursos.

As recomendações estarão em qualquer um dos seguintes status: ativas, rejeitadas, pendentes ou resolvidas. As recomendações resolvidas ficam disponíveis por 365 dias.

É possível ver ou ignorar as recomendações. É possível aplicar uma recomendação ativa baseada em configuração imediatamente, programá-la para a próxima janela de manutenção ou descartá-la. Para recomendações proativas baseadas em limites e baseadas em machine learning, é necessário analisar a causa sugerida do problema e, depois, realizar as ações recomendadas para corrigir o problema.

As recomendações são aceitas nas seguintes Regiões da AWS:

- Leste dos EUA (Ohio)
- Leste dos EUA (Norte da Virgínia)
- Oeste dos EUA (Norte da Califórnia)
- Oeste dos EUA (Oregon)
- Ásia-Pacífico (Mumbai)
- Ásia-Pacífico (Seul)
- Ásia-Pacífico (Singapura)
- Ásia-Pacífico (Sydney)
- Ásia-Pacífico (Tóquio)
- Canadá (Central)
- Europa (Frankfurt)
- Europa (Irlanda)
- Europa (Londres)
- Europa (Paris)
- Europa (Estocolmo)
- América do Sul (São Paulo)

Aprenda a visualizar, aplicar, ignorar e modificar recomendações do Amazon RDS nas seções a seguir.

Tópicos

- Visualizar as recomendações Amazon RDS
- Aplicar as recomendações do Amazon RDS
- Ignorar as recomendações do Amazon RDS
- Modificar as recomendações ignoradas do Amazon RDS para recomendações ativas
- Recomendações da referência do Amazon RDS

Visualizar as recomendações Amazon RDS

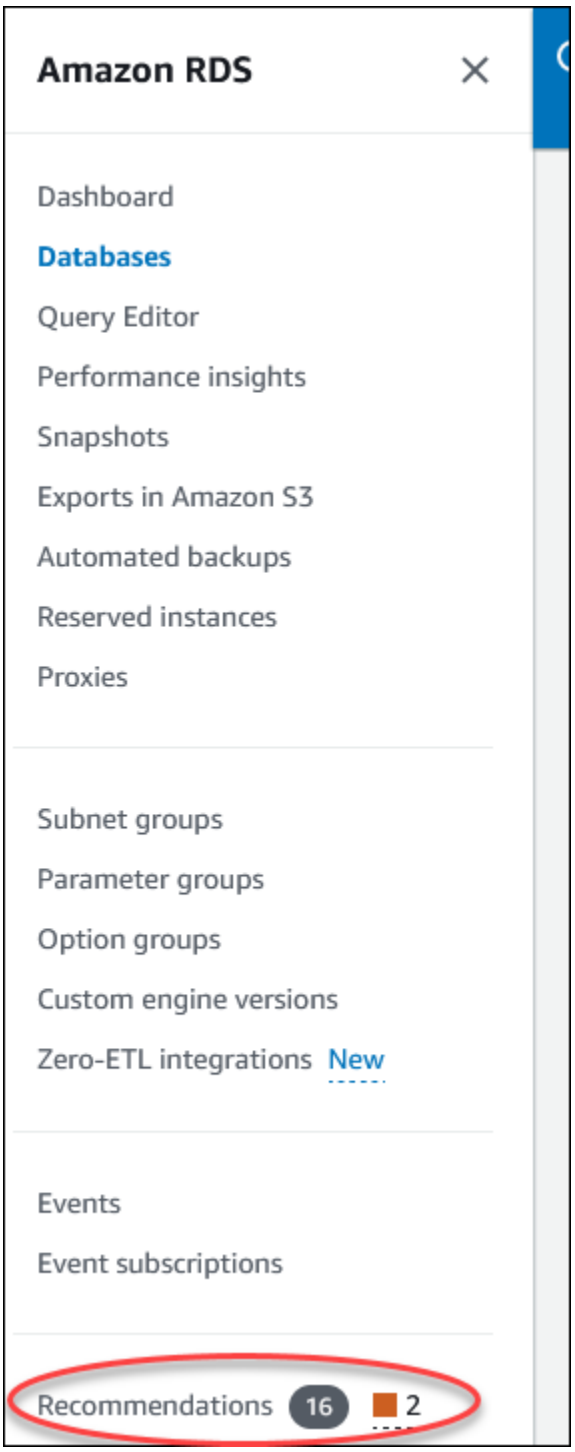
Usando o console do Amazon RDS, é possível visualizar as recomendações do Amazon RDS para os recursos de banco de dados.

Console

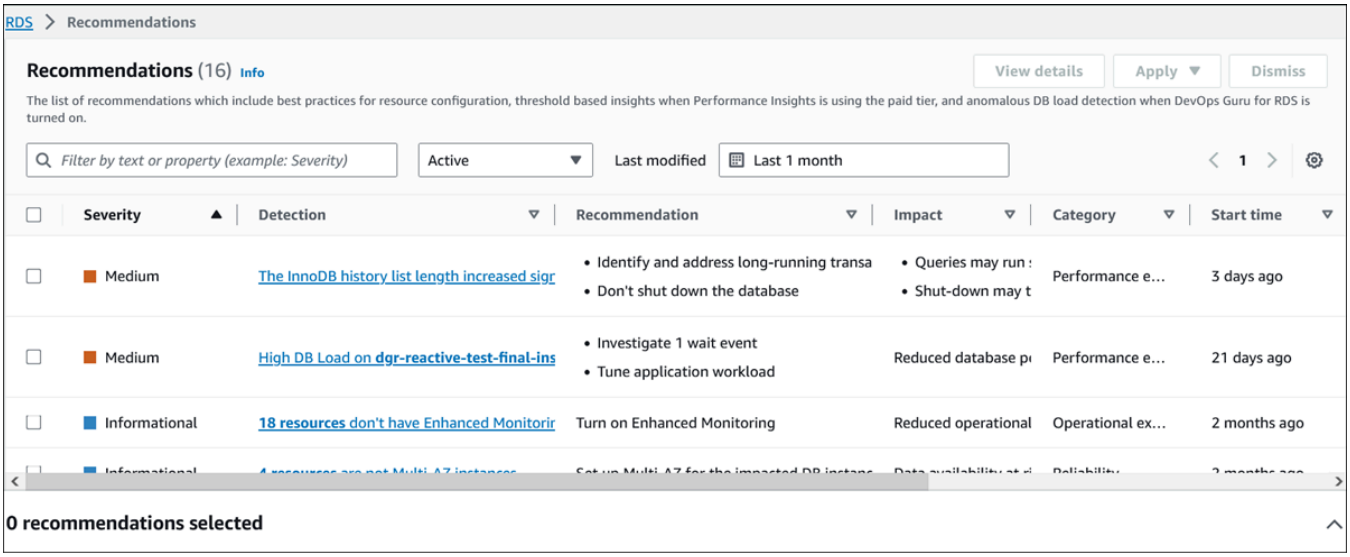
Como visualizar recomendações do Amazon RDS

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, execute qualquer uma das seguintes opções:

   - Selecione Recomendações. O número de recomendações ativas para os recursos e o número de recomendações com maior gravidade geradas no último mês estão disponíveis ao lado de Recomendações. Para encontrar o número de recomendações ativas para cada gravidade, selecione o número que mostra a maior gravidade.

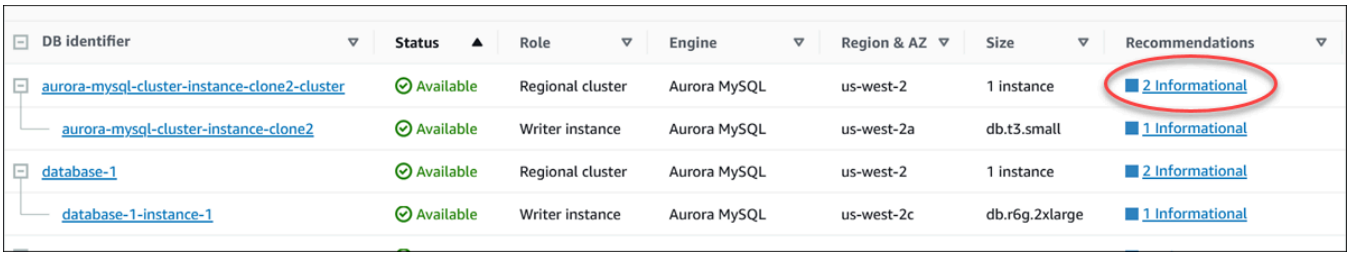


Por padrão, a página Recomendações exibe uma lista de novas recomendações no último mês. O Amazon RDS fornece recomendações para todos os recursos na conta e as classifica de acordo com a gravidade.

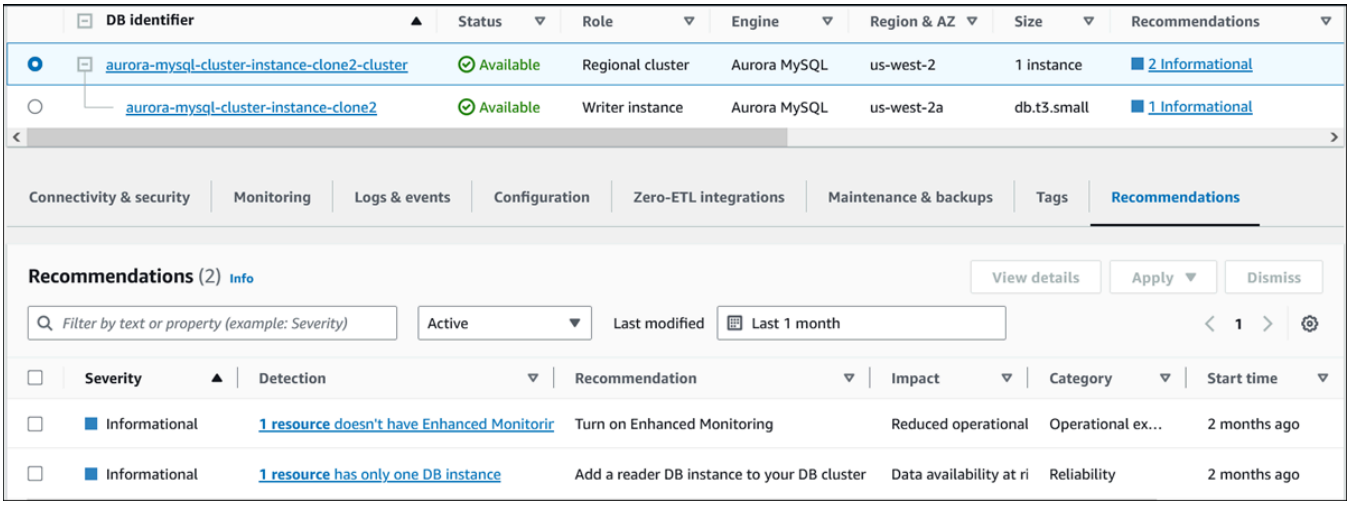


É possível selecionar uma recomendação para ver uma seção na parte inferior da página que contém os recursos afetados e detalhes de como a recomendação será aplicada.

- Na página Bancos de dados, selecione Recomendações para um recurso.



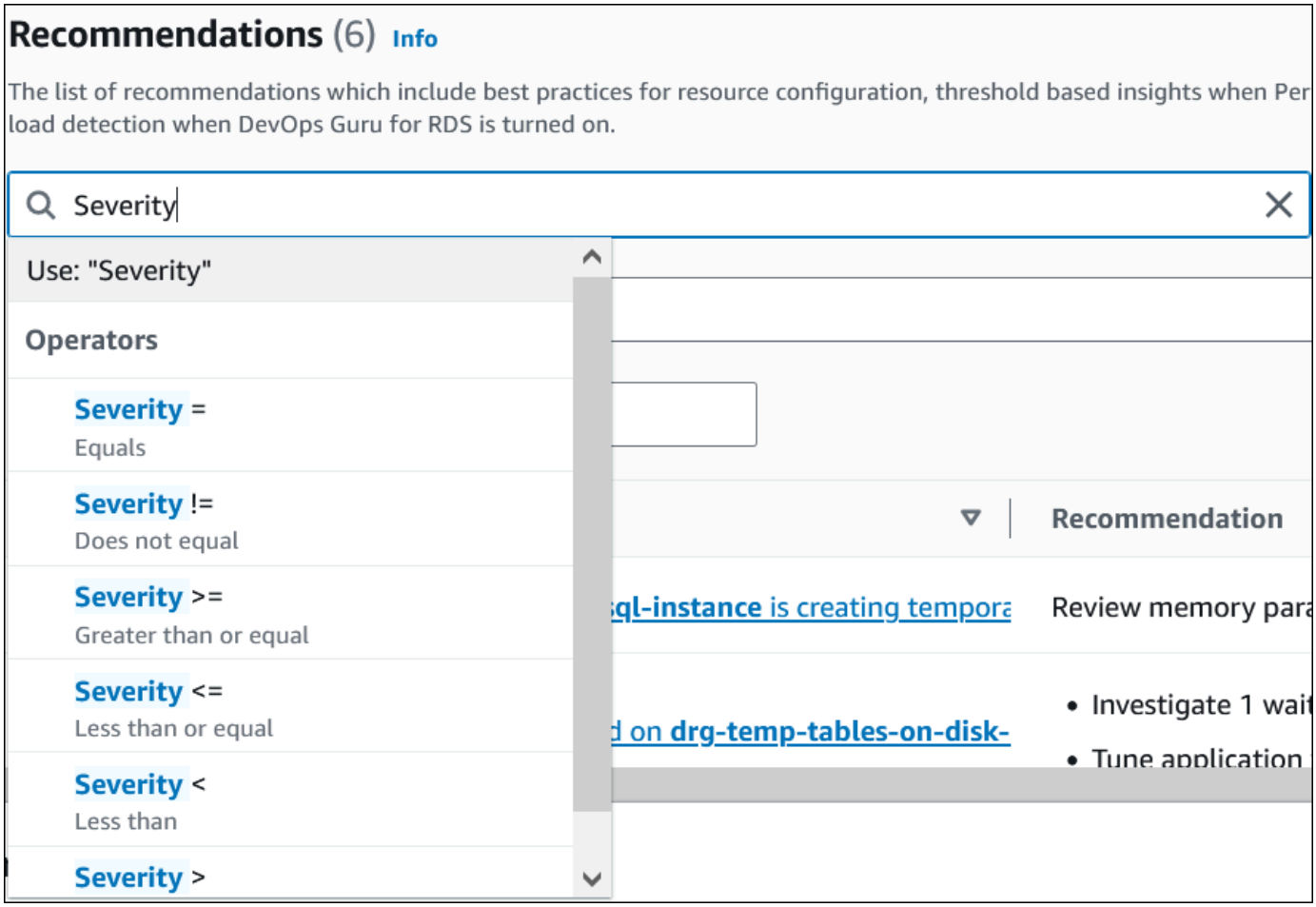
A guia Recomendações exibe as recomendações e os detalhes do recurso selecionado.



Os detalhes a seguir estão disponíveis para as recomendações:

- Gravidade: o nível de implicação do problema. Os níveis de gravidade são Alto, Médio, Baixo e Informativo.
- Detecção: o número de recursos afetados e uma breve descrição do problema. Selecione este link para ver a recomendação e os detalhes da análise.
- Recomendação: uma breve descrição da ação recomendada a ser aplicada.
- Impacto: uma breve descrição do possível impacto quando a recomendação não é aplicada.
- Categoria: o tipo de recomendação. As categorias são Eficiência de performance, Segurança, Confiabilidade, Otimização de custos, Excelência operacional e Sustentabilidade.
- Status: o status atual da recomendação. Os status possíveis são Todos, Ativo, Dispensado, Resolvido e Pendente.
- Horário de início: a hora em que o problema começou. Por exemplo, 18 horas atrás.
- Última modificação: a hora em que a recomendação foi atualizada pela última vez pelo sistema devido a uma alteração na Gravidade ou a hora em que você respondeu à recomendação. Por exemplo, 10 horas atrás.
- Horário de término: a hora em que o problema terminou. A hora não será exibida para nenhum problema contínuo.
- Identificador do recurso: o nome de um ou mais recursos.

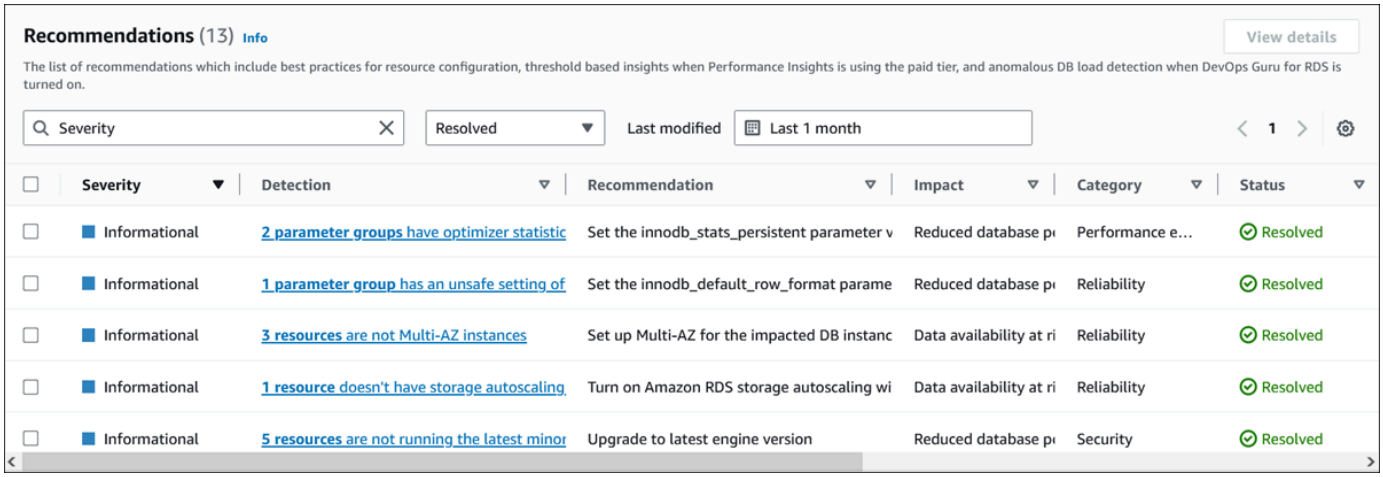
3. (Opcional) Selecione os operadores de Gravidade ou Categoria no campo para filtrar a lista de recomendações.



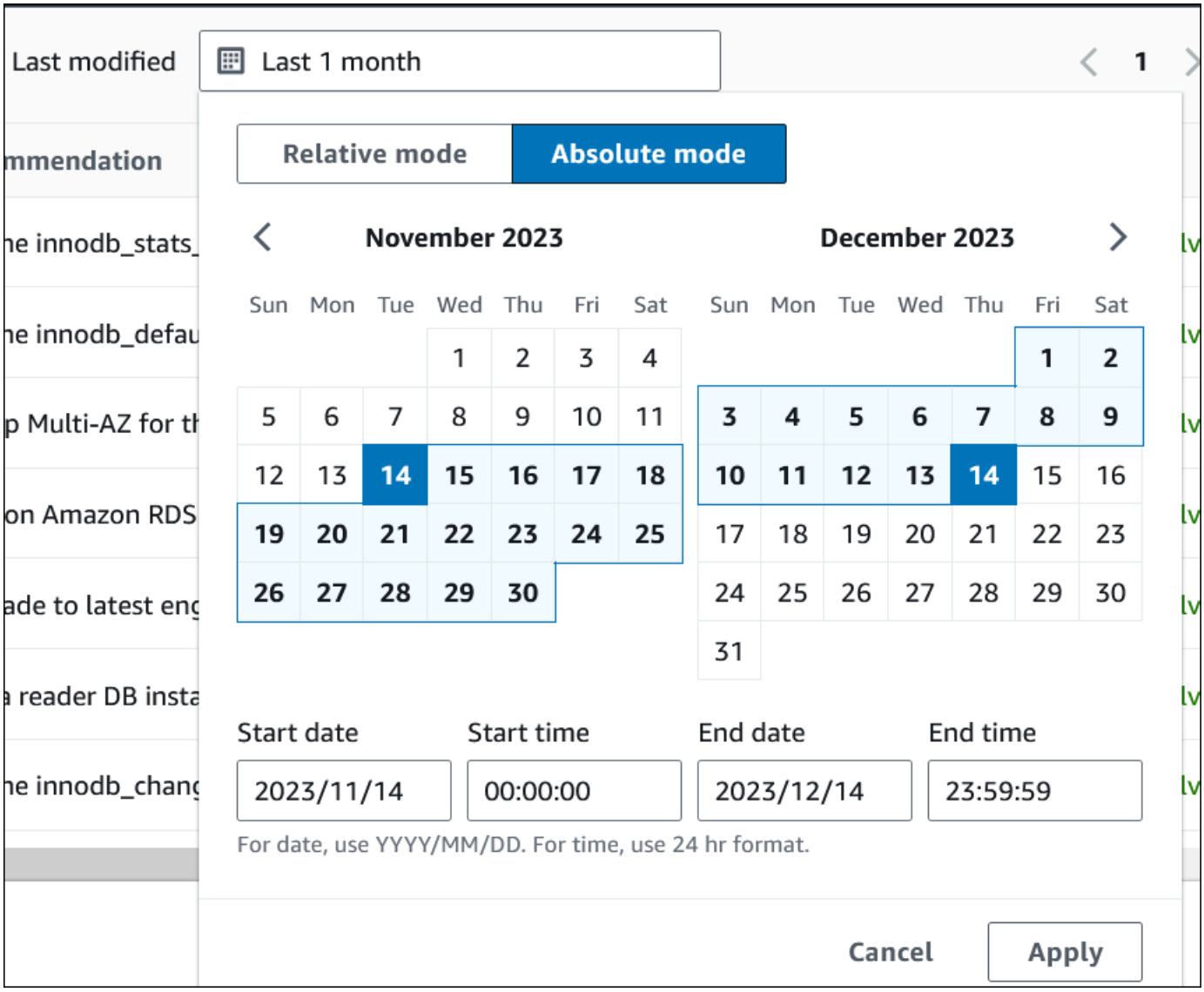
As recomendações para a operação selecionada são exibidas.

4. (Opcional) Selecione qualquer um dos seguintes status de recomendação:

   - Ativo (padrão): mostra as recomendações atuais que você pode aplicar, programar para a próxima janela de manutenção ou dispensar.
   - Todos: mostra todas as recomendações com o status atual.
   - Dispensado: mostra as recomendações dispensadas.
   - Resolvido: mostra as recomendações resolvidas.
   - Pendente: mostra as recomendações cujas ações recomendadas estão em andamento ou programadas para a próxima janela de manutenção.



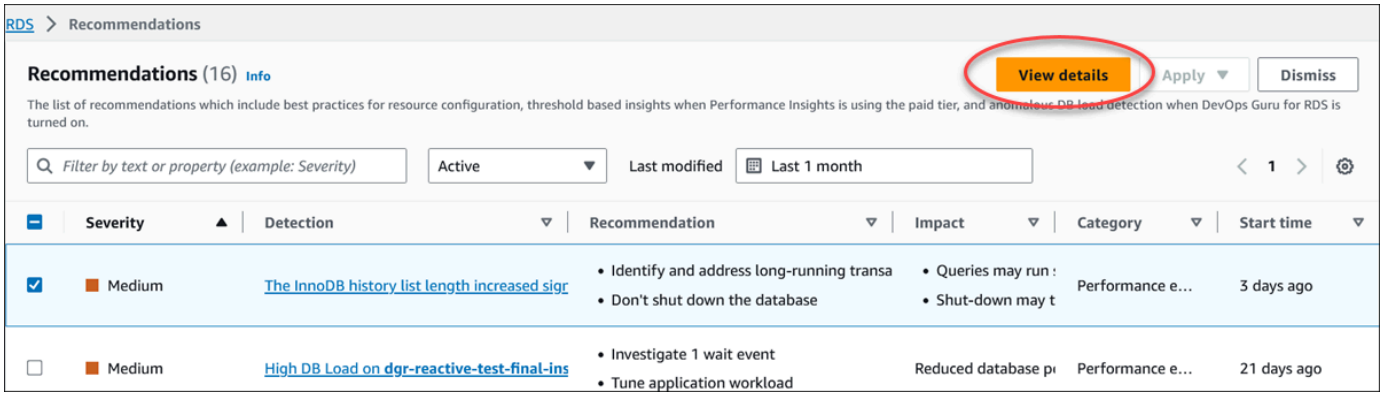
5. (Opcional) Selecione Modo relativo ou Modo absoluto em Última modificação para modificar o período. A página Recomendações exibe as recomendações geradas no período. O período padrão é o último mês. No Modo absoluto, é possível escolher o período ou inserir a hora nos campos Data de início e Data de término.



As recomendações para o período definido são exibidas.

Observe que é possível ver todas as recomendações de recursos na conta definindo o intervalo como Todos.

6. (Opcional) Selecione Preferências à direita para personalizar os detalhes a serem exibidos. É possível escolher um tamanho de página, quebrar as linhas do texto e exibir ou ocultar as colunas.
7. (Opcional) Selecione uma recomendação e, depois, escolha Visualizar detalhes.



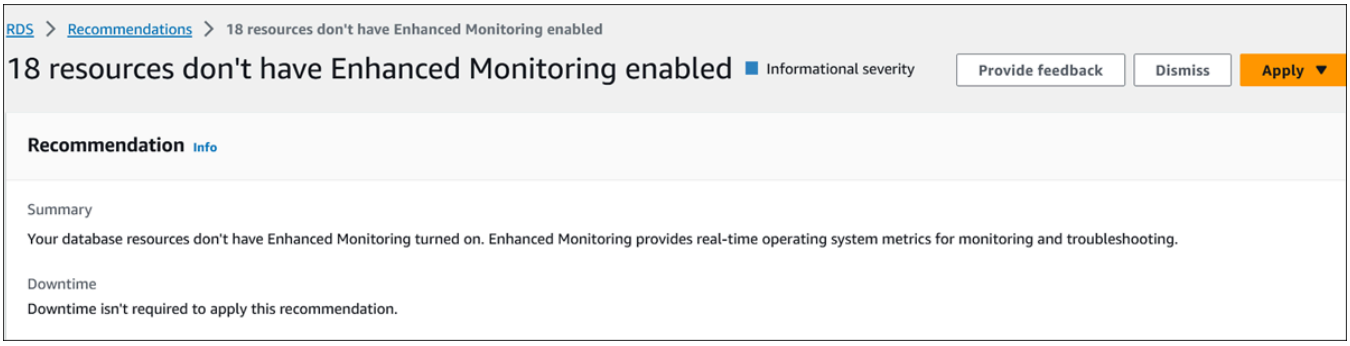
A página de detalhes da recomendação é exibida. O título fornece a contagem total dos recursos com o problema detectado e a gravidade.

Para ter informações sobre os componentes na página de detalhes de uma recomendação reativa baseada em anomalias, consulte Viewing reactive anomalies no Guia do usuário do Amazon DevOps Guru.

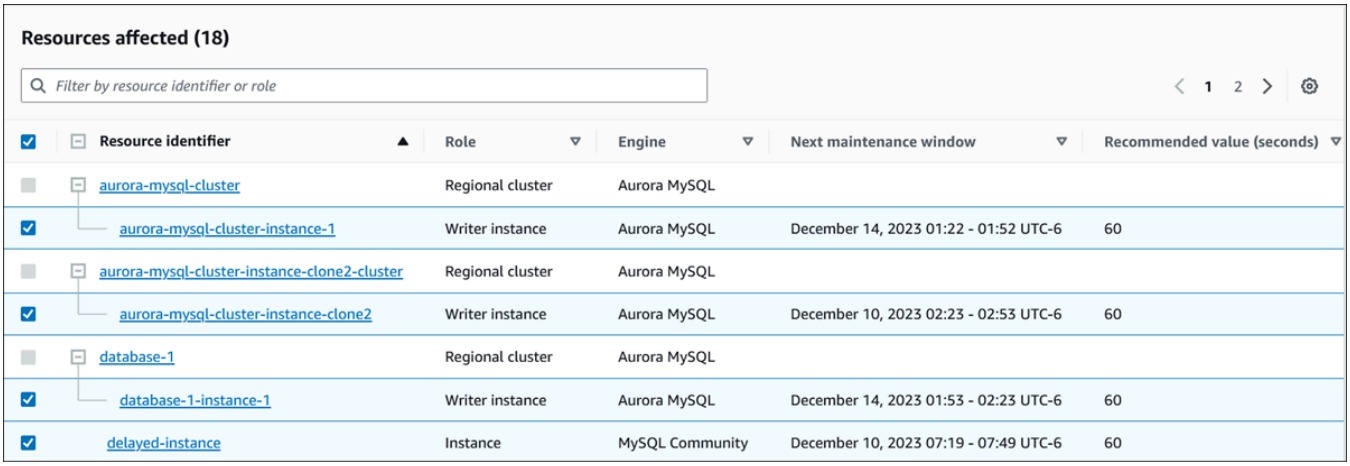
Para ter informações sobre os componentes na página de detalhes de uma recomendação proativa baseada em limites, consulte Visualizar as recomendações proativas do Performance Insights.

As outras recomendações automatizadas exibem os seguintes componentes na página de detalhes da recomendação:

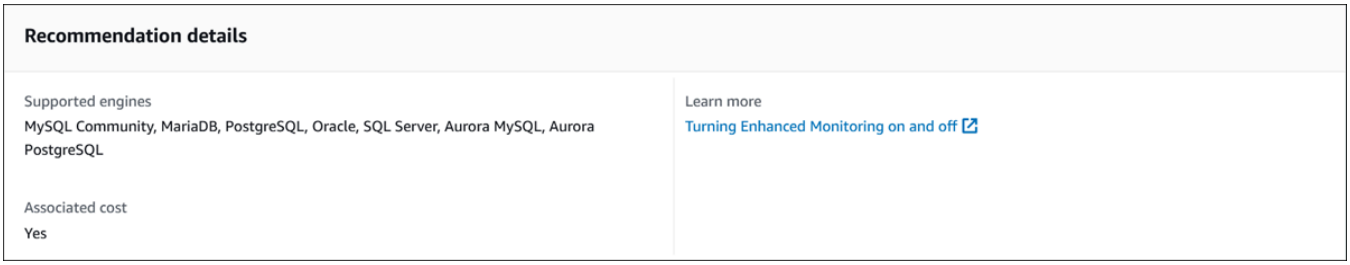
- Recomendação: um resumo da recomendação e se é necessário tempo de inatividade para aplicar a recomendação.



- Recursos afetados: detalhes dos recursos afetados.



- Detalhes da recomendação: informações do mecanismo compatível, qualquer custo associado necessário para aplicar a recomendação e link da documentação para saber mais.



CLI

Para visualizar as recomendações do Amazon RDS das instâncias de banco de dados, use o comando a seguir em AWS CLI.

```
aws rds describe-db-recommendations
```

API do RDS

Para visualizar as recomendações do Amazon RDS usando a API do Amazon RDS, use a operação DescribeDBRecommendations.

Aplicar as recomendações do Amazon RDS

Para aplicar as recomendações do Amazon RDS usando o console do Amazon RDS, selecione uma recomendação baseada na configuração ou um recurso afetado na página de detalhes. Depois, opte por aplicar a recomendação imediatamente ou programá-la para a próxima janela de manutenção. Talvez seja necessário que o recurso seja reiniciado para que a alteração tenha efeito. Para algumas

recomendações de grupos de parâmetros de banco de dados, talvez seja necessário reiniciar os recursos.

As recomendações proativas baseadas em limites ou reativas baseadas em anomalias não terão a opção de aplicação e poderão precisar de análise adicional.

Console

Como aplicar uma recomendação baseada em configuração

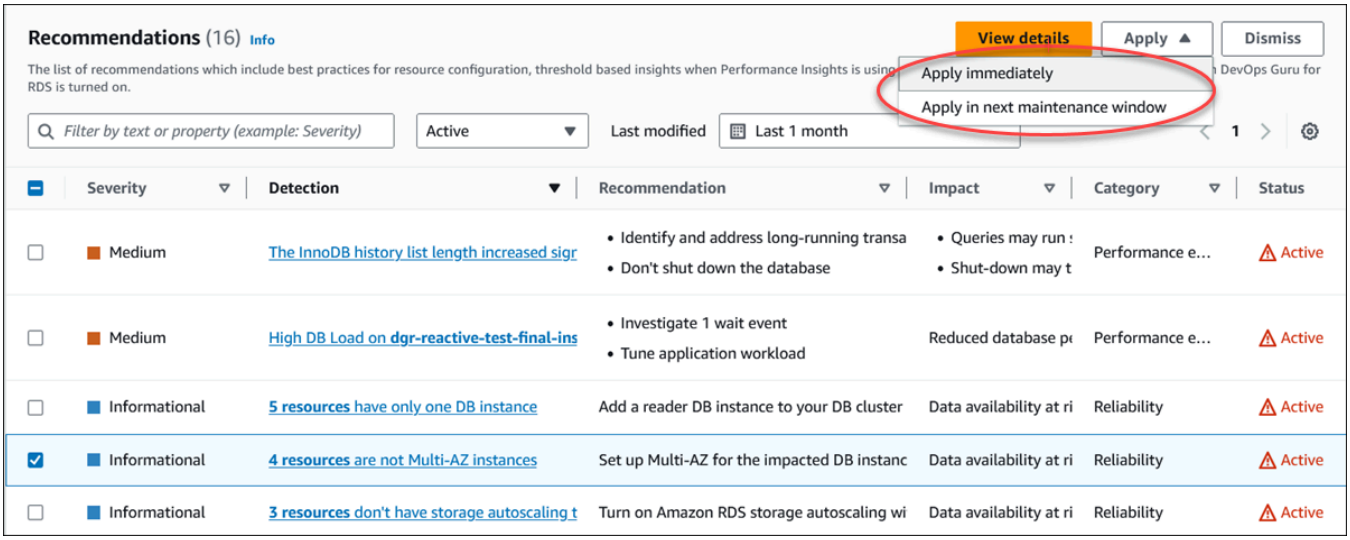
1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, execute qualquer uma das seguintes opções:
   - Selecione Recomendações.

     A página Recomendações aparece com a lista de todas as recomendações.
   - Selecione Bancos de dados e, depois, escolha Recomendações para um recurso na página de bancos de dados.

     Os detalhes são exibidos na guia Recomendações da recomendação selecionada.
   - Selecione Detecção para uma recomendação ativa na página Recomendações ou na guia Recomendações na página Bancos de dados.

     A página de detalhes da recomendação é exibida.
3. Selecione uma recomendação ou um ou mais recursos afetados na página de detalhes da recomendação e faça o seguinte:
   - Selecione Aplicar e, depois, escolha Aplicar imediatamente para aplicar a recomendação imediatamente.
   - Selecione Aplicar e, depois, escolha Aplicar durante a próxima janela de manutenção programada para programar na próxima janela de manutenção.

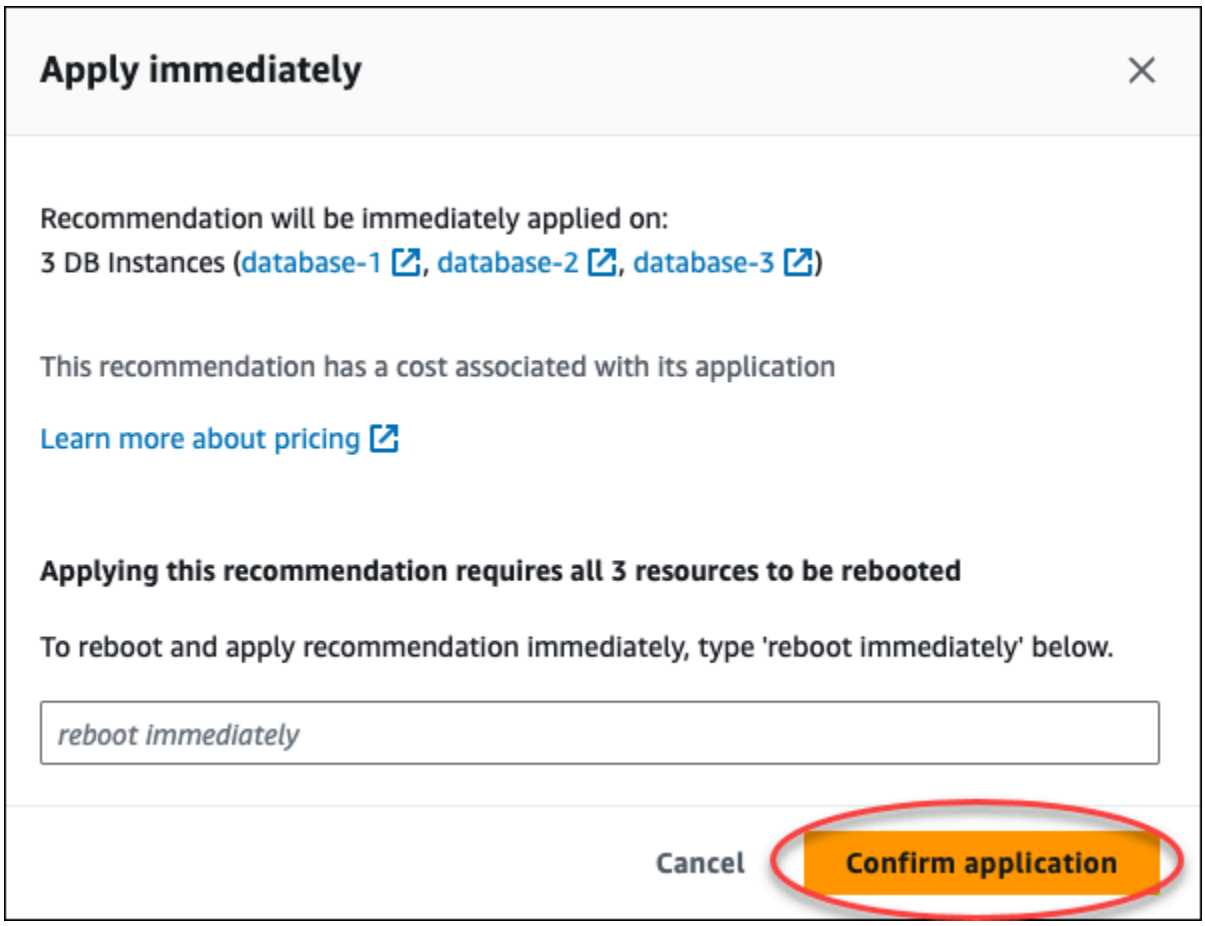
     O status da recomendação selecionada é atualizado para pendente até a próxima janela de manutenção.



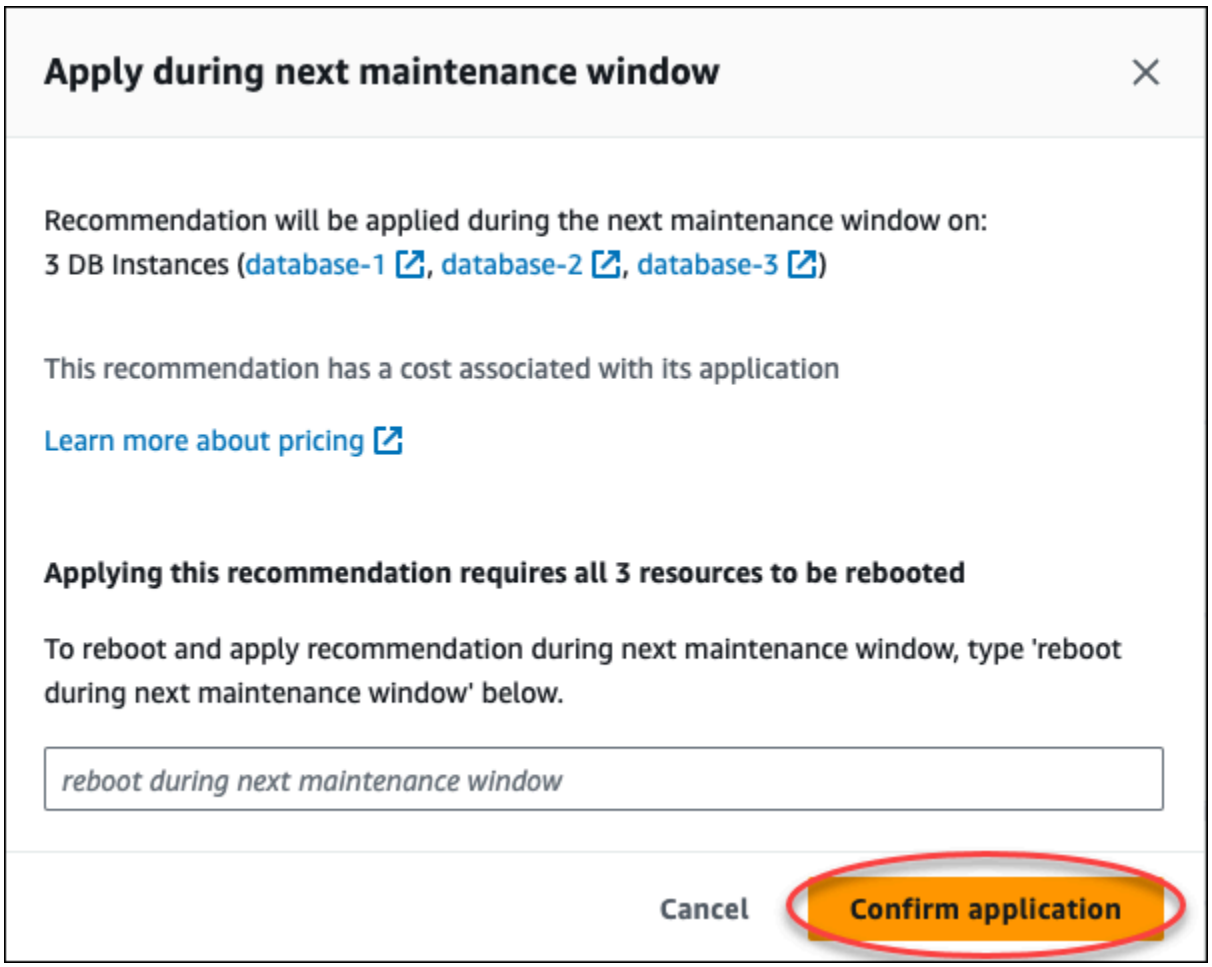
Uma janela de confirmação é exibida.

4. Selecione Confirmar aplicação para aplicar a recomendação. Essa janela confirma se os recursos precisam de uma reinicialização automática ou manual para que as alterações tenham efeito.

   O exemplo a seguir mostra a janela de confirmação para aplicar a recomendação imediatamente.

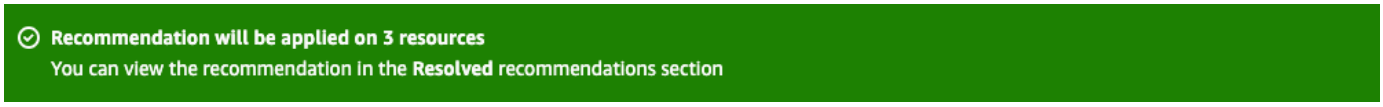


O exemplo a seguir mostra a janela de confirmação para programar a aplicação da recomendação na próxima janela de manutenção.



Um banner exibe uma mensagem quando a recomendação aplicada é bem-sucedida ou falhou.

O exemplo a seguir mostra o banner com a mensagem de êxito.



O exemplo a seguir mostra o banner com a mensagem de falha.



API do RDS

Para aplicar uma recomendação do RDS baseada em configuração usando a API do Amazon RDS

1. Use a operação DescribeDBRecommendations. `RecommendedActions` na saída pode ter uma ou mais ações recomendadas.
2. Use o objeto RecommendedAction para cada ação recomendada da etapa 1. A saída contém `Operation` e `Parameters`.

   O exemplo a seguir mostra a saída com uma ação recomendada.

```
   "RecommendedActions": [
       {
           "ActionId": "0b19ed15-840f-463c-a200-b10af1b552e3",
           "Title": "Turn on auto backup", // localized
           "Description": "Turn on auto backup for my-mysql-instance-1", //
localized
           "Operation": "ModifyDbInstance",
           "Parameters": [
               {
                   "Key": "DbInstanceIdentifier",
                   "Value": "my-mysql-instance-1"
               },
               {
                   "Key": "BackupRetentionPeriod",
                   "Value": "7"
               }
           ],
           "ApplyModes": ["immediately", "next-maintenance-window"],
           "Status": "applied"
       },
       ... // several others
   ],
```

3. Use `operation` para cada ação recomendada da saída na etapa 2 e insira os valores `Parameters`.
4. Depois que a operação na etapa 2 for bem-sucedida, use a operação ModifyDBRecommendation para modificar o status da recomendação.

Ignorar as recomendações do Amazon RDS

É possível ignorar uma ou mais recomendações do Amazon RDS usando o console do Amazon RDS, a AWS CLI ou a API do Amazon RDS.

Console

Como dispensar uma ou mais recomendações

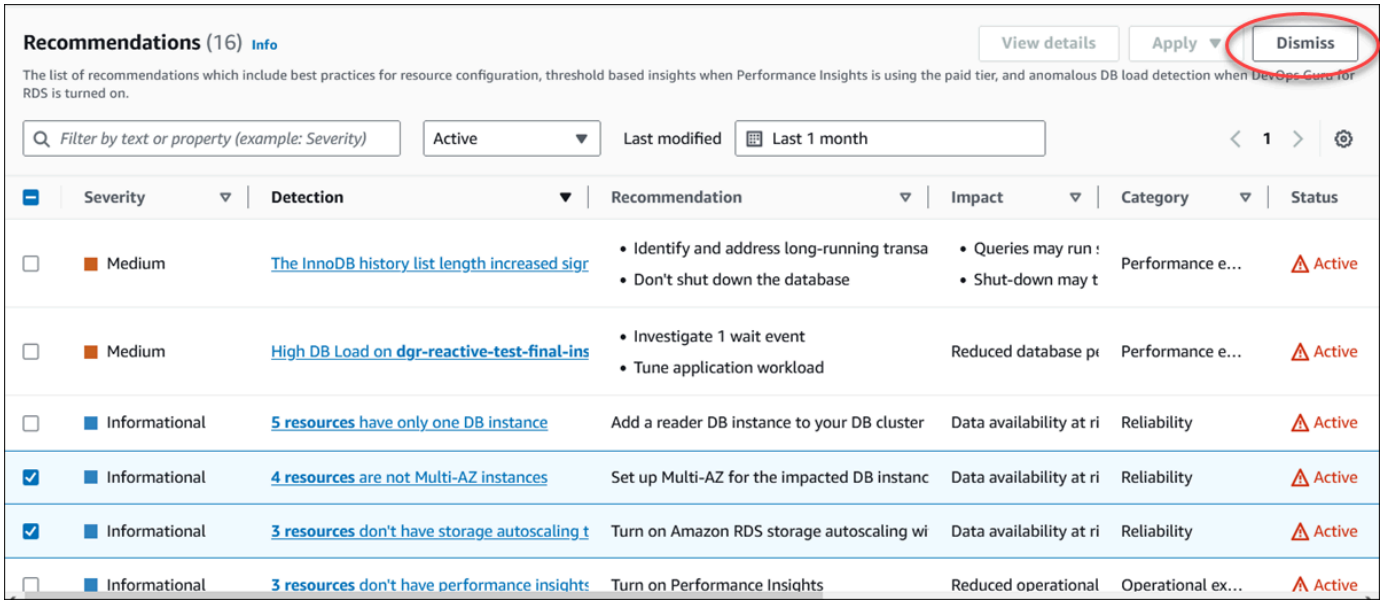
1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, execute qualquer uma das seguintes opções:
   - Selecione Recomendações.

     A página Recomendações aparece com a lista de todas as recomendações.
   - Selecione Bancos de dados e, depois, escolha Recomendações para um recurso na página de bancos de dados.

     Os detalhes são exibidos na guia Recomendações da recomendação selecionada.
   - Selecione Detecção para uma recomendação ativa na página Recomendações ou na guia Recomendações na página Bancos de dados.

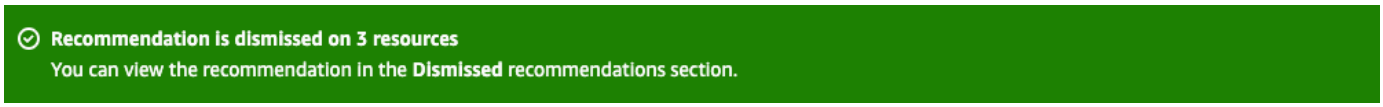
     A página de detalhes da recomendação exibe a lista dos recursos afetados.
3. Escolha uma ou mais recomendações ou um ou mais recursos afetados na página de detalhes da recomendação e selecione Dispensar.

   O exemplo a seguir mostra a página Recomendações com várias recomendações ativas selecionadas para serem dispensadas.



Um banner exibe uma mensagem quando uma ou mais recomendações selecionadas são dispensadas.

O exemplo a seguir mostra o banner com a mensagem de êxito.



O exemplo a seguir mostra o banner com a mensagem de falha.



CLI

Como dispensar uma recomendação do RDS usando a AWS CLI

1. Execute o comando `aws rds describe-db-recommendations --filters "Name=status,Values=active"`.

   A saída fornece uma lista de recomendações no status `active`.
2. Encontre o `recommendationId` para a recomendação que você deseja dispensar na etapa 1.
3. Execute o comando `>aws rds modify-db-recommendation --status dismissed --recommendationId <ID>` com o `recommendationId` da etapa 2 para dispensar a recomendação.

API do RDS

Para dispensar uma recomendação do RDS usando a API do Amazon RDS, use a operação ModifyDBRecommendation.

Modificar as recomendações ignoradas do Amazon RDS para recomendações ativas

É possível modificar uma ou mais recomendações ignoradas do Amazon RDS para recomendações ativas usando o console do Amazon RDS, a AWS CLI ou a API do Amazon RDS.

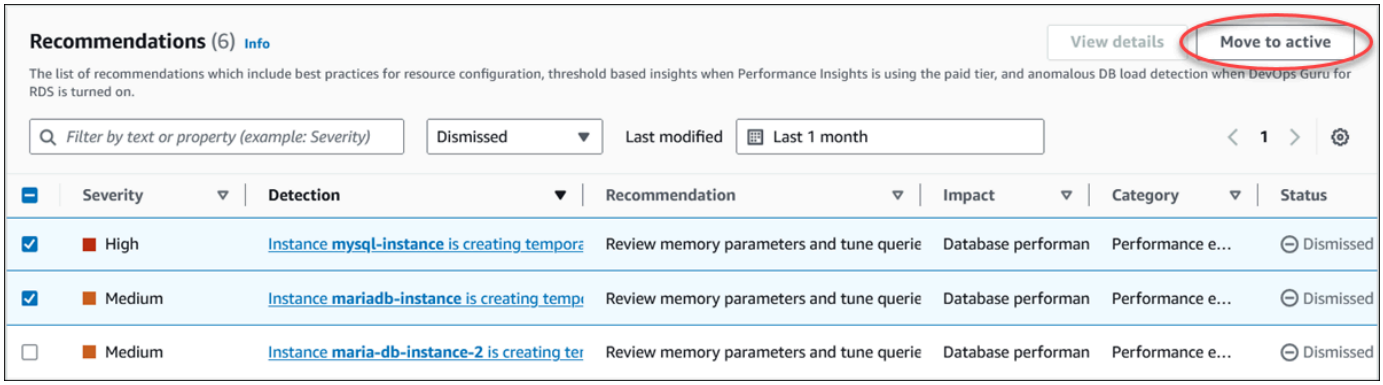
Console

Como mover uma ou mais recomendações dispensadas para recomendações ativas

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, execute qualquer uma das seguintes opções:
   - Selecione Recomendações.

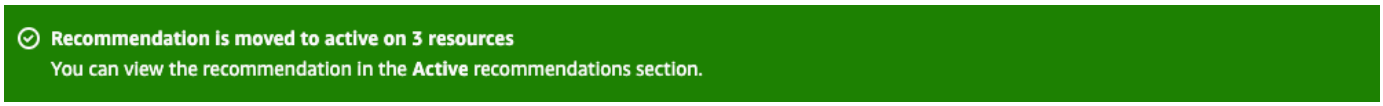
     A página Recomendações exibe uma lista de recomendações classificadas pela gravidade de todos os recursos da conta.
   - Selecione Bancos de dados e, depois, escolha Recomendações para um recurso na página de bancos de dados.

     A guia Recomendações exibe as recomendações e os detalhes do recurso selecionado.
3. Escolha uma ou mais recomendações dispensadas na lista e selecione Mover para ativo.



Um banner exibe uma mensagem de êxito ou falha ao mover as recomendações selecionadas do status dispensado para ativo.

O exemplo a seguir mostra o banner com a mensagem de êxito.



O exemplo a seguir mostra o banner com a mensagem de falha.



CLI

Como alterar uma recomendação dispensada do RDS para uma recomendação ativa usando a AWS CLI

1. Execute o comando `aws rds describe-db-recommendations --filters "Name=status,Values=dismissed"`.

   A saída fornece uma lista de recomendações no status `dismissed`.
2. Encontre `recommendationId` para a recomendação cujo status você deseja alterar da etapa 1.
3. Execute o comando `>aws rds modify-db-recommendation --status active --recommendationId <ID>` com o `recommendationId` da etapa 2 para alterar para a recomendação ativa.

API do RDS

Para alterar uma recomendação do RDS para uma recomendação ativa usando a API do Amazon RDS, use a operação ModifyDBRecommendation.

Recomendações da referência do Amazon RDS

O Amazon RDS gera recomendações para um recurso quando esse recurso é criado ou modificado. É possível encontrar exemplos de recomendações do Amazon RDS na tabela a seguir.

Tipo	Descrição	Recomendação	Tempo de inatividade necessário	Mais informações
O volume magnético está em uso.	As instâncias de banco de dados estão usando armazenamento magnético. O armazenamento magnético não é recomendado para a maioria das instâncias de banco de dados. Escolha um tipo de armazenamento diferente: de uso geral (SSD) ou IOPS provisionadas.	Selecione um tipo de armazenamento diferente: de uso geral (SSD) ou IOPS provisionadas.	Sim	Volumes da geração anterior na documentação do Amazon EC2.
Os backups automatizados do recurso estão desativados	Os backups automatizados não estão ativados para as instâncias de banco de dados. Backups automatizados são recomendados porque permitem a recuperação para um ponto no tempo de instâncias de banco de dados.	Ative os backups automatizados com um período de retenção de até 14 dias.	Sim	Ativar backups automáticos Desmistificar os custos de armazenamento de backup do Amazon RDS no blog de bancos de dados da AWS

Tipo	Descrição	Recomendação	Tempo de inatividade necessário	Mais informações
A atualização da versão secundária do mecanismo é necessária.	Os recursos de banco de dados não estão executando a versão secundária mais recente do mecanismo de banco de dados. A versão secundária mais recente contém as correções de segurança mais recentes e outras melhorias.	Atualize para a versão mais recente do mecanismo.	Sim	Atualizar a versão de mecanismo de uma instância de banco de dados
O monitoramento aprimorado está desativado.	Os recursos de banco de dados não têm o monitoramento aprimorado ativado. O monitoramento avançado fornece métricas do sistema operacional em tempo real para monitoramento e solução de problemas.	Ative o monitoramento aprimorado.	Não	Monitorar métricas do SO com o monitoramento avançado

Tipo	Descrição	Recomendação	Tempo de inatividade necessário	Mais informações
A criptografia de armazenamento está desativada.	O Amazon RDS é compatível com a criptografia em repouso para todos os mecanismos de banco de dados usando as chaves gerenciadas no AWS Key Management Service (AWS KMS). Em uma instância de banco de dados ativa com criptografia do Amazon RDS, os dados armazenados em repouso no armazenamento são criptografados, de forma semelhante aos backups automatizados, réplicas de leitura e snapshots. Se a criptografia não estiver ativada durante a criação de uma instância de banco de dados, será necessário criar e restaurar uma cópia	Ative a criptografia de dados em repouso para a instância de banco de dados.	Sim	Segurança no Amazon RDS Copiar um snapshot de banco de dados para o Amazon RDS

Tipo	Descrição	Recomendação	Tempo de inatividade necessário	Mais informações
	criptografada do snapshot descriptografado da instância de banco de dados antes de ativar a criptografia.			
O Performance Insights está desativado	O Performance Insights monitora a carga da instância de banco de dados para ajudar a analisar e solucionar problemas de performance do banco de dados. Recomendamos ativar o Performance Insights.	Ative o Performance Insights.	Não	Monitorar a carga de banco de dados com o Performance Insights no Amazon RDS

Tipo	Descrição	Recomendação	Tempo de inatividade necessário	Mais informações
As instâncias de banco de dados têm o ajuste de escala automático do armazenamento desativado.	O ajuste de escala automático do armazenamento não está ativado para a instância de banco de dados. Quando há um aumento na workload, o ajuste de escala automático do armazenamento do RDS escala automaticamente a capacidade de armazenamento, sem tempo de inatividade.	Ative o ajuste de escala automático do armazenamento do Amazon RDS com um limite máximo de armazenamento especificado.	Não	Gerenciar a capacidade automaticamente com a escalabilidade automática de armazenamento do Amazon RDS
A atualização das versões principais dos recursos do RDS é necessária.	Bancos de dados com a versão principal atual do mecanismo de banco de dados não serão aceitos. Recomendamos atualizar para a versão principal mais recente, que inclui novas funcionalidades e aprimoramentos.	Atualize para a versão principal mais recente do mecanismo de banco de dados.	Sim	Atualizar a versão de mecanismo de uma instância de banco de dados Usar implantações azul/verde do Amazon RDS para atualizações de banco de dados

Tipo	Descrição	Recomendação	Tempo de inatividade necessário	Mais informações
A atualização da classe de instância de recursos do RDS é necessária.	A instância de banco de dados está executando uma classe de instância de banco de dados de uma geração anterior. Substituímos as classes de instância de banco de dados de uma geração anterior por classes de instância de banco de dados com melhor custo, performance ou ambos. Recomendamos executar a instância de banco de dados com uma classe de instância de banco de dados de uma geração mais recente.	Atualize a classe de instância de banco de dados.	Sim	Mecanismos de banco de dados compatíveis para classes de instância de banco de dados

Tipo	Descrição	Recomendação	Tempo de inatividade necessário	Mais informações
Recursos do RDS usando a edição final do suporte do mecanismo sob licença incluída	Recomendamos atualizar a versão principal para a versão mais recente do mecanismo aceita pelo Amazon RDS para continuar com o suporte de licença atual. A versão do mecanismo do banco de dados não será compatível com a licença atual.	Recomendamos atualizar o banco de dados para a versão mais recente compatível com o Amazon RDS para continuar usando o modelo licenciado.	Sim	Atualizações da versão principal do Oracle

Tipo	Descrição	Recomendação	Tempo de inatividade necessário	Mais informações
Instâncias de banco de dados que não usam a implantação multi-AZ	Recomendamos que você use a implantação multi-AZ. As implantações multi-AZ aumentam a disponibilidade e a durabilidade da instância de banco de dados.	Configurar multi-AZ para instâncias de banco de dados afetadas	Não Não ocorre uma inatividade durante esta alteraçã. No entanto há um possíve impacto na perform ce. Para ter mais informaes, consult Conver uma instânc de banco	Preços do multi-AZ do Amazon RDS

Tipo	Descrição	Recomendação	Tempo de inatividade necessário	Mais informações
			de dados em uma implantação multi-AZ para o Amazon RDS.	

Tipo	Descrição	Recomendação	Tempo de inatividade necessário	Mais informações
Os parâmetros de memória do banco de dados estão divergindo do padrão.	Os parâmetros de memória das instâncias de banco de dados são significativamente diferentes dos valores padrão. Essas configurações podem afetar a performance e causar erros. Recomendamos definir os parâmetros de memória personalizados da instância de banco de dados como os valores padrão no grupo de parâmetros de banco de dados.	Redefina os parâmetros de memória para os valores padrão.	Não	Best practices for configuring performance parameters for Amazon RDS for MySQL no Blog de bancos de dados da AWS

Tipo	Descrição	Recomendação	Tempo de inatividade necessário	Mais informações
O parâmetro `InnoDB_Change_Buffering` usando menos do que o valor ideal	O buffer de alterações permite que uma instância de banco de dados do MySQL adie algumas gravações necessárias para manter índices secundários. Esse recurso foi útil em ambientes com discos lentos. A alteração na configuração do buffer melhorou um pouco a performance do banco de dados, mas causou um atraso na recuperação de falhas e longos tempos de desligamento durante a atualização. Definido como `OFF` por padrão na versão 8.4 do MySQL.	Defina o valor do parâmetro `InnoDB_Change_Buffering` como `NONE` nos grupos de parâmetros do banco de dados.	Não	Best practices for configuring performance parameters for Amazon RDS for MySQL no Blog de bancos de dados da AWS

Tipo	Descrição	Recomendação	Tempo de inatividade necessário	Mais informações
O parâmetro de cache de consulta está ativado.	Quando as alterações exigirem que o cache de consultas seja limpo, a instância de banco de dados parecerá paralisada. A maioria das workloads não se beneficia de um cache de consultas. O cache de consultas foi removido do MySQL versão 8.0 e posterior. Recomendamos definir o parâmetro query_cache_type como 0.	Defina o valor do parâmetro `query_cache_type` como `0` nos grupos de parâmetros do banco de dados.	Sim	Best practices for configuring performance parameters for Amazon RDS for MySQL no Blog de bancos de dados da AWS

Tipo	Descrição	Recomendação	Tempo de inatividade necessário	Mais informações
O parâmetro `log_output` está definido como tabela.	Quando `log_output` está definido como `TABLE`, mais armazenamento é usado do que quando `log_output` está definido como `FILE`. Recomendamos definir o parâmetro como `FILE` para não atingir o limite de tamanho do armazenamento. Definido como `FILE` por padrão no MySQL 8.4 e versões posteriores.	Defina o valor do parâmetro `log_output` como `FILE` nos grupos de parâmetros do banco de dados.	Não	Arquivos de log do banco de dados MySQL

Tipo	Descrição	Recomendação	Tempo de inatividade necessário	Mais informações
Grupos de parâmetros que não usam páginas grandes.	Páginas grandes podem aumentar a escalabilidade do banco de dados, mas a instância de banco de dados não está usando páginas grandes. Recomendamos definir o valor do parâmetro `use_large_pages` como `ONLY` no grupo de parâmetros de banco de dados para a instância de banco de dados.	Defina o valor do parâmetro `use_large_pages` como `ONLY` nos grupos de parâmetros do banco de dados.	Sim	Ativar o HugePages para uma instância do RDS para Oracle

Tipo	Descrição	Recomendação	Tempo de inatividade necessário	Mais informações
O parâmetro `autovacuum` está desativado.	O parâmetro autovacuum está desativado para instâncias de banco de dados. Desativar o autovacuum aumenta o inchaço da tabela e do índice e afeta a performance. Recomendamos que você ative o autovacuum nos grupos de parâmetros de banco de dados.	Ative o parâmetro autovacuum nos grupos de parâmetros do de banco de dados.	Não	Understanding autovacuum in Amazon RDS for PostgreSQL environments no Blog de bancos de dados da AWS

Tipo	Descrição	Recomendação	Tempo de inatividade necessário	Mais informações
O parâmetro `synchronous_commit` está desativado.	Quando o `synchronous_commit` parâmetro é desativado, os dados podem ser perdidos em uma falha no banco de dados. A durabilidade do banco de dados está em risco. Recomendamos que você ative o parâmetro `synchronous_commit` .	Ative o parâmetro `synchronous_commit` nos grupos de parâmetros do banco de dados.	Sim	Parâmetros do Amazon Aurora PostgreSQL: replicação, segurança e registro em log no blog de banco de dados da AWS

Tipo	Descrição	Recomendação	Tempo de inatividade necessário	Mais informações
O parâmetro `track_counts` está desativado.	Quando o parâmetro `track_counts` estiver desativado, o banco de dados não coletará as estatísticas de atividade do banco de dados. O autovacuum exige que essas estatísticas funcionem corretamente. Recomendamos que você defina o parâmetro `track_counts` como 1.	Defina o parâmetro `track_counts` como 1.	Não	Estatísticas de tempo de execução do PostgreSQL

Tipo	Descrição	Recomendação	Tempo de inatividade necessário	Mais informações
O parâmetro `enable_indexonlyscan` está desativado.	O planejador ou o otimizador de consultas não pode usar o plano de analisar somente o índice quando ele está desativado. Recomendamos definir o valor do parâmetro `enable_indexonlyscan` como 1.	Defina o valor do parâmetro `enable_indexonlyscan` como 1.	Não	Configuração do método Planner para PostgreSQL
O parâmetro `enable_indexscan` está desativado.	O planejador ou o otimizador de consultas não pode usar o plano de analisar o índice quando ele está desativado. Recomendamos que você defina o valor `enable_indexscan` como 1.	Defina o valor do parâmetro `enable_indexscan` como 1.	Não	Configuração do método Planner para PostgreSQL

Tipo	Descrição	Recomendação	Tempo de inatividade necessário	Mais informações
O parâmetro `innodb_flush_log_at_trx` está desativado.	O valor do parâmetro `innodb_flush_log_at_trx` da instância de banco de dados não é um valor seguro. Esse parâmetro controla a persistência das operações de confirmação no disco. Recomendamos que você defina o parâmetro `innodb_flush_log_at_trx` como 1.	Defina o valor do parâmetro `innodb_flush_log_at_trx` como 1.	Não	Best practices for configuring performance parameters for Amazon RDS for MySQL no Blog de bancos de dados da AWS

Tipo	Descrição	Recomendação	Tempo de inatividade necessário	Mais informações
O parâmetro `sync_binlog` está desativado.	A sincronização do log binário com o disco não é aplicada antes que as confirmações das transações sejam reconhecidas na instância de banco de dados. Recomendamos definir o valor do parâmetro `sync_binlog` como 1.	Defina o valor do parâmetro `sync_binlog` como 1.	Não	Best practices for configuring replication parameters for Amazon RDS for MySQL no Blog de bancos de dados da AWS

Tipo	Descrição	Recomendação	Tempo de inatividade necessário	Mais informações
O parâmetro `innodb_stats_persistent` está desativado.	Sua instância de banco de dados não está configurada para manter as estatísticas do InnoDB no disco. Quando as estatísticas não são armazenadas, elas são recalculadas sempre que a instância é reiniciada e a tabela é acessada. Isso causa variações no plano de execução da consulta. Você pode modificar o valor desse parâmetro global no nível da tabela. Recomendamos definir o valor do parâmetro `innodb_stats_persistent` como `ON`.	Defina o valor do parâmetro `innodb_stats_persistent` como `ON`.	Não	Best practices for configuring performance parameters for Amazon RDS for MySQL no Blog de bancos de dados da AWS

Tipo	Descrição	Recomendação	Tempo de inatividade necessário	Mais informações
O parâmetro `innodb_open_files` é baixo.	O parâmetro `innodb_open_files` controla o número de arquivos que o InnoDB pode abrir ao mesmo tempo. O InnoDB abre todos os arquivos de log e tablespace do sistema quando o mysqld está em execução. Sua instância de banco de dados tem um valor baixo para o número máximo de arquivos que o InnoDB pode abrir ao mesmo tempo. Recomendamos que você defina o parâmetro `innodb_open_files` com um valor mínimo de 65.	Defina o parâmetro `innodb_open_files` como um valor mínimo de 65.	Sim	Arquivos abertos do InnoDB para MySQL

Tipo	Descrição	Recomendação	Tempo de inatividade necessário	Mais informações
O parâmetro `max_user_connections` é baixo.	Sua instância de banco de dados tem um valor baixo para o número máximo de conexões simultâneas para cada conta de banco de dados. Recomendamos definir o parâmetro `max_user_connections` como um número maior que 5.	Aumente o valor do parâmetro `max_user_connections` para um número maior que 5.	Sim	Definir limites de recursos da conta para MySQL

Tipo	Descrição	Recomendação	Tempo de inatividade necessário	Mais informações
As réplicas de leitura são abertas no modo gravável.	A instância de banco de dados tem uma réplica de leitura no modo de gravação, que permite que os clientes realizem atualizações. Recomendamos que você defina o parâmetro `read_only` como `TrueIfReplica` para que as réplicas de leitura não estejam no modo gravável.	Defina o valor do parâmetro `read_only` como `TrueIfReplica` .	Não	Best practices for configuring replication parameters for Amazon RDS for MySQL no Blog de bancos de dados da AWS

Tipo	Descrição	Recomendação	Tempo de inatividade necessário	Mais informações
A configuração do parâmetro `innodb_default_row_format` não é segura.	A instância de banco de dados encontra um problema conhecido: uma tabela criada em uma versão do MySQL inferior à 8.0.26 com o valor `row_format` definido como `COMPACT` ou `REDUNDANT` ficará inacessível e irrecuperável se o índice exceder 767 bytes. Recomendamos definir o valor do parâmetro `innodb_default_row_format` como `DYNAMIC`.	Defina o valor do parâmetro `innodb_default_row_format` como `DYNAMIC`.	Não	Alterações feitas no MySQL 8.0.26

Tipo	Descrição	Recomendação	Tempo de inatividade necessário	Mais informações
O parâmetro `general_logging` está ativado.	O registro em log geral é ativado para a instância de banco de dados. Essa configuração é útil para solucionar problemas no banco de dados. No entanto, ativar o registro em log geral aumenta a quantidade de operações de E/S e o espaço de armazenamento alocado, o que pode causar contenção e degradação da performance. Confira os requisitos para uso do registro em log geral. Recomendamos definir o valor do parâmetro `general_logging` como `0`.	Confira os requisitos para uso do registro em log geral. Se não for obrigatório, recomendamos definir o valor do parâmetro `general_logging` como `0`.	Não	Visão geral dos logs de banco de dados do RDS para MySQL

Tipo	Descrição	Recomendação	Tempo de inatividade necessário	Mais informações
Instância do RDS subprovisionada para a capacidade de memória do sistema	Recomendamos ajustar as consultas para usar menos memória ou usar um tipo de instância de banco de dados com maior memória alocada. Quando a instância está com pouca memória, a performance do banco de dados é afetada.	Usar uma instância de banco de dados com maior capacidade de memória	Sim	Escalar a instância do Amazon RDS vertical e horizontalmente no blog de bancos de dados da AWS Tipos de instância do Amazon RDS Preços do Amazon RDS

Tipo	Descrição	Recomendação	Tempo de inatividade necessário	Mais informações
Instância do RDS subprovisionada para a capacidade de CPU do sistema	Recomendamos ajustar as consultas para utilizar menos CPU ou modificar a instância de banco de dados para utilizar uma classe de instância de banco de dados com mais vCPUs alocadas. A performance do banco de dados pode diminuir quando uma instância de banco de dados está com pouca CPU.	Usar uma instância de banco de dados com maior capacidade de CPU	Sim	Escalar a instância do Amazon RDS vertical e horizontalmente no blog de bancos de dados da AWS Tipos de instância do Amazon RDS Preços do Amazon RDS

Tipo	Descrição	Recomendação	Tempo de inatividade necessário	Mais informações
Os recursos do RDS não estão utilizando o grupo de conexões corretamente	Recomendamos habilitar o Amazon RDS Proxy para agrupar e compartilhar com eficiência as conexões de banco de dados existentes. Se você já estiver usando um proxy para o banco de dados, configure-o corretamente para melhorar o grupo de conexões e o balanceamento de carga em várias instâncias de banco de dados. O RDS Proxy pode ajudar a reduzir o risco de esgotamento da conexão e o tempo de inatividade, enquanto melhora a disponibilidade e a escalabilidade.	Habilitar o RDS Proxy ou modificar a configuração de proxy existente	Não	Escalar a instância do Amazon RDS vertical e horizontalmente no blog de bancos de dados da AWS Amazon RDS Proxy Preços do Amazon RDS Proxy

Tipo	Descrição	Recomendação	Tempo de inatividade necessário	Mais informações
As instâncias do RDS estão criando objetos temporários excessivos	Recomendamos que você ajuste a workload para evitar a criação excessiva de objetos temporários ou alterne para classes de instância do RDS que comportem leituras otimizadas. As Leituras otimizadas pelo RDS melhoram a performance do banco de dados para workloads que envolvem grande número de objetos temporários e/ou grandes objetos temporários. Avalie a workload para determinar se o uso de uma instância com Leituras otimizadas pelo RDS beneficia a workload de banco de dados.	Usar um tipo de instância de banco de dados com Leituras otimizadas pelo RDS	Sim	Tipos de instância do Amazon RDS Melhorar a performance das consultas para o RDS para MySQL com Leituras otimizadas pelo Amazon RDS Melhorar a performance das consultas para o RDS para MariaDB com Leituras otimizadas pelo Amazon RDS Melhorar a performance das consultas para o RDS para PostgreSQL com Leituras otimizadas pelo Amazon RDS

Tipo	Descrição	Recomendação	Tempo de inatividade necessário	Mais informações
As instâncias do RDS são subprovisionadas para a capacidade de IOPS	Recomendamos ajustar a workload do banco de dados para reduzir as IOPS ou aumentar a escala verticalmente da instância de banco de dados para um tipo com um limite de IOPS padrão mais alto. A instância de banco de dados atual não é compatível com IOPS provisionadas ou a workload do banco de dados tem alta utilização de IOPS.	Usar um tipo de instância de banco de dados com limites de IOPS padrão mais altos	Sim	Tipos de instância do Amazon RDS Armazenamento de instâncias de banco de dados do Amazon RDS Carga de banco de dados

Tipo	Descrição	Recomendação	Tempo de inatividade necessário	Mais informações
Instâncias do RDS têm volumes subprovisionados do Amazon EBS	Recomendamos ajustar a workload do banco de dados para reduzir as IOPS ou aumentar as IOPS provisionadas do banco de dados. Quando a utilização das IOPS se aproxima das IOPS provisionadas, o desempenho do banco de dados pode diminuir.	Provisionar mais IOPS para a instância de banco de dados	Sim	Tipos de instância do Amazon RDS Armazenamento de instâncias de banco de dados do Amazon RDS Carga de banco de dados

Tipo	Descrição	Recomendação	Tempo de inatividade necessário	Mais informações
As instâncias do RDS são subprovisionadas para a capacidade de throughput	Recomendamos ajustar a workload do banco de dados para reduzir o throughput ou aumentar o throughput provisionado para o banco de dados. Quando a utilização do throughput se aproxima do throughput provisionado, o desempenho do banco de dados pode ser afetado.	Provisionar mais throughput para a instância de banco de dados	Sim	Tipos de instância do Amazon RDS Armazenamento de instâncias de banco de dados do Amazon RDS Carga de banco de dados

Tipo	Descrição	Recomendação	Tempo de inatividade necessário	Mais informações
As instâncias do RDS são subprovisionadas para E/S do EBS	Recomendamos ajustar a workload de banco de dados para reduzir as operações de E/S ou modificar a instância de banco de dados para usar os volumes io2 Block Express do Amazon RDS, projetados para workloads de banco de dados que exigem alta performance, alto throughput e baixa latência. Com a workload atual, o banco de dados pode não conseguir processar as operações de E/S na taxa necessária, o que pode levar à degradação da performance.	Usar os volumes io2 Block Express do Amazon RDS para a instância do RDS	Não	Armazenamento de instâncias de banco de dados do Amazon RDS Métricas do Amazon CloudWatch para o Amazon RDS Provisioned IOPS SSD volumes no Guia do usuário do Amazon EBS

Visualizar métricas no console do Amazon RDS

O Amazon RDS se integra ao Amazon CloudWatch para exibir várias métricas de instância de banco de dados do RDS no console do RDS. As descrições dessas métricas estão disponíveis em Referência de métricas do Amazon RDS.

Para a instância de banco de dados, as seguintes categorias de métricas são monitoradas:

- CloudWatch: mostra as métricas do Amazon CloudWatch relacionadas ao RDS disponíveis no console do RDS. Também é possível visualizá-las no console do CloudWatch. Cada métrica inclui um gráfico que mostra a métrica monitorada durante um período específico. Veja uma lista completa de métricas do CloudWatch em Métricas do Amazon CloudWatch para o Amazon RDS.
- Monitoramento avançado: mostra um resumo das métricas do sistema operacional quando o monitoramento avançado de sua instância de banco de dados do RDS está ativado. O RDS fornece as métricas do monitoramento avançado à sua conta do Amazon CloudWatch Logs. Cada métrica de sistema operacional inclui um gráfico que mostra a métrica monitorada durante um período específico. Para obter uma visão geral, consulte Monitorar métricas do SO com o monitoramento avançado. Veja uma lista das métricas do monitoramento avançado em Métricas do sistema operacional em monitoramento avançado.
- OS Process list (Lista de processos de SO): mostra os detalhes de cada processo em execução na instância de banco de dados selecionada.
- Performance Insights: abre o painel do Amazon RDS Performance Insights relacionado a uma instância de banco de dados. Para ter uma visão geral do Performance Insights, consulte Monitorar a carga de banco de dados com o Performance Insights no Amazon RDS. Veja uma lista das métricas do Performance Insights em Métricas do Amazon CloudWatch para Insights de Performance do Amazon RDS.

O Amazon RDS agora fornece uma visão consolidada das métricas do Insights de Performance e do CloudWatch no painel do Insights de Performance. O Insights de Performance deve estar ativado para que a instância de banco de dados use essa visualização. Você pode escolher a nova visualização de monitoramento na guia Monitoramento ou Insights de Performance no painel de navegação. Para ver as instruções para escolher essa visualização, consulte Visualizar métricas combinadas com o painel do Insights de Performance.

Visualizar métricas combinadas com o painel do Insights de Performance

Important

A AWS anunciou a data de fim da vida útil do Insights de Performance: 30 de novembro de 2025. Após essa data, o Amazon RDS não oferecerá mais suporte à experiência do console, aos períodos de retenção flexíveis (de 1 a 24 meses) e aos preços associados do Insights de Performance. A API do Insights de Performance continuará existindo sem alterações de preço. Os custos da API do Insights de Performance aparecerão em sua fatura da com o custo do CloudWatch Database Insights.

Recomendamos que você atualize quaisquer instâncias de banco de dados usando o nível pago do Insights de Performance para o modo avançado do Database Insights antes de 30 de novembro de 2025. Para ter informações sobre como atualizar para o modo avançado do Database Insights, consulte Ativação do modo Avançado do Database Insights para Amazon RDS.

Se você não tomar nenhuma medida, as instâncias de banco de dados que estão usando o Insights de Performance usarão como padrão o modo padrão do Database Insights. Com o modo padrão do Database Insights, você pode perder o acesso ao histórico de dados de desempenho após sete dias e talvez não consiga usar planos de execução e recursos de análise sob demanda no console do Amazon RDS. Após 30 de novembro de 2025, somente o modo avançado do Database Insights permitirá planos de execução e análise sob demanda.

Com o CloudWatch Database Insights, você pode monitorar a carga do banco de dados para sua frota de bancos de dados e analisar e solucionar problemas de desempenho em grande escala. Para ter mais informações sobre o Database Insights, consulte Monitoramento de bancos de dados do Amazon RDS com o CloudWatch Database Insights. Para obter informações de definição de preço, consulte Definição de preço do Amazon CloudWatch.

O Amazon RDS fornece uma visão consolidada das métricas do Insights de Performance e do CloudWatch para sua instância de banco de dados no painel do Insights de Performance. É possível usar o painel pré-configurado ou criar um painel personalizado. O painel pré-configurado fornece as métricas mais usadas para ajudar a diagnosticar problemas de performance em um mecanismo de banco de dados. Como alternativa, é possível criar um painel do personalizado com as métricas de um mecanismo de banco de dados que atenda aos seus requisitos de análise. Depois, use esse

painel para todas as instâncias de banco de dados desse tipo de mecanismo de banco de dados em sua conta da AWS.

Você pode escolher a visualização de monitoramento na guia Monitoramento ou Insights de Performance no painel de navegação.

O Insights de Performance deve estar ativado para que a instância de banco de dados visualize as métricas combinadas no painel do Insights de Performance. Para obter mais informações sobre como ativar o Insights de Performance, consulte Ativar e desativar o Insights de Performance do Amazon RDS.

Nas seções a seguir, é possível aprender a exibir as métricas do Insights de Performance e do CloudWatch.

Tópicos

- Escolher a nova visualização de monitoramento na guia Monitoramento
- Escolher a nova visualização de monitoramento na página do Insights de Performance
- Criar um painel personalizado com o Insights de Performance
- Escolher o painel predefinido com o Insights de Performance

Escolher a nova visualização de monitoramento na guia Monitoramento

Important

A AWS anunciou a data de fim da vida útil do Insights de Performance: 30 de novembro de 2025. Após essa data, o Amazon RDS não oferecerá mais suporte à experiência do console, aos períodos de retenção flexíveis (de 1 a 24 meses) e aos preços associados do Insights de Performance. A API do Insights de Performance continuará existindo sem alterações de preço. Os custos da API do Insights de Performance aparecerão em sua fatura da com o custo do CloudWatch Database Insights.
Recomendamos que você atualize quaisquer instâncias de banco de dados usando o nível pago do Insights de Performance para o modo avançado do Database Insights antes de 30 de novembro de 2025. Para ter informações sobre como atualizar para o modo avançado do Database Insights, consulte Ativação do modo Avançado do Database Insights para Amazon RDS.
Se você não tomar nenhuma medida, as instâncias de banco de dados que estão usando o Insights de Performance usarão como padrão o modo padrão do Database Insights. Com

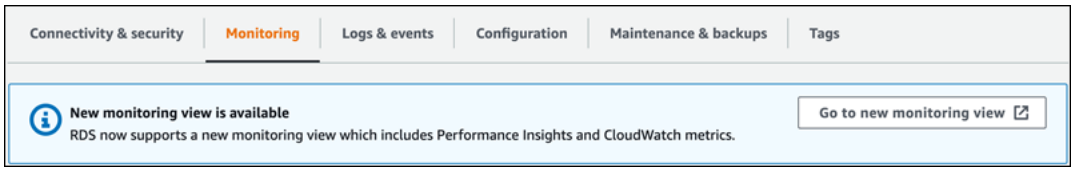
o modo padrão do Database Insights, você pode perder o acesso ao histórico de dados de desempenho após sete dias e talvez não consiga usar planos de execução e recursos de análise sob demanda no console do Amazon RDS. Após 30 de novembro de 2025, somente o modo avançado do Database Insights permitirá planos de execução e análise sob demanda.
Com o CloudWatch Database Insights, você pode monitorar a carga do banco de dados para sua frota de bancos de dados e analisar e solucionar problemas de desempenho em grande escala. Para ter mais informações sobre o Database Insights, consulte Monitoramento de bancos de dados do Amazon RDS com o CloudWatch Database Insights. Para obter informações de definição de preço, consulte Definição de preço do Amazon CloudWatch.

No console do Amazon RDS, é possível escolher a nova visualização de monitoramento para visualizar as métricas do Insights de Performance e do CloudWatch para sua instância de banco de dados.

Como escolher a nova visualização de monitoramento na guia Monitoramento

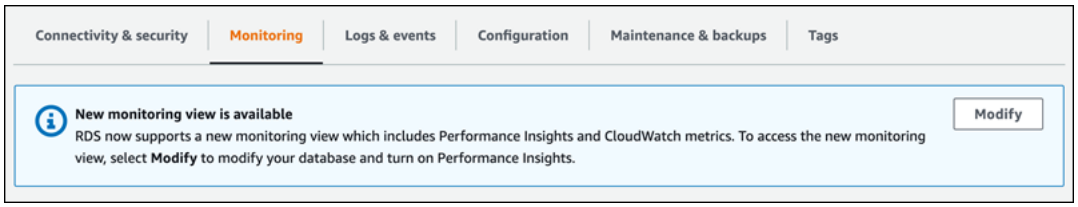
1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação à esquerda, selecione Bancos de dados.
3. Escolha a instância de banco de dados que você deseja monitorar.
4. Role para baixo e escolha a guia Monitoramento.

   Um banner aparece com a opção de escolher a nova visualização de monitoramento. O exemplo a seguir mostra o banner para escolher a nova visualização de monitoramento.



5. Escolha Ir para a nova visualização de monitoramento para abrir o painel do Insights de Performance com métricas do Insights de Performance e do CloudWatch para a instância de banco de dados.
6. (Opcional) Se o Insights de Performance estiver desativado para a instância de banco de dados, um banner aparecerá com a opção de modificar o cluster de banco de dados e ativar o Insights de Performance.

O exemplo a seguir mostra o banner para modificar o cluster de banco de dados na guia Monitoramento.



Escolha Modificar para modificar a instância de banco de dados e ativar o Insights de Performance. Para obter mais informações sobre como ativar o Insights de Performance, consulte Ativar e desativar o Insights de Performance do Amazon RDS

Escolher a nova visualização de monitoramento na página do Insights de Performance

Important

A AWS anunciou a data de fim da vida útil do Insights de Performance: 30 de novembro de 2025. Após essa data, o Amazon RDS não oferecerá mais suporte à experiência do console, aos períodos de retenção flexíveis (de 1 a 24 meses) e aos preços associados do Insights de Performance. A API do Insights de Performance continuará existindo sem alterações de preço. Os custos da API do Insights de Performance aparecerão em sua fatura da com o custo do CloudWatch Database Insights.

Recomendamos que você atualize quaisquer instâncias de banco de dados usando o nível pago do Insights de Performance para o modo avançado do Database Insights antes de 30 de novembro de 2025. Para ter informações sobre como atualizar para o modo avançado do Database Insights, consulte Ativação do modo Avançado do Database Insights para Amazon RDS.

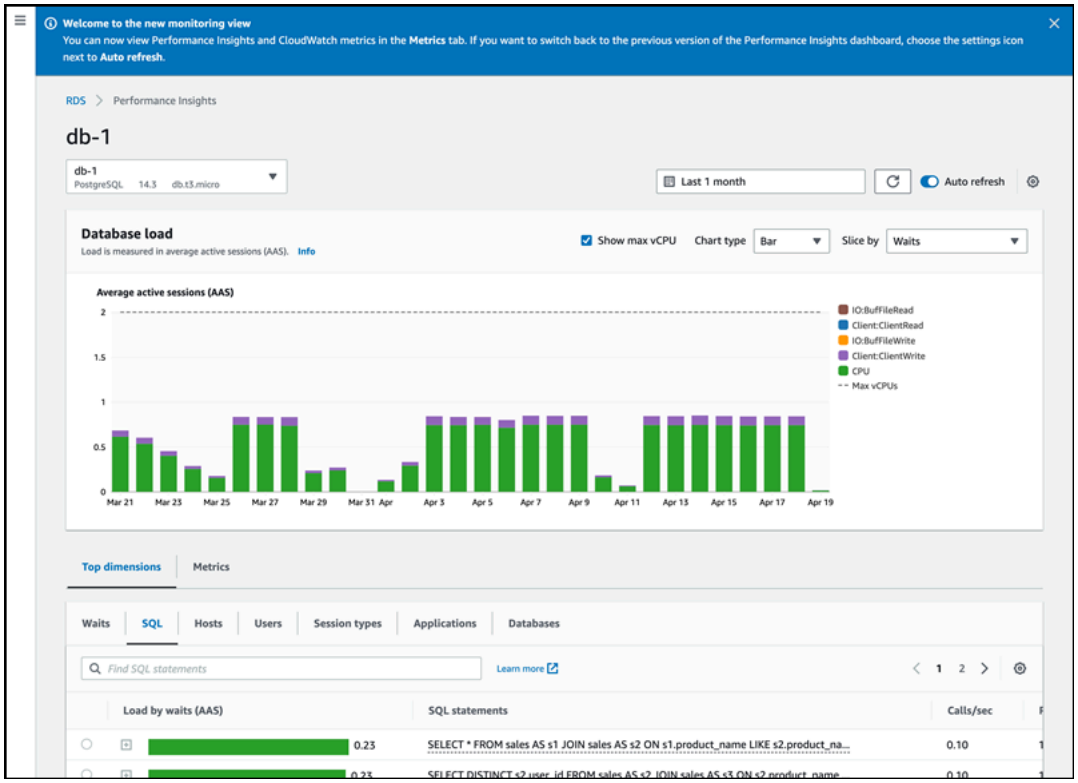
Se você não tomar nenhuma medida, as instâncias de banco de dados que estão usando o Insights de Performance usarão como padrão o modo padrão do Database Insights. Com o modo padrão do Database Insights, você pode perder o acesso ao histórico de dados de desempenho após sete dias e talvez não consiga usar planos de execução e recursos de análise sob demanda no console do Amazon RDS. Após 30 de novembro de 2025, somente o modo avançado do Database Insights permitirá planos de execução e análise sob demanda.

Com o CloudWatch Database Insights, você pode monitorar a carga do banco de dados para sua frota de bancos de dados e analisar e solucionar problemas de desempenho em grande escala. Para ter mais informações sobre o Database Insights, consulte Monitoramento de bancos de dados do Amazon RDS com o CloudWatch Database Insights. Para obter informações de definição de preço, consulte Definição de preço do Amazon CloudWatch.

No console do Amazon RDS, é possível escolher a nova visualização de monitoramento para visualizar as métricas do Insights de Performance e do CloudWatch para sua instância de banco de dados.

Como escolher a nova visualização de monitoramento com o Insights de Performance no painel de navegação

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Performance Insights.
3. Escolha uma instância de banco de dados para visualizar o painel do Insights de Performance que mostra as métricas do Insights de Performance e do CloudWatch para a instância de banco de dados.



Criar um painel personalizado com o Insights de Performance

Important

A AWS anunciou a data de fim da vida útil do Insights de Performance: 30 de novembro de 2025. Após essa data, o Amazon RDS não oferecerá mais suporte à experiência do console, aos períodos de retenção flexíveis (de 1 a 24 meses) e aos preços associados do Insights de Performance. A API do Insights de Performance continuará existindo sem alterações de preço. Os custos da API do Insights de Performance aparecerão em sua fatura da com o custo do CloudWatch Database Insights.

Recomendamos que você atualize quaisquer instâncias de banco de dados usando o nível pago do Insights de Performance para o modo avançado do Database Insights antes de 30 de novembro de 2025. Para ter informações sobre como atualizar para o modo avançado do Database Insights, consulte Ativação do modo Avançado do Database Insights para Amazon RDS.

Se você não tomar nenhuma medida, as instâncias de banco de dados que estão usando o Insights de Performance usarão como padrão o modo padrão do Database Insights. Com o modo padrão do Database Insights, você pode perder o acesso ao histórico de dados de desempenho após sete dias e talvez não consiga usar planos de execução e recursos de análise sob demanda no console do Amazon RDS. Após 30 de novembro de 2025, somente o modo avançado do Database Insights permitirá planos de execução e análise sob demanda.

Com o CloudWatch Database Insights, você pode monitorar a carga do banco de dados para sua frota de bancos de dados e analisar e solucionar problemas de desempenho em grande escala. Para ter mais informações sobre o Database Insights, consulte Monitoramento de bancos de dados do Amazon RDS com o CloudWatch Database Insights. Para obter informações de definição de preço, consulte Definição de preço do Amazon CloudWatch.

Na nova visualização de monitoramento, é possível criar um painel personalizado com as métricas necessárias para atender aos seus requisitos de análise.

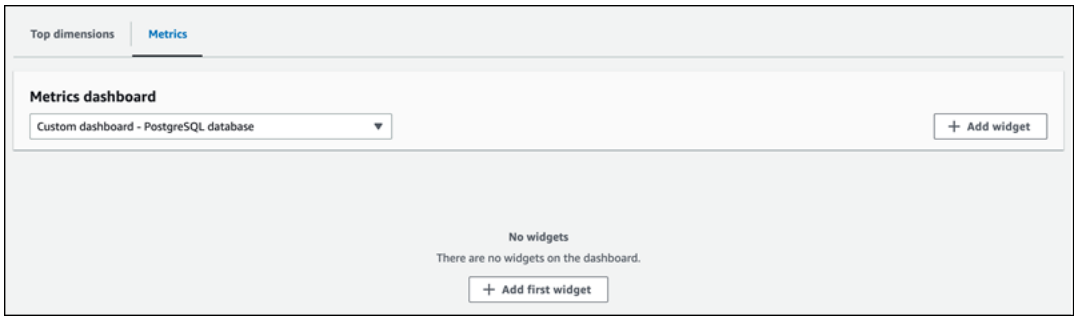
É possível criar um painel personalizado selecionando métricas do Insights de Performance e do CloudWatch para sua instância de banco de dados. É possível usar esse painel personalizado para outras instâncias de banco de dados do mesmo tipo de mecanismo de banco de dados em sua conta da AWS.

> **Note**
>
> O painel personalizado permite até 50 métricas.

Use o menu de configurações do widget para editar ou excluir o painel e mover ou redimensionar a janela do widget.

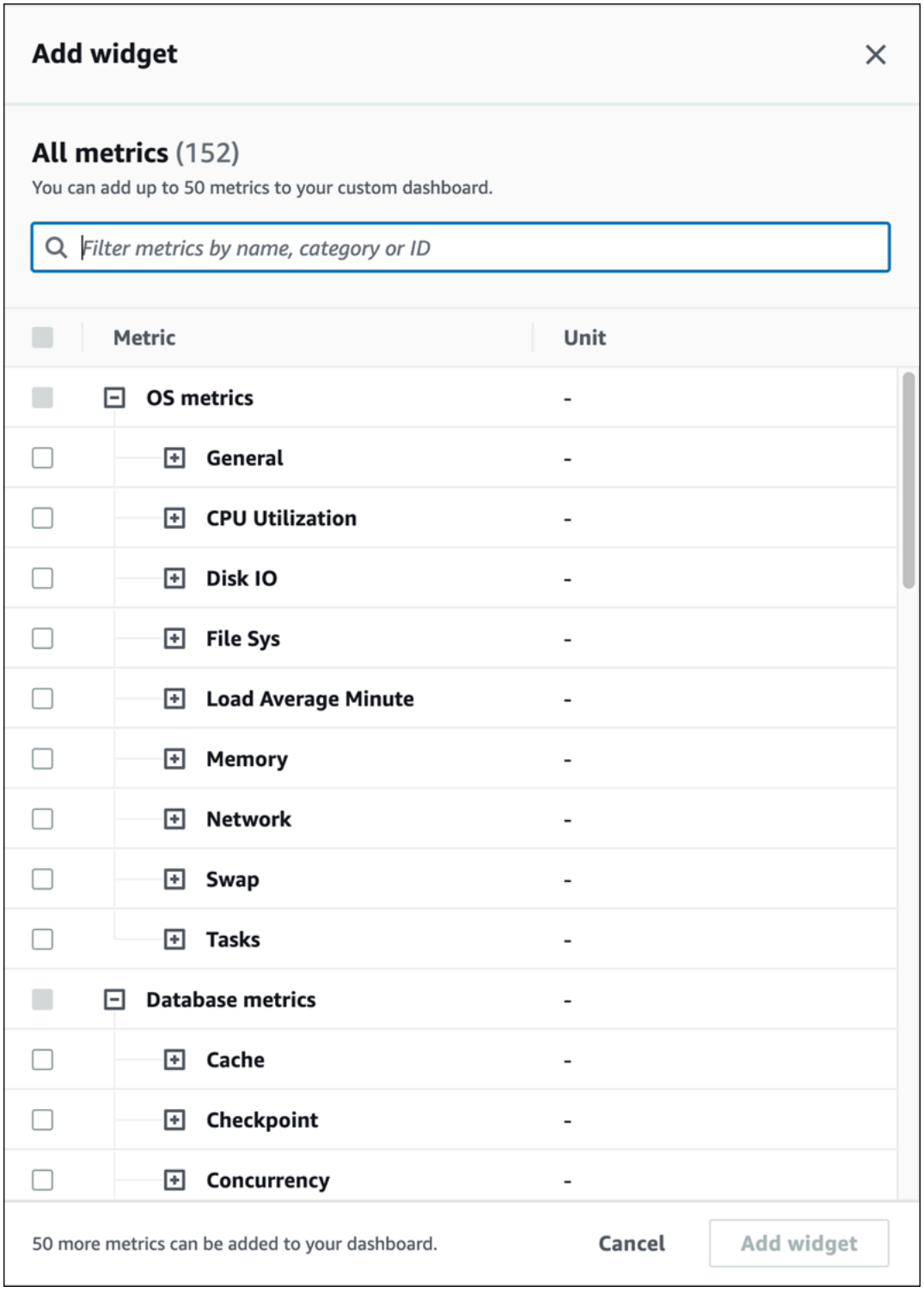
Como criar um painel personalizado com o Insights de Performance no painel de navegação

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Performance Insights.
3. Escolha uma instância de banco de dados.
4. Role para baixo até a guia Métricas na janela.
5. Selecione o painel personalizado na lista suspensa. O exemplo a seguir mostra a criação do painel personalizado.



6. Escolha Adicionar widget para abrir a janela Adicionar widget. É possível abrir e visualizar as métricas do sistema operacional (SO) disponíveis, as métricas do banco de dados e as métricas do CloudWatch na janela.

   O exemplo a seguir mostra a janela Adicionar widget com as métricas.



7. Selecione as métricas que você deseja visualizar no painel e escolha Adicionar widget. É possível usar o campo de pesquisa para encontrar uma métrica específica.

   As métricas selecionadas aparecem no painel.

8. (Opcional) Se você quiser modificar ou excluir o painel, escolha o ícone de configurações no canto superior direito do widget e selecione uma das ações a seguir no menu.

   - Editar modifique a lista de métricas na janela. Escolha Atualizar widget depois de selecionar as métricas para o painel.
   - Excluir: exclui o widget. Selecione Excluir na janela de confirmação.

Escolher o painel predefinido com o Insights de Performance

Important

A AWS anunciou a data de fim da vida útil do Insights de Performance: 30 de novembro de 2025. Após essa data, o Amazon RDS não oferecerá mais suporte à experiência do console, aos períodos de retenção flexíveis (de 1 a 24 meses) e aos preços associados do Insights de Performance. A API do Insights de Performance continuará existindo sem alterações de preço. Os custos da API do Insights de Performance aparecerão em sua fatura da com o custo do CloudWatch Database Insights.
Recomendamos que você atualize quaisquer instâncias de banco de dados usando o nível pago do Insights de Performance para o modo avançado do Database Insights antes de 30 de novembro de 2025. Para ter informações sobre como atualizar para o modo avançado do Database Insights, consulte Ativação do modo Avançado do Database Insights para Amazon RDS.
Se você não tomar nenhuma medida, as instâncias de banco de dados que estão usando o Insights de Performance usarão como padrão o modo padrão do Database Insights. Com o modo padrão do Database Insights, você pode perder o acesso ao histórico de dados de desempenho após sete dias e talvez não consiga usar planos de execução e recursos de análise sob demanda no console do Amazon RDS. Após 30 de novembro de 2025, somente o modo avançado do Database Insights permitirá planos de execução e análise sob demanda.
Com o CloudWatch Database Insights, você pode monitorar a carga do banco de dados para sua frota de bancos de dados e analisar e solucionar problemas de desempenho em grande escala. Para ter mais informações sobre o Database Insights, consulte Monitoramento de bancos de dados do Amazon RDS com o CloudWatch Database Insights. Para obter informações de definição de preço, consulte Definição de preço do Amazon CloudWatch.

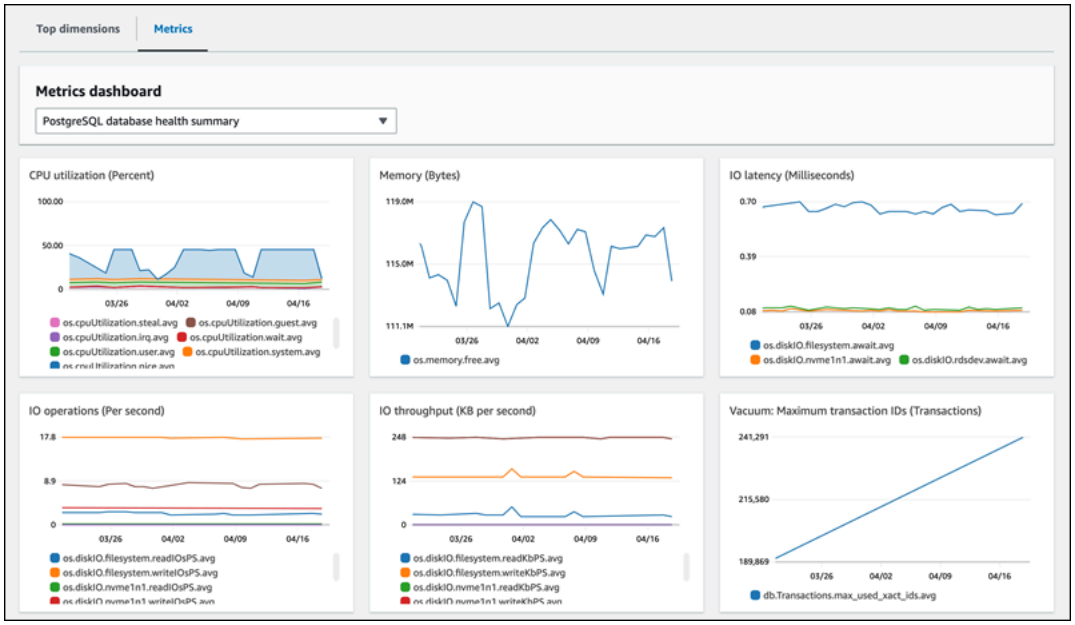
É possível visualizar as métricas mais usadas com o painel pré-configurado. Esse painel ajuda a diagnosticar problemas de performance com um mecanismo de banco de dados e a reduzir o tempo médio de recuperação de horas para minutos.

> **Note**
>
> Esse painel não pode ser editado.

Como escolher o painel predefinido com o Insights de Performance no painel de navegação

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Performance Insights.
3. Escolha uma instância de banco de dados.
4. Role para baixo até a guia Métricas na janela
5. Selecione um painel pré-configurado na lista suspensa.

   É possível visualizar as métricas da instância de banco de dados no painel. O exemplo a seguir mostra um painel de métricas pré-configurado.



Monitorar métricas do Amazon RDS com o Amazon CloudWatch

O Amazon CloudWatch é um repositório de métricas. O repositório coleta e processa dados brutos do Amazon RDS como métricas legíveis, quase em tempo real. Para obter uma lista completa de métricas do Amazon RDS enviadas ao CloudWatch, consulte Referência de métricas para o Amazon RDS.

Para analisar e solucionar problemas de performance de banco de dados em grande escala, use CloudWatch Database Insights.

Tópicos

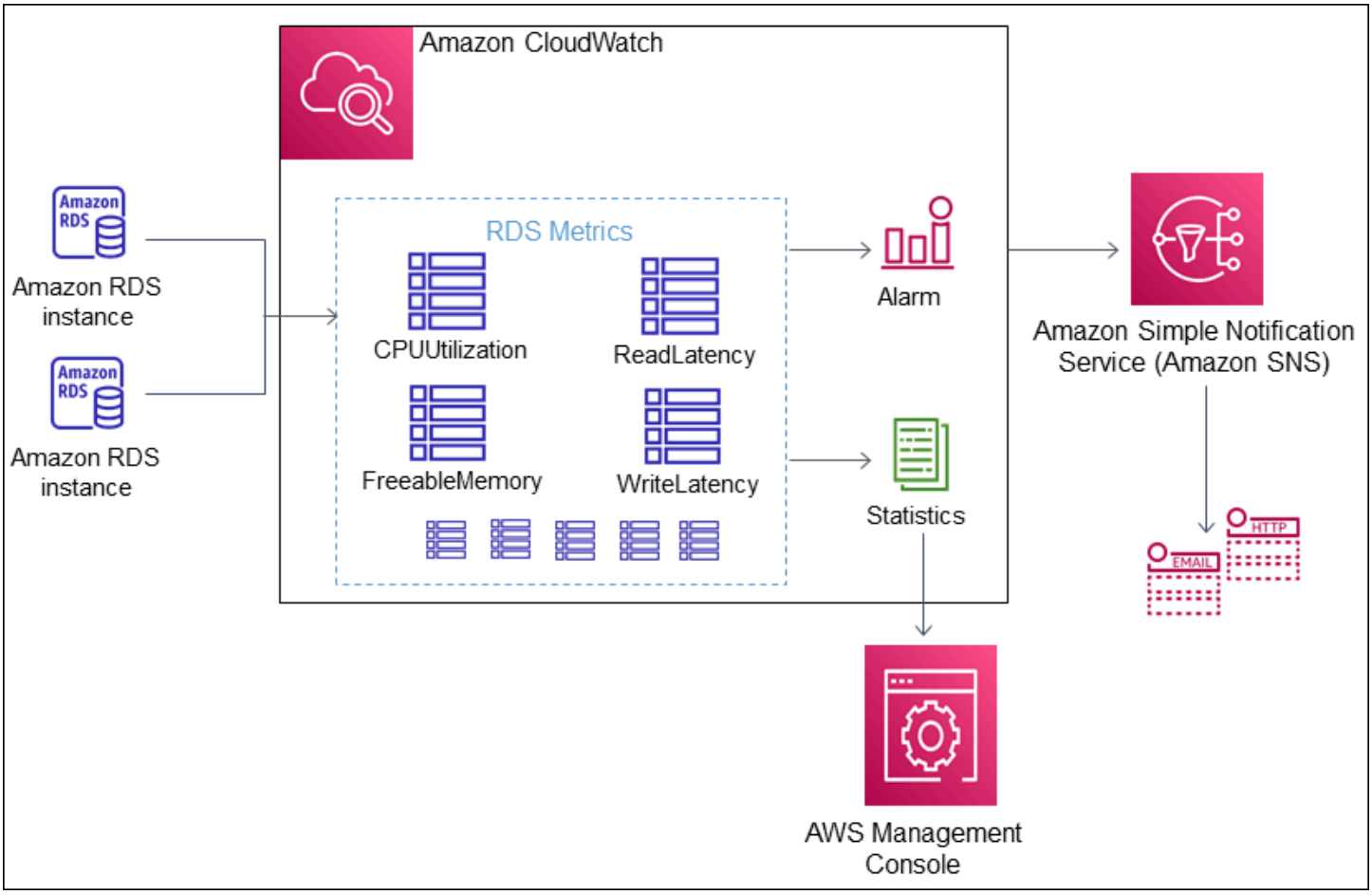
- Visão geral do Amazon RDS e do Amazon CloudWatch
- Visualizar métricas de de instância de banco de dados no console do CloudWatch e na AWS CLI.
- Exportar as métricas do Performance Insights para o CloudWatch
- Criação de alarmes CloudWatch para monitorar Amazon RDS
- Tutorial: criar um alarme do Amazon CloudWatch para atraso de réplica do cluster de banco de dados multi-AZ para o Amazon RDS

Visão geral do Amazon RDS e do Amazon CloudWatch

Por padrão, o Amazon RDS envia dados de métrica automaticamente para o CloudWatch em períodos de um minuto. Por exemplo, a métrica `CPUUtilization` registra a porcentagem de utilização da CPU para uma instância de banco de dados ao longo do tempo. Pontos de dados com um período de 60 segundos (1 minuto) ficam disponíveis por 15 dias. Isso significa que você pode acessar informações históricas e ter uma perspectiva melhor sobre a performance da aplicação ou do serviço web.

Agora você pode exportar painéis de métricas do Performance Insights do Amazon RDS para o Amazon CloudWatch. Você pode exportar os painéis de métricas pré-configurados ou personalizados como um novo painel ou adicioná-los a um painel existente do CloudWatch. O painel exportado está disponível para visualização no console do CloudWatch. Para obter mais informações sobre como exportar os painéis de métricas do Performance Insights para o CloudWatch, consulte. Exportar as métricas do Performance Insights para o CloudWatch

Conforme mostrado no diagrama a seguir, você pode configurar alarmes para suas métricas do CloudWatch. Por exemplo, é possível criar um alarme que sinalize quando a utilização da CPU para uma instância é superior a 70%. Você pode configurar o Amazon Simple Notification Service para enviar um e-mail quando o limite for ultrapassado.



O Amazon RDS publica os seguintes tipos de métrica no Amazon CloudWatch:

- Métricas para suas instâncias de banco de dados do RDS

  Para obter uma tabela dessas métricas, consulte Métricas do Amazon CloudWatch para o Amazon RDS.
- Métricas do Performance Insights

  Para obter uma tabela dessas métricas, consulte Métricas do Amazon CloudWatch para Insights de Performance do Amazon RDS e Métricas de contadores do Performance Insights.
- Métricas de monitoramento aprimoradas (publicadas no Amazon CloudWatch Logs)

  Para obter uma tabela dessas métricas, consulte Métricas do sistema operacional em monitoramento avançado.
- Métricas de uso para as cotas de serviço do Amazon RDS em sua Conta da AWS

Para obter uma tabela dessas métricas, consulte Métricas de uso do Amazon CloudWatch para Amazon RDS. Para obter mais informações sobre cotas do Amazon RDS, consulte Cotas e restrições do Amazon RDS.

Para obter mais informações sobre o CloudWatch, consulte O que é o Amazon CloudWatch? no Guia do usuário do Amazon CloudWatch. Para obter mais informações sobre a retenção de métricas do CloudWatch, consulte Retenção de métricas.

Visualizar métricas de de instância de banco de dados no console do CloudWatch e na AWS CLI.

Veja a seguir detalhes sobre como visualizar métricas de sua instância de banco de dados usando o CloudWatch. Para obter informações sobre o monitoramento de métricas para o sistema operacional da sua instância de banco de dados em tempo real usando o CloudWatch Logs, consulte Monitorar métricas do SO com o monitoramento avançado.

Ao usar recursos do Amazon RDS, o Amazon RDS envia métricas e dimensões ao Amazon CloudWatch a cada minuto.

Agora você pode exportar painéis de métricas do Performance Insights do Amazon RDS para o Amazon CloudWatch e visualizar essas métricas no console do CloudWatch. Para obter mais informações sobre como exportar os painéis de métricas do Performance Insights para o CloudWatch, consulte. Exportar as métricas do Performance Insights para o CloudWatch

É possível usar os procedimentos a seguir para visualizar as métricas do Amazon RDS no console do CloudWatch e na CLI.

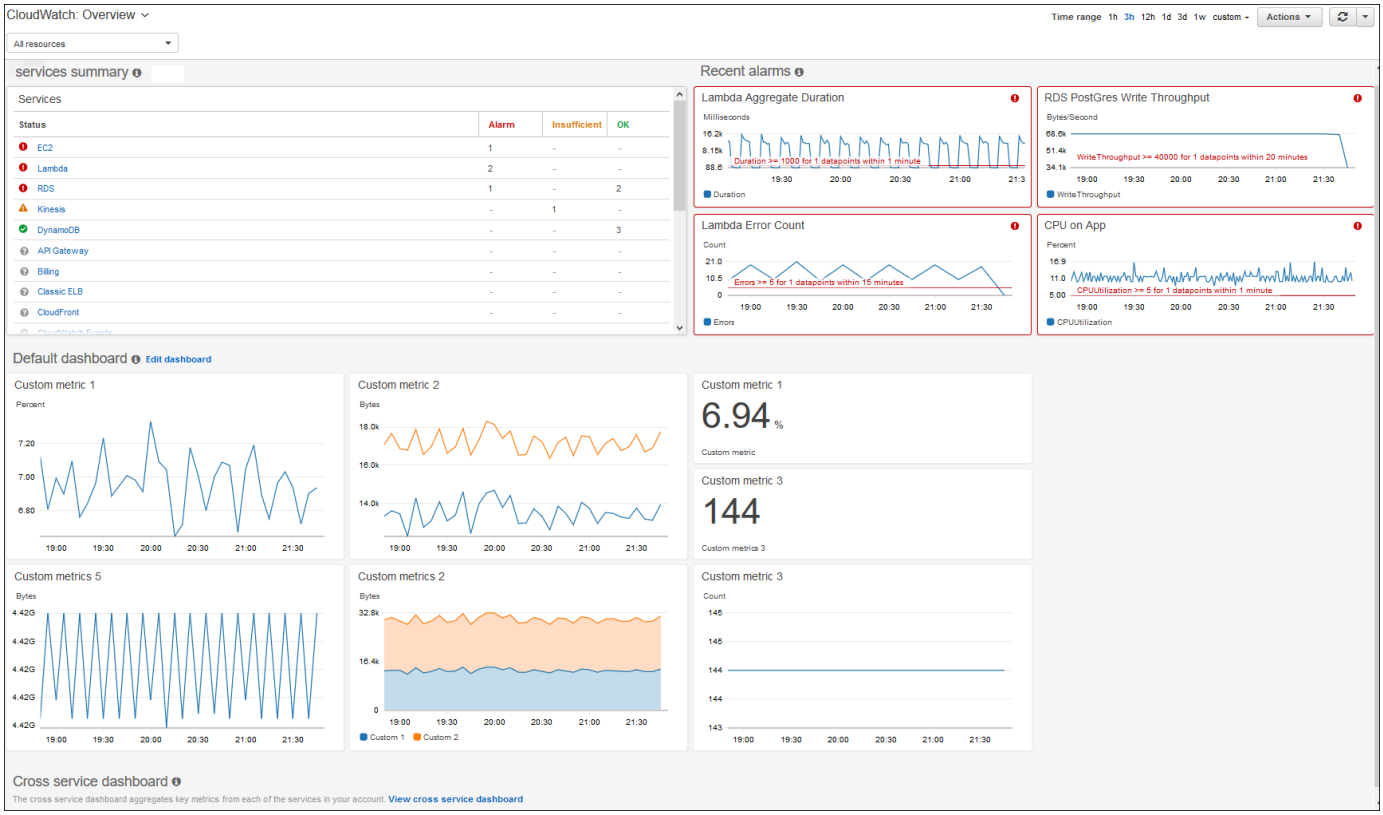
Console

Para visualizar as métricas usando o console do Amazon CloudWatch

As métricas são agrupadas primeiro pelo namespace do serviço e, em seguida, por várias combinações de dimensão dentro de cada namespace.

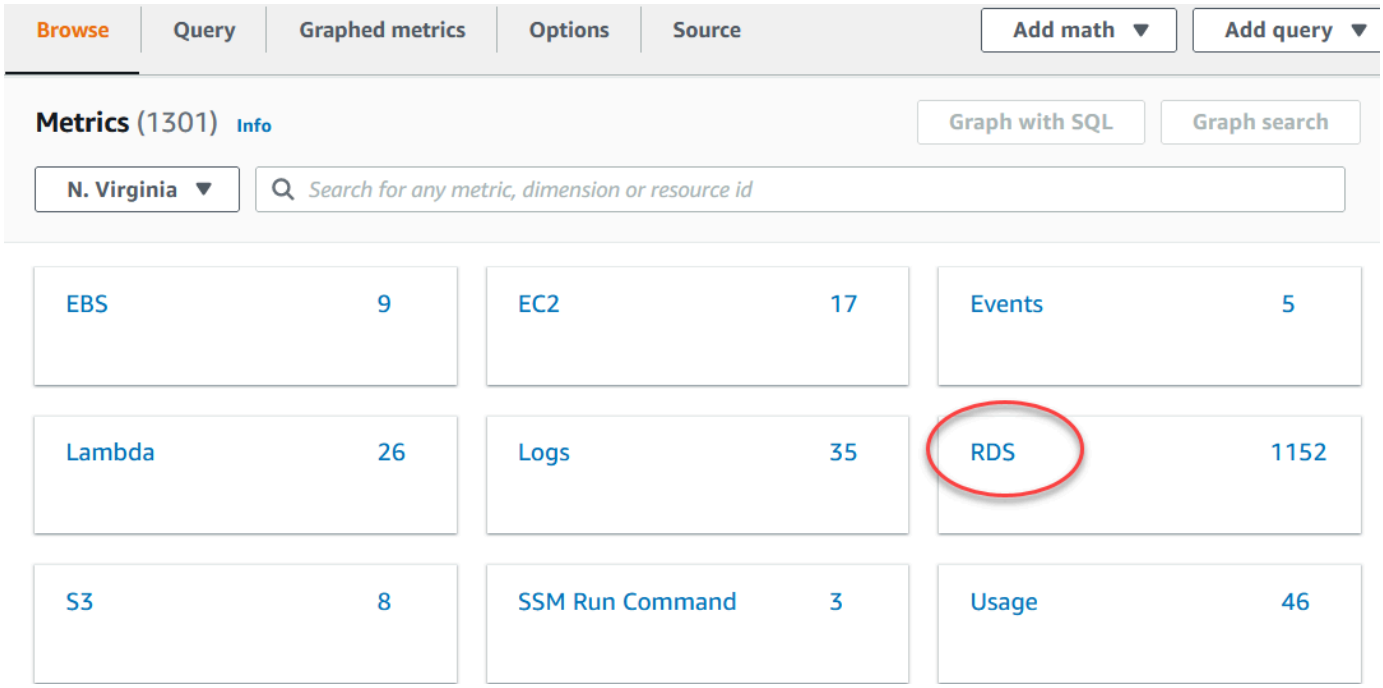
1. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/.

   A página inicial de visão geral do CloudWatch é exibida.



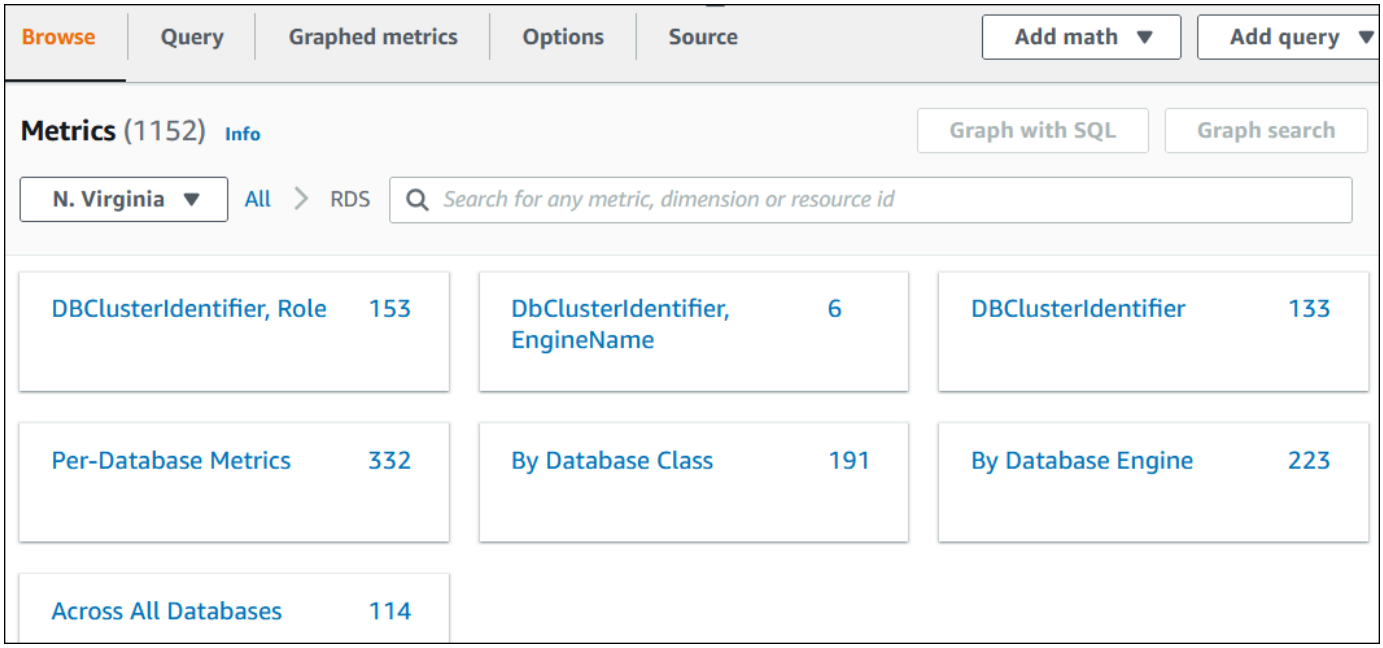
2. Se necessário, altere a Região da AWS. Na barra de navegação, escolha a Região da AWS na qual seus recursos da AWS estão localizados. Para obter mais informações, consulte Regiões e endpoints da .

3. No painel de navegação, escolha Metrics (Métricas) e, em seguida, All metrics (Todas as métricas).

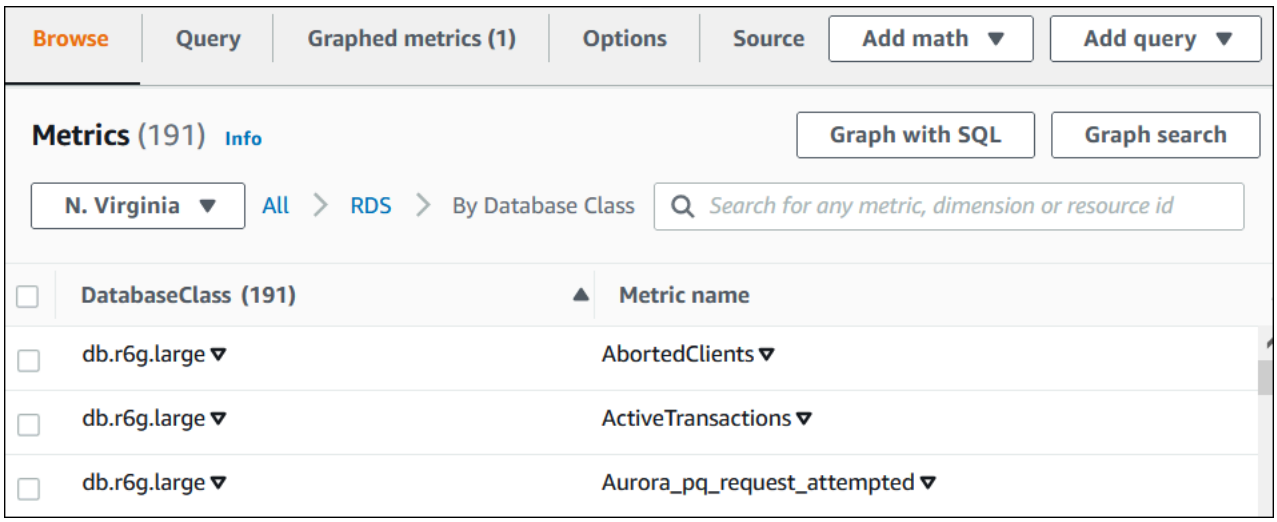


4. Role para baixo e escolha o namespace de métrica RDS.

   A página exibe as dimensões do Amazon RDS. Para obter descrições dessas dimensões, consulte Dimensões do Amazon CloudWatch para o Amazon RDS.



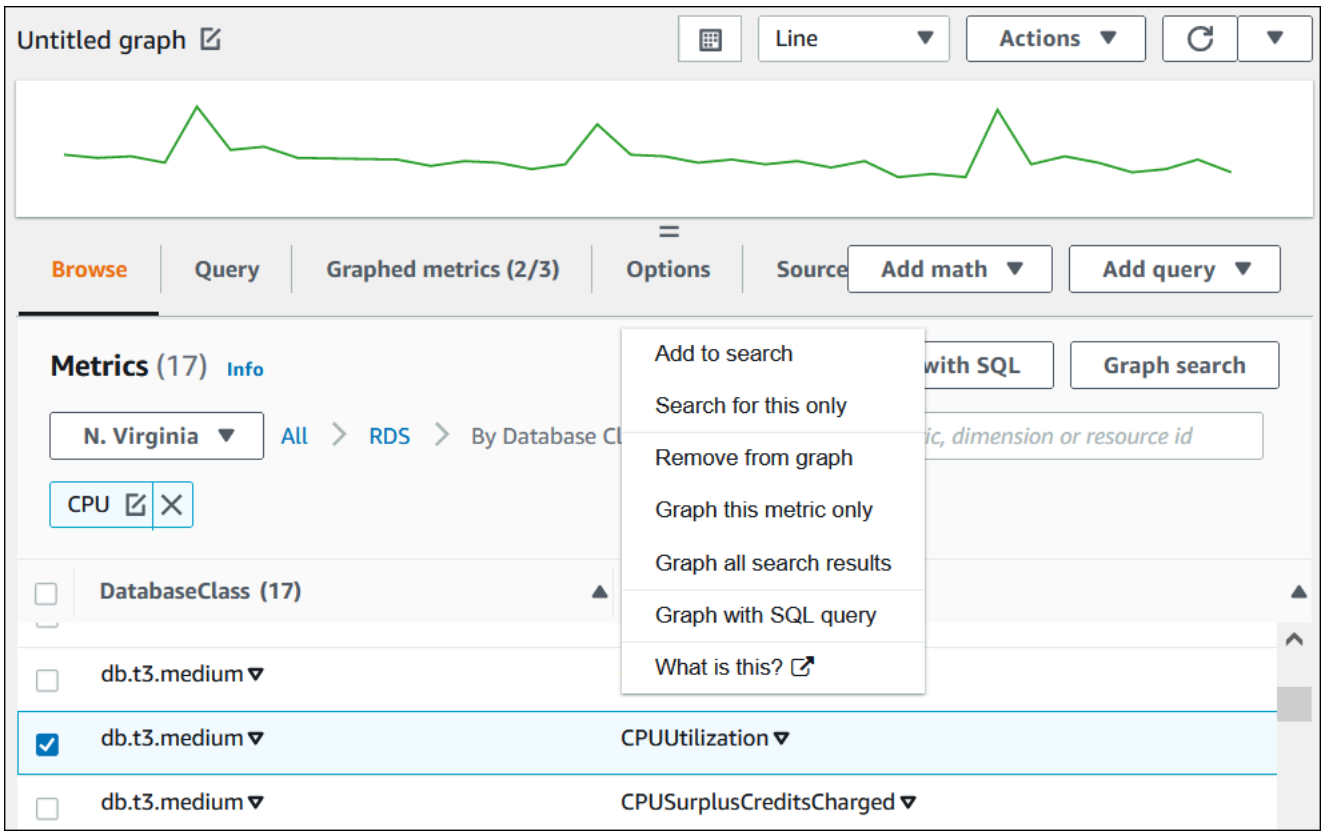
5. Escolha uma dimensão de métrica; por exemplo, By Database Class (Por classe de banco de dados).



6. Realize uma destas ações:

   - Para classificar a métrica, use o cabeçalho da coluna.
   - Para criar um gráfico de uma métrica, marque a caixa de seleção ao lado da métrica.
   - Para filtrar por recurso, escolha o ID do recurso e Add to search (Adicionar à pesquisa).
   - Para filtrar por métrica, escolha o nome da métrica e Add to search (Adicionar à pesquisa).

   O exemplo a seguir filtra com base na classe db.t3.medium e faz um gráfico da métrica CPUUtilization.



AWS CLI

Para obter informações sobre métricas usando a AWS CLI, use o comando `list-metrics` do CloudWatch. No exemplo a seguir, você lista todas as métricas no namespace `AWS/RDS`.

```
aws cloudwatch list-metrics --namespace AWS/RDS
```

Para receber os dados de métricas, use o comando `get-metric-data`.

O exemplo a seguir obtém estatísticas `CPUUtilization` para a instância `my-instance` durante o período específico de 24 horas, com granularidade de 5 minutos.

Crie um arquivo JSON `CPU_metric.json` com o conteúdo apresentado a seguir.

```
{
   "StartTime" : "2023-12-25T00:00:00Z",
   "EndTime" : "2023-12-26T00:00:00Z",
   "MetricDataQueries" : [{
     "Id" : "cpu",
     "MetricStat" : {
```

```
    "Metric" : {
        "Namespace" : "AWS/RDS",
        "MetricName" : "CPUUtilization",
        "Dimensions" : [{ "Name" : "DBInstanceIdentifier" , "Value" : my-instance}]
    },
       "Period" : 360,
       "Stat" : "Minimum"
     }
   }]
}
```

Example

Para Linux, macOS ou Unix:

```
aws cloudwatch get-metric-data \
    --cli-input-json file://CPU_metric.json
```

Para Windows:

```
aws cloudwatch get-metric-data ^
     --cli-input-json file://CPU_metric.json
```

A saída da amostra é exibida da seguinte maneira:

```
{
    "MetricDataResults": [
        {
            "Id": "cpu",
            "Label": "CPUUtilization",
            "Timestamps": [
                "2023-12-15T23:48:00+00:00",
                "2023-12-15T23:42:00+00:00",
                "2023-12-15T23:30:00+00:00",
                "2023-12-15T23:24:00+00:00",
                ...
            ],
            "Values": [
                13.299778337027714,
                13.677507543049558,
                14.24976250395827,
                13.02521708695145,
```

```
                ...
            ],
            "StatusCode": "Complete"
        }
    ],
    "Messages": []
}
```

Para obter mais informações, consulte Obter as estatísticas de uma métrica no Guia do usuário do Amazon CloudWatch.

Exportar as métricas do Performance Insights para o CloudWatch

O Performance Insights permite que você exporte o painel de métricas pré-configurado ou personalizado da sua instância de banco de dados para o Amazon CloudWatch. Você pode exportar o painel de métricas como um novo painel ou adicioná-lo a um painel existente do CloudWatch. Ao optar por adicionar o painel a um painel existente do CloudWatch, você pode criar um rótulo de cabeçalho para que as métricas apareçam em uma seção separada no painel do CloudWatch.

Você pode visualizar as métricas exportadas no console do CloudWatch. Se você adicionar novas métricas a um painel de métricas do Performance Insights depois de exportá-lo, deverá exportá-lo novamente para visualizar as novas métricas no console do CloudWatch.

Você também pode selecionar um widget de métrica no painel do Performance Insights e visualizar os dados de métricas no console do CloudWatch.

Para obter mais informações sobre como exibir métricas de replicação no console do CloudWatch, consulte Visualizar métricas de de instância de banco de dados no console do CloudWatch e na AWS CLI..

Nas seções a seguir, exporte as métricas do Insights de Performance para o CloudWatch como um painel novo ou existente e visualize as métricas do Insights de Performance no CloudWatch.

Tópicos

- Exportar métricas do Performance Insights como um novo painel para o CloudWatch
- Adicionar as métricas do Performance Insights a um painel existente do CloudWatch
- Visualizar um widget de métrica do Performance Insights no CloudWatch

Exportar métricas do Performance Insights como um novo painel para o CloudWatch

Escolha um painel de métricas pré-configurado ou personalizado no painel do Performance Insights e exporte-o como um novo painel no CloudWatch. Você pode visualizar o painel exportado no console do CloudWatch.

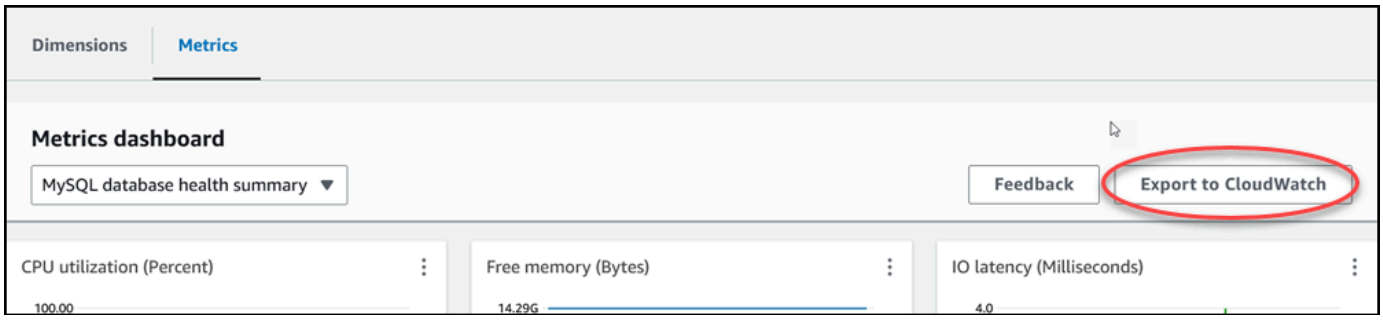
Para exportar métricas do Performance Insights como um novo painel no CloudWatch

1. Abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Performance Insights.
3. Escolha uma instância de banco de dados.

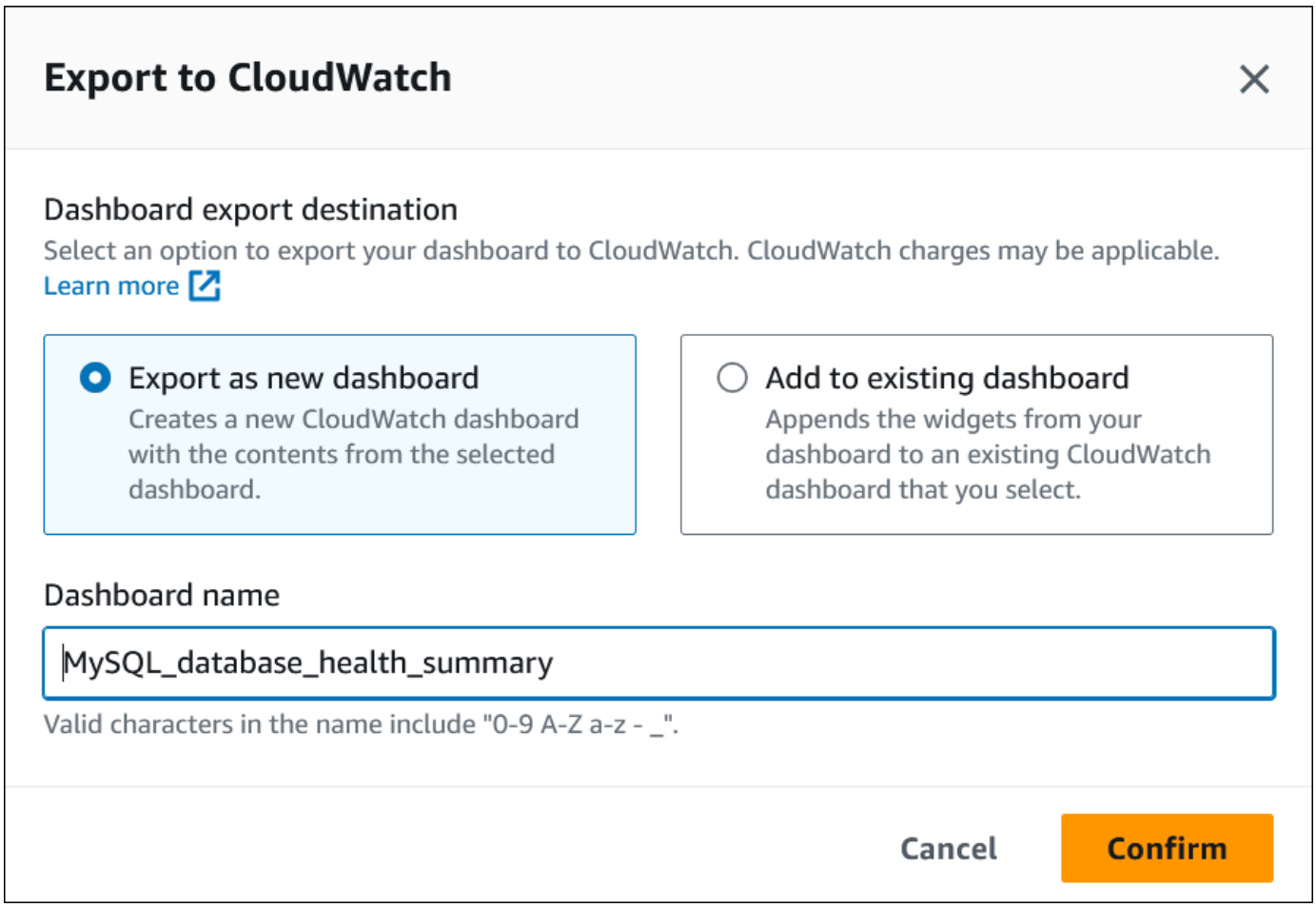
   O painel do Insights de Performance é exibido para a instância de banco de dados.
4. Role para baixo e escolha Métricas.

   Por padrão, é exibido o painel pré-configurado com as métricas do Performance Insights.
5. Escolha um painel pré-configurado ou personalizado e, em seguida, escolha Exportar para o CloudWatch.

   A janela Exportar para o CloudWatch é exibida.

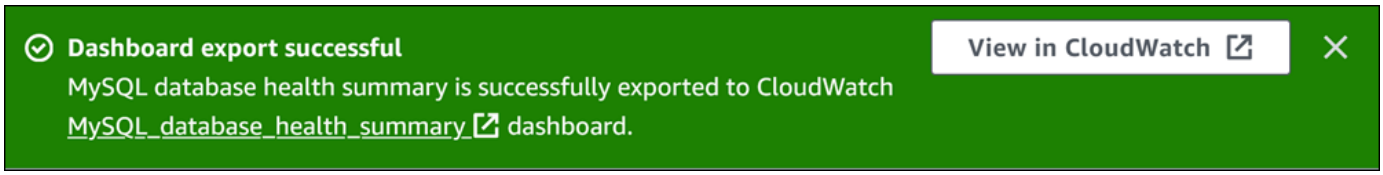


6. Escolha Exportar como novo painel.



7. Insira um nome para o novo painel no campo Nome do painel e escolha Confirmar.

   Um banner exibe uma mensagem após a exportação do painel ser bem-sucedida.



8. Escolha o link ou Exibir no CloudWatch no banner para visualizar o painel de métricas no console do CloudWatch.

Adicionar as métricas do Performance Insights a um painel existente do CloudWatch

Adicione um painel de métricas pré-configurado ou personalizado a um painel existente do CloudWatch. Você pode adicionar um rótulo ao painel de métricas para aparecer em uma seção separada no painel do CloudWatch.

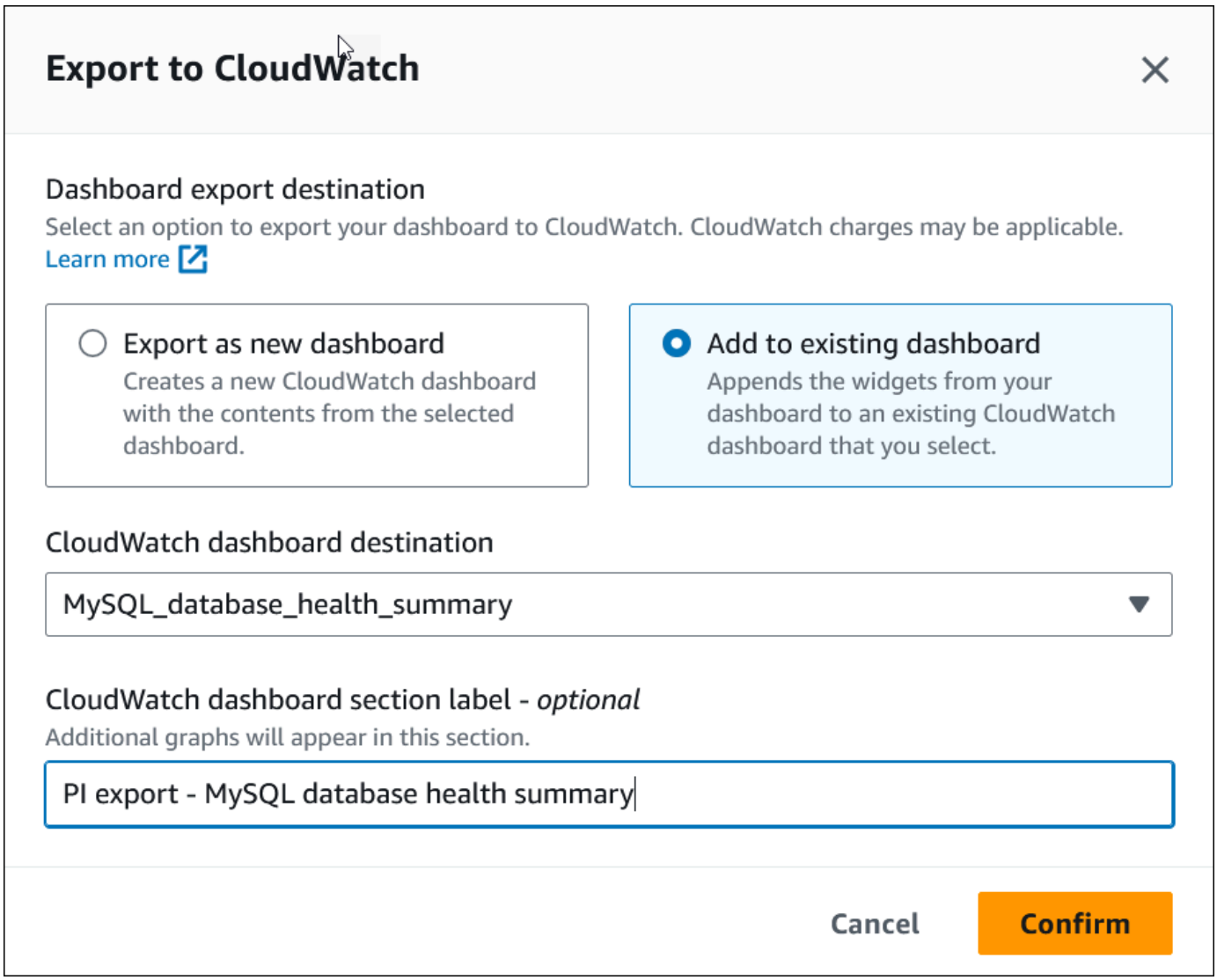
Para exportar as métricas para um painel existente do CloudWatch

1. Abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Performance Insights.
3. Escolha uma instância de banco de dados.

   O painel do Insights de Performance é exibido para a instância de banco de dados.
4. Role para baixo e escolha Métricas.

   Por padrão, é exibido o painel pré-configurado com as métricas do Performance Insights.
5. Escolha o painel pré-configurado ou personalizado e, em seguida, escolha Exportar para o CloudWatch.

   A janela Exportar para o CloudWatch é exibida.
6. Escolha Adicionar ao painel existente.



7. Especifique o destino e o rótulo do painel e escolha Confirmar.

   - Destino do painel do CloudWatch: escolha um painel existente do CloudWatch.
   - Rótulo da seção do painel do CloudWatch - opcional: insira um nome para as métricas do Performance Insights que aparecerão nesta seção no painel do CloudWatch.

   Um banner exibe uma mensagem após a exportação do painel ser bem-sucedida.

8. Escolha o link ou Exibir no CloudWatch no banner para visualizar o painel de métricas no console do CloudWatch.

Visualizar um widget de métrica do Performance Insights no CloudWatch

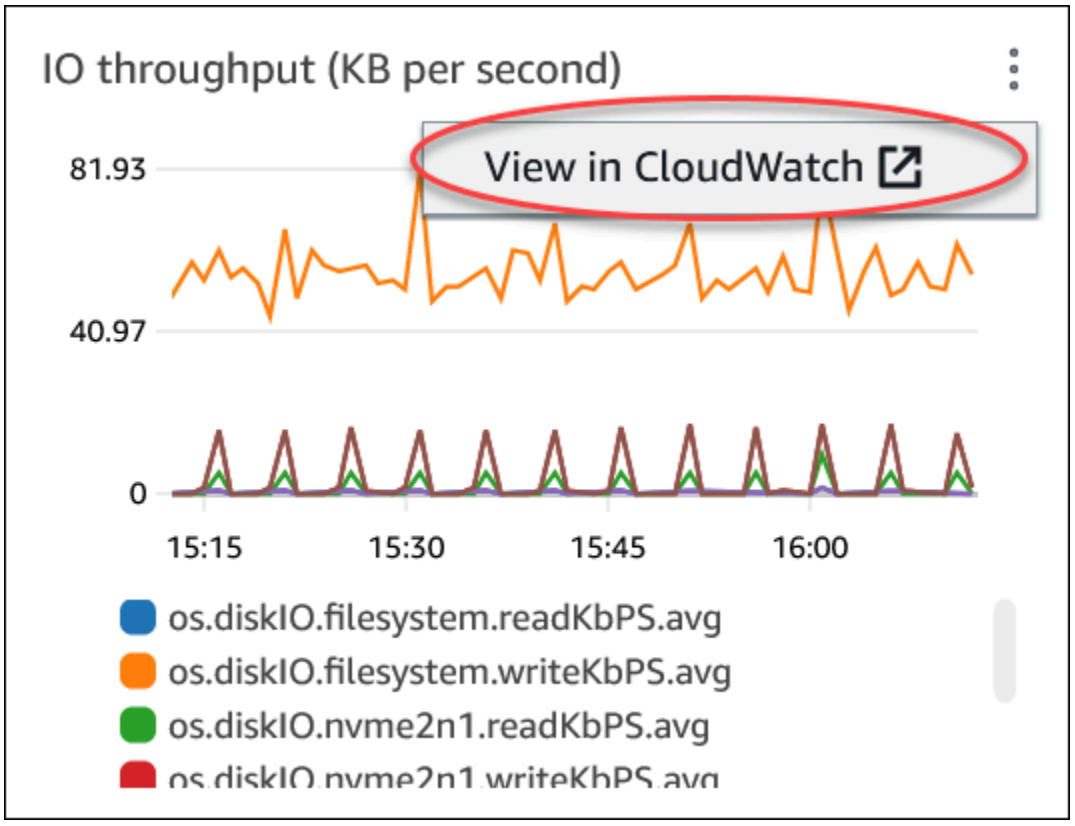
Selecione um widget de métrica no painel do Performance Insights do Amazon RDS e visualize os dados de métricas no console do CloudWatch.

Para exportar um widget de métrica e visualizar os dados das métricas no console do CloudWatch

1. Abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Performance Insights.
3. Escolha uma instância de banco de dados.

   O painel do Insights de Performance é exibido para a instância de banco de dados.
4. Role para baixo até Métricas.

   Por padrão, é exibido o painel pré-configurado com as métricas do Performance Insights.
5. Escolha um widget de métrica e, em seguida, escolha Exibir no CloudWatch no menu.



   Os dados da métrica são exibidos no console do CloudWatch.

Criação de alarmes CloudWatch para monitorar Amazon RDS

Você pode criar um alarme do CloudWatch que envia uma mensagem do Amazon SNS quando o alarme muda de estado. Um alarme observa uma única métrica por um período tempo que você especifica. O alarme também pode realizar uma ou mais ações com base no valor da métrica relativa a um limite especificado durante vários períodos. A ação é uma notificação enviada para um tópico do Amazon SNS ou por uma política do Amazon EC2 Auto Scaling.

Os alertas invocam ações apenas para alterações de estado mantidas. Os alarmes do CloudWatch não invocam ações somente porque estão em um determinado estado. O estado deve ter sido alterado e mantido por um período especificado.

É possível usar a função matemática métrica DB_PERF_INSIGHTS no console do CloudWatch para consultar métricas do contador do Amazon RDS do Insights de Performance. A função DB_PERF_INSIGHTS também inclui a métrica DBLoad em intervalos de menos de um minuto. Também é possível definir alarmes do CloudWatch para essas métricas.

Para obter mais detalhes sobre como criar um alarme, consulte Crie um alarme para as métricas do contador do Performance Insights a partir de um banco de dados da AWS.

Para definir um alarme usando a AWS CLI

- Chame `put-metric-alarm`. Para obter mais informações, consulte Referência de comandos da AWS CLI.

Para definir um alarme usando a API do CloudWatch

- Chame `PutMetricAlarm`. Para obter mais informações, consulte a Referência da API do Amazon CloudWatch.

Para obter mais informações sobre a definição de tópicos do Amazon SNS e a criação de alarmes, consulte Usar alarmes do Amazon CloudWatch.

Tutorial: criar um alarme do Amazon CloudWatch para atraso de réplica do cluster de banco de dados multi-AZ para o Amazon RDS

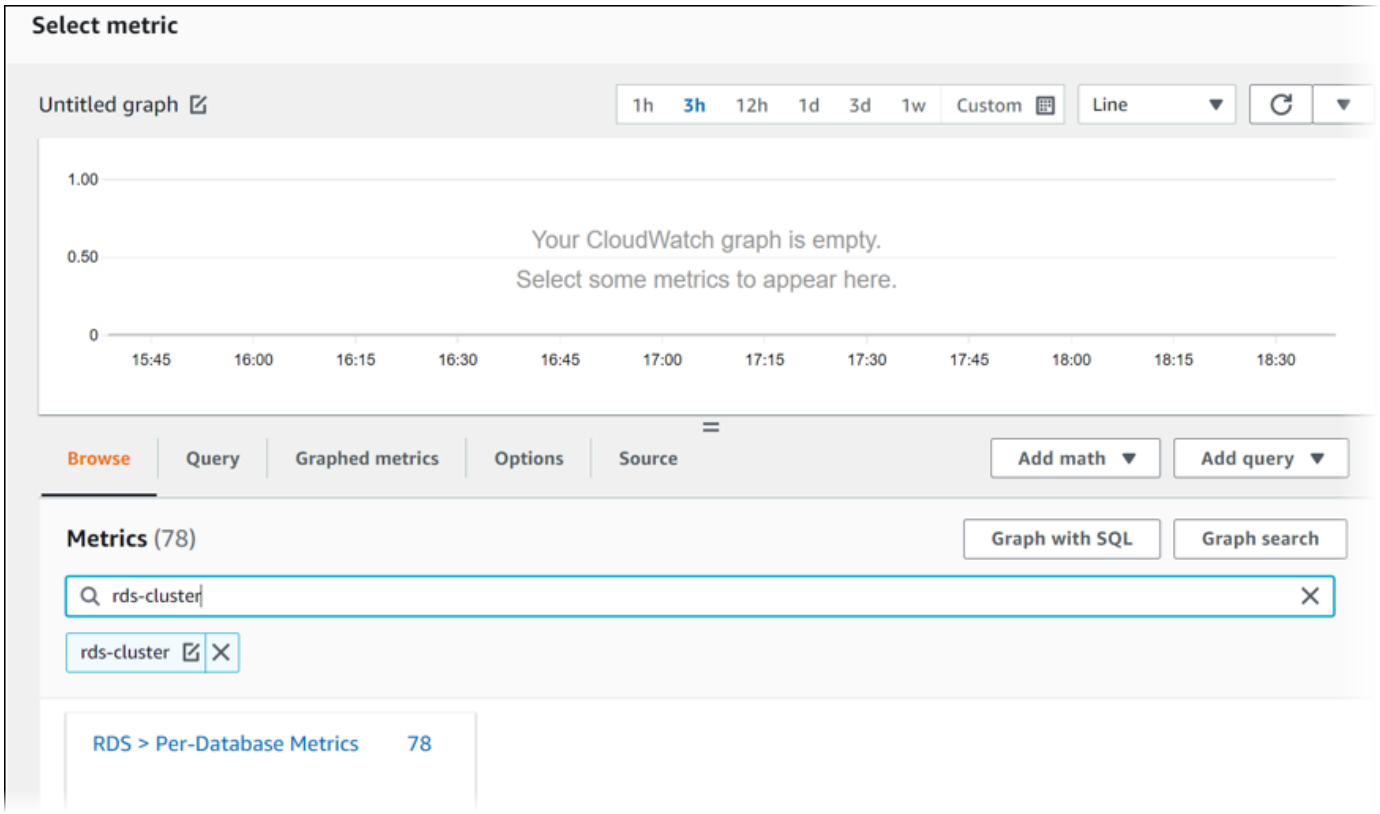
Você pode criar um alarme do Amazon CloudWatch que envia uma mensagem do Amazon SNS quando o atraso de réplica de um cluster de banco de dados multi-AZ excede um limite. Um alarme

observa uma métrica `ReplicaLag` ao longo do período que você especificar. A ação é uma notificação enviada para um tópico do Amazon SNS ou por uma política do Amazon EC2 Auto Scaling.

Para definir um alarme do CloudWatch para o atraso de réplica do cluster de banco de dados multi-AZ

1. Faça login no AWS Management Console e abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Alarms (Alarmes), All alarms (Todos os alarmes).
3. Selecione Create alarm (Criar alarme).
4. Na página Specify metric and conditions (Especificar métrica e condições), selecione Select metric (Selecionar métrica).
5. Na caixa de pesquisa, insira o nome do seu cluster de banco de dados multi-AZ e pressione Enter.

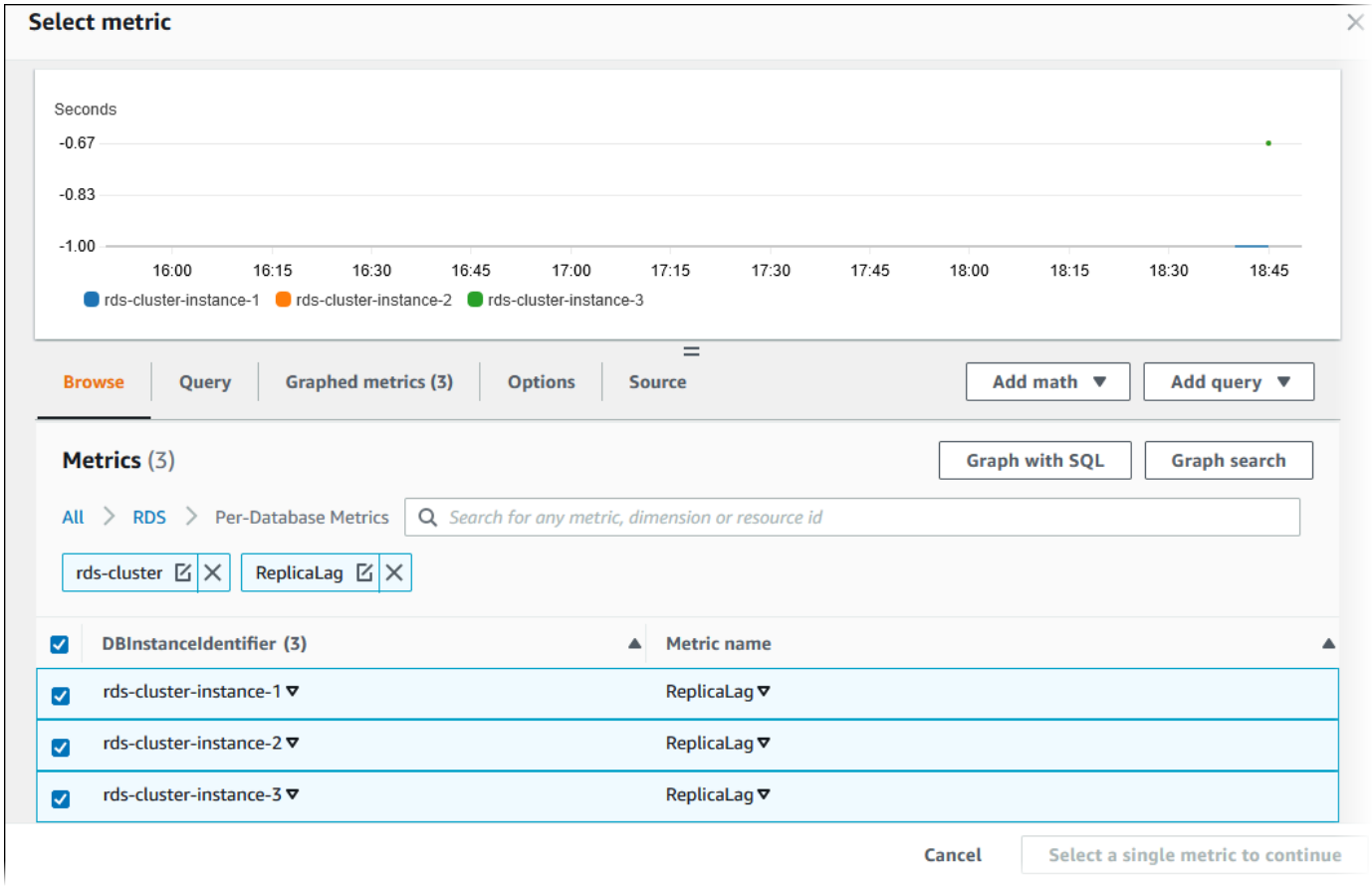
   A imagem a seguir mostra a página Select metric (Selecionar métrica) com um cluster de banco de dados multi-AZ chamado `rds-cluster` inserido.



6. Escolha RDS, Per-Database Metrics (Métricas por banco de dados).

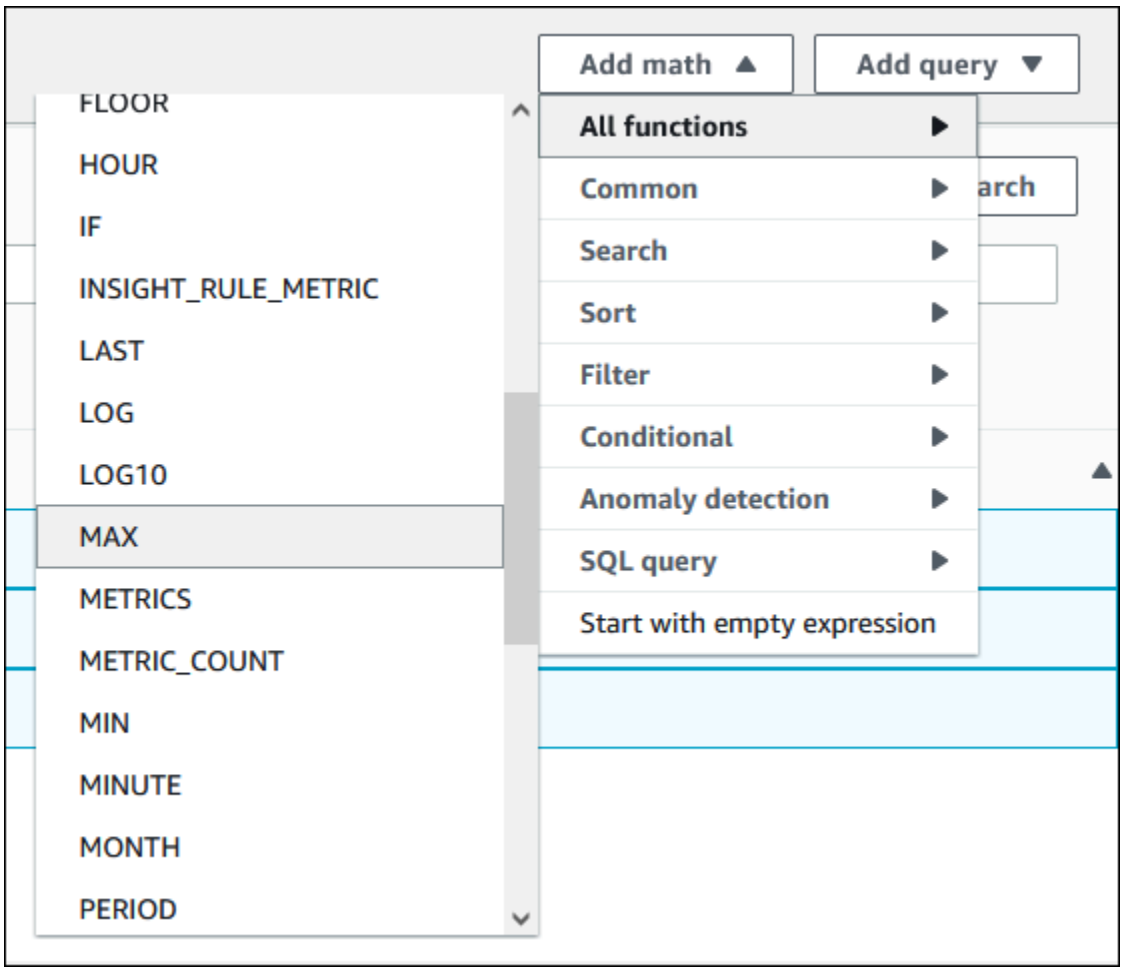
7. Na caixa de pesquisa, insira **`ReplicaLag`** e pressione Enter. Depois selecione cada instância de banco de dados no cluster de banco de dados.

   A imagem a seguir mostra a página Select metric (Selecionar métrica) com as instâncias de banco de dados selecionadas referentes à métrica ReplicaLag.



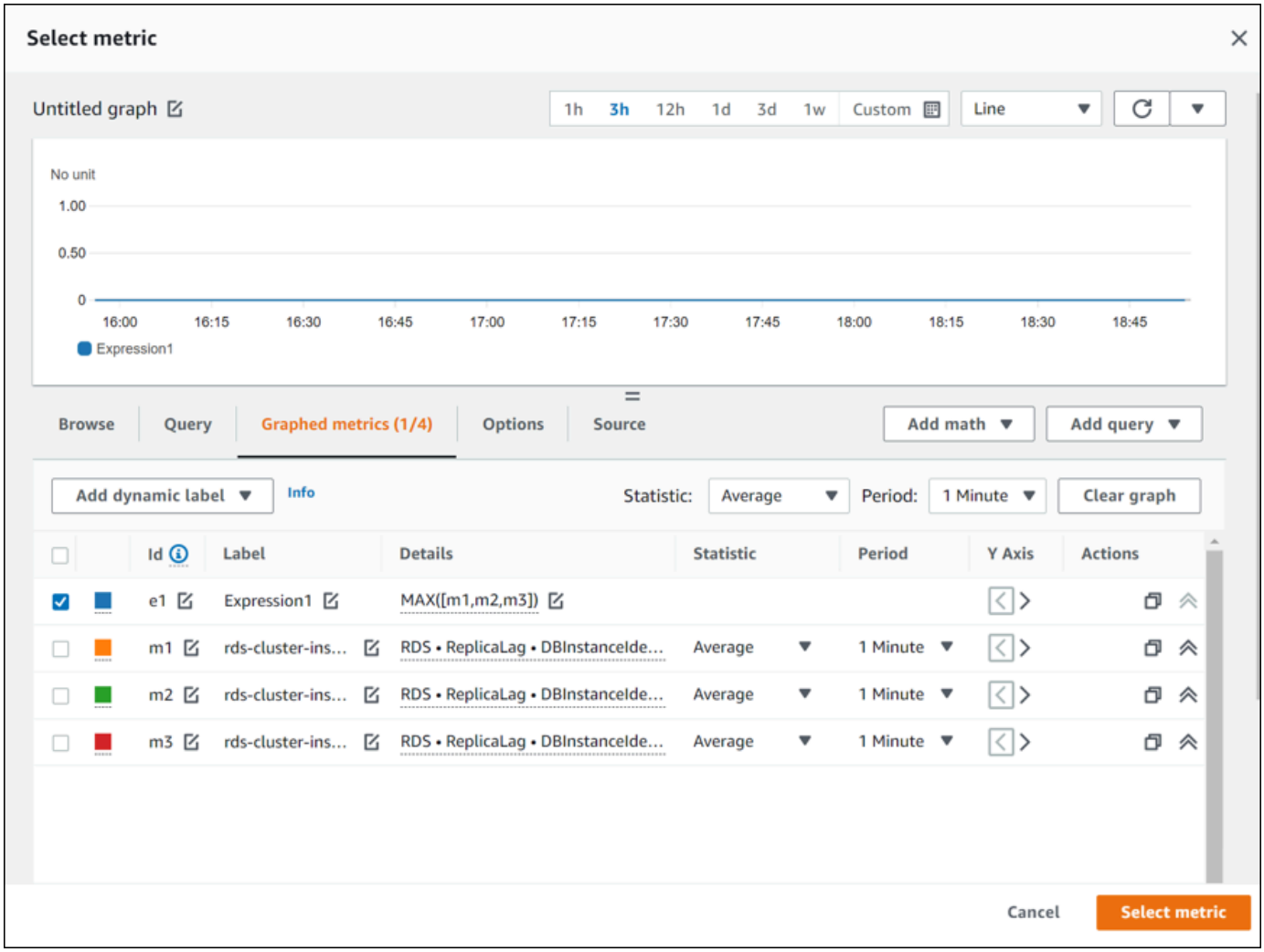
   Este alarme considera o atraso de réplica de todas as três instâncias de banco de dados no cluster de banco de dados multi-AZ. O alarme responde quando qualquer instância de banco de dados excede o limite. Ele usa uma expressão matemática que retorna o valor máximo das três métricas. Comece classificando por nome da métrica e, em seguida, escolha todas as três métricas ReplicaLag.

8. Em Add math (Adicionar matemática), escolha All functions (Todas as funções), MAX.



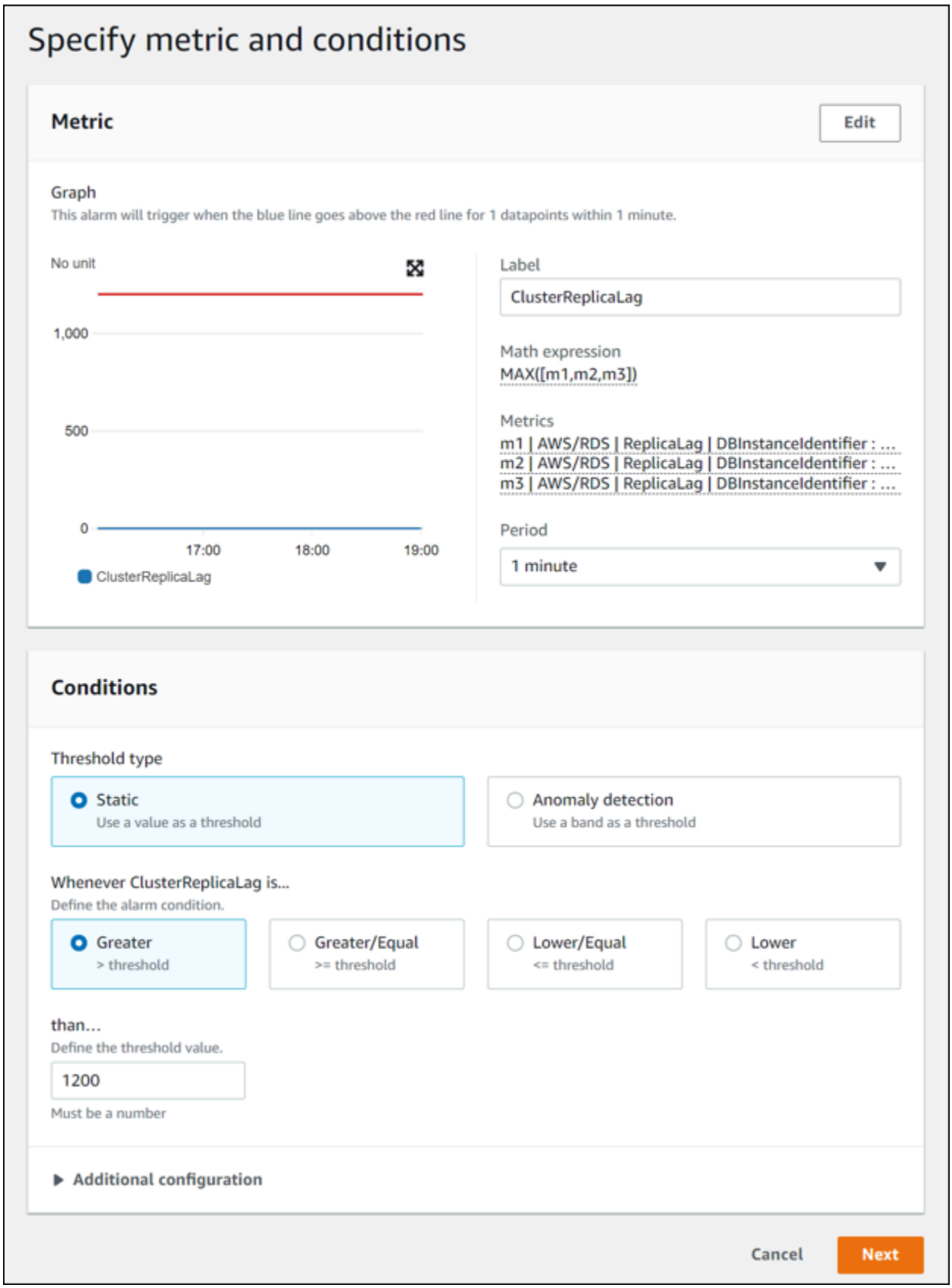
9. Escolha a guia Graphed metrics (Métricas em gráficos) e edite os detalhes de Expression1 para **`MAX([m1,m2,m3])`**.
10. Para todas as três métricas ReplicaLag, altere Period (Período) para 1 minute (1 minuto).
11. Limpe a seleção de todas as métricas, exceto de Expression1.

    A página Select metric (Selecionar métrica) deve ser semelhante à imagem a seguir.



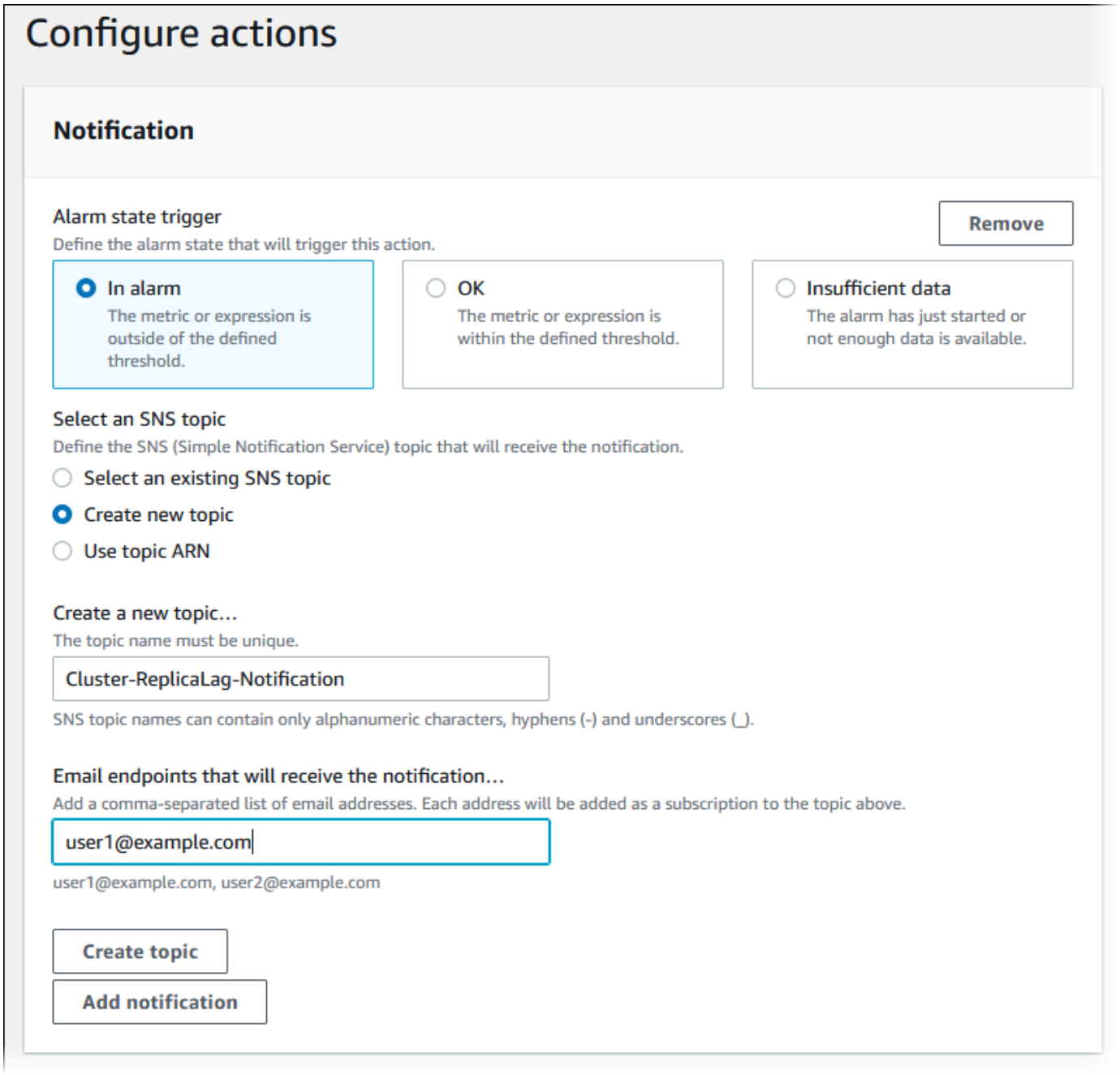
12. Escolha Selecionar métrica.
13. Na página Specify metric and conditions, altere o rótulo para um nome significativo, como **`ClusterReplicaLag`**, e insira um número de segundos em Define the threshold value (Definir o valor limite). Para este tutorial, insira **`1200`** segundos (20 minutos). Você pode ajustar esse valor para os requisitos de sua workload.

    A página Specify metric and conditions (Especificar métrica e condições) deve ser semelhante à imagem a seguir.

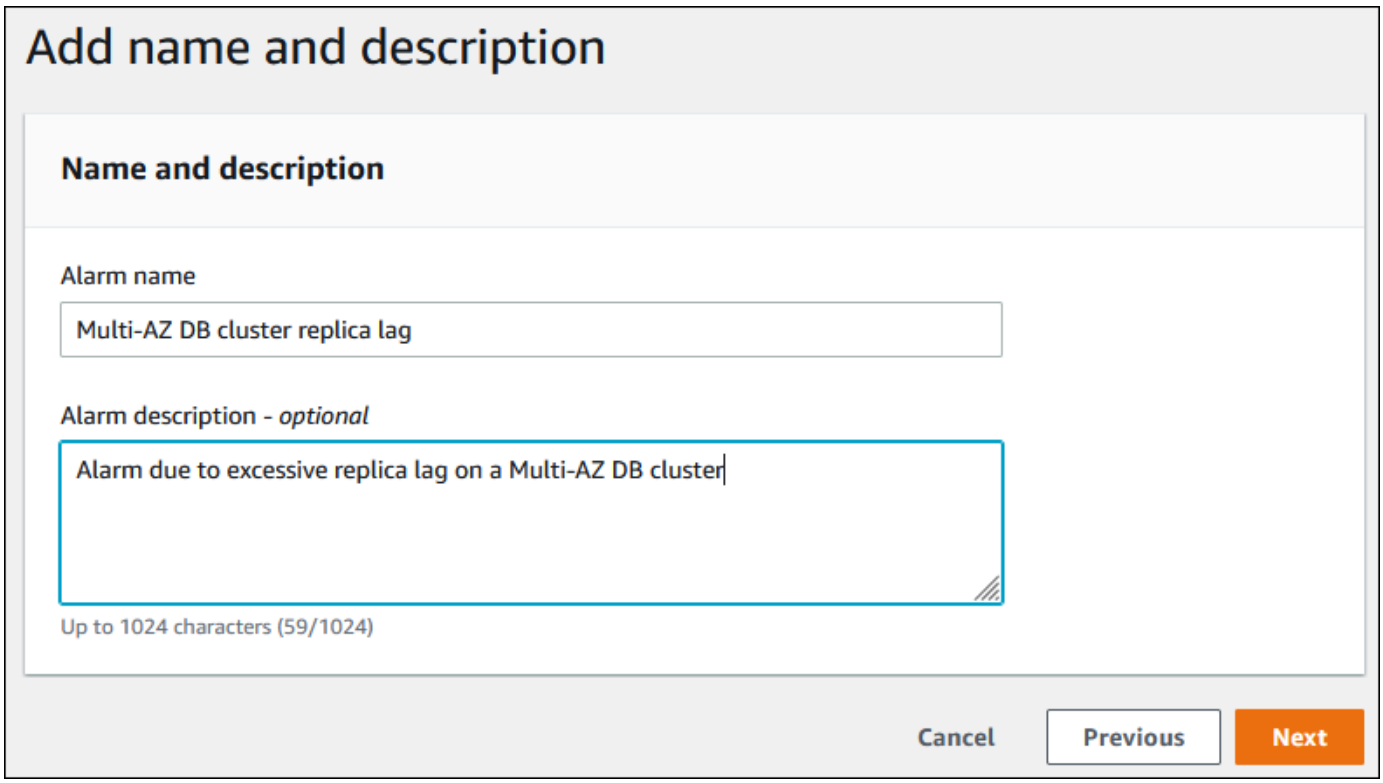


14. Selecione Next (Próximo), e a página Configure actions (Configurar ações) é exibida.

15. Mantenha In alarm (Em alarme) selecionado, escolha Create new topic (Criar tópico) e insira o nome do tópico e um endereço de e-mail válido.



16. Escolha Create topic (Criar tópico) e, em seguida, escolha Next (Próximo).
17. No painel Add name and description (Adicionar nome e descrição), insira o Alarm name (Nome do alarme) e a Alarm description (Descrição do alarme) e, em seguida, escolha Next (Próximo).



18. Visualize o alarme que você está prestes a criar na página Preview and create (Visualizar e criar) e escolha Create alarm (Criar alarme).

Monitoramento de bancos de dados do Amazon RDS com o CloudWatch Database Insights

Monitore a carga do banco de dados da sua frota de instâncias de banco de dados do Amazon RDS com o Database Insights. A carga do banco de dados avalia o nível de atividade de sessão no banco de dados. Você pode usar o Database Insights para analisar e solucionar problemas de performance de seus bancos de dados do Amazon RDS em grande escala.

Com o Database Insights, você pode visualizar a carga do banco de dados na sua frota e filtrá-la por esperas, instruções SQL, hosts ou usuários.

Por padrão, o RDS habilita o modo Padrão do Database Insights para os bancos de dados do Amazon RDS.

Para obter informações sobre o uso do Database Insights no console do Amazon CloudWatch, consulte CloudWatch Database Insights no Guia do usuário do Amazon CloudWatch.

Preços

Para obter informações sobre preços, consulte Amazon CloudWatch Pricing.

Tópicos

- O mecanismo de banco de dados do Amazon RDS, a região e a classe de instância são compatíveis com o Database Insights
- Ativação do modo Avançado do Database Insights para Amazon RDS
- Ativação do modo Standard do Database Insights para Amazon RDS
- Configurar o banco de dados para monitorar consultas SQL lentas com o Database Insights para Amazon RDS
- Considerações sobre o Database Insights para Amazon RDS

O mecanismo de banco de dados do Amazon RDS, a região e a classe de instância são compatíveis com o Database Insights

A tabela a seguir fornece mecanismos de banco de dados do Amazon RDS que são compatíveis com o Database Insights.

Mecanismo de banco de dados do Amazon RDS	Versões do mecanismo e regiões compatíveis	Limitações de classes de instâncias
Amazon RDS para MariaDB	Para ter mais informações sobre a disponibilidade de versões e regiões do Database Insights com o RDS para MariaDB, consulte Regiões e mecanismos de banco de dados compatíveis com o Insights de Performance no Amazon RDS.	O Database Insights não comporta as seguintes classes de instância: • db.t2.micro • db.t2.small • db.t3.micro • db.t3.small • db.t4g.micro • db.t4g.small
RDS para MySQL	Para ter mais informações sobre a disponibilidade de versões e regiões do Database Insights com o RDS para MySQL, consulte Regiões e mecanismos de banco de dados compatíveis com o Insights de Performance no Amazon RDS.	O Database Insights não comporta as seguintes classes de instância: • db.t2.micro • db.t2.small • db.t3.micro • db.t3.small • db.t4g.micro • db.t4g.small
Amazon RDS para Microsoft SQL Server	Para ter mais informações sobre a disponibilidade de versões e regiões do Database Insights	N/D

Mecanismo de banco de dados do Amazon RDS	Versões do mecanismo e regiões compatíveis	Limitações de classes de instâncias
	com o RDS para SQL Server, consulte Regiões e mecanismos de banco de dados compatíveis com o Insights de Performance no Amazon RDS.	
Amazon RDS para PostgreSQL	Para ter mais informações sobre a disponibilidade de versões e regiões do Database Insights com o RDS para PostgreSQL, consulte Regiões e mecanismos de banco de dados compatíveis com o Insights de Performance no Amazon RDS.	N/D
Amazon RDS para Oracle	Para ter mais informações sobre a disponibilidade de versões e regiões do Database Insights com o RDS para Oracle, consulte Regiões e mecanismos de banco de dados compatíveis com o Insights de Performance no Amazon RDS.	N/D

Mecanismo de banco de dados do Amazon RDS, região e classe de instância compatíveis com os recursos do Database Insights

A tabela a seguir apresenta os mecanismos de banco de dados do Amazon RDS que são compatíveis com recursos do Database Insights.

Recurso	Nível de preço	Regiões compatíveis	Mecanismos de banco de dados compatíveis	Classes de instância compatíveis
Estatísticas SQL para Performance Insights	Todos	Todos	Todos	Todos
Analisar planos de execução do Oracle usando o painel do Insights de Performance para o Amazon RDS	Todos	Todos	RDS para Oracle	Todos
Analisar a performance do banco de dados por um período	Somente nível pago	Todos	• RDS para MariaDB • RDS para MySQL • RDS para PostgreSQL.	Todos
Visualizar as recomendações proativas do Performance Insights	Somente nível pago	• Leste dos EUA (Ohio) • Leste dos EUA (Norte da Virgínia) • Oeste dos EUA (Norte da Califórnia) • Oeste dos EUA (Oregon)	Todos	Todos

Recurso	Nível de preço	Regiões compatíveis	Mecanismos de banco de dados compatíveis	Classes de instância compatíveis
		• Ásia-Pacífico (Mumbai) • Ásia-Pacífico (Seul) • Ásia-Pacífico (Singapura) • Ásia-Pacífico (Sydney) • Ásia-Pacífico (Tóquio) • Canadá (Central) • Europa (Frankfurt) • Europa (Irlanda) • Europa (Londres) • Europa (Paris) • Europa (Estocolmo) • América do Sul (São Paulo)		

Regiões em que o Amazon RDS oferece suporte ao Database Insights

O Amazon RDS oferece suporte ao Database Insights nas Regiões da AWS a seguir.

- Leste dos EUA (Norte da Virgínia)

- Leste dos EUA (Ohio)
- Oeste dos EUA (Norte da Califórnia)
- Oeste dos EUA (Oregon)
- África (Cidade do Cabo)
- Ásia-Pacífico (Hong Kong)
- Ásia-Pacífico (Hyderabad)
- Ásia-Pacífico (Jacarta)
- Ásia-Pacífico (Malásia)
- Ásia-Pacífico (Melbourne)
- Ásia-Pacífico (Mumbai)
- Ásia-Pacífico (Osaka)
- Ásia-Pacífico (Seul)
- Ásia-Pacífico (Singapura)
- Ásia-Pacífico (Sydney)
- Ásia-Pacífico (Tóquio)
- Canadá (Central)
- Oeste do Canadá (Calgary)
- Europa (Frankfurt)
- Europa (Irlanda)
- Europa (Londres)
- Europa (Milão)
- Europe (Paris)
- Europa (Espanha)
- Europa (Estocolmo)
- Europa (Zurique)
- Israel (Tel Aviv)
- Oriente Médio (Bahrein)
- Oriente Médio (Emirados Árabes Unidos)

- South America (São Paulo)
- AWS GovCloud (Leste dos EUA)
- AWS GovCloud (Oeste dos EUA)

Ativação do modo Avançado do Database Insights para Amazon RDS

Para ativar o modo Avançado do Database Insights para Amazon RDS, use os procedimentos a seguir.

Ativar ou desativar o modo Avançado do Database Insights ao criar uma instância de banco de dados ou cluster de banco de dados multi-AZ

Ative o modo Avançado do Database Insights ao criar um banco de dados para Amazon RDS.

Console

No console, você pode ativar o modo Avançado do Database Insights ao criar uma instância de banco de dados ou um cluster de banco de dados multi-AZ.

Como ativar o modo Avançado do Database Insights ao criar uma instância de banco de dados ou um cluster de banco de dados multi-AZ usando o console

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. Escolha Databases (Bancos de dados).
3. Selecione Criar banco de dados.
4. Na seção Database Insights, selecione Modo avançado. Em seguida, escolha as seguintes opções:
   - Retention (Retenção) – a duração do período de retenção de dados do Performance Insights. O período de retenção deve ser de 15 meses para o modo avançado do Database Insights.
   - AWS KMS key: especifique a sua chave do KMS. O Performance Insights criptografa todos os possíveis dados sigilosos usando a sua chave do KMS. Os dados são criptografados em repouso e em trânsito. Para obter mais informações, consulte Criptografar recursos do Amazon RDS.
5. Selecione Criar banco de dados.

AWS CLI

Para ativar o modo Avançado do Database Insights ao criar uma instância de banco de dados ou um cluster de banco de dados multi-AZ, chame o comando create-db-instance ou create-db-cluster da AWS CLI e forneça os seguintes valores:

- `--database-insights-mode advanced` para ativar o modo Avançado do Database Insights.
- `--engine`: o mecanismo de banco de dados para a instância de banco de dados.
- `--db-instance-identifier`, o identificador da instância de banco de dados, ou `--db-cluster-identifier`, o identificar do cluster de banco de dados multi-AZ.
- `--enable-performance-insights` para ativar o Performance Insights para o Database Insights.
- `--performance-insights-retention-period`: o período de retenção de dados do cluster de banco de dados multi-AZ ou da instância de banco de dados. Para ativar o Database Insights, o período de retenção deve ser de pelo menos 465 dias.

O exemplo a seguir habilita o modo Avançado do Database Insights ao criar uma instância de banco de dados.

Para Linux, macOS ou Unix:

```
aws rds create-db-instance \
    --database-insights-mode advanced \
    --engine postgresql \
    --db-instance-identifier sample-db-identifier \
    --enable-performance-insights \
    --performance-insights-retention-period 465
```

Para Windows:

```
aws rds create-db-instance ^
    --database-insights-mode advanced ^
    --engine postgresql ^
    --db-instance-identifier sample-db-identifier ^
    --enable-performance-insights ^
    --performance-insights-retention-period 465
```

RDS API

Para ativar o modo Avançado do Database Insights ao criar um cluster de banco de dados multi-AZ ou uma instância de banco de dados, especifique os parâmetros a seguir para a operação de API CreateDBInstance ou CreateDBCluster do Amazon RDS.

- `DatabaseInsightsMode` para `advanced`
- `EnablePerformanceInsights` para `True`
- `PerformanceInsightsRetentionPeriod` para pelo menos 465 dias

Ativar ou desativar o modo Avançado do Database Insights ao modificar uma instância de banco de dados ou cluster de banco de dados multi-AZ

Ative o Database Insights ao modificar um banco de dados para o Amazon RDS. A modificação de uma instância de banco de dados para habilitar o modo avançado do Database Insights não causa tempo de inatividade.

> **Note**
>
> Para habilitar o Database Insights, cada instância de banco de dados em um cluster de banco de dados multi-AZ deve ter as mesmas configurações do Insights de Performance e do monitoramento avançado.

Console

No console, você pode ativar o modo Avançado do Database Insights ao modificar uma instância de banco de dados ou um cluster de banco de dados multi-AZ.

Como ativar o modo Avançado do Database Insights ao modificar uma instância de banco de dados ou um cluster de banco de dados multi-AZ usando o console

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. Escolha Databases (Bancos de dados).
3. Escolha uma instância de banco de dados ou um cluster de banco de dados multi-AZ e escolha Modify (Modificar).

4. Na seção Database Insights, selecione Modo avançado. Em seguida, escolha as seguintes opções:

   - Retention (Retenção) – a duração do período de retenção de dados do Performance Insights. O período de retenção deve ser de 15 meses para o modo avançado do Database Insights.
   - AWS KMS key: especifique a sua chave do KMS. O Performance Insights criptografa todos os possíveis dados sigilosos usando a sua chave do KMS. Os dados são criptografados em repouso e em trânsito. Para obter mais informações, consulte Criptografar recursos do Amazon RDS.
5. Escolha Continue.
6. Em Programação das modificações, selecione Aplicar imediatamente. Se você escolher Aplicar durante a próxima janela de manutenção programada, seu banco de dados vai ignorar essa configuração e ativar imediatamente o modo Avançado do Database Insights.
7. Escolha Modificar instância.

AWS CLI

Para ativar o modo avançado do Database Insights ao modificar uma instância de banco de dados ou um cluster de banco de dados multi-AZ, chame o comando modify-db-instance ou modify-db-cluster da AWS CLI e forneça os seguintes valores:

- `--database-insights-mode advanced` para ativar o modo Avançado do Database Insights.
- `--db-instance-identifier`, o identificador da instância de banco de dados, ou `--db-cluster-identifier`, o identificar do cluster de banco de dados multi-AZ.
- `--enable-performance-insights` para ativar o Performance Insights para o Database Insights.
- `--performance-insights-retention-period`: o período de retenção de dados da instância de banco de dados. Para ativar o modo Avançado do Database Insights, o período de retenção deve ser de pelo menos 465 dias.

O exemplo a seguir habilita o modo Avançado do Database Insights ao modificar uma instância de banco de dados.

Para Linux, macOS ou Unix:

```
aws rds modify-db-instance \
    --database-insights-mode advanced \
    --db-instance-identifier sample-db-identifier \
    --enable-performance-insights \
    --performance-insights-retention-period 465
```

Para Windows:

```
aws rds modify-db-instance ^
    --database-insights-mode advanced ^
    --db-instance-identifier sample-db-identifier ^
    --enable-performance-insights ^
    --performance-insights-retention-period 465
```

RDS API

Para ativar o modo Avançado do Database Insights ao modificar uma instância de banco de dados ou um cluster de banco de dados multi-AZ, especifique os seguintes parâmetros para a operação de API ModifyDBInstance ou ModifyDBCluster do Amazon RDS.

- `DatabaseInsightsMode` para `advanced`
- `EnablePerformanceInsights` para `True`
- `PerformanceInsightsRetentionPeriod` para pelo menos 465 dias

Ativação do modo Standard do Database Insights para Amazon RDS

Para ativar o modo Standard do Database Insights para Amazon RDS, use os procedimentos a seguir.

Ativar ou desativar o modo Standard do Database Insights ao criar uma instância de banco de dados ou cluster de banco de dados multi-AZ

Ative o modo Standard do Database Insights ao criar um banco de dados para Amazon RDS.

Console

No console, você pode ativar o modo Padrão do Database Insights ao criar uma instância de banco de dados ou um cluster de banco de dados multi-AZ.

Como ativar o modo Standard do Database Insights ao criar uma instância de banco de dados ou um cluster de banco de dados multi-AZ usando o console

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. Escolha Databases (Bancos de dados).
3. Selecione Criar banco de dados.
4. Na seção Database Insights, selecione Modo Standard. Em seguida, escolha uma das opções a seguir para ativar ou desativar o Insights de Performance:
   - Para desativar o Insights de Performance, desmarque a opção Habilitar Insights de Performance.
   - Para ativar o Insights de Performance, selecione Habilitar Insights de Performance. Para configurar o Performance Insights, especifique as seguintes opções:
     - Retention (Retenção) – a duração do período de retenção de dados do Performance Insights. O período de retenção deve ser de pelo menos sete dias.
     - AWS KMS key: especifique a sua chave do KMS. O Performance Insights criptografa todos os possíveis dados sigilosos usando a sua chave do KMS. Os dados são criptografados em repouso e em trânsito. Para obter mais informações, consulte Criptografar recursos do Amazon RDS.
5. Selecione Criar banco de dados.

AWS CLI

Para ativar o modo Padrão do Database Insights ao criar uma instância de banco de dados ou um cluster de banco de dados multi-AZ, chame o comando create-db-instance ou create-db-cluster da AWS CLI e forneça os seguintes valores:

- `--database-insights-mode standard` para ativar o modo Standard do Database Insights.
- `--engine`: o mecanismo de banco de dados para a instância de banco de dados.
- `--db-instance-identifier`, o identificador da instância de banco de dados, ou `--db-cluster-identifier`, o identificar do cluster de banco de dados multi-AZ.
- `--enable-performance-insights` ou `--no-enable-performance-insights` para ativar ou desativar o Performance Insights. Se você especificar `--enable-performance-insights`, também deverá especificar o `--performance-insights-retention-period`,

que é o período de retenção de dados do da instância de banco de dados. O período de retenção deve ser de pelo menos sete dias.

O exemplo a seguir habilita o modo Padrão do Database Insights e o Insights de Performance ao criar uma instância de banco de dados.

Para Linux, macOS ou Unix:

```
aws rds create-db-instance \
    --database-insights-mode standard \
    --engine postgresql \
    --db-instance-identifier sample-db-identifier \
    --enable-performance-insights \
    --performance-insights-retention-period 7
```

Para Windows:

```
aws rds create-db-instance ^
    --database-insights-mode standard ^
    --engine postgresql ^
    --db-instance-identifier sample-db-identifier ^
    --enable-performance-insights ^
    --performance-insights-retention-period 7
```

O exemplo a seguir habilita o modo Padrão do Database Insights e desabilita o Insights de Performance ao criar uma instância de banco de dados.

Para Linux, macOS ou Unix:

```
aws rds create-db-instance \
    --database-insights-mode standard \
    --engine postgresql \
    --db-instance-identifier sample-db-identifier \
    --no-enable-performance-insights
```

Para Windows:

```
aws rds create-db-instance ^
    --database-insights-mode standard ^
    --engine postgresql ^
```

```
    --db-instance-identifier sample-db-identifier ^
    --no-enable-performance-insights
```

RDS API

Para ativar o modo Standard do Database Insights ao criar uma instância de banco de dados ou um cluster de banco de dados multi-AZ, especifique os seguintes parâmetros para sua operação de API do Amazon RDS CreateDBInstance ou CreateDBCluster.

- `DatabaseInsightsMode` para `standard`
- `EnablePerformanceInsights` como `True` ou `False`. Se você definir `EnablePerformanceInsights` como `True`, deverá definir `PerformanceInsightsRetentionPeriod` como pelo menos sete dias.

Ativar ou desativar o modo Standard do Database Insights ao modificar uma instância de banco de dados ou cluster de banco de dados multi-AZ

Ative o modo Standard do Database Insights ao modificar um banco de dados para o Amazon RDS. A modificação de um uma instância de banco de dados para habilitar o modo padrão do Database Insights não causa tempo de inatividade.

> **Note**
>
> Para habilitar o Database Insights, cada instância de banco de dados em um cluster de banco de dados multi-AZ deve ter as mesmas configurações do Insights de Performance e do monitoramento avançado.

Console

No console, você pode ativar o modo Padrão do Database Insights ao modificar uma instância de banco de dados ou um cluster de banco de dados multi-AZ.

Para ativar o modo Standard do Database Insights ao modificar uma instância de banco de dados ou um cluster de banco de dados multi-AZ usando o console

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.

2. Escolha Databases (Bancos de dados).
3. Escolha uma instância de banco de dados ou um cluster de banco de dados multi-AZ e escolha Modify (Modificar).
4. Na seção Database Insights, selecione Modo Standard. Depois, escolha uma das seguintes opções:
   - Para desativar o Insights de Performance, desmarque a opção Habilitar Insights de Performance.
   - Para ativar o Insights de Performance, selecione Habilitar Insights de Performance. Para configurar o Performance Insights, especifique as seguintes opções:
     - Retention (Retenção) – a duração do período de retenção de dados do Performance Insights. O período de retenção deve ser de pelo menos sete dias.
     - AWS KMS key: especifique a sua chave do KMS. O Performance Insights criptografa todos os possíveis dados sigilosos usando a sua chave do KMS. Os dados são criptografados em repouso e em trânsito. Para obter mais informações, consulte Criptografar recursos do Amazon RDS.
5. Escolha Continue.
6. Em Programação das modificações, selecione Aplicar imediatamente. Se você escolher Aplicar durante a próxima janela de manutenção programada, seu banco de dados vai ignorar essa configuração e ativar imediatamente o modo Padrão do Database Insights.
7. Escolha Modificar instância.

AWS CLI

Para ativar o modo Padrão do Database Insights ao modificar uma instância de banco de dados ou um cluster de banco de dados multi-AZ, chame o comando modify-db-instance ou modify-db-cluster da AWS CLI e forneça os seguintes valores:

- `--database-insights-mode standard` para ativar o modo Standard do Database Insights.
- `--db-instance-identifier`, o identificador da instância de banco de dados, ou `--db-cluster-identifier`, o identificar do cluster de banco de dados multi-AZ.
- `--enable-performance-insights` ou `--no-enable-performance-insights` para ativar ou desativar o Performance Insights. Se você especificar `--enable-performance-insights`, também deverá especificar o `--performance-insights-retention-period`,

que é o período de retenção de dados do cluster de banco de dados multi-AZ ou da instância de banco de dados. O período de retenção deve ser de pelo menos sete dias.

O exemplo a seguir habilita o modo Padrão do Database Insights e o Insights de Performance ao modificar uma instância de banco de dados.

Para Linux, macOS ou Unix:

```
aws rds modify-db-instance \
    --database-insights-mode standard \
    --db-instance-identifier sample-db-identifier \
    --enable-performance-insights \
    --performance-insights-retention-period 7
```

Para Windows:

```
aws rds modify-db-instance ^
    --database-insights-mode standard ^
    --db-instance-identifier sample-db-identifier ^
    --enable-performance-insights ^
    --performance-insights-retention-period 7
```

O exemplo a seguir habilita o modo Padrão do Database Insights e desabilita o Insights de Performance ao modificar uma instância de banco de dados.

Para Linux, macOS ou Unix:

```
aws rds modify-db-instance \
    --database-insights-mode standard \
    --db-instance-identifier sample-db-identifier \
    --no-enable-performance-insights
```

Para Windows:

```
aws rds modify-db-instance ^
    --database-insights-mode standard ^
    --db-instance-identifier sample-db-identifier ^
    --no-enable-performance-insights
```

RDS API

Para ativar o modo Standard do Database Insights ao modificar uma instância de banco de dados ou um cluster de banco de dados multi-AZ, especifique os seguintes parâmetros para a operação de API ModifyDBInstance ou ModifyDBCluster do Amazon RDS.

- `DatabaseInsightsMode` para `standard`
- `EnablePerformanceInsights` como `True` ou `False`. Se você definir `EnablePerformanceInsights` como `True`, deverá definir `PerformanceInsightsRetentionPeriod` como pelo menos sete dias.

Configurar o banco de dados para monitorar consultas SQL lentas com o Database Insights para Amazon RDS

Para monitorar consultas SQL lentas em seu banco de dados, você pode usar a seção Consultas SQL lentas no painel do Database Insights. Antes de você configurar seu banco de dados para monitorar consultas SQL lentas, a seção Consultas SQL lentas se apresenta em branco.

Para ter mais informações sobre o monitoramento de consultas SQL lentas no painel do Database Insights, consulte Visualizar o painel de instância de banco de dados do CloudWatch Database Insights no Guia do usuário do Amazon CloudWatch.

Para configurar o banco de dados para monitorar consultas SQL lentas com o Database Insights, conclua as seguintes etapas:

1. Habilite as exportações de log para o CloudWatch Logs.
2. Crie ou modifique o grupo de parâmetros de banco de dados para sua instância de banco de dados.

Para ter informações sobre como configurar exportações de log, consulte Publicação de logs de banco de dados no Amazon CloudWatch Logs no Guia do usuário do Amazon RDS.

Para criar ou modificar seu grupo de parâmetros de banco de dados, consulte os tópicos a seguir.

- Criar um grupo de parâmetros de banco de dados no Amazon RDS
- Modificar parâmetros em um grupo de parâmetros de banco de dados no Amazon RDS

RDS for MariaDB

Para configurar sua instância de banco de dados do RDS para MariaDB e monitorar consultas SQL lentas, defina os parâmetros a seguir.

- `log_slow_query`: definir como `1`
- `log_slow_query_time`: definir como `1.0`
- `log_output`: definir como `FILE`

RDS for MySQL

Para configurar sua instância de banco de dados do RDS para MySQL e monitorar consultas SQL lentas, defina os parâmetros a seguir.

- `slow_query_log`: definir como `1`
- `long_query_time`: definir como `1.0`
- `log_output`: definir como `FILE`

RDS for PostgreSQL

Para configurar sua instância de banco de dados do RDS para PostgreSQL e monitorar consultas SQL lentas, defina os parâmetros a seguir. Observe que, ao definir esses parâmetros, o desempenho da instância de banco de dados pode diminuir.

- `log_min_duration_statement`: definir como `1000`
- `log_statement`: definir como `none`
- `log_destination`: definir como `stderr`

> **Note**
>
> Para o RDS para MySQL, você pode configurar o parâmetro `long_query_time` com granularidade de 1 microssegundo. Por exemplo, você pode definir esse parâmetro como `0.000001`. Dependendo da quantidade de consultas na instância de banco de dados, o valor do parâmetro `long_query_time` pode reduzir o desempenho. Comece com o valor `1.0` e ajuste-o com base na sua workload. Quando você define esse parâmetro como `0`, o Database Insights registra em log todas as consultas.

Para ter informações sobre os logs do RDS para MariaDB, RDS para MySQL e RDS para PostgreSQL, consulte o seguinte:

- Arquivos de log do banco de dados MariaDB
- Arquivos de log do banco de dados MySQL
- Arquivos de log do banco de dados do RDS para PostgreSQL

Considerações sobre o Database Insights para Amazon RDS

Confira a seguir considerações sobre o Database Insights para Amazon RDS.

- Não é possível gerenciar o Database Insights para uma instância de banco de dados em um cluster de banco de dados multi-AZ.
- Para ativar o modo Avançado do Database Insights, você deve ativar o Insights de Performance e definir o período de retenção do Insights de Performance para pelo menos 465 dias (15 meses). Não há custo adicional para definir o período de retenção do Insights de Performance como 15 meses, além do custo do Database Insights. Para ter informações sobre os preços do Database Insights, consulte Definição de preço do Amazon CloudWatch.
- Para habilitar o Database Insights, cada instância de banco de dados em um cluster de banco de dados multi-AZ deve ter as mesmas configurações do Insights de Performance e do monitoramento avançado.
- A modificação de uma instância de banco de dados para habilitar qualquer um dos modos do Database Insights não causa tempo de inatividade.

Monitorar a carga de banco de dados com o Performance Insights no Amazon RDS

Important

A AWS anunciou a data de fim da vida útil do Insights de Performance: 30 de novembro de 2025. Após essa data, o Amazon RDS não oferecerá mais suporte à experiência do console, aos períodos de retenção flexíveis (de 1 a 24 meses) e aos preços associados do Insights de Performance. A API do Insights de Performance continuará existindo sem alterações de preço. Os custos da API do Insights de Performance aparecerão em sua fatura da com o custo do CloudWatch Database Insights.

Recomendamos que você atualize quaisquer instâncias de banco de dados usando o nível pago do Insights de Performance para o modo avançado do Database Insights antes de 30 de novembro de 2025. Para ter informações sobre como atualizar para o modo avançado do Database Insights, consulte Ativação do modo Avançado do Database Insights para Amazon RDS.

Se você não tomar nenhuma medida, as instâncias de banco de dados que estão usando o Insights de Performance usarão como padrão o modo padrão do Database Insights. Com o modo padrão do Database Insights, você pode perder o acesso ao histórico de dados de desempenho após sete dias e talvez não consiga usar planos de execução e recursos de análise sob demanda no console do Amazon RDS. Após 30 de novembro de 2025, somente o modo avançado do Database Insights permitirá planos de execução e análise sob demanda.

Com o CloudWatch Database Insights, você pode monitorar a carga do banco de dados para sua frota de bancos de dados e analisar e solucionar problemas de desempenho em grande escala. Para ter mais informações sobre o Database Insights, consulte Monitoramento de bancos de dados do Amazon RDS com o CloudWatch Database Insights. Para obter informações de definição de preço, consulte Definição de preço do Amazon CloudWatch.

O Performance Insights expande os recursos de monitoramento do Amazon RDS existentes para ilustrar e ajudar você a analisar a performance do banco de dados. Com o painel do Performance Insights, você pode visualizar a carga do banco de dados em sua carga de instâncias de banco de dados do Amazon RDS e filtrá-la por esperas, instruções SQL, hosts ou usuários. Para obter informações sobre como usar o Performance Insights com Amazon DocumentDB, consulte o Guia do desenvolvedor do Amazon DocumentDB.

Tópicos

- Visão geral do Performance Insights no Amazon RDS
- Ativar e desativar o Insights de Performance do Amazon RDS
- Visão geral do Performance Schema para o Insights de Performance no Amazon RDS para MariaDB ou MySQL
- Configurar políticas de acesso para o Performance Insights
- Análise de métricas usando o painel do Performance Insights
- Visualizar as recomendações proativas do Performance Insights
- Recuperar métricas com a API do Insights de Performance do Amazon RDS
- Registrar em log as chamadas do Performance Insights usando o AWS CloudTrail
- API do Performance Insights e endpoints da VPC de interface (AWS PrivateLink)

Visão geral do Performance Insights no Amazon RDS

Important

A AWS anunciou a data de fim da vida útil do Insights de Performance: 30 de novembro de 2025. Após essa data, o Amazon RDS não oferecerá mais suporte à experiência do console, aos períodos de retenção flexíveis (de 1 a 24 meses) e aos preços associados do Insights de Performance. A API do Insights de Performance continuará existindo sem alterações de preço. Os custos da API do Insights de Performance aparecerão em sua fatura da com o custo do CloudWatch Database Insights.

Recomendamos que você atualize quaisquer instâncias de banco de dados usando o nível pago do Insights de Performance para o modo avançado do Database Insights antes de 30 de novembro de 2025. Para ter informações sobre como atualizar para o modo avançado do Database Insights, consulte Ativação do modo Avançado do Database Insights para Amazon RDS.

Se você não tomar nenhuma medida, as instâncias de banco de dados que estão usando o Insights de Performance usarão como padrão o modo padrão do Database Insights. Com o modo padrão do Database Insights, você pode perder o acesso ao histórico de dados de desempenho após sete dias e talvez não consiga usar planos de execução e recursos de análise sob demanda no console do Amazon RDS. Após 30 de novembro de 2025, somente o modo avançado do Database Insights permitirá planos de execução e análise sob demanda.

Com o CloudWatch Database Insights, você pode monitorar a carga do banco de dados para sua frota de bancos de dados e analisar e solucionar problemas de desempenho em grande escala. Para ter mais informações sobre o Database Insights, consulte Monitoramento de bancos de dados do Amazon RDS com o CloudWatch Database Insights. Para obter informações de definição de preço, consulte Definição de preço do Amazon CloudWatch.

Por padrão, o RDS ativa o Insights de Performance no assistente de criação do console para mecanismos do Amazon RDS. Se você tiver mais de um banco de dados em uma instância de banco de dados, o Performance Insights agregará dados de performance.

É possível encontrar uma visão geral do Performance Insights para Amazon RDS no vídeo a seguir.

Como usar o Performance Insights para analisar a performance do Amazon Aurora PostgreSQL

Important

Os tópicos a seguir descrevem o uso do Amazon RDS Performance Insights com mecanismos de banco de dados não Aurora. Para obter informações sobre o uso do Amazon RDS Performance Insights com o Amazon Aurora, consulte Usar o Amazon RDS Performance Insights no Guia do usuário do Amazon Aurora.

Tópicos

- Carga de banco de dados
- Máximo de CPU
- O mecanismo de banco de dados do Amazon RDS, a região e a classe de instância são compatíveis com o Performance Insights
- Preços e retenção de dados para o Performance Insights

Carga de banco de dados

Carga do banco do dados mede o nível de atividade de sessão no banco de dados. `DBLoad` é a métrica principal no Insights de Performance, e o Insights de Performance coleta a carga do banco de dados a cada segundo.

Tópicos

- Sessões ativas
- Média de sessões ativas
- Média de execuções ativas
- Dimensões

Sessões ativas

Uma sessão de base de dados relacional representa o diálogo de uma aplicação com um banco de dados relacional. Uma sessão ativa é uma conexão que enviou trabalho para o mecanismo de banco de dados e está aguardando uma resposta.

Uma sessão fica ativa quando está em execução na CPU ou aguardando a disponibilidade de um recurso para que ela possa continuar. Por exemplo, uma sessão ativa pode esperar que uma página (ou um bloco) seja lida na memória e, depois, consumir CPU enquanto faz a leitura dos dados na página.

Média de sessões ativas

A média de sessões ativas (AAS) é a unidade da métrica `DBLoad` no Performance Insights. Ele mede quantas sessões estão ativas simultaneamente no banco de dados.

A cada segundo, o Insights de Performance faz uma amostra do número de sessões executando simultaneamente uma consulta. Para cada sessão ativa, o Insights de Performance coleta os seguintes dados:

- Declaração do SQL
- Estado da sessão (em execução na CPU ou em espera)
- Host
- Usuário executando o SQL

O Insights de Performance calcula a AAS dividindo o número total de sessões pelo número total de amostras por um período específico. Por exemplo, a tabela a seguir mostra cinco amostras consecutivas de uma consulta em execução em intervalos de um segundo.

Amostra	Número de sessões que executam a consulta	AAS	Cálculo
1	2	2	2 sessões no total/1 amostra
2	0	1	2 sessões no total/2 amostras
3	4	2	6 sessões no total/3 amostras
4	0	1.5	6 sessões no total/4 amostras
5	4	2	10 sessões no total/5 amostras

No exemplo anterior, a carga do banco de dados para o intervalo de tempo foi de 2 AAS. Essa medida significa que, em média, duas sessões estavam ativas em determinado momento durante o intervalo em que as cinco amostras foram obtidas.

Média de execuções ativas

A média de execuções ativas (AAE) por segundo está relacionada ao AAS. Para calcular os AAE, o Performance Insights divide o tempo total de execução de uma consulta pelo intervalo de tempo. A tabela a seguir mostra o cálculo de AAE para a mesma consulta na tabela anterior.

Tempo decorrido (s)	Tempo de execução total (s)	AAE	Cálculo
60	120	2	120 segundos de execução/60 segundos decorridos
120	120	1	120 segundos de execução/120 segundos decorridos

Tempo decorrido (s)	Tempo de execução total (s)	AAE	Cálculo
180	380	2.11	380 segundos de execução/180 segundos decorridos
240	380	1,58	380 segundos de execução/240 segundos decorridos
300	600	2	600 segundos de execução/300 segundos decorridos

Na maioria dos casos, o AAS e o AAE de uma consulta são aproximadamente os mesmos. No entanto, como as entradas para os cálculos são diferentes fontes de dados, os cálculos geralmente variam ligeiramente.

Dimensões

A métrica `db.load` é diferente das outras métricas da série temporal, pois você pode fragmentá-la em subcomponentes chamados de dimensões. Você pode pensar em dimensões como “pedaços” de categorias para as diferentes características da métrica `DBLoad`.

Quando você está diagnosticando problemas de performance, as seguintes dimensões geralmente são as mais úteis:

Tópicos

- Eventos de espera
- SQL principal
- Planos

Para obter uma lista completa de dimensões dos mecanismos Amazon RDS, consulte Carga de banco de dados separada por dimensões.

Eventos de espera

Um evento de espera faz com que uma instrução SQL aguarde que um evento específico aconteça antes que ele possa continuar a execução. Eventos de espera são uma dimensão, ou categoria, importante para a carga do banco de dados, pois indicam onde o trabalho está impedido.

Todas as sessões ativas estão em um estado de espera ou de execução na CPU. Por exemplo, sessões consomem CPU quando procuram um buffer na memória, realizam um cálculo ou executam um código processual. Quando as sessões não estão consumindo CPU, elas podem estar aguardando a liberação de um buffer de memória, a leitura de um arquivo de dados ou a gravação em um log. Quanto mais tempo uma sessão aguardar recursos, menos tempo ela será executada na CPU.

Ao ajustar um banco de dados, muitas vezes você tenta descobrir os recursos que as sessões estão aguardando. Por exemplo, dois ou três eventos de espera podem representar 90% da carga do banco de dados. Essa medida significa que, em média, as sessões ativas estão passando a maior parte do tempo aguardando um pequeno número de recursos. Se você conseguir descobrir a causa dessas esperas, poderá tentar uma solução.

Os eventos de espera variam de acordo com o mecanismo de banco de dados:

- Para obter informações sobre todos os eventos de espera do MariaDB e do MySQL, consulte Tabelas de resumo de eventos de espera na documentação do MySQL.
- Para obter informações sobre todos os eventos de espera do PostgreSQL, consulte The Statistics Collector > Wait Event tables (Coletor de estatísticas > Tabelas de eventos de espera) na documentação do PostgreSQL.
- Para obter mais informações sobre todos os eventos de espera do Oracle, consulte Descriptions of Wait Events (Descrições de ventos de espera) na documentação do Oracle.
- Para obter informações sobre todos os eventos de espera do SQL Server, consulte Tipos de esperas na documentação do SQL Server.

> **Note**
>
> No Oracle, às vezes, os processos em segundo plano funcionam sem uma instrução SQL associada. Nesses casos, o Performance Insights relata o tipo de processo em segundo plano concatenado com dois-pontos e a classe de espera associada a esse processo em segundo plano. Os tipos de processos em segundo plano incluem `LGWR`, `ARC0`, `PMON` e assim por diante.

Por exemplo, quando o arquivador está realizando E/S, o relatório do Performance Insights é semelhante a `ARC1:System I/O`. Às vezes, o tipo de processo em segundo plano também está ausente, e o Performance Insights só informa a classe de espera, por exemplo, `:System I/O`.

SQL principal

Enquanto eventos de espera mostram gargalos, o gráfico Top SQL (SQL principal) mostra quais consultas estão contribuindo mais para a carga do banco de dados. Por exemplo, muitas consultas podem estar em execução no banco de dados, mas uma única consulta pode consumir 99% da carga do banco de dados. Nesse caso, a carga alta pode indicar um problema com a consulta.

Por padrão, o console do Performance Insights exibe as consultas de SQL principal que estão contribuindo para a carga do banco de dados. O console também mostra estatísticas relevantes para cada instrução. Para diagnosticar problemas de performance para uma instrução específica, você pode examinar seu plano de execução.

Planos

Um plano de execução, também chamado simplesmente de plano, é uma sequência de etapas que acessam dados. Por exemplo, um plano para unir tabelas `t1` e `t2` pode percorrer todas as linhas em `t1` e comparar cada linha com uma linha em `t2`. Em um banco de dados relacional, um otimizador é um código interno que determina o plano mais eficiente para uma consulta SQL.

Em relação a instâncias de banco de dados, o Insights de Performance coleta planos de execução automaticamente. Para diagnosticar problemas de performance do SQL, examine os planos capturados para consultas de SQL com uso elevado de recursos. Os planos mostram como o banco de dados analisou e executou consultas.

Para saber como analisar a carga de banco de dados, consulte:

- Oracle: Analisar planos de execução do Oracle usando o painel do Insights de Performance para o Amazon RDS
- SQL Server: Analisar planos de execução do SQL Server usando o painel do Insights de Performance para o Amazon RDS

Captura de planos

A cada cinco minutos, o Insights de Performance identifica as consultas com uso mais intenso de recursos e captura os planos. Assim, você não precisa coletar e gerenciar manualmente um grande número de planos. Em vez disso, você pode usar o SQL principal para focar nos planos para as consultas mais problemáticas.

> **Note**
>
> O Performance Insights não captura planos para consultas cujo texto exceda o limite máximo de texto de consulta coletável. Para ter mais informações, consulte Acessar mais texto SQL no painel do Performance Insights..

O período de retenção para planos de execução é o mesmo dos seus dados do Performance Insights. A configuração de retenção no nível gratuito é Default (7 days) (Padrão (7 dias)). Para reter seus dados de performance por mais tempo, especifique entre 1 e 24 meses. Para obter mais informações sobre os períodos de retenção, consulte Preços e retenção de dados para o Performance Insights.

Consultas de resumo

O SQL principal mostra as consultas de resumo por padrão. Uma consulta de resumo não tem um plano, mas todas as consultas que usam valores literais têm planos. Por exemplo, uma consulta de resumo pode incluir o texto `WHERE `email`=?`. O resumo pode conter duas consultas, uma com o texto `WHERE email=user1@example.com` e outro com `WHERE email=user2@example.com`. Cada uma dessas consultas literais pode incluir vários planos.

Ao selecionar uma consulta de resumo, o console mostra todos os planos para declarações secundárias do resumo selecionado. Assim, você não precisa examinar todas as instruções filho para encontrar o plano. Você pode ver planos que não estão na lista exibida das 10 principais instruções filho. O console mostra planos para todas as consultas filho para as quais os planos foram coletados, independentemente de as consultas estarem entre as dez principais.

Máximo de CPU

No painel, o gráfico Carga de banco de dados coleta, agrega e exibe informações da sessão. Para ver se as sessões ativas estão excedendo o máximo de CPU, observe sua relação com a linha Máx. vCPU. O Insights Performance determina o valor de Máx. vCPU pelo número de núcleos de vCPU (CPUs virtuais) da instância de banco de dados.

Um processo pode ser executado em uma vCPU por vez. Se o número de processos exceder o número de vCPUs, os processos começarão a ser colocados em fila. Quando a colocação em fila aumenta, a performance do banco de dados diminui. Se a carga de banco de dados estiver com frequência acima da linha Máx. vCPU e o estado de espera primário for CPU, isso indicará que a CPU está sobrecarregada. Nesse caso, convém limitar as conexões com a instância, ajustar todas as consultas SQL com uma alta carga de CPU ou considerar uma classe de instância maior. As instâncias altas e consistentes de qualquer estado de espera indicam que pode haver problemas de gargalos ou de contenção de recursos que você deve resolver. Isso pode ser válido mesmo quando a carga do banco de dados não ultrapassa a linha de Máx. vCPU.

O mecanismo de banco de dados do Amazon RDS, a região e a classe de instância são compatíveis com o Performance Insights

Important

A AWS anunciou a data de fim da vida útil do Insights de Performance: 30 de novembro de 2025. Após essa data, o Amazon RDS não oferecerá mais suporte à experiência do console, aos períodos de retenção flexíveis (de 1 a 24 meses) e aos preços associados do Insights de Performance. A API do Insights de Performance continuará existindo sem alterações de preço. Os custos da API do Insights de Performance aparecerão em sua fatura da com o custo do CloudWatch Database Insights.

Recomendamos que você atualize quaisquer instâncias de banco de dados usando o nível pago do Insights de Performance para o modo avançado do Database Insights antes de 30 de novembro de 2025. Para ter informações sobre como atualizar para o modo avançado do Database Insights, consulte Ativação do modo Avançado do Database Insights para Amazon RDS.

Se você não tomar nenhuma medida, as instâncias de banco de dados que estão usando o Insights de Performance usarão como padrão o modo padrão do Database Insights. Com o modo padrão do Database Insights, você pode perder o acesso ao histórico de dados de desempenho após sete dias e talvez não consiga usar planos de execução e recursos de análise sob demanda no console do Amazon RDS. Após 30 de novembro de 2025, somente o modo avançado do Database Insights permitirá planos de execução e análise sob demanda.

Com o CloudWatch Database Insights, você pode monitorar a carga do banco de dados para sua frota de bancos de dados e analisar e solucionar problemas de desempenho em grande escala. Para ter mais informações sobre o Database Insights, consulte Monitoramento

de bancos de dados do Amazon RDS com o CloudWatch Database Insights. Para obter informações de definição de preço, consulte Definição de preço do Amazon CloudWatch.

A tabela a seguir fornece mecanismos de banco de dados do Amazon RDS que são compatíveis com o Insights de Performance.

> **Note**
>
> Para o Amazon Aurora, consulte Suporte ao mecanismo de bancos de dados Amazon Aurora para Performance Insights no Guia do usuário do Amazon Aurora.

Mecanismo de banco de dados do Amazon RDS	Versões do mecanismo e regiões compatíveis	Limitações de classes de instâncias
Amazon RDS para MariaDB	Para obter mais informações sobre a disponibilidade de versões e regiões do Performance Insights com o RDS para MariaDB, consulte Regiões e mecanismos de banco de dados compatíveis com o Insights de Performance no Amazon RDS.	O Insights de Performance não tem suporte para as seguintes classes de instâncias: • db.t2.micro • db.t2.small • db.t3.micro • db.t3.small • db.t4g.micro • db.t4g.small
RDS para MySQL	Para obter mais informações sobre a disponibilidade de versões e regiões do Performance Insights com o RDS para MySQL, consulte	O Insights de Performance não tem suporte para as seguintes classes de instâncias: •

Mecanismo de banco de dados do Amazon RDS	Versões do mecanismo e regiões compatíveis	Limitações de classes de instâncias
	Regiões e mecanismos de banco de dados compatíveis com o Insights de Performance no Amazon RDS.	• db.t2.micro • db.t2.small • db.t3.micro • db.t3.small • db.t4g.micro • db.t4g.small
Amazon RDS para Microsoft SQL Server	Para obter mais informações sobre a disponibilidade de versões e regiões do Performance Insights com o RDS para SQL Server, consulte Regiões e mecanismos de banco de dados compatíveis com o Insights de Performance no Amazon RDS.	N/D
Amazon RDS para PostgreSQL	Para obter mais informações sobre a disponibilidade de versões e regiões do Performance Insights com o RDS para PostgreSQL, consulte Regiões e mecanismos de banco de dados compatíveis com o Insights de Performance no Amazon RDS.	N/D

Mecanismo de banco de dados do Amazon RDS	Versões do mecanismo e regiões compatíveis	Limitações de classes de instâncias
Amazon RDS para Oracle	Para obter mais informações sobre a disponibilidade de versões e regiões do Performance Insights com o RDS para Oracle, consulte Regiões e mecanismos de banco de dados compatíveis com o Insights de Performance no Amazon RDS.	N/D

O mecanismo de banco de dados do Amazon RDS, a região e a classe de instância são compatíveis com atributos do Insights de Performance.

A tabela a seguir fornece mecanismos de banco de dados do Amazon RDS que são compatíveis com atributos do Insights de Performance.

Recurso	Nível de preço	Regiões compatíveis	Mecanismos de banco de dados compatíveis	Classes de instância compatíveis
Estatísticas SQL para Performance Insights	Todos	Todos	Todos	Todos
Analisar planos de execução do Oracle usando o painel do Insights de Performance para o Amazon RDS	Todos	Todos	RDS para Oracle	Todos

Recurso	Nível de preço	Regiões compatíveis	Mecanismos de banco de dados compatíveis	Classes de instância compatíveis
Analisar a performance do banco de dados por um período	Somente nível pago	Todos	RDS para PostgreSQL	Todos

Recurso	Nível de preço	Regiões compatíveis	Mecanismos de banco de dados compatíveis	Classes de instância compatíveis
Visualizar as recomendações proativas do Performance Insights	Somente nível pago	• Leste dos EUA (Ohio) • Leste dos EUA (Norte da Virgínia) • Oeste dos EUA (Norte da Califórnia) • Oeste dos EUA (Oregon) • Ásia-Pacífico (Mumbai) • Ásia-Pacífico (Seul) • Ásia-Pacífico (Singapura) • Ásia-Pacífico (Sydney) • Ásia-Pacífico (Tóquio) • Canadá (Central) • Europa (Frankfurt) • Europa (Irlanda) • Europa (Londres) • Europa (Paris)	Todos	Todos

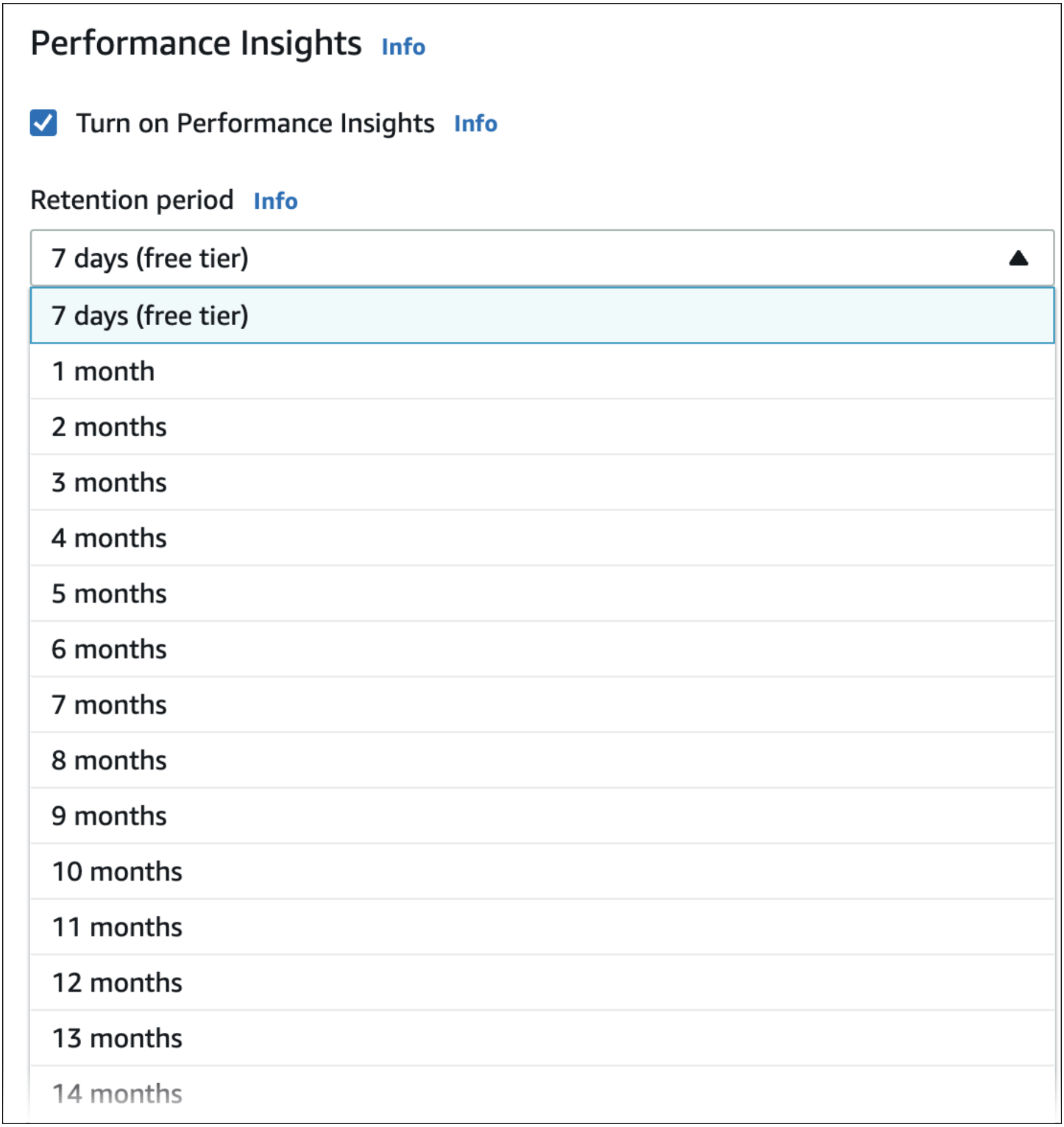
Recurso	Nível de preço	Regiões compatíveis	Mecanismos de banco de dados compatíveis	Classes de instância compatíveis
		• Europa (Estocolmo) • América do Sul (São Paulo)		

Preços e retenção de dados para o Performance Insights

Por padrão, o Performance Insights oferece um nível gratuito que inclui 7 dias de histórico de dados de performance e 1 milhão de solicitações de API por mês. Você também pode comprar períodos de retenção mais longos. Para obter informações completas sobre custos, consulte Definição de preço do Performance Insights.

No console do RDS, você pode escolher qualquer um dos seguintes períodos de retenção para seus dados do Performance Insights:

- Default (7 days) (Padrão (7 dias))
- ***n*** meses, em que ***n*** é um número entre 1 e 24



Para saber como definir um período de retenção usando a AWS CLI, consulte ???.

Ativar e desativar o Insights de Performance do Amazon RDS

Important

A AWS anunciou a data de fim da vida útil do Insights de Performance: 30 de novembro de 2025. Após essa data, o Amazon RDS não oferecerá mais suporte à experiência do console, aos períodos de retenção flexíveis (de 1 a 24 meses) e aos preços associados do Insights de Performance. A API do Insights de Performance continuará existindo sem alterações de preço. Os custos da API do Insights de Performance aparecerão em sua fatura da com o custo do CloudWatch Database Insights.

Recomendamos que você atualize quaisquer instâncias de banco de dados usando o nível pago do Insights de Performance para o modo avançado do Database Insights antes de 30 de novembro de 2025. Para ter informações sobre como atualizar para o modo avançado do Database Insights, consulte Ativação do modo Avançado do Database Insights para Amazon RDS.

Se você não tomar nenhuma medida, as instâncias de banco de dados que estão usando o Insights de Performance usarão como padrão o modo padrão do Database Insights. Com o modo padrão do Database Insights, você pode perder o acesso ao histórico de dados de desempenho após sete dias e talvez não consiga usar planos de execução e recursos de análise sob demanda no console do Amazon RDS. Após 30 de novembro de 2025, somente o modo avançado do Database Insights permitirá planos de execução e análise sob demanda.

Com o CloudWatch Database Insights, você pode monitorar a carga do banco de dados para sua frota de bancos de dados e analisar e solucionar problemas de desempenho em grande escala. Para ter mais informações sobre o Database Insights, consulte Monitoramento de bancos de dados do Amazon RDS com o CloudWatch Database Insights. Para obter informações de definição de preço, consulte Definição de preço do Amazon CloudWatch.

É possível ativar o Insights de Performance para a instância de banco de dados ou o cluster de banco de dados multi-AZ durante a criação. Se necessário, você pode desativá-lo posteriormente modificando a instância de banco de dados no console. A ativação e a desativação do Performance Insights não causa tempo de inatividade, reinicialização ou failover.

> **Note**
>
> O Performance Schema é uma ferramenta de performance opcional usada pelo Amazon RDS para MariaDB ou MySQL. Se você ativar ou desativar o Performance Schema, será necessário reinicializar. No entanto, se você ativar ou desativar o Performance Insights, não será necessário reinicializar. Para obter mais informações, consulte Visão geral do Performance Schema para o Insights de Performance no Amazon RDS para MariaDB ou MySQL.

O agente do Performance Insights consome CPU e memória limitadas no host do banco de dados. Quando a carga do banco de dados é alta, o agente limita o impacto sobre a performance coletando dados com menos frequência.

Console

No console, ative ou desative o Performance Insights ao criar ou modificar uma instância de banco de dados ou cluster de banco de dados multi-AZ.

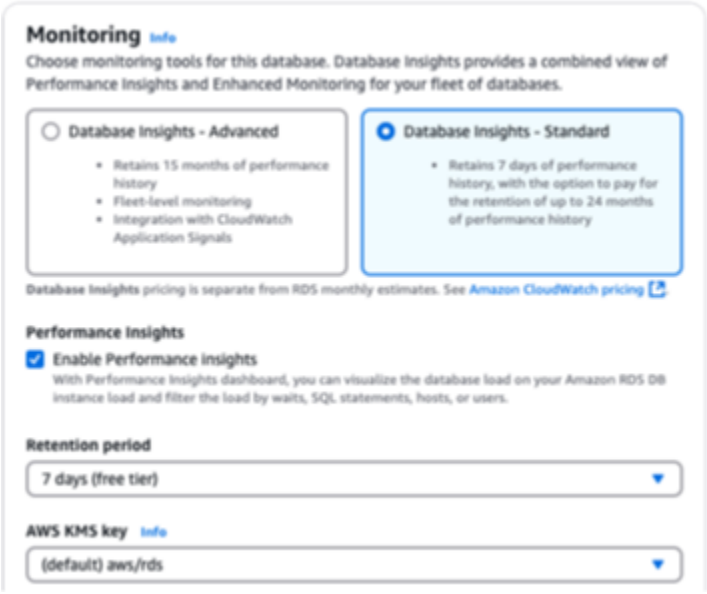
Ativar ou desativar o Insights de Performance ao criar uma instância de banco de dados ou cluster de banco de dados multi-AZ

Depois de criar uma instância de banco de dados ou cluster de banco de dados multi-AZ, o Amazon RDS habilita o Insights de Performance por padrão. Para desativar o Insights de Performance, escolha a opção Database Insights: padrão e desmarque a opção Habilitar Insights de Performance.

Para obter mais informações, consulte os tópicos a seguir.

- Para criar uma instância de banco de dados, siga as instruções do seu mecanismo de banco de dados em Criar uma instância de banco de dados do Amazon RDS.
- Para criar um cluster de banco de dados multi-AZ, siga as instruções relacionadas ao seu mecanismo de banco de dados em Criar um cluster de banco de dados multi-AZ para o Amazon RDS.

A captura de tela a seguir mostra a seção Performance Insights.



Se você escolher Enable Performance Insights (Habilitar o Performance Insights) terá as seguintes opções:

- Retenção (apenas para o modo padrão do Database Insights): o intervalo de tempo para reter os dados do Insights de Performance. A configuração de retenção no nível gratuito é Default (7 days) (Padrão (7 dias)). Para reter seus dados de performance por mais tempo, especifique entre 1 e 24 meses. Para obter mais informações sobre os períodos de retenção, consulte Preços e retenção de dados para o Performance Insights.
- AWS KMS key: especifica a sua AWS KMS key. O Performance Insights criptografa todos os possíveis dados sigilosos usando a sua chave do KMS. Os dados são criptografados em repouso e em trânsito. Para obter mais informações, consulte Alterar uma política do AWS KMS para o Insights de Performance.

Ativar ou desativar o Insights de Performance ao modificar uma instância de banco de dados ou um cluster de banco de dados multi-AZ

No console, é possível modificar uma instância de banco de dados ou um cluster de banco de dados multi-AZ para gerenciar o Insights de Performance.

Como gerenciar o Insights de Performance para uma instância de banco de dados ou um cluster de banco de dados multi-AZ usando o console

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.

2. Escolha Databases (Bancos de dados).
3. Escolha uma instância de banco de dados ou um cluster de banco de dados multi-AZ e selecione Modificar.
4. Para ativar o Insights de Performance, selecione Habilitar Insights de Performance.
   Para desativar o Insights de Performance, escolha a opção Database Insights: padrão e desmarque a opção Habilitar Insights de Performance.

   Se você escolher Enable Performance Insights (Habilitar o Performance Insights) terá as seguintes opções:

   - Retenção (apenas para o modo padrão do Database Insights): o intervalo de tempo para reter os dados do Insights de Performance. A configuração de retenção no nível gratuito é Default (7 days) (Padrão (7 dias)). Para reter seus dados de performance por mais tempo, especifique entre 1 e 24 meses. Para obter mais informações sobre os períodos de retenção, consulte Preços e retenção de dados para o Performance Insights.
   - AWS KMS key: especifique a sua chave do KMS. O Performance Insights criptografa todos os possíveis dados sigilosos usando a sua chave do KMS. Os dados são criptografados em repouso e em trânsito. Para obter mais informações, consulte Criptografar recursos do Amazon RDS.
5. Escolha Continue.
6. Em Scheduling of Modifications (Programação de modificações), escolha Apply immediately (Aplicar imediatamente). Se você escolher Apply during the next scheduled maintenance window (Aplicar durante a próxima janela de manutenção agendada), sua instância ignorará essa configuração e ativará o Performance Insights imediatamente.
7. Escolha Modify instance (Modificar instância).

AWS CLI

Ao usar o comando create-db-instance da AWS CLI, ative o Insights de Performance especificando `--enable-performance-insights` e definindo `--database-insights-mode` como `advanced` ou `standard`. Para desativar o Insights de Performance, especifique `--no-enable-performance-insights` e `database-insights-mode` como `standard`.

Você também pode especificar esses valores usando os seguintes comandos da AWS CLI:

- create-db-cluster
- modify-db-cluster

- create-db-instance-read-replica
- modify-db-instance
- restore-db-instance-from-s3

Quando você ativa o Performance Insights na CLI, é possível especificar o número de dias para retenção dos dados do Performance Insights com a opção `--performance-insights-retention-period`. Você pode especificar 7, *mês* * 31 (em que *mês* é um número de 1 a 23) ou 731. Por exemplo, se você quiser reter seus dados de desempenho por 3 meses, especifique 93, que é 3 * 31. O padrão são 7 dias. Para obter mais informações sobre os períodos de retenção, consulte Preços e retenção de dados para o Performance Insights.

O exemplo a seguir ativa o Performance Insights para `sample-db-cluster` e especifica que os dados do Performance Insights serão retidos por 93 dias (3 meses).

Para Linux, macOS ou Unix:

```
aws rds modify-db-cluster \
 --database-insights-mode standard \
    --db-cluster-identifier sample-db-instance \
    --enable-performance-insights \
    --performance-insights-retention-period 93
```

Para Windows:

```
aws rds modify-db-cluster ^
 --database-insights-mode standard ^
    --db-cluster-identifier sample-db-instance ^
    --enable-performance-insights ^
    --performance-insights-retention-period 93
```

Se você especificar um período de retenção como 94 dias, que não é um valor válido, o RDS emitirá um erro.

```
An error occurred (InvalidParameterValue) when calling the CreateDBInstance
 operation:
Invalid Performance Insights retention period. Valid values are: [7, 31, 62, 93,
 124, 155, 186, 217,
248, 279, 310, 341, 372, 403, 434, 465, 496, 527, 558, 589, 620, 651, 682, 713, 731]
```

> **Note**
>
> É possível ativar o Insights de Performance para uma instância em um cluster de banco de dados no qual o Insights de Performance não é gerenciado no nível do cluster.

RDS API

Ao criar uma instância de banco de dados usando a operação CreateDBInstance da API do Amazon RDS, ative o Performance Insights definindo `EnablePerformanceInsights` como `True`. Para desativar o Insights de Performance, defina `EnablePerformanceInsights` como `False` e `DatabaseInsightsMode` como `standard`.

Também é possível especificar o valor `EnablePerformanceInsights` usando as seguintes operações da API:

- CreateDBCluster (cluster de banco de dados multi-AZ)
- ModifyDBCluster (cluster de banco de dados multi-AZ)
- ModifyDBInstance
- CreateDBInstanceReadReplica
- RestoreDBInstanceFromS3

Quando você ativar o Performance Insights, é possível especificar a quantidade de tempo, em dias, para a retenção de dados do Performance Insights com o parâmetro `PerformanceInsightsRetentionPeriod`. Você pode especificar 7, *`mês`* * 31 (em que *`mês`* é um número de 1 a 23) ou `731`. Por exemplo, se você quiser reter seus dados de desempenho por 3 meses, especifique `93`, que é 3 * 31. O padrão são 7 dias. Para obter mais informações sobre os períodos de retenção, consulte Preços e retenção de dados para o Performance Insights.

Visão geral do Performance Schema para o Insights de Performance no Amazon RDS para MariaDB ou MySQL

O Performance Schema é um recurso opcional para monitorar a performance do tempo de execução do Amazon RDS para MariaDB ou MySQL em um nível inferior de detalhes. O Performance Schema foi projetado para causar impacto mínimo na performance do banco de dados. O Performance Insights é um recurso separado que você pode usar com ou sem o Performance Schema.

Tópicos

- Visão Geral do Performance Schema
- Performance Insights e Performance Schema
- Gerenciamento automático do Performance Schema pelo Performance Insights
- Efeito de uma reinicialização no Performance Schema
- Determinar se o Performance Insights está gerenciando o Performance Schema
- Ativar o Performance Schema para o Amazon RDS para MariaDB ou MySQL

Visão Geral do Performance Schema

O Performance Schema monitora os eventos nos bancos de dados MariaDB e MySQL. Um evento é uma ação do servidor de banco de dados que consome tempo e foi instrumentada para que as informações de tempo possam ser coletadas. São exemplos de evento:

- Chamadas de função
- Aguarda o sistema operacional
- Estágios da execução SQL
- Grupos de instruções SQL

O mecanismo de armazenamento `PERFORMANCE_SCHEMA` é um mecanismo para implementar o recurso Performance Schema. Esse mecanismo coleta dados de eventos usando instrumentação no código-fonte do banco de dados. O mecanismo armazena eventos em tabelas somente na memória no banco de dados `performance_schema`. É possível consultar `performance_schema` assim como quaisquer outras tabelas. Para obter mais informações, consulte MySQL Performance Schema no Guia de referência do MySQL.

Performance Insights e Performance Schema

O Performance Insights e o Performance Schema são recursos separados, mas estão conectados. O comportamento do Performance Insights para Amazon RDS para MariaDB ou MySQL depende se o Performance Schema está ativado e, em caso afirmativo, se o Performance Insights gerencia o Performance Schema automaticamente. A tabela a seguir descreve o comportamento.

Performance Schema ativado	Modo de gerenciamento do Performance Insights	Comportamento do Performance Insights
Sim	Automatico	• Coleta informações de monitoramento detalhadas e de nível inferior. • Coleta métricas de sessão ativas a cada segundo. • Exibe carga de banco de dados categorizada por eventos de espera detalhados, que você pode usar para identificar gargalos.
Sim	Manual	• Coleta eventos de espera e métricas por SQL • Coleta métricas de sessão ativas a cada segundo. • Relata estados de usuário, como inserção e envio, que não ajudam a identificar gargalos.
Não	N/D	• Não coleta eventos de espera, métricas por SQL nem outras informações detalhadas de monitoramento de nível inferior. • Coleta métricas de sessão ativas a cada cinco segundos em vez de cada segundo. • Relata estados de usuário, como inserção e envio, que não ajudam a identificar gargalos.

Gerenciamento automático do Performance Schema pelo Performance Insights

Quando você cria uma instância de banco de dados do Amazon RDS para MariaDB ou MySQL com o Performance Insights ativado, o Performance Schema também é ativado. Nesse caso, o

Performance Insights gerencia automaticamente seus parâmetros do Performance Schema. Essa é a configuração recomendada.

Quando o Insights de Performance gerencia o Performance Schema automaticamente, a Origem de `performance_schema` é `System default`.

> **Note**
>
> O gerenciamento automático do esquema de performance não é compatível com a classe de instância t4g.medium.

Se você alterar o valor do parâmetro `performance_schema` manualmente e, posteriormente, quiser reverter para o gerenciamento automático, consulte Ativar o Performance Schema para o Amazon RDS para MariaDB ou MySQL.

> **Important**
>
> Quando o Performance Insights ativa o Performance Schema, ele não altera os valores do grupo de parâmetros. No entanto, os valores são alterados nas instâncias de banco de dados que estão em execução. A única forma de ver os valores alterados é executar o comando `SHOW GLOBAL VARIABLES`.

Efeito de uma reinicialização no Performance Schema

O Performance Insights e o Performance Schema diferem em seus requisitos para reinicializações de instâncias de banco de dados:

Performance Schema

Para ativar ou desativar esse recurso, você deve reinicializar a instância de banco de dados.

Performance Insights

Para ativar ou desativar esse recurso, não é necessário reinicializar a instância de banco de dados.

Se o Performance Schema não estiver ativado no momento e você ativar o Performance Insights sem reinicializar a instância de banco de dados, o Performance Schema não será ativado.

Determinar se o Performance Insights está gerenciando o Performance Schema

Para descobrir se, no momento, o Insights de Performance está gerenciando o Performance Schema nas versões principais compatíveis do mecanismo, consulte a tabela a seguir.

Configuração do parâmetro performance_schema	Configuração da coluna Source (Fonte)	O Performance Insights está gerenciando o Performance Schema?
`0`	`System default`	Sim
`0` ou `1`	`Modified`	Não

No procedimento a seguir, você determinará se o Performance Insights está gerenciando o Performance Schema automaticamente.

Como saber se o Performance Insights está gerenciando automaticamente o Performance Schema

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. Escolha Parameter groups (Grupos de parâmetros).
3. Selecione o nome do grupo de parâmetros para a instância de banco de dados.
4. Insira **`performance_schema`** na barra de pesquisa.
5. Confira se o valor de Origem é o padrão do sistema e Valor é 0. Nesse caso, o Performance Insights está gerenciando o Performance Schema automaticamente.

   No exemplo mostrado aqui, o Performance Insights não está gerenciando o Performance Schema automaticamente.



Ativar o Performance Schema para o Amazon RDS para MariaDB ou MySQL

Suponha que o Performance Insights esteja ativado para sua instância de banco de dados ou o cluster de banco de dados multi-AZ, mas no momento, não está gerenciando o Performance Schema. Se você quiser permitir que o Performance Insights gerencie o Performance Schema automaticamente, conclua as etapas a seguir.

Como configurar o Performance Schema para gerenciamento automático

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. Escolha Parameter groups (Grupos de parâmetros).
3. Selecione o nome do grupo de parâmetros de sua instância de banco de dados ou cluster de banco de dados multi-AZ.
4. Escolha Editar.
5. Insira **`performance_schema`** na barra de pesquisa.
6. Selecione o parâmetro `performance_schema`.
7. Selecione Definir como valor padrão.
8. Confirme escolhendo Definir valores como padrão.
9. Escolha Salvar alterações.
10. Reinicialize a instância de banco de dados ou o cluster de banco de dados multi-AZ.

> **Important**
>
> Sempre que habilitar ou desabilitar o Performance Schema, você deverá reinicializar a instância de banco de dados ou o cluster de banco de dados multi-AZ.

Para obter mais informações sobre como modificar os parâmetros da instância, consulte Modificar parâmetros em um grupo de parâmetros de banco de dados no Amazon RDS. Para obter mais informações sobre o painel, consulte Análise de métricas usando o painel do Performance Insights. Para obter mais informações sobre o esquema de desempenho do MySQL, consulte MySQL Performance Schema (para 8.0) na MySQL Performance Schema (para 8.4) na documentação do MySQL.

Configurar políticas de acesso para o Performance Insights

Para acessar o Performance Insights, é necessário que uma entidade principal tenha as permissões apropriadas do AWS Identity and Access Management (IAM). Você pode conceder acesso das seguintes maneiras:

- Anexe a política gerenciada `AmazonRDSPerformanceInsightsReadOnly` para um conjunto de permissões ou perfil para acessar todas as operações somente de leitura da API do Insights de Performance. Anexe as seguintes permissões do CloudWatch: `GetMetricStatistics`, `ListMetrics` e `GetMetricData`. Para ter mais informações sobre as permissões do CloudWatch, consulte Referência de permissões do Amazon CloudWatch.
- Anexe a política gerenciada `AmazonRDSPerformanceInsightsFullAccess` para um conjunto de permissões ou perfil para acessar todas as operações da API do Insights de Performance. Anexe as seguintes permissões do CloudWatch: `GetMetricStatistics`, `ListMetrics` e `GetMetricData`. Para ter mais informações sobre as permissões do CloudWatch, consulte Referência de permissões do Amazon CloudWatch.
- Crie uma política do IAM personalizada e anexe ela a um conjunto de permissões ou perfil.

Se você especificou uma chave gerenciada pelo cliente quando ativou o Insights de Performance, verifique se os usuários em sua conta têm as permissões `kms:Decrypt` e `kms:GenerateDataKey` na AWS KMS key.

Nas seções a seguir, anexe uma política gerenciada da AWS a uma entidade principal do IAM, crie uma política personalizada do IAM, altere uma política do AWS KMS e conceda acesso detalhado ao Insights de Performance.

Tópicos

- Anexar a política AmazonRDSPerformanceInsightsReadOnly a uma entidade principal do IAM
- Anexar a política AmazonRDSPerformanceInsightsFullAccess a uma entidade principal do IAM
- Criação de uma política de IAM personalizada para o Performance Insights
- Alterar uma política do AWS KMS para o Insights de Performance
- Atribuir acesso refinado para o Insights de Performance

Anexar a política AmazonRDSPerformanceInsightsReadOnly a uma entidade principal do IAM

A `AmazonRDSPerformanceInsightsReadOnly` é uma política gerenciada pela AWS que concede acesso a todas as operações somente leitura da API do Insights de Performance do Amazon RDS.

Se você anexar `AmazonRDSPerformanceInsightsReadOnly` a um conjunto de permissões ou perfil, o destinatário poderá usar o Insights de Performance com outros atributos do console.

Para ter mais informações, consulte Política gerenciada pela AWS: AmazonRDSPerformanceInsightsReadOnly.

Anexar a política AmazonRDSPerformanceInsightsFullAccess a uma entidade principal do IAM

A `AmazonRDSPerformanceInsightsFullAccess` é uma política gerenciada pela AWS que concede acesso a todas as operações da API do Insights de Performance do Amazon RDS.

Se você anexar `AmazonRDSPerformanceInsightsFullAccess` a um conjunto de permissões ou perfil, o destinatário poderá usar o Insights de Performance com outros atributos do console.

Para ter mais informações, consulte Política gerenciada pela AWS: AmazonRDSPerformanceInsightsFullAccess.

Criação de uma política de IAM personalizada para o Performance Insights

Para usuários que não têm a política `AmazonRDSPerformanceInsightsReadOnly` ou `AmazonRDSPerformanceInsightsFullAcce` é possível conceder acesso ao Insights de Performance criando ou modificando uma política do IAM gerenciada pelo usuário. Quando você anexa a política a um conjunto de permissões ou perfil do IAM, o destinatário pode usar o Performance Insights.

Para criar uma política personalizada

1. Abra o console do IAM, em https://console.aws.amazon.com/iam/.
2. No painel de navegação, escolha Policies.
3. Escolha Criar política.
4. Na página Criar política, selecione a guia JSON.

5. Copie e cole o texto fornecido na seção do documento de política JSON no Guia de referência de políticas gerenciadas da AWS para AmazonRDSPerformanceInsightsReadOnly ou a política AmazonRDSPerformanceInsightsFullAccess.
6. Escolha Revisar política.
7. Forneça um nome para a política e, se preferir, uma descrição. Em seguida, escolha Criar política.

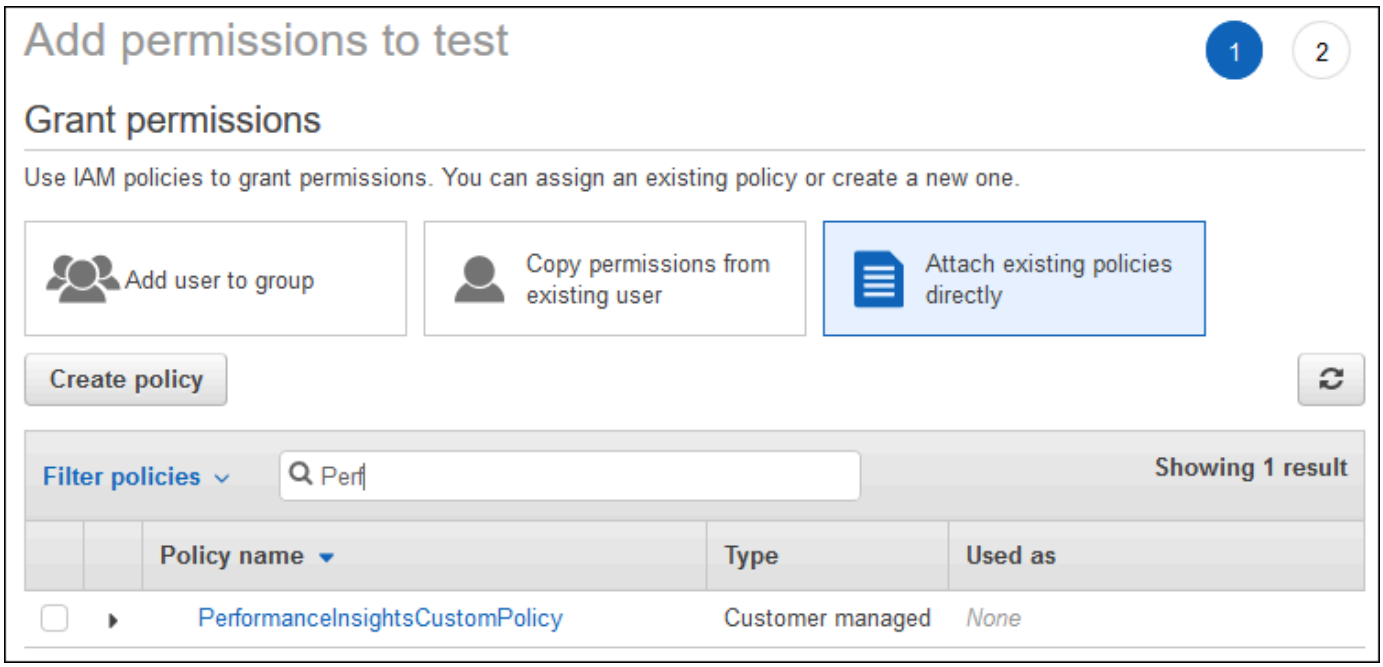
Agora você pode anexar a política a um conjunto de permissões ou perfil. O procedimento a seguir pressupõe que você já tem um usuário disponível para essa finalidade.

Como anexar a política a um usuário

1. Abra o console do IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação, escolha Usuários.
3. Escolha um usuário existente na lista.

> **Important**
>
> Para usar o Performance Insights, você deve ter acesso ao Amazon RDS e à política personalizada. Por exemplo, a política predefinida `AmazonRDSPerformanceInsightsReadOnly` concede acesso somente leitura ao Amazon RDS. Para obter mais informações, consulte Gerenciamento do acesso usando políticas.

4. Na página Summary (Resumo), escolha Add permissions (Adicionar permissões).
5. Escolha Anexar políticas existentes diretamente. Em Pesquisar, digite os primeiros caracteres do nome da política, conforme mostrado na imagem a seguir.



6. Escolha a política e, em seguida, escolha Próximo: revisar.
7. Escolha Adicionar permissões.

Alterar uma política do AWS KMS para o Insights de Performance

O Performance Insights usa uma AWS KMS key para criptografar dados sigilosos. Ao habilitar o Performance Insights por meio da API ou do console, você poderá seguir um destes procedimentos:

- Escolha o Chave gerenciada pela AWS padrão.

  O Amazon RDS usa a Chave gerenciada pela AWS para a sua nova instância de banco de dados. O Amazon RDS cria uma Chave gerenciada pela AWS para a sua Conta da AWS. A sua Conta da AWS tem uma Chave gerenciada pela AWS diferente para o Amazon RDS para cada Região da AWS.

- Escolha uma chave gerenciada pelo cliente.

  Se você especificar uma chave gerenciada pelo cliente, os usuários em sua conta que chamam a API do Performance Insights precisarão das permissões `kms:Decrypt` e `kms:GenerateDataKey` na chave do KMS. Você pode configurar essas permissões por meio de políticas do IAM. No entanto, recomendamos que você gerencie essas permissões por meio da política de chaves do KMS. Para obter mais informações, consulte Políticas de chaves no AWS KMS no Guia do desenvolvedor do AWS Key Management Service.

Example

O exemplo a seguir mostra como adicionar instruções à sua política da chaves do KMS. Essas instruções permitem acesso ao Performance Insights. Dependendo de como você usa a chave KMS, talvez você queira alterar algumas restrições. Antes de adicionar instruções à política, remova todos os comentários.

```
{
"Version" : "2012-10-17",
 "Id" : "your-policy",
 "Statement" : [ {
    //This represents a statement that currently exists in your policy.
 }
 ....,
 //Starting here, add new statement to your policy for Performance Insights.
 //We recommend that you add one new statement for every RDS instance
{
    "Sid" : "Allow viewing RDS Performance Insights",
    "Effect": "Allow",
    "Principal": {
        "AWS": [
            //One or more principals allowed to access Performance Insights
            "arn:aws:iam::444455556666:role/Role1"
        ]
    },
    "Action": [
        "kms:Decrypt",
        "kms:GenerateDataKey"
    ],
    "Resource": "*",
    "Condition" : {
        "StringEquals" : {
            //Restrict access to only RDS APIs (including Performance Insights).
            //Replace region with your AWS Region.
            //For example, specify us-west-2.
            "kms:ViaService" : "rds.region.amazonaws.com"
        },
        "ForAnyValue:StringEquals": {
            //Restrict access to only data encrypted by Performance Insights.
            "kms:EncryptionContext:aws:pi:service": "rds",
            "kms:EncryptionContext:service": "pi",

            //Restrict access to a specific RDS instance.
```

```
            //The value is a DbiResourceId.
           "kms:EncryptionContext:aws:rds:db-id": "db-AAAAABBBBBCCCCDDDDDEEEEE"
        }
    }
}
```

Como o Insights de Performance usa a chave do AWS KMS gerenciada pelo cliente

O Insights de Performance usa chaves gerenciadas pelo cliente para criptografar dados sigilosos. Ao ativar o Insights de Performance, você pode fornecer uma chave do AWS KMS por meio da API. O Insights de Performance cria permissões do KMS nessa chave. Ele usa a chave e executa as operações necessárias para processar dados sigilosos. Os dados sigilosos incluem campos como usuário, banco de dados, aplicação e texto de consulta SQL. O Insights de Performance garante que os dados permaneçam criptografados tanto em repouso quanto em trânsito.

Como o Insights de Performance e o IAM funcionam com o AWS KMS

O IAM concede permissões para APIs específicas. O Insights de Performance tem as seguintes APIs públicas, que você pode restringir usando políticas do IAM:

- `DescribeDimensionKeys`
- `GetDimensionKeyDetails`
- `GetResourceMetadata`
- `GetResourceMetrics`
- `ListAvailableResourceDimensions`
- `ListAvailableResourceMetrics`

Você pode usar as solicitações de API a seguir para obter dados sigilosos.

- `DescribeDimensionKeys`
- `GetDimensionKeyDetails`
- `GetResourceMetrics`

Quando você usa a API para obter dados sigilosos, o Insights de Performance utiliza as credenciais do chamador. Essa verificação garante que o acesso a dados sigilosos seja limitado àqueles com acesso à chave do KMS.

Ao chamar essas APIs, você precisa de permissões para chamar a API por meio da política do IAM e de permissões para invocar a ação `kms:decrypt` por meio da política de chave AWS KMS.

A API `GetResourceMetrics` pode retornar dados sigilosos e não sigilosos. Os parâmetros da solicitação determinam se a resposta deve incluir dados sigilosos. A API retorna dados sigilosos quando a solicitação inclui uma dimensão confidencial nos parâmetros filtrar ou agrupar por.

Para obter mais informações sobre as dimensões que você pode usar com a API `GetResourceMetrics`, consulte DimensionGroup.

Example Exemplos

O seguinte exemplo solicita dados sigilosos para o grupo `db.user`:

```
POST / HTTP/1.1
Host: <Hostname>
Accept-Encoding: identity
X-Amz-Target: PerformanceInsightsv20180227.GetResourceMetrics
Content-Type: application/x-amz-json-1.1
User-Agent: <UserAgentString>
X-Amz-Date: <Date>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=<Headers>,
 Signature=<Signature>
Content-Length: <PayloadSizeBytes>
{
  "ServiceType": "RDS",
  "Identifier": "db-ABC1DEFGHIJKL2MNOPQRSTUV3W",
  "MetricQueries": [
    {
      "Metric": "db.load.avg",
      "GroupBy": {
        "Group": "db.user",
        "Limit": 2
      }
    }
  ],
  "StartTime": 1693872000,
  "EndTime": 1694044800,
  "PeriodInSeconds": 86400
}
```

Example

O seguinte exemplo solicita dados não sigilosos para o grupo `db.load.avg`:

```
POST / HTTP/1.1
Host: <Hostname>
Accept-Encoding: identity
X-Amz-Target: PerformanceInsightsv20180227.GetResourceMetrics
Content-Type: application/x-amz-json-1.1
User-Agent: <UserAgentString>
X-Amz-Date: <Date>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=<Headers>,
 Signature=<Signature>
Content-Length: <PayloadSizeBytes>
{
    "ServiceType": "RDS",
    "Identifier": "db-ABC1DEFGHIJKL2MNOPQRSTUV3W",
    "MetricQueries": [
        {
            "Metric": "db.load.avg"
        }
    ],
    "StartTime": 1693872000,
    "EndTime": 1694044800,
    "PeriodInSeconds": 86400
}
```

Atribuir acesso refinado para o Insights de Performance

O controle de acesso refinado oferece formas adicionais de controlar o acesso para o Insights de Performance. Esse controle de acesso pode permitir ou negar acesso a dimensões individuais para as ações `GetResourceMetrics`, `DescribeDimensionKeys` e `GetDimensionKeyDetails` do Insights de Performance. Para usar o acesso refinado, especifique as dimensões na política do IAM usando chaves de condição. A avaliação do acesso segue a lógica de avaliação da política do IAM. Para obter mais informações, consulte Lógica da avaliação de política no Guia do usuário do IAM. Se a declaração de política do IAM não especificar nenhuma dimensão, a declaração controlará o acesso a todas as dimensões da ação especificada. Para ver a lista de dimensões disponíveis, consulte DimensionGroup.

Para descobrir as dimensões que as credenciais estão autorizadas a acessar, use o parâmetro `AuthorizedActions` em `ListAvailableResourceDimensions` e especifique a ação. Os valores permitidos para `AuthorizedActions` são os seguintes:

- `GetResourceMetrics`
- `DescribeDimensionKeys`
- `GetDimensionKeyDetails`

Por exemplo, se você especificar `GetResourceMetrics` como o parâmetro `AuthorizedActions`, `ListAvailableResourceDimensions` exibirá a lista de dimensões que a ação `GetResourceMetrics` está autorizada a acessar. Se você especificar várias ações no parâmetro `AuthorizedActions`, `ListAvailableResourceDimensions` exibirá uma interseção de dimensões que essas ações estão autorizadas a acessar.

Example

O exemplo a seguir concede acesso às dimensões especificadas para as ações `GetResourceMetrics` e `DescribeDimensionKeys`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowToDiscoverDimensions",
            "Effect": "Allow",
            "Action": [
                "pi:ListAvailableResourceDimensions"
            ],
            "Resource": [
                "arn:aws:pi:us-east-1:123456789012:metrics/rds/db-
ABC1DEFGHIJKL2MNOPQRSTUV3W"
            ]
        },
        {
            "Sid": "SingleAllow",
            "Effect": "Allow",
            "Action": [
                "pi:GetResourceMetrics",
                "pi:DescribeDimensionKeys"
            ],
            "Resource": [
```

```
                "arn:aws:pi:us-east-1:123456789012:metrics/rds/db-
ABC1DEFGHIJKL2MNOPQRSTUV3W"
            ],
            "Condition": {
                "ForAllValues:StringEquals": {
                    // only these dimensions are allowed. Dimensions not included in
                    // a policy with "Allow" effect will be denied
                    "pi:Dimensions": [
                        "db.sql_tokenized.id",
                        "db.sql_tokenized.statement"
                    ]
                }
            }
        }


    ]
}
```

Veja abaixo a resposta para a dimensão solicitada:

```
 // ListAvailableResourceDimensions API
// Request
{
    "ServiceType": "RDS",
    "Identifier": "db-ABC1DEFGHIJKL2MNOPQRSTUV3W",
    "Metrics": [ "db.load" ],
    "AuthorizedActions": ["DescribeDimensionKeys"]
}

// Response
{
    "MetricDimensions": [ {
        "Metric": "db.load",
        "Groups": [
            {
                "Group": "db.sql_tokenized",
                "Dimensions": [
                    { "Identifier": "db.sql_tokenized.id" },
                  //  { "Identifier": "db.sql_tokenized.db_id" }, // not included
 because not allows in the IAM Policy
```

```
                    { "Identifier": "db.sql_tokenized.statement" }
                ]
            }

        ] }
    ]
}
```

O exemplo a seguir especifica uma permissão e duas negações de acesso às dimensões.

```
{
    "Version": "2012-10-17",
    "Statement": [
          {
            "Sid": "AllowToDiscoverDimensions",
            "Effect": "Allow",
            "Action": [
                "pi:ListAvailableResourceDimensions"
            ],
            "Resource": [
                "arn:aws:pi:us-east-1:123456789012:metrics/rds/db-
ABC1DEFGHIJKL2MNOPQRSTUV3W"
            ]
          },

          {
            "Sid": "O01AllowAllWithoutSpecifyingDimensions",
            "Effect": "Allow",
            "Action": [
                "pi:GetResourceMetrics",
                "pi:DescribeDimensionKeys"
            ],
            "Resource": [
                "arn:aws:pi:us-east-1:123456789012:metrics/rds/db-
ABC1DEFGHIJKL2MNOPQRSTUV3W"
            ]
        },

        {
            "Sid": "O01DenyAppDimensionForAll",
            "Effect": "Deny",
            "Action": [
                "pi:GetResourceMetrics",
```

```
                "pi:DescribeDimensionKeys"
            ],
            "Resource": [
                "arn:aws:pi:us-east-1:123456789012:metrics/rds/db-
ABC1DEFGHIJKL2MNOPQRSTUV3W"
            ],
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "pi:Dimensions": [
                        "db.application.name"
                    ]
                }
            }
        },

        {
            "Sid": "O01DenySQLForGetResourceMetrics",
            "Effect": "Deny",
            "Action": [
                "pi:GetResourceMetrics"
            ],
            "Resource": [
                "arn:aws:pi:us-east-1:123456789012:metrics/rds/db-
ABC1DEFGHIJKL2MNOPQRSTUV3W"
            ],
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "pi:Dimensions": [
                        "db.sql_tokenized.statement"
                    ]
                }
            }
        }
    ]
}
```

Veja abaixo a resposta para as dimensões solicitadas:

```
   // ListAvailableResourceDimensions API
// Request
{
```

```
    "ServiceType": "RDS",
    "Identifier": "db-ABC1DEFGHIJKL2MNOPQRSTUV3W",
    "Metrics": [ "db.load" ],
    "AuthorizedActions": ["GetResourceMetrics"]
}

// Response
{
    "MetricDimensions": [ {
        "Metric": "db.load",
        "Groups": [
            {
                "Group": "db.application",
                "Dimensions": [

                  // removed from response because denied by the IAM Policy
                  //  { "Identifier": "db.application.name" }
                ]
            },
            {
                "Group": "db.sql_tokenized",
                "Dimensions": [
                    { "Identifier": "db.sql_tokenized.id" },
                    { "Identifier": "db.sql_tokenized.db_id" },

                  // removed from response because denied by the IAM Policy
                  //  { "Identifier": "db.sql_tokenized.statement" }
                ]
            },
            ...
        ] }
    ]
}
```

```
// ListAvailableResourceDimensions API
// Request
{
    "ServiceType": "RDS",
    "Identifier": "db-ABC1DEFGHIJKL2MNOPQRSTUV3W",
    "Metrics": [ "db.load" ],
    "AuthorizedActions": ["DescribeDimensionKeys"]
}
```

```
// Response
{
    "MetricDimensions": [ {
        "Metric": "db.load",
        "Groups": [
            {
                "Group": "db.application",
                "Dimensions": [
                  // removed from response because denied by the IAM Policy
                  //  { "Identifier": "db.application.name" }
                ]
            },
            {
                "Group": "db.sql_tokenized",
                "Dimensions": [
                    { "Identifier": "db.sql_tokenized.id" },
                    { "Identifier": "db.sql_tokenized.db_id" },

                  // allowed for DescribeDimensionKeys because our IAM Policy
                  // denies it only for GetResourceMetrics
                    { "Identifier": "db.sql_tokenized.statement" }
                ]
            },
            ...
        ] }
    ]
}
```

Análise de métricas usando o painel do Performance Insights

> **Important**
>
> A AWS anunciou a data de fim da vida útil do Insights de Performance: 30 de novembro de 2025. Após essa data, o Amazon RDS não oferecerá mais suporte à experiência do console, aos períodos de retenção flexíveis (de 1 a 24 meses) e aos preços associados do Insights de Performance. A API do Insights de Performance continuará existindo sem alterações de preço. Os custos da API do Insights de Performance aparecerão em sua fatura da com o custo do CloudWatch Database Insights.
> Recomendamos que você atualize quaisquer instâncias de banco de dados usando o nível pago do Insights de Performance para o modo avançado do Database Insights antes de 30

de novembro de 2025. Para ter informações sobre como atualizar para o modo avançado do Database Insights, consulte Ativação do modo Avançado do Database Insights para Amazon RDS.

Se você não tomar nenhuma medida, as instâncias de banco de dados que estão usando o Insights de Performance usarão como padrão o modo padrão do Database Insights. Com o modo padrão do Database Insights, você pode perder o acesso ao histórico de dados de desempenho após sete dias e talvez não consiga usar planos de execução e recursos de análise sob demanda no console do Amazon RDS. Após 30 de novembro de 2025, somente o modo avançado do Database Insights permitirá planos de execução e análise sob demanda.

Com o CloudWatch Database Insights, você pode monitorar a carga do banco de dados para sua frota de bancos de dados e analisar e solucionar problemas de desempenho em grande escala. Para ter mais informações sobre o Database Insights, consulte Monitoramento de bancos de dados do Amazon RDS com o CloudWatch Database Insights. Para obter informações de definição de preço, consulte Definição de preço do Amazon CloudWatch.

O painel do Performance Insights contém informações de performance do banco de dados para ajudar você a analisar e solucionar problemas de performance. Na página principal do painel, você pode visualizar informações sobre a carga do banco de dados. Você pode separar a carga de banco de dados por dimensões como eventos de espera ou SQL.

Painel do Performance Insights

- Visão geral do painel do Performance Insights
- Acessar o painel do Performance Insights.
- Analisar a carga do banco de dados por eventos de espera
- Analisar a performance do banco de dados por um período
- Analisar consultas com a guia SQL principal no Insights de Performance
- Analisar a carga superior de PDBs da Oracle
- Analisar planos de execução usando o painel do Insights de Performance para o Amazon RDS

Visão geral do painel do Performance Insights

O painel é a maneira mais fácil de interagir com o Performance Insights. O exemplo a seguir mostra o painel de uma instância de banco de dados do PostgreSQL.

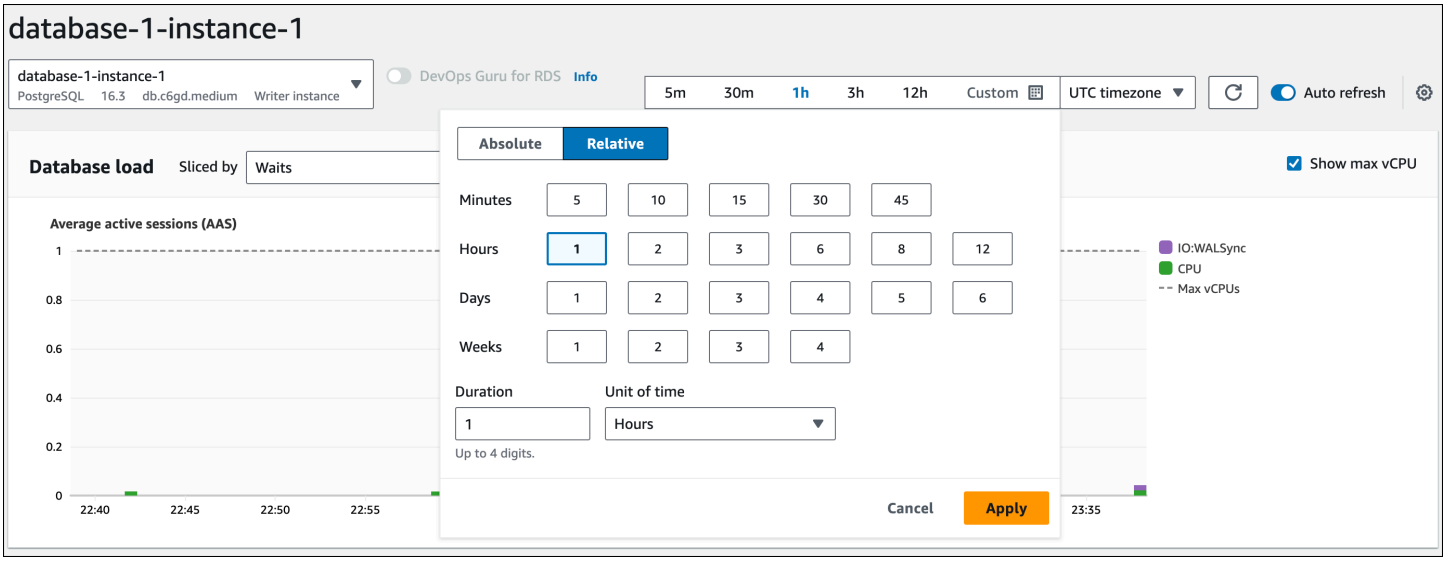


Tópicos

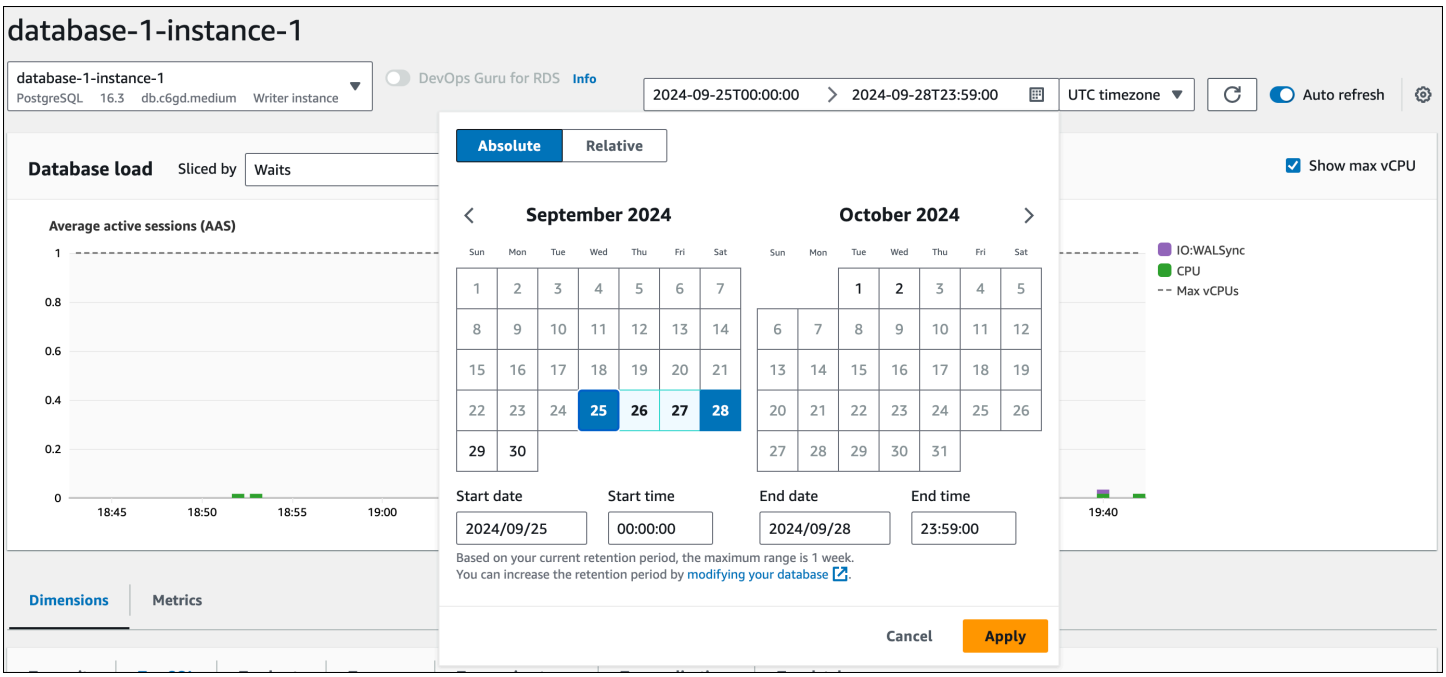
- Filtro de intervalo de tempo
- Gráfico de métricas de contador
- Gráfico de carga do banco de dados
- Tabela Top dimensions (Principais dimensões)

Filtro de intervalo de tempo

Por padrão, o painel do Performance Insights exibe a carga de banco de dados da última hora. Você pode ajustar esse intervalo para cinco minutos ou dois anos. Também é possível selecionar um intervalo relativo personalizado.



Você pode selecionar um intervalo absoluto com data e hora de início e término. O exemplo a seguir mostra o intervalo de tempo que começa à meia-noite de 25/9/24 e termina às 23h59 de 28/9/24.



Por padrão, o fuso horário do painel do Insights de Performance é o Tempo Universal Coordenado (UTC). Também é possível escolher o fuso horário local.

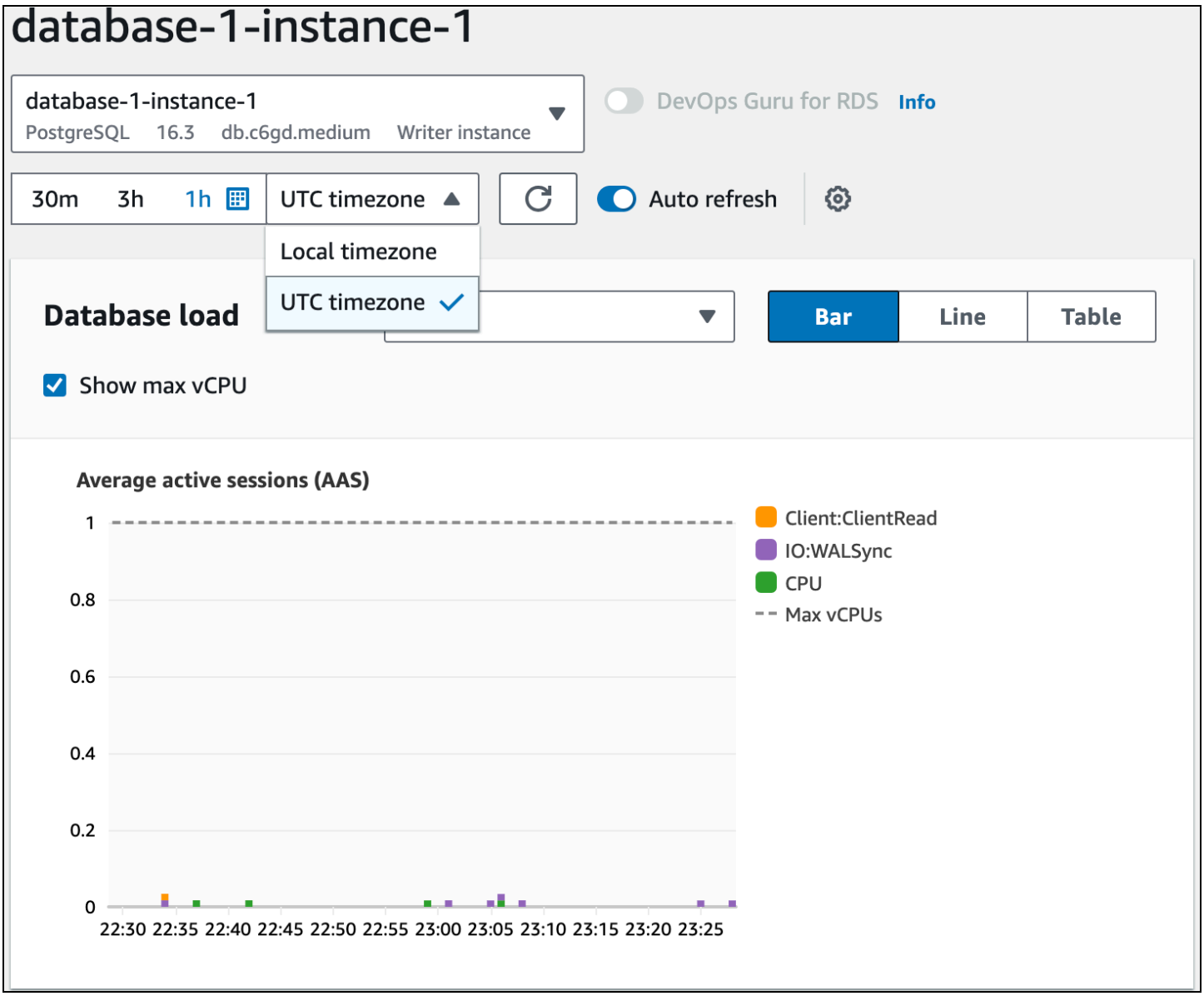


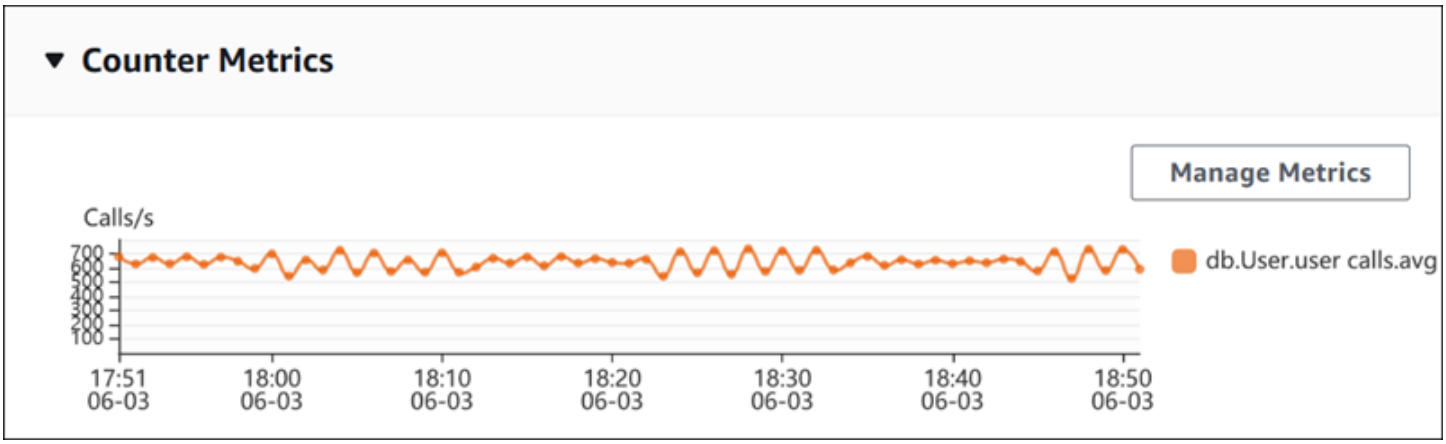
Gráfico de métricas de contador

Com métricas de contador, você pode personalizar o painel do Performance Insights para incluir até 10 gráficos adicionais. Esses gráficos mostram uma seleção de dezenas de métricas de performance do sistema operacional e do banco de dados. Você pode correlacionar essas informações à carga do banco de dados para ajudar a identificar e analisar problemas de performance.

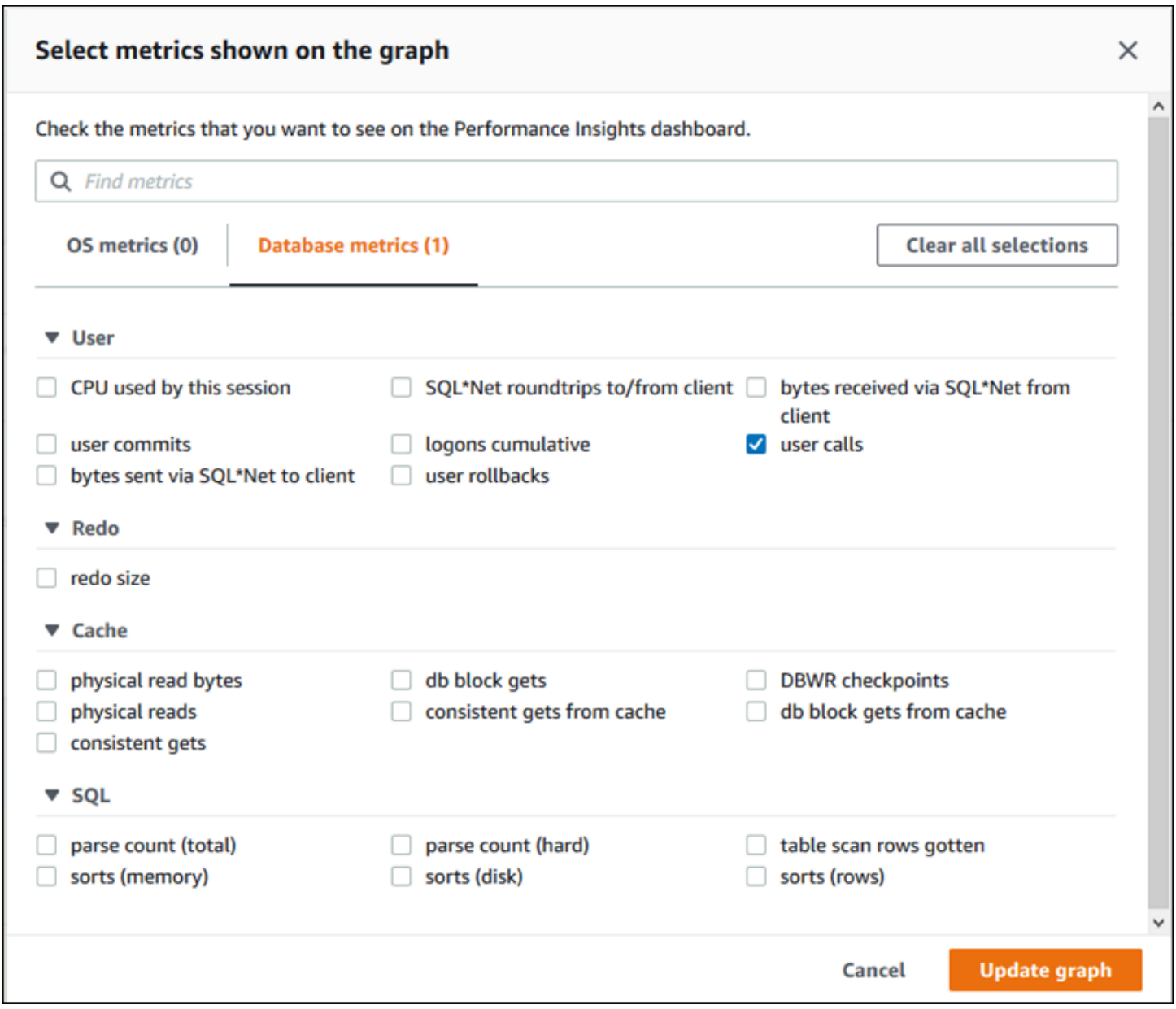
O gráfico Métricas de contador exibe dados dos contadores de performance. As métricas padrão dependem do mecanismo de banco de dados:

- MySQL e MariaDB – `db.SQL.Innodb_rows_read.avg`
- Oracle – `db.User.user calls.avg`

- Microsoft SQL Server – `db.Databases.Active Transactions(_Total).avg`
- PostgreSQL – `db.Transactions.xact_commit.avg`



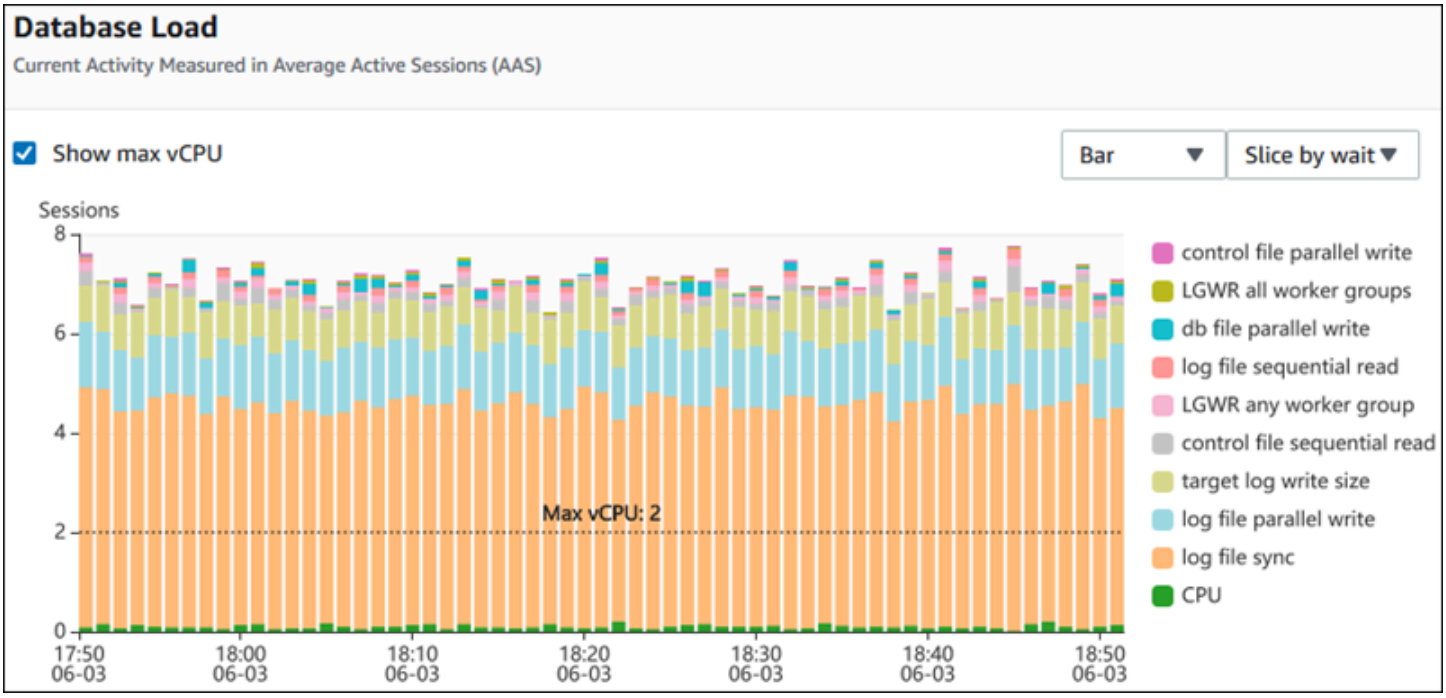
Para alterar os contadores de performance, escolha Manage Metrics (Gerenciar métricas). É possível selecionar várias Métricas de SO ou Métricas de banco de dados, conforme mostrado na captura de tela a seguir. Para ver detalhes de qualquer métrica, passe o mouse sobre o nome da métrica.



Para obter descrições das métricas de contador que você pode adicionar a cada mecanismo de banco de dados, consulte Métricas de contadores do Performance Insights.

Gráfico de carga do banco de dados

O gráfico Database load (Carga do banco de dados) mostra como a atividade do banco de dados se compara à capacidade da instância de banco de dados representada pela linha Max vCPU (Máximo de vCPU). Por padrão, o gráfico de linhas empilhadas representa a carga do banco de dados como sessões ativas médias por unidade de tempo. A carga do banco de dados é separada (agrupada) por estados de espera.

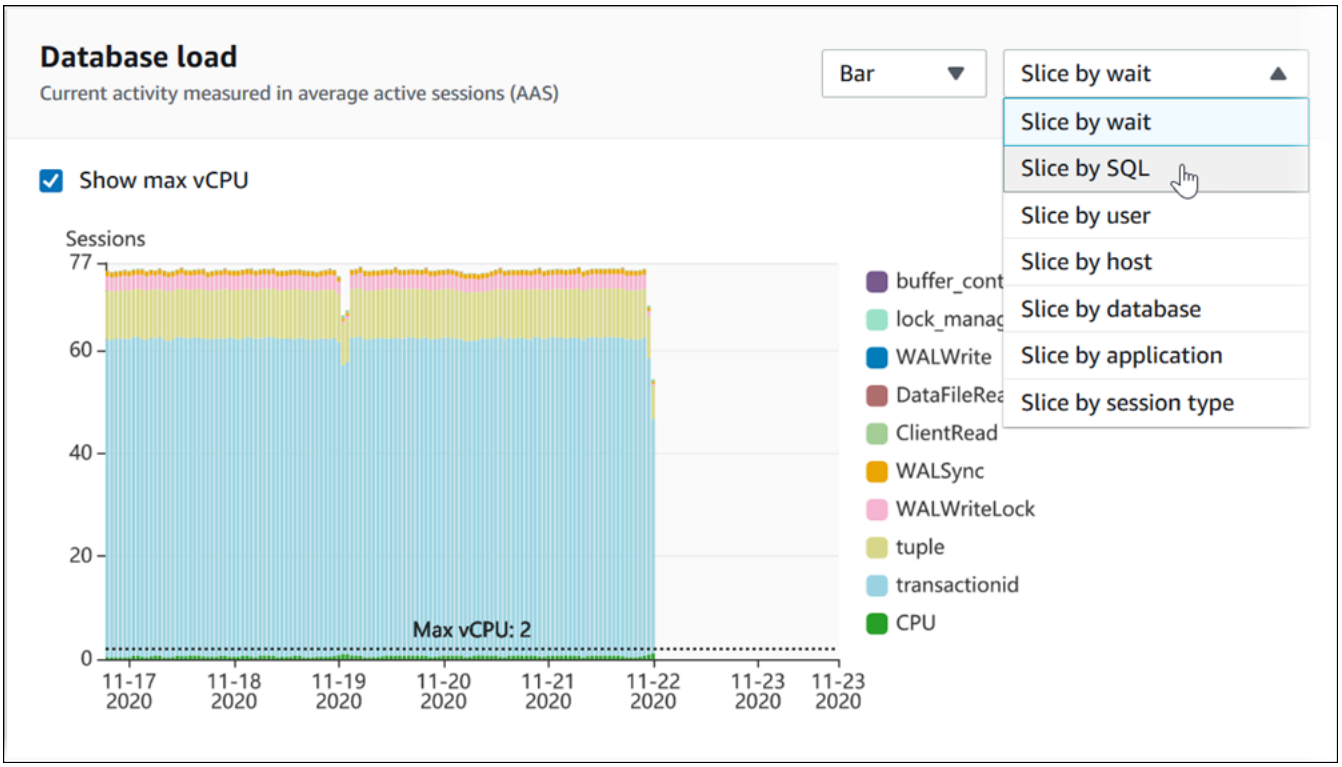


Carga de banco de dados separada por dimensões

Você pode optar por exibir a carga como sessões ativas agrupadas por quaisquer dimensões aceitas. A tabela a seguir mostra quais dimensões são aceitas pelos diferentes mecanismos.

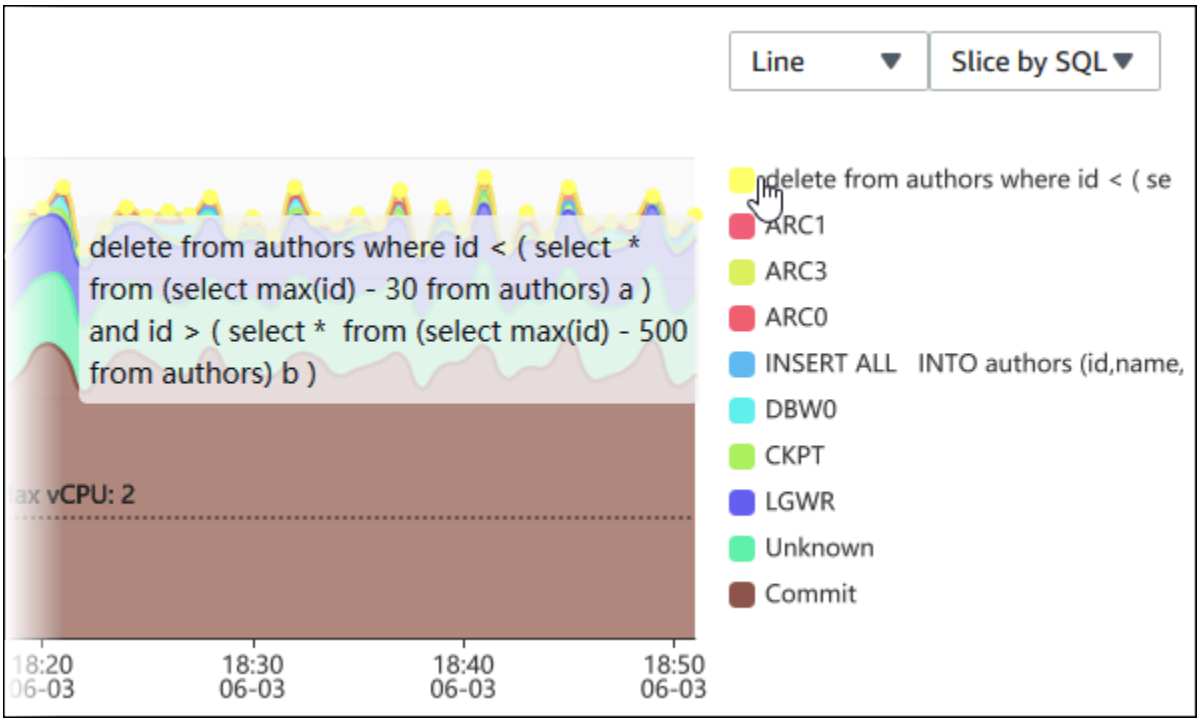
Dimensão	Oracle	SQL Server	PostgreSQL	MySQL
Host	Sim	Sim	Sim	Sim
SQL	Sim	Sim	Sim	Sim
Usuário	Sim	Sim	Sim	Sim
Espera	Sim	Sim	Sim	Sim
Planos	Sim	Não	Não	Não
Aplicação	Não	Não	Sim	Não
Banco de dados	Não	Não	Sim	Sim
Tipo de sessão	Não	Não	Sim	Não

A imagem a seguir mostra as dimensões de uma instância de banco de dados do PostgreSQL.



Detalhes de carga de banco de dados para um item de dimensão

Para ver detalhes sobre um item de carga de banco de dados dentro de uma dimensão, passe o mouse sobre o nome do item. A imagem a seguir mostra detalhes de uma instrução SQL.



Para ver detalhes de qualquer item do período selecionado na legenda, passe o mouse sobre esse item.

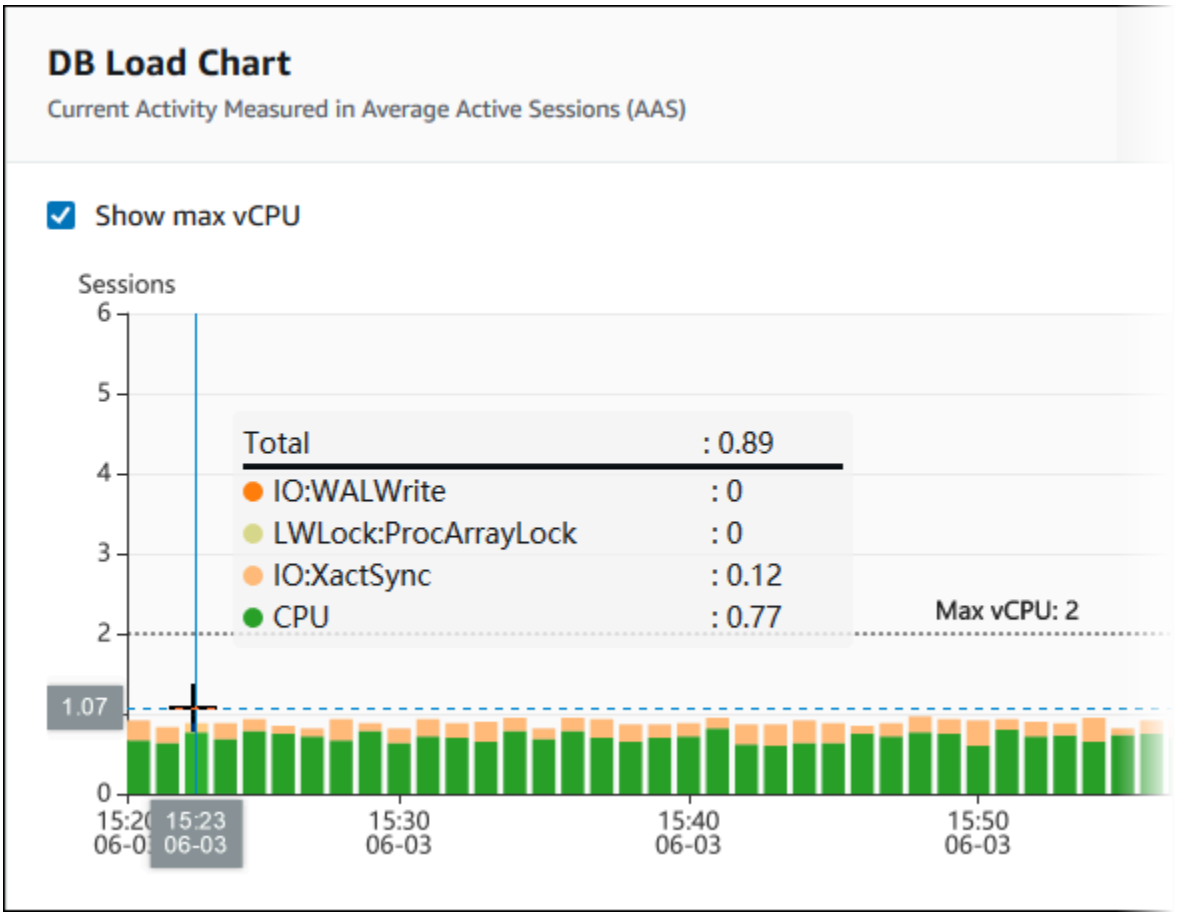
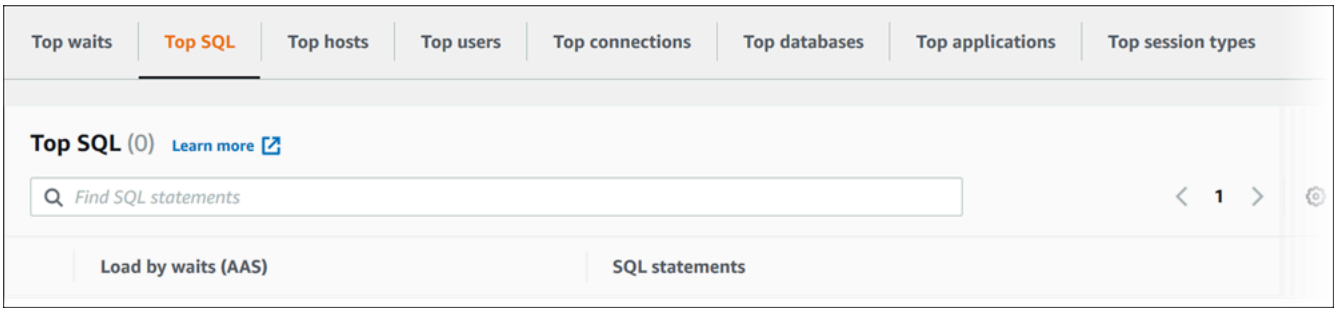


Tabela Top dimensions (Principais dimensões)

A tabela Principais dimensões separa a carga do banco de dados com base em diferentes dimensões. Uma dimensão é uma categoria ou “pedaços” de diferentes características de uma carga de banco de dados. Se a dimensão for SQL, Top SQL (SQL principal) mostrará as instruções SQL que mais contribuem para a carga do banco de dados.



Escolha qualquer uma das guias de dimensão a seguir.

Tab	Descrição	Mecanismos com suporte
SQL principal	As instruções SQL que estão sendo executadas no momento	Todos
Esperas principais	O evento para o qual o backend do banco de dados está aguardando	Todos
Hosts principais	O nome do host do cliente conectado	Todos
Principais usuários	O usuário conectado ao banco de dados	Todos
Principais bancos de dados	O nome do banco de dados ao qual o cliente está conectado	Somente PostgreSQL, MySQL, MariaDB e SQL Server
Principais aplicações	O nome da aplicação que está conectada ao banco de dados	Somente PostgreSQL e SQL Server
Principais tipos de sessão	O tipo da sessão atual	Apenas PostgreSQL

Para aprender a analisar consultas utilizando a guia Top SQL (SQL principal), consulte Visão geral da guia Top SQL (SQL principal).

Acessar o painel do Performance Insights.

O Amazon RDS fornece uma visão consolidada das métricas do Insights de Performance e do CloudWatch no painel do Insights de Performance.

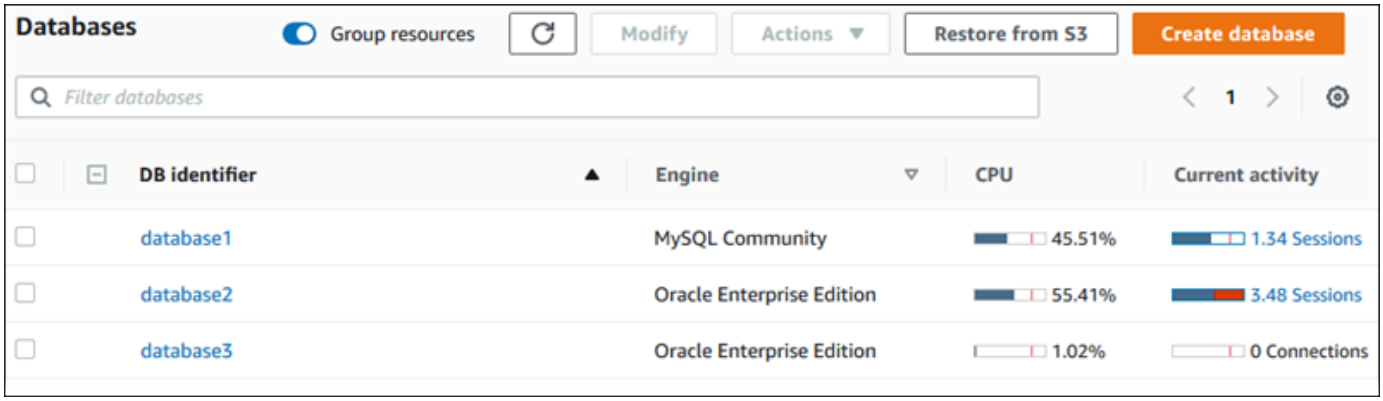
Para acessar o painel do Performance Insights, use o procedimento a seguir.

Para visualizar o painel do Performance Insights no Console de gerenciamento da AWS

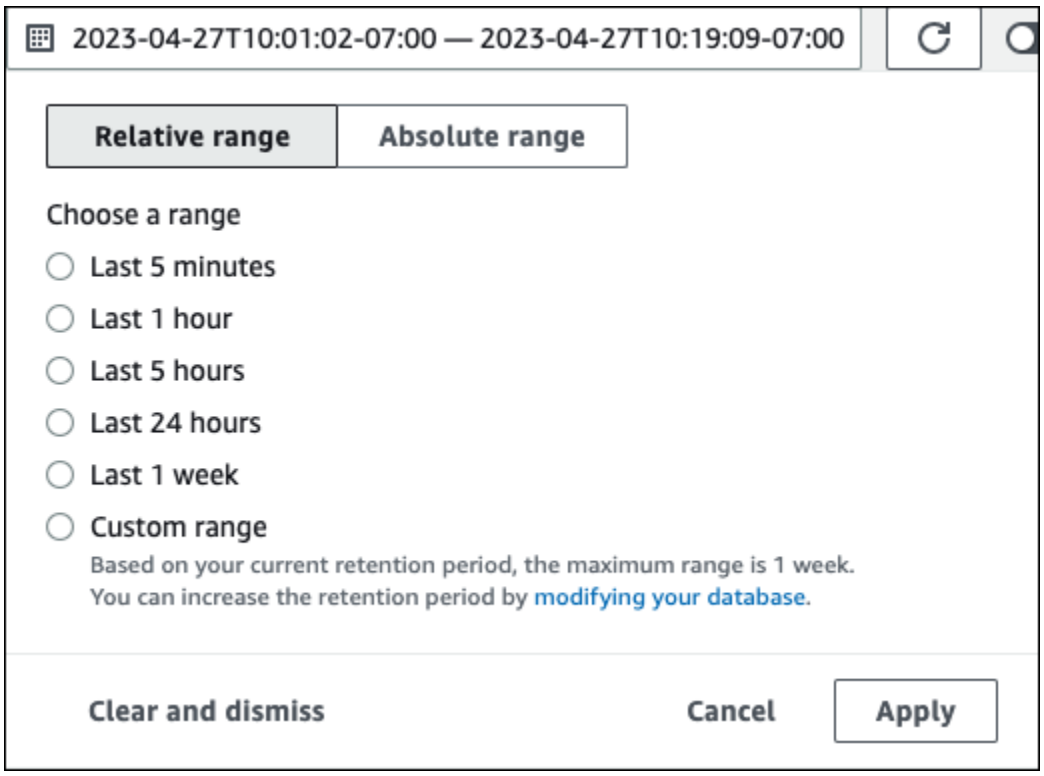
1. Abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Performance Insights.

3. Escolha uma instância de banco de dados.

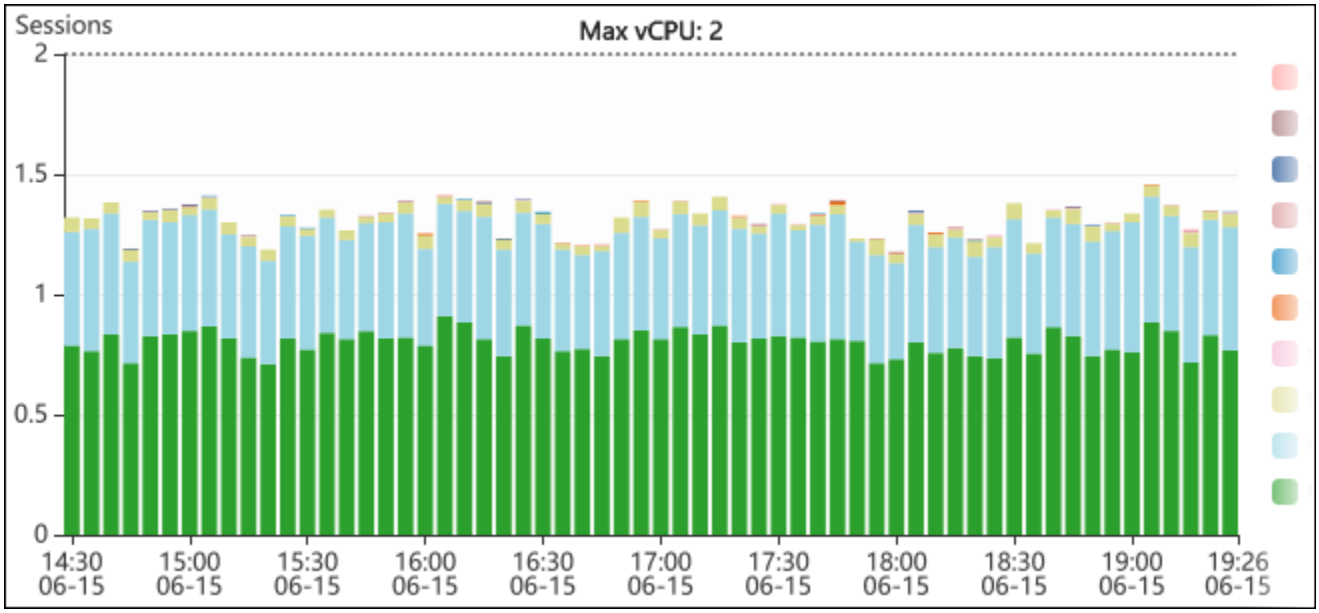
   Para as instâncias de banco de dados com o Insights de Performance ativado, você também pode acessar o painel do Insights de Performance escolhendo o item Sessões na lista de instâncias de banco de dados. Em Atividade atual, o item Sessões mostra a carga de banco de dados em sessões ativas médias nos últimos cinco minutos. A carga é mostrada graficamente por meio de barras. Quando a barra está vazia, a instância de banco de dados está ociosa. À medida que a carga aumenta, a barra é preenchida com a cor azul. Quando a carga ultrapassa o número de CPUs virtuais (vCPUs) na classe da instância de banco de dados, a barra se torna vermelha, indicando um possível gargalo.



4. (Opcional) Escolha a data ou o intervalo de tempo no canto superior direito e especifique um intervalo de tempo relativo ou absoluto diferente. Agora você pode especificar um período e gerar um relatório de análise de performance do banco de dados. O relatório fornece as recomendações e os insights identificados. Para ter mais informações, consulte Criar um relatório de análise de performance no Insights de Performance.

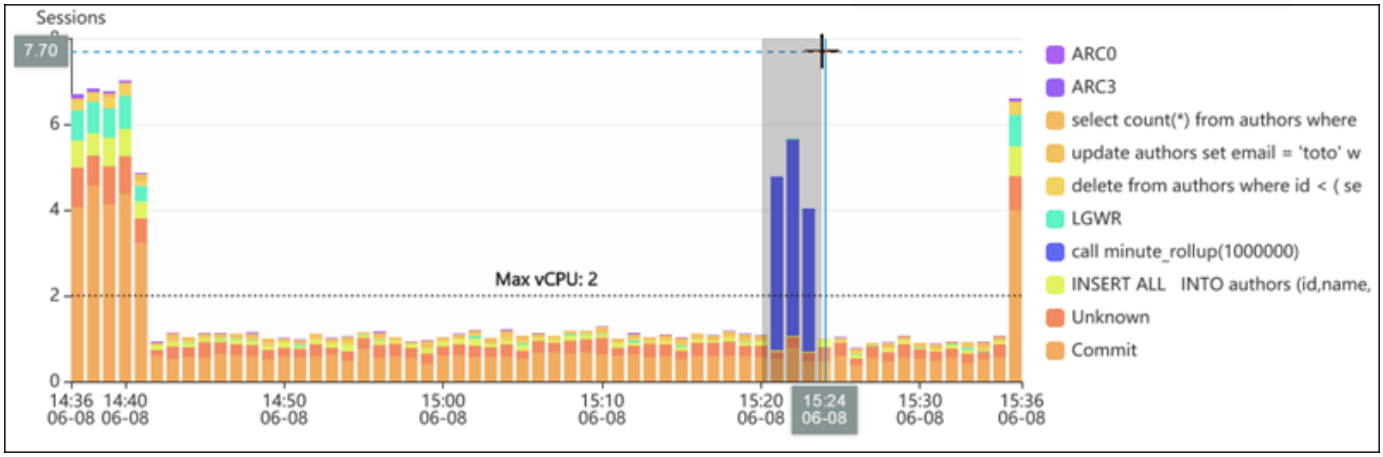


Na captura de tela a seguir, o intervalo da carga do banco de dados é de 5 horas.

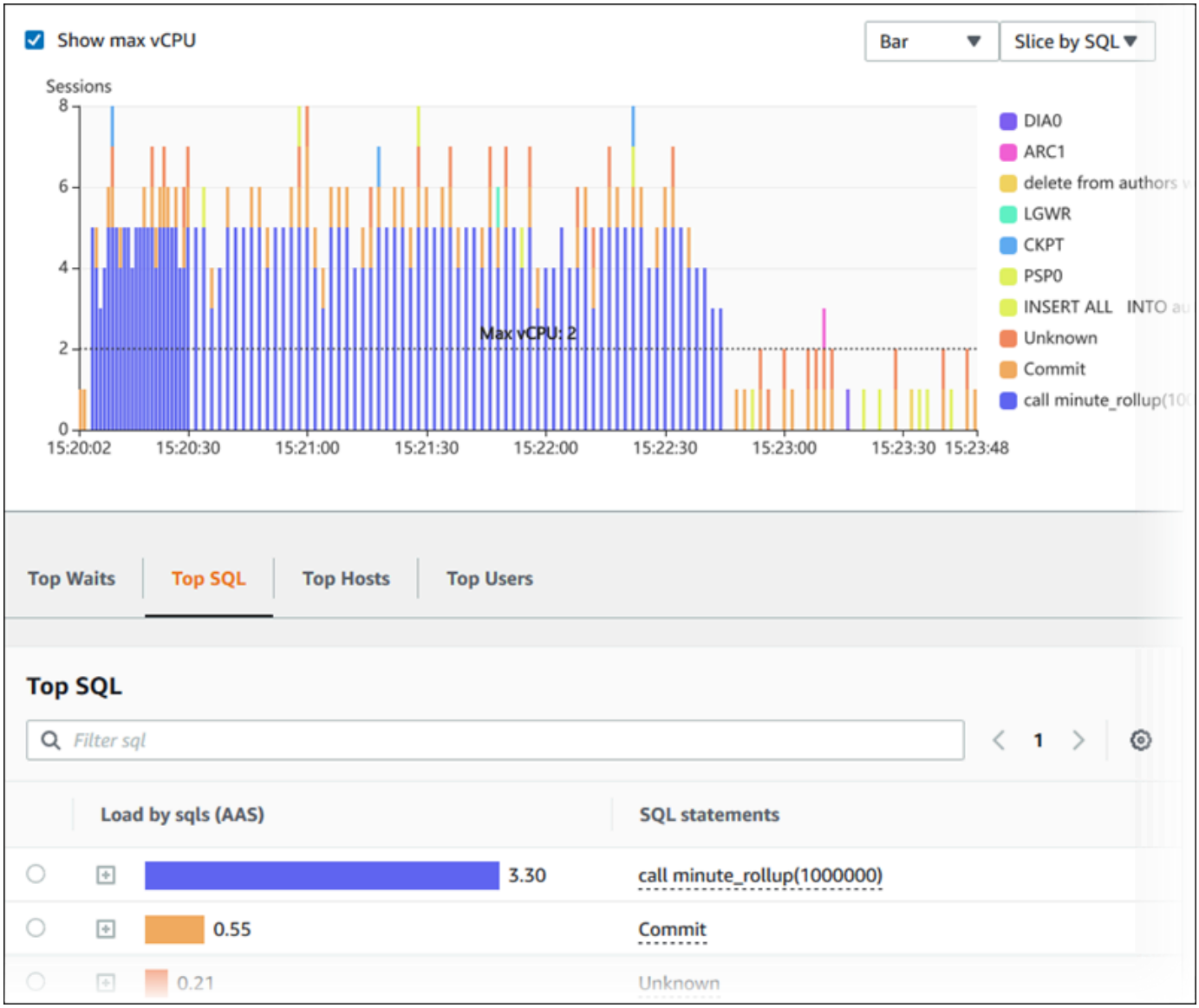


5. (Opcional) Para ampliar uma parte do grafo de carga de banco de dados, escolha a hora de início e arraste até o final do período desejado.

   A área selecionada é destacada no grafo de carga de banco de dados.



Ao soltar o mouse, o grafo de carga de banco de dados amplia a Região da AWS selecionada e a tabela Top dimensions (Dimensões principais) é recalculada.



6. (Opcional) Para atualizar seus dados automaticamente, selecione Atualização automática.



O painel de Insights de Performance é atualizado automaticamente com novos dados. A taxa de atualização depende da quantidade de dados exibida:

- 5 minutos atualiza a cada 10 segundos.
- 1 hora atualiza a cada 5 minutos.
- 5 horas atualiza a cada 5 minutos.
- 24 horas atualiza a cada 30 minutos.
- 1 semana atualiza a cada hora.

- 1 mês atualiza todos os dias.

Analisar a carga do banco de dados por eventos de espera

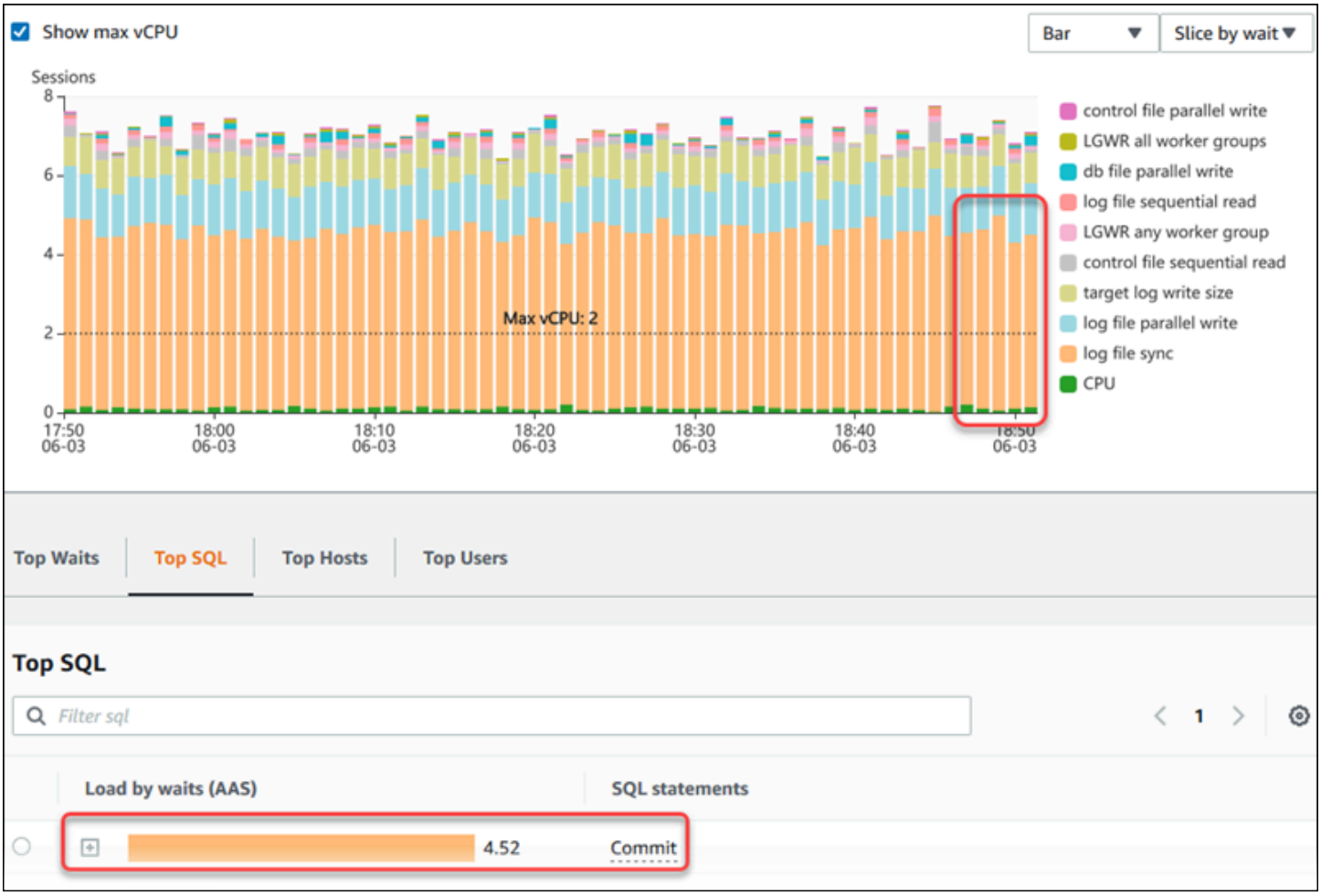
Se o gráfico Database load (Carga do banco de dados) mostrar um gargalo, você poderá descobrir de onde vem essa carga. Para fazer isso, examine a tabela de principais itens de carga abaixo do gráfico Database load (Carga do banco de dados). Escolha um item específico, como uma consulta SQL ou um usuário, para ver detalhes sobre ele.

A carga de banco de dados agrupada por espera e as principais consultas SQL compõem a visualização padrão do painel do Performance Insights. Em geral, essa combinação fornece os melhores insights sobre problemas de performance. A carga de banco de dados agrupada por espera mostra se há algum gargalo de recursos ou de concorrências no banco de dados. Nesse caso, a guia SQL da tabela Top Load Items (Principais itens de carga) mostra quais consultas estão gerando essa carga.

Seu fluxo de trabalho típico para diagnosticar problemas de performance é o seguinte:

1. Analise o gráfico Carregamento do banco de dados e veja se há casos de cargas de banco de dados que estejam ultrapassando a linha Máximo de CPU.
2. Se houver, examine o gráfico Database load (Carga do banco de dados) e identifique quais estados de espera são os principais responsáveis por isso.
3. Identifique as consultas resumidas que estão gerando a carga examinando quais consultas na guia SQL da tabela Top Load Items (Principais itens de carga) estão contribuindo mais para aqueles estados de espera. Você pode identificar essas consultas na coluna DB Load by Wait (Carga de banco de dados por espera).
4. Escolha uma dessas consultas resumidas na guia SQL para expandi-la e exibir as consultas secundárias que a compõem.

Por exemplo, no painel a seguir, as esperas de sincronização de arquivos de log compõem a maior parte da carga de banco de dados. A espera de Todos os grupos de operador de LGWR também é alta. O gráfico Top SQL (SQL principal) exibe o que está provocando as esperas de sincronização de arquivos de log: instruções `COMMIT` frequentes. Nesse caso, a confirmação com menos frequência reduzirá a carga de banco de dados.



Analisar a performance do banco de dados por um período

Analise a performance do banco de dados com análise sob demanda criando um relatório de análise de performance por um período. Visualize os relatórios de análise de performance para descobrir problemas de performance, como gargalos de recursos ou alterações em uma consulta na instância de banco de dados. O painel do Insights de Performance permite que você selecione um período e crie um relatório de análise de performance. Também é possível adicionar uma ou mais tags ao relatório.

Para usar esse atributo, você deve usar o período de retenção do nível pago. Para ter mais informações, consulte Preços e retenção de dados para o Performance Insights.

O relatório está disponível na guia Relatórios de análise de performance: novo para selecionar e visualizar. O relatório contém os insights, as métricas relacionadas e as recomendações para resolver o problema de performance. O relatório fica disponível para visualização durante o período de retenção do Insights de Performance.

O relatório será excluído se a hora de início do período de análise do relatório estiver fora do período de retenção. Você também pode excluir o relatório antes que o período de retenção termine.

Para detectar os problemas de performance e gerar o relatório de análise para sua instância de banco de dados, você deve ativar o Insights de Performance. Para obter mais informações sobre como ativar o Insights de Performance, consulte Ativar e desativar o Insights de Performance do Amazon RDS.

Para ter informações sobre compatibilidade de regiões, mecanismos de banco de dados e classes de instância com esse recurso, consulte O mecanismo de banco de dados do Amazon RDS, a região e a classe de instância são compatíveis com atributos do Insights de Performance..

Nas seções a seguir, é possível criar, visualizar, adicionar tags e excluir um relatório de análise de performance.

Tópicos

- Criar um relatório de análise de performance no Insights de Performance
- Visualizar um relatório de análise de performance no Insights de Performance
- Adicionar tags a um relatório de análise de performance no Insights de Performance
- Excluir um relatório de análise de performance no Insights de Performance

Criar um relatório de análise de performance no Insights de Performance

Você pode criar um relatório de análise de performance para um período específico no painel do Insights de Performance. Você pode selecionar um período e adicionar uma ou mais tags ao relatório de análise.

O período de análise pode variar de cinco minutos a seis dias. Deve haver pelo menos 24 horas de dados de performance antes do horário de início da análise.

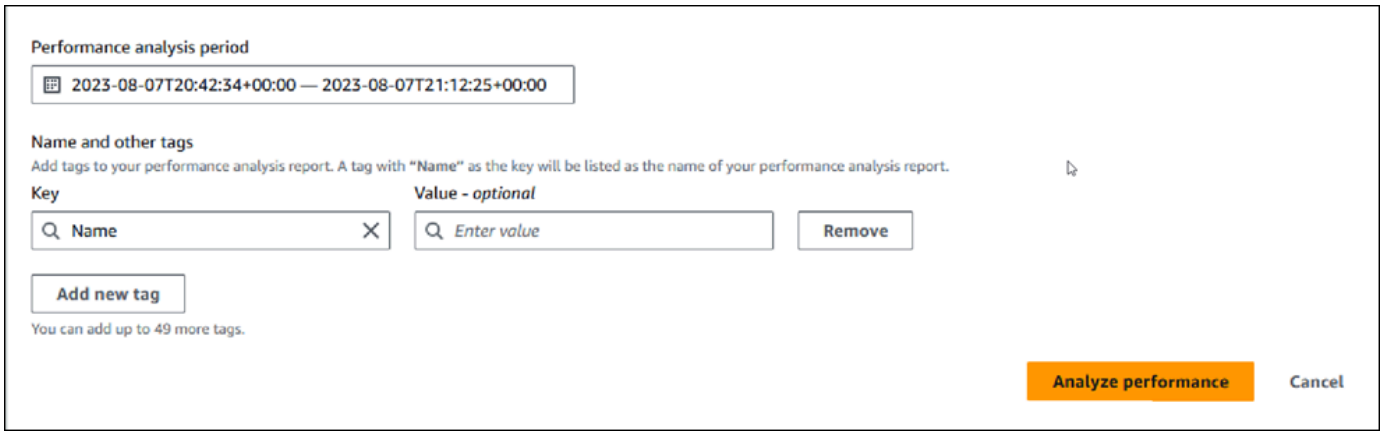
Para ter informações sobre compatibilidade de regiões, mecanismos de banco de dados e classes de instância com esse recurso, consulte O mecanismo de banco de dados do Amazon RDS, a região e a classe de instância são compatíveis com atributos do Insights de Performance..

Como criar um relatório de análise de performance para um período

1. Abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Performance Insights.

3. Escolha uma instância de banco de dados.
4. Selecione Analisar performance na seção Carga do banco de dados no painel do Insights de Performance.

   Os campos para definir o período e adicionar uma ou mais tags ao relatório de análise de performance são exibidos.

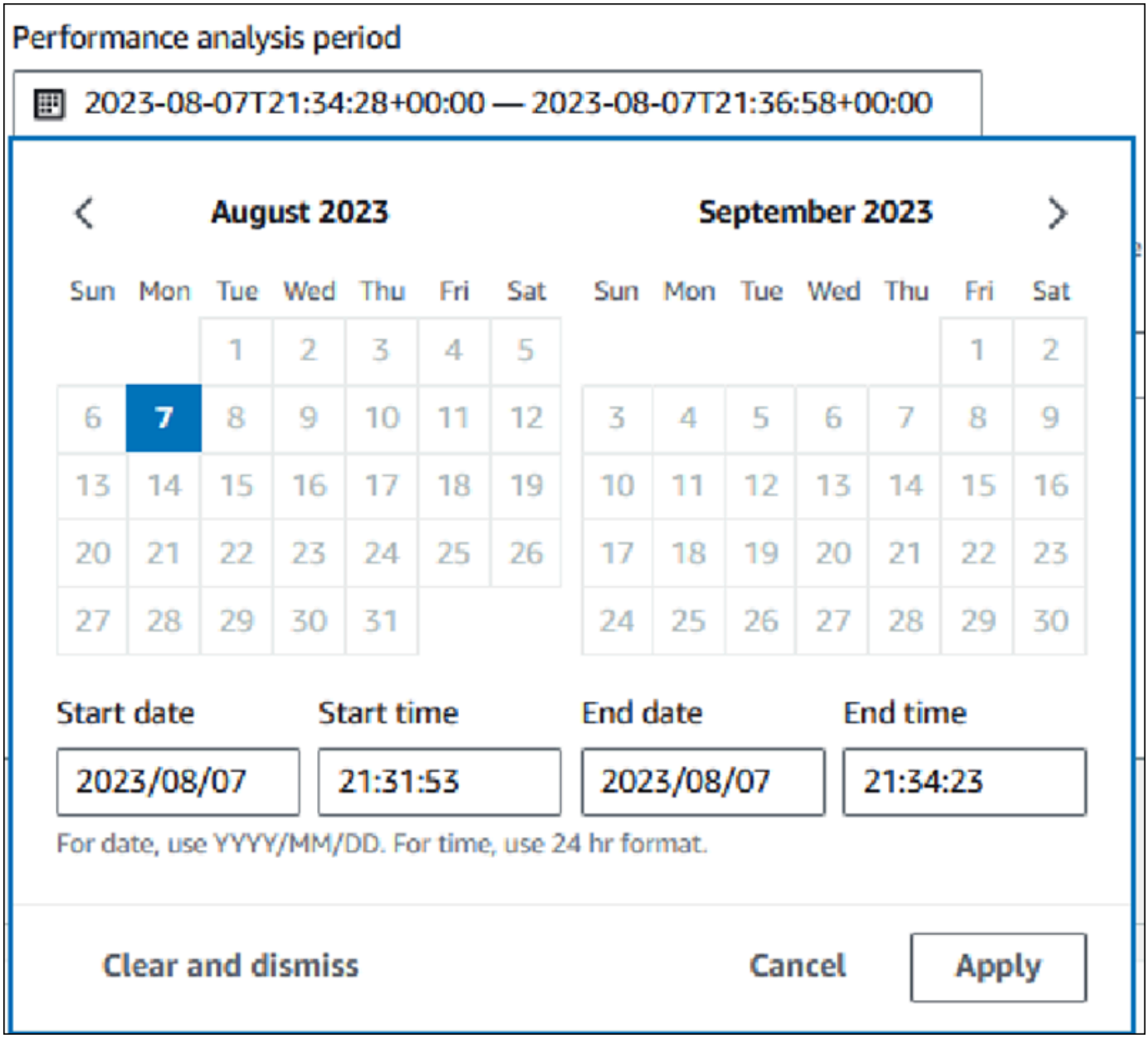


5. Escolha um período. Se definir um período no Intervalo relativo ou no Intervalo absoluto no canto superior direito, você poderá inserir ou selecionar a data e a hora do relatório de análise somente nesse período. Se você selecionar o período de análise fora desse período, uma mensagem de erro será exibida.

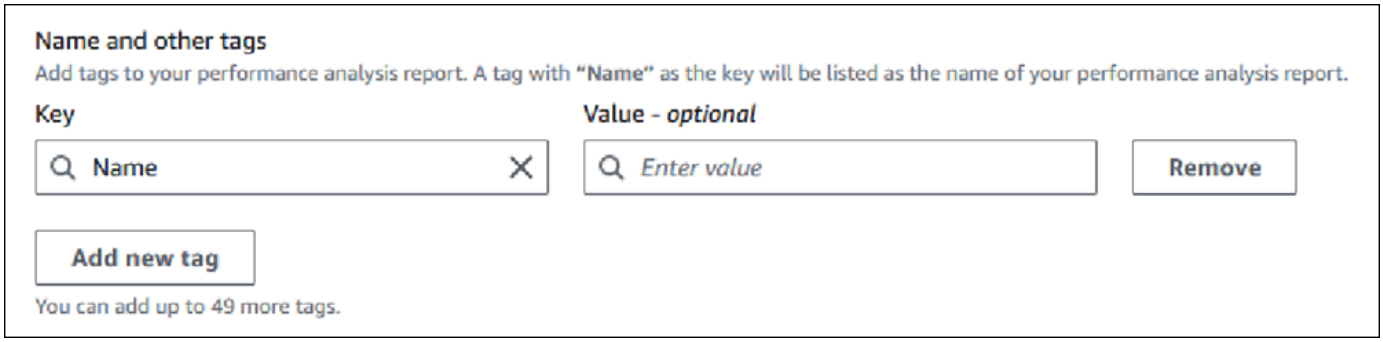
   Para definir o período, você pode fazer o seguinte:

   - Pressione e arraste qualquer um dos controles deslizantes no gráfico de carga do banco de dados.

     A caixa Período de análise de performance exibe o período selecionado, e o gráfico de carga do banco de dados destaca o período selecionado.
   - Selecione a Data de início, o Horário de início, a Data de término e a Horário de término na caixa Período de análise de performance.



6. (Opcional) Digite Chave e Valoropcionais para adicionar uma tag para o relatório.



7. Selecione Analisar performance.

   Um banner exibe uma mensagem se a geração do relatório foi bem-sucedida ou falhou. A mensagem também fornece o link para visualizar o relatório.

   O exemplo a seguir mostra o banner com a mensagem de sucesso na criação do relatório.



   O relatório está disponível na guia Relatórios de análise de performance: novo.

Você pode criar um relatório de análise de desempenho usando a AWS CLI. Para obter um exemplo de como criar um relatório usando a AWS CLI, consulte Criar um relatório de análise de performance para um período.

Visualizar um relatório de análise de performance no Insights de Performance

A guia Relatórios de análise de performance: novo lista todos os relatórios criados para a instância de banco de dados. Os itens a seguir são exibidos para cada relatório:

- ID: identificador exclusivo do relatório.
- Nome: chave de tag adicionada ao relatório.
- Hora da criação do relatório: hora em que você criou o relatório.
- Hora de início da análise: hora de início da análise no relatório.
- Hora de término da análise: hora de término da análise no relatório.

Como visualizar um relatório de análise de performance

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Performance Insights.
3. Selecione uma instância de banco de dados para a qual você deseja visualizar o relatório de análise.
4. Role para baixo e selecione a guia Relatórios de análises de performance: novo no painel do Insights de Performance.

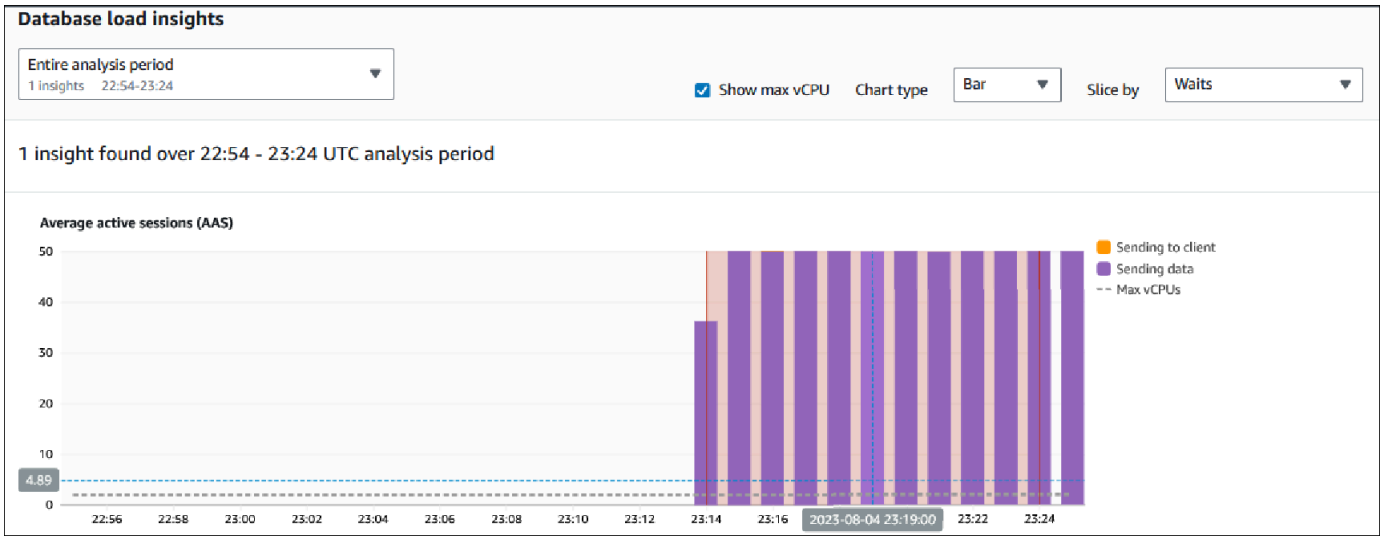
   Todos os relatórios de análise para os diferentes períodos são exibidos.

5. Selecione ID do relatório que você deseja visualizar.

   O gráfico de carga do banco de dados exibirá todo o período de análise por padrão se mais de um insight for identificado. Se o relatório tiver identificado um insight, o gráfico de carga do banco de dados exibirá o insight por padrão.

   O painel também lista as tags do relatório na seção Tags.

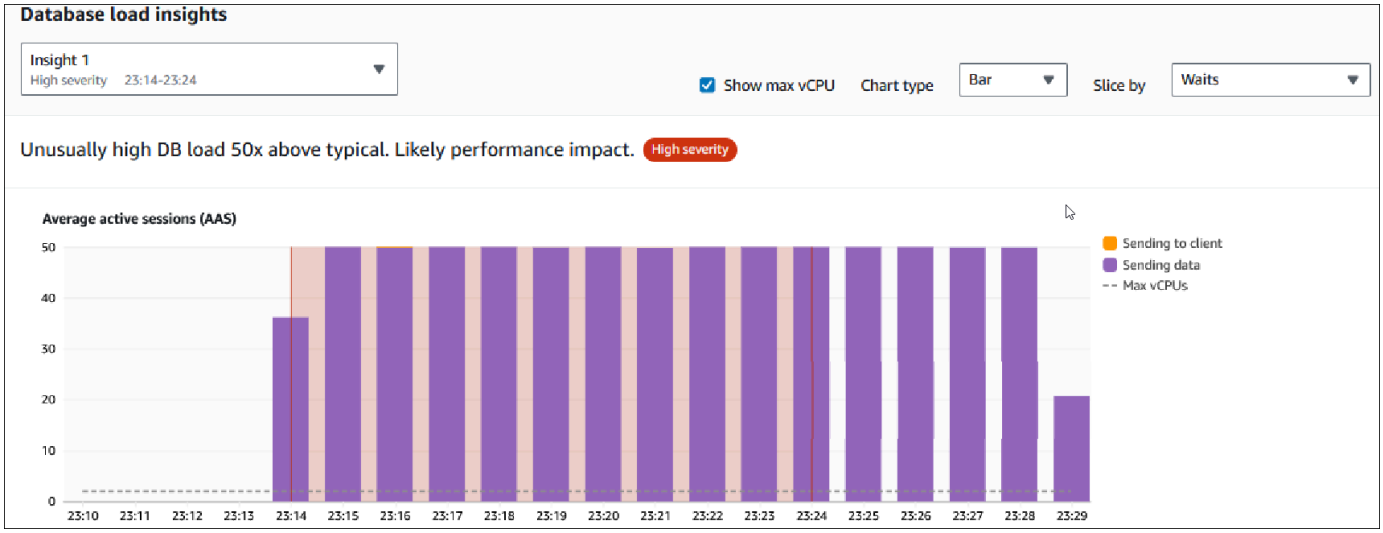
   O exemplo a seguir mostra todo o período de análise do relatório.

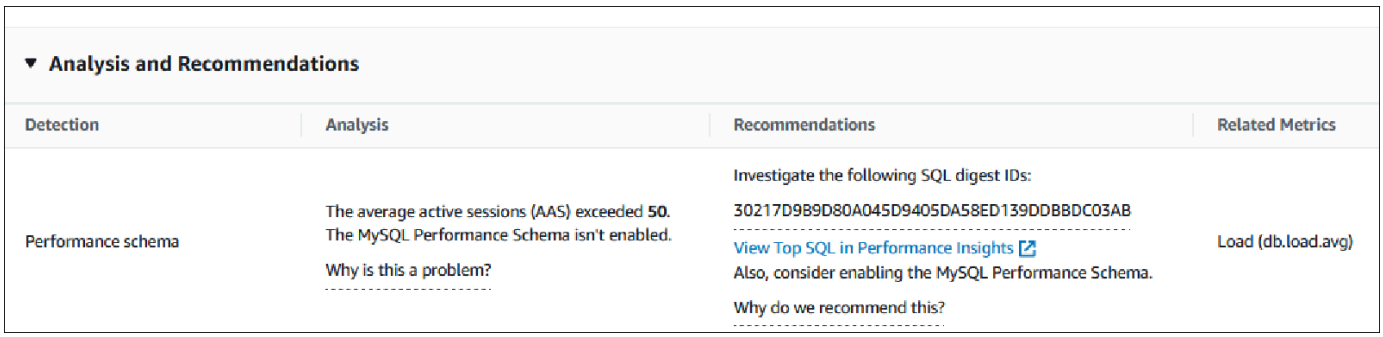


6. Selecione o insight na lista Insights da carga do banco de dados que você deseja visualizar se mais de um insight for identificado no relatório.

   O painel exibe a mensagem de insight, o gráfico de carga do banco de dados destacando o período do insight, a análise e as recomendações, bem como a lista de tags do relatório.

   O exemplo a seguir mostra o insight da carga do banco de dados no relatório.





Adicionar tags a um relatório de análise de performance no Insights de Performance

Você pode adicionar uma tag ao criar ou visualizar um relatório. Você pode adicionar até 50 tags a um relatório.

Você precisa de permissões para adicionar as tags. Para receber mais informações sobre as políticas de acesso para o Insights de Performance, consulte Configurar políticas de acesso para o Performance Insights.

Para adicionar uma ou mais tags ao criar um relatório, consulte a etapa 6 do procedimento Criar um relatório de análise de performance no Insights de Performance.

Como adicionar uma ou mais tags ao visualizar um relatório

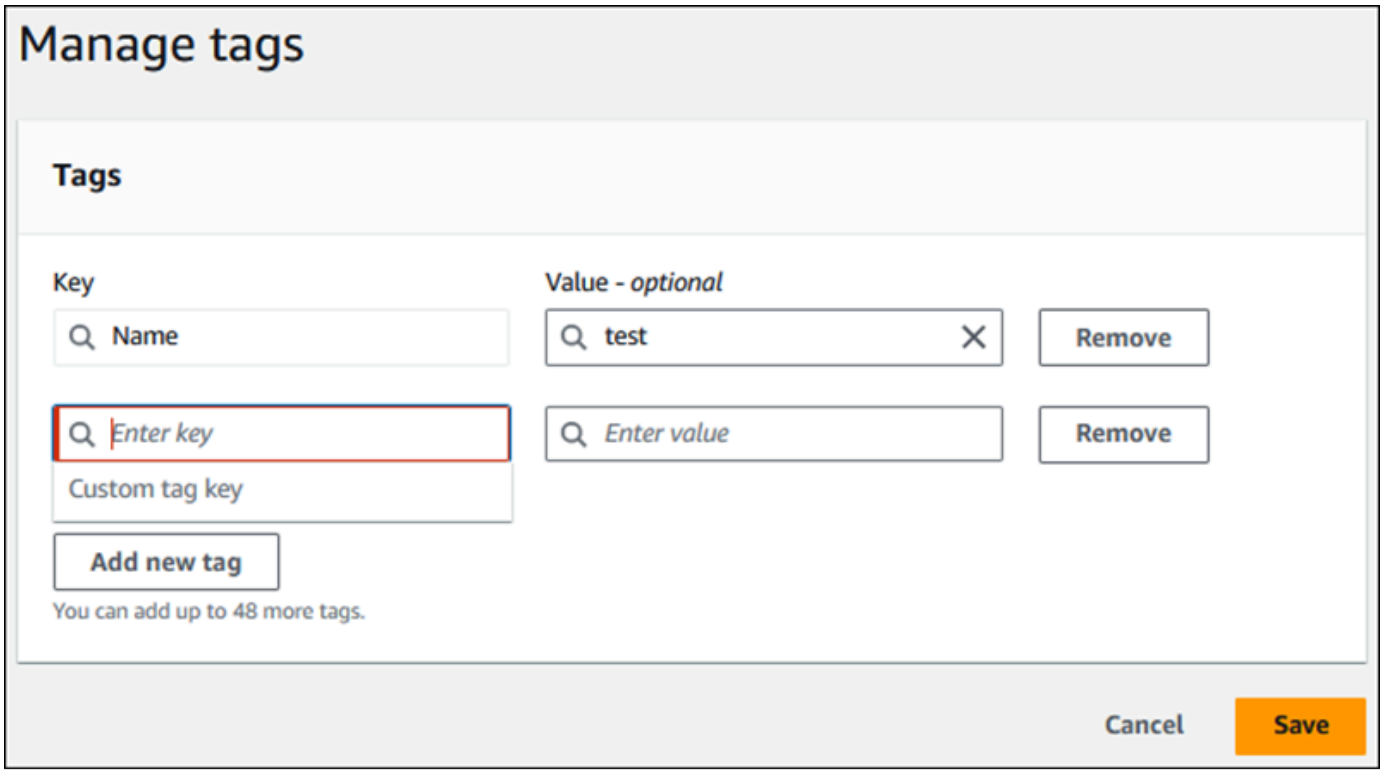
1. Abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Performance Insights.
3. Escolha uma instância de banco de dados.

O painel do Insights de Performance é exibido para a instância de banco de dados.

4. Role para baixo e selecione a guia Relatórios de análises de performance: novo.
5. Escolha o relatório ao qual você deseja adicionar as tags.

   O painel exibe o relatório.
6. Role para baixo até Tags e selecione Gerenciar tags.
7. Selecione Adicionar nova tag.
8. Insira a Chave e o Valor: opcional e selecione Adicionar nova tag.

   O exemplo a seguir fornece a opção de adicionar uma nova tag ao relatório selecionado.



   Uma nova tag é criada para o relatório.

   A lista de tags do relatório é exibida na seção Tags no painel. Se você quiser remover uma tag do relatório, selecione Remover ao lado da tag.

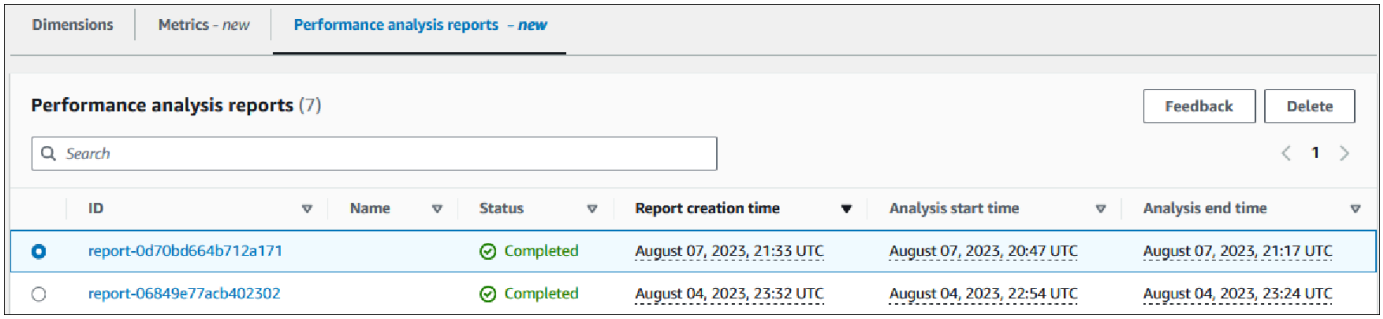
Excluir um relatório de análise de performance no Insights de Performance

Você pode excluir um relatório da lista de relatórios exibida na guia Relatórios de análise de performance ou ao visualizar um relatório.

Para excluir um relatório

1. Abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Performance Insights.
3. Escolha uma instância de banco de dados.

   O painel do Insights de Performance é exibido para a instância de banco de dados.
4. Role para baixo e selecione a guia Relatórios de análises de performance: novo.
5. Selecione o relatório que você deseja excluir e escolha Excluir no canto superior direito.



   Uma janela de confirmação é exibida. O relatório é excluído depois que você escolhe confirmar.
6. (Opcional) Selecione ID do relatório que você deseja excluir.

   Na página de relatório, selecione Excluir no canto superior direito.

   Uma janela de confirmação é exibida. O relatório é excluído depois que você escolhe confirmar.

Analisar consultas com a guia SQL principal no Insights de Performance

No painel do Amazon RDS Performance Insights, é possível encontrar informações sobre consultas recentes em execução na guia Top SQL (SQL principal), na tabela Top dimensions (Principais dimensões). É possível utilizar essas informações para ajustar suas consultas.

Tópicos

- Visão geral da guia Top SQL (SQL principal)
- Acessar mais texto SQL no painel do Performance Insights.
- Visualizar estatísticas SQL no painel do Performance Insights

Visão geral da guia Top SQL (SQL principal)

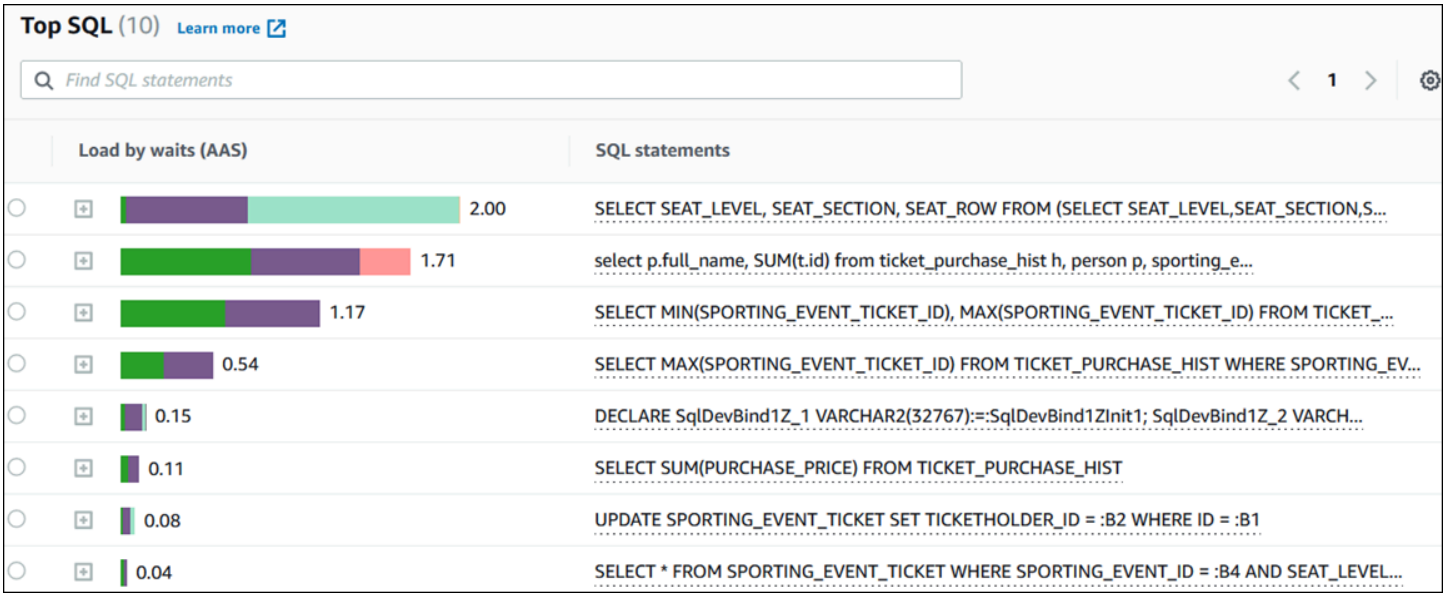
Por padrão, a guia Top SQL (SQL principal) mostra as 25 consultas que mais estão contribuindo para a carga do banco de dados. Para ajudar a ajustar as consultas, é possível analisar informações como o texto da consulta e as estatísticas de SQL. Você também pode escolher as estatísticas que deseja tornar visíveis na guia Top SQL (SQL principal).

Tópicos

- Texto SQL
- Estatísticas SQL
- Load by waits (AAS) (Carga por esperas)
- Visualizar informações do SQL
- Escolher preferências de estatísticas

Texto SQL

Por padrão, cada linha na tabela Top SQL (SQL principal) mostra 500 bytes de texto para cada declaração.



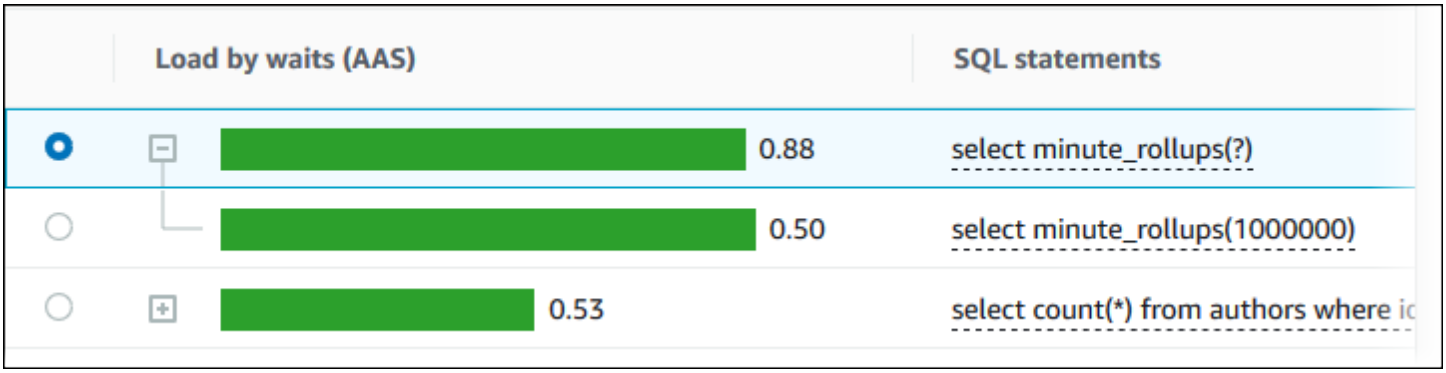
Para saber como ver mais do que os 500 bytes padrão de texto SQL, consulte Acessar mais texto SQL no painel do Performance Insights..

Um resumo SQL é formado por várias consultas reais com estruturas semelhantes, mas que possivelmente apresentam valores literais diferentes. O resumo substitui valores codificados por

um ponto de interrogação. Por exemplo, um resumo pode ser `SELECT * FROM emp WHERE lname= ?`. Esse resumo pode incluir as seguintes consultas subordinadas:

```
SELECT * FROM emp WHERE lname = 'Sanchez'
SELECT * FROM emp WHERE lname = 'Olagappan'
SELECT * FROM emp WHERE lname = 'Wu'
```

Para ver as instruções SQL literais em um resumo, escolha a consulta e depois o sinal de mais (+). No exemplo a seguir, a consulta selecionada é de resumo.



> **Note**
>
> Um resumo SQL agrupa instruções SQL semelhantes, mas não edita informações confidenciais.

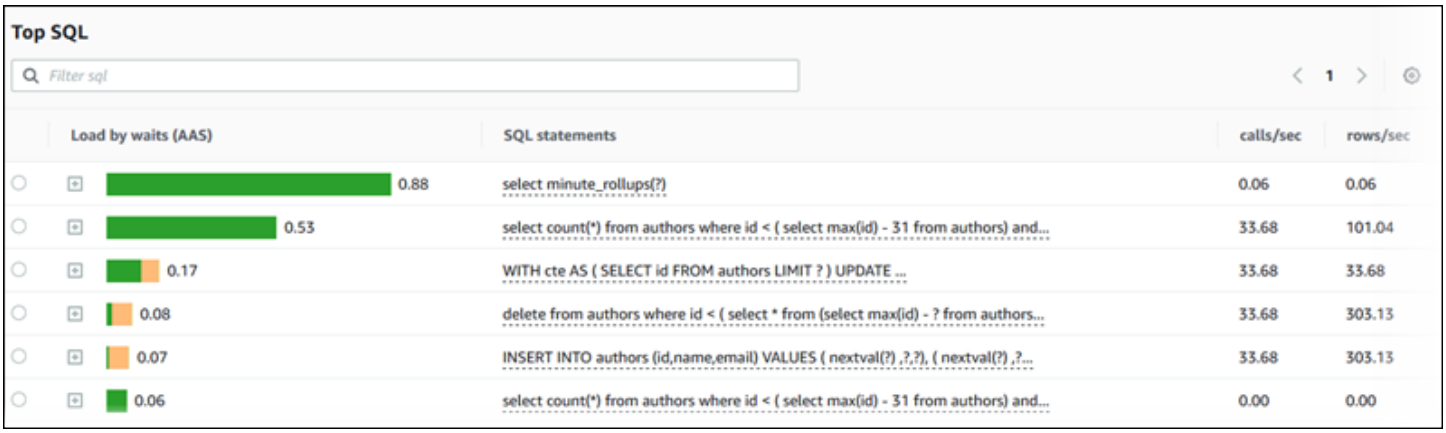
O Performance Insights pode mostrar o texto Oracle SQL como Unknown (Desconhecido). O texto tem esse status nas seguintes situações:

- Um usuário de banco de dados Oracle diferente de SYS está ativo, mas não está executando SQL no momento. Por exemplo, quando uma consulta paralela é concluída, o coordenador de consultas aguarda que os processos auxiliares enviem suas estatísticas de sessão. Durante a espera, o texto da consulta é exibido como Unknown (Desconhecido).
- Para uma instância do RDS para Oracle na Standard Edition, o Oracle Resource Manager limita o número de threads paralelos. O processo em segundo plano que realiza esse trabalho faz com que o texto da consulta seja exibido como Unknown (Desconhecido).

Estatísticas SQL

Estatísticas SQL são métricas de performance sobre consultas SQL. Por exemplo, o Performance Insights pode exibir execuções por segundo ou linhas processadas por segundo. O Performance Insights apenas coleta estatísticas para as consultas mais comuns. Normalmente, elas correspondem às principais consultas por carga mostradas no painel do Performance Insights.

Todas as linhas da tabela Top SQL (SQL principal) mostram estatísticas relevantes para a instrução ou o resumo SQL, como mostra exemplo a seguir.



O Performance Insights pode relatar `0.00` e - (unknown) (desconhecido) para estatísticas SQL. Essa situação ocorre nas seguintes condições:

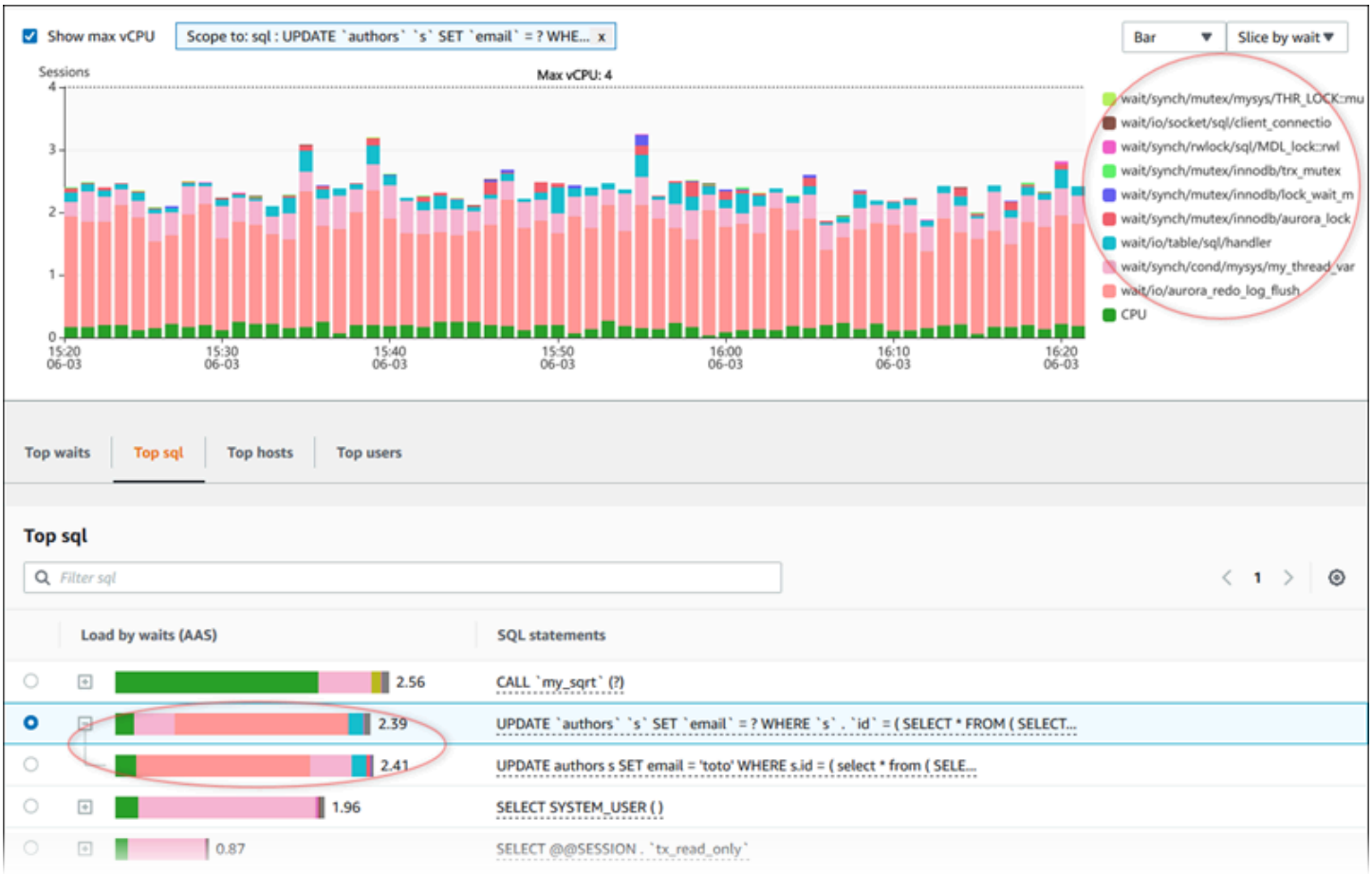
- Existe apenas uma amostra. Por exemplo, o Performance Insights calcula as taxas de alteração para consultas do RDS PostgreSQL com base em várias amostras da visualização `pg_stat_statements`. Quando uma workload é executada por um curto período, o Performance Insights consegue coletar apenas uma amostra, o que significa que ele não consegue calcular uma taxa de alteração. O valor desconhecido é representado com um traço (-).
- Duas amostras têm os mesmos valores. O Performance Insights não consegue calcular uma taxa de alteração porque não ocorreu nenhuma alteração, portanto, ele relata a taxa como `0.00`.
- Uma instrução do RDS PostgreSQL não tem um identificador válido. O PostgreSQL cria um identificador para uma instrução somente após a análise. Assim, pode existir uma instrução nas estruturas internas na memória do PostgreSQL sem identificador. Como o Performance Insights cria amostras de estruturas internas na memória uma vez por segundo, consultas de baixa latência podem aparecer apenas para uma única amostra. Se o identificador de consultas não estiver disponível para essa amostra, o Performance Insights não conseguirá associar essa instrução às suas estatísticas. O valor desconhecido é representado com um traço (-).

Para obter uma descrição das estatísticas SQL para o mecanismos do Amazon RDS, consulte Estatísticas SQL para Performance Insights.

Load by waits (AAS) (Carga por esperas)

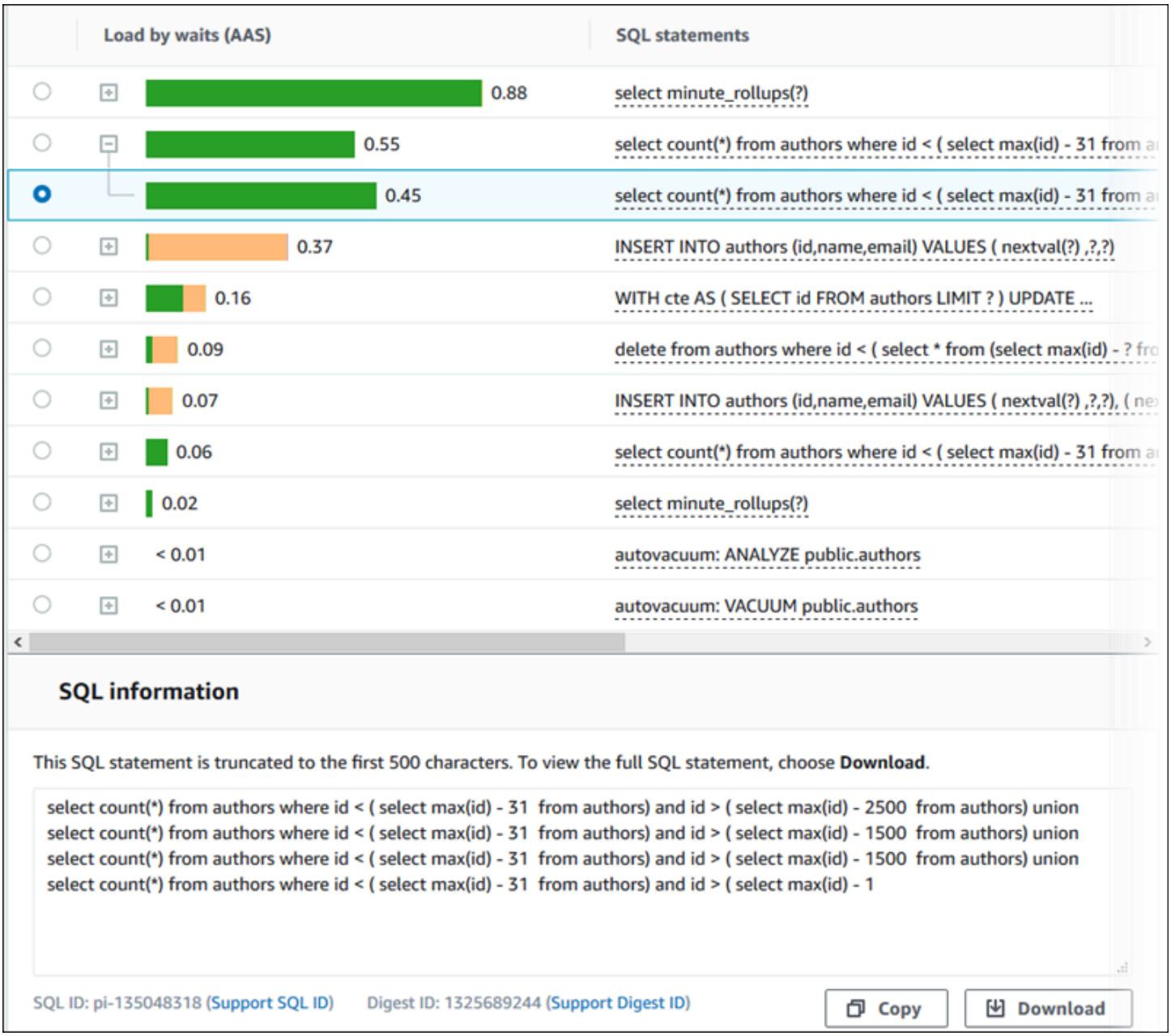
Em Top SQL (SQL principal), a coluna Load by waits (AAS) (Carga por esperas) mostra a porcentagem da carga do banco de dados associada a cada item de carga principal. Essa coluna reflete a carga desse item por qualquer agrupamento atualmente selecionado no Gráfico de carga de banco de dados. Para obter mais informações sobre média de sessões ativas (AAS), consulte Média de sessões ativas.

Por exemplo, é possível agrupar o gráfico DB load (Carga do banco de dados) com base em estados de espera. Examine consultas SQL na tabela de itens de carga principal. Nesse caso, a barra DB Load by Waits (Carga de banco de dados por espera) é dimensionada, segmentada e codificada por cores para mostrar com quanto de um determinado estado de espera a consulta está contribuindo. Ela também mostra quais estados de espera estão afetando a consulta selecionada.



Visualizar informações do SQL

Na tabela Top SQL (SQL principal), é possível abrir uma instrução para visualizar suas informações. As informações são exibidas no painel inferior.



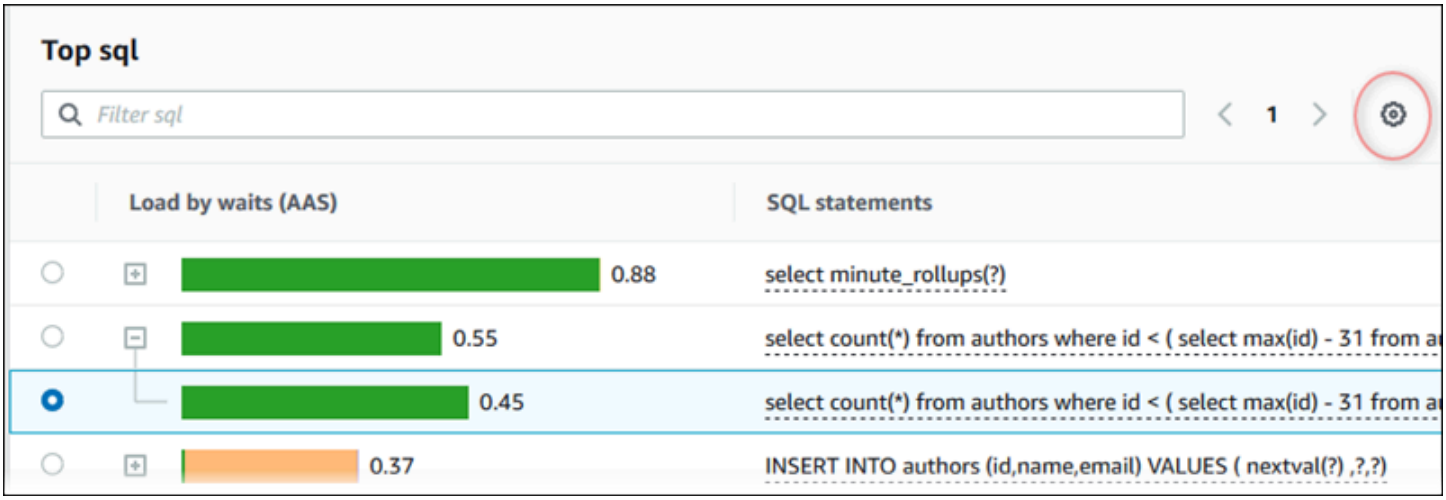
Os seguintes tipos de identificadores (IDs) associados a instruções SQL:

- ID SQL de suporte: Um valor de hash do ID SQL. Esse valor só se destina a referenciar um ID SQL quando você está trabalhando com o AWS Support. AWS O Support não tem acesso a IDs SQL reais e ao texto SQL.

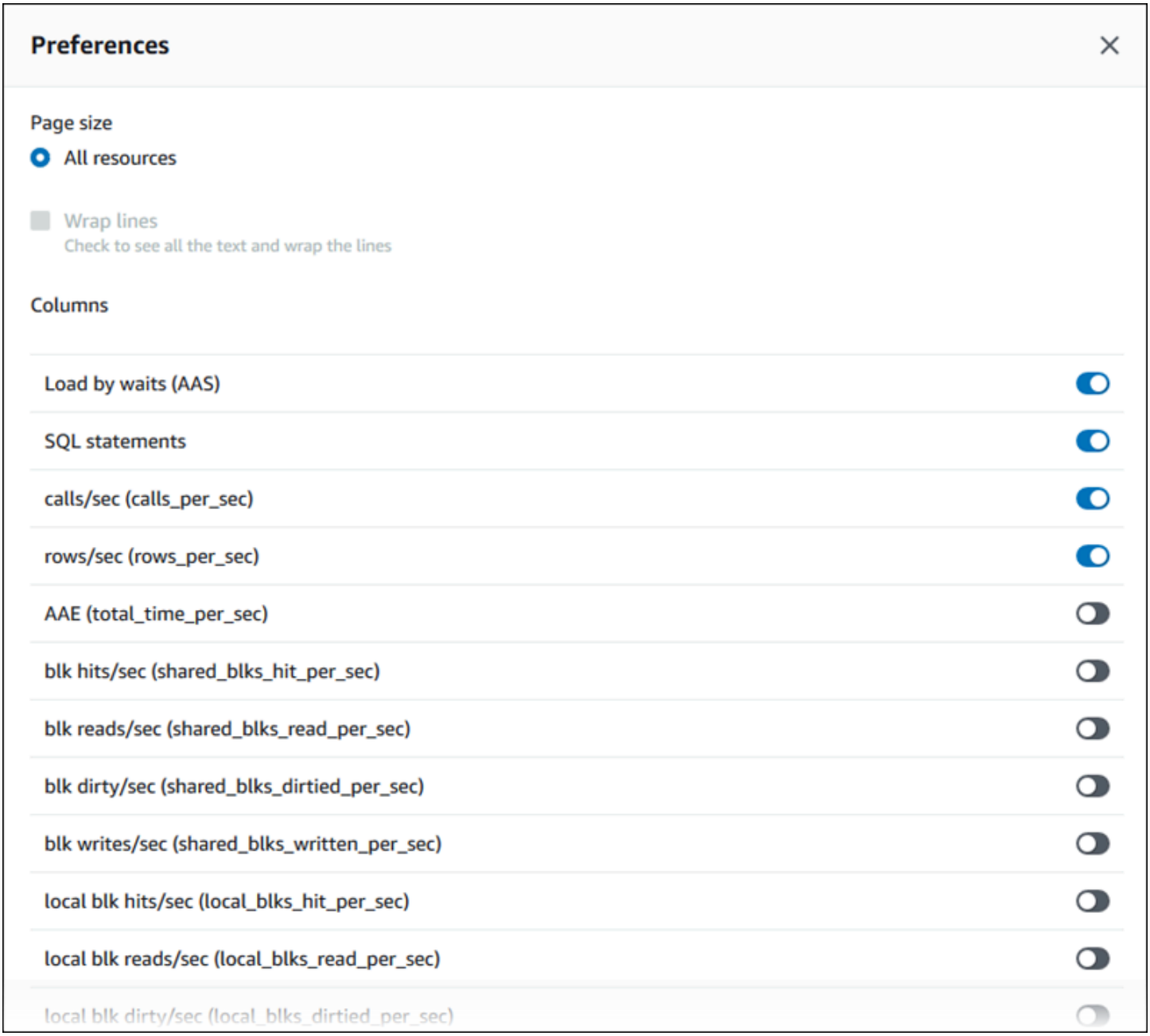
- ID de arquivo de resumo de suporte: um valor de hash do ID de arquivo de resumo. Esse valor apenas se destina como referência a um ID de arquivo de resumo quando você está trabalhando com o AWS Support. AWS O Support não tem acesso a IDs de arquivo de resumo reais e ao texto SQL.

Escolher preferências de estatísticas

É possível controlar as estatísticas exibidas na guia Top SQL (SQL principal) selecionando o ícone Preferences (Preferências).



Quando você escolhe o ícone Preferences (Preferências), a janela Preferences (Preferências) abre. O screenshot a seguir é um exemplo da janela Preferences (Preferências).



Para habilitar as estatísticas que você deseja tornar visíveis na guia Top SQL (SQL principal), use o mouse para rolar até o final da janela e depois escolha Continue (Continuar).

Para obter mais informações sobre estatísticas por segundo ou por chamada para os mecanismos do Amazon RDS, consulte a seção de estatísticas SQL específicas do mecanismo emEstatísticas SQL para Performance Insights

Acessar mais texto SQL no painel do Performance Insights.

Por padrão, cada linha na tabela Top SQL (SQL principal) mostra 500 bytes de texto SQL para cada instrução SQL.



Quando uma instrução SQL excede 500 bytes, você pode visualizar mais texto na seção SQL text (Texto SQL) abaixo da tabela Top SQL (SQL principal). Nesse caso, o comprimento máximo para o texto exibido em SQL text (Texto SQL) é de 4 KB. Esse limite é introduzido pelo console e está sujeito aos limites definidos pelo mecanismo de banco de dados. Para salvar o texto mostrado em SQL text (Texto SQL), escolha Download.

Tópicos

- Limites de tamanho de texto para mecanismos Amazon RDS
- Definir o limite do texto SQL para instâncias de banco de dados do Amazon RDS para PostgreSQL
- Visualizar e baixar texto SQL no painel do Performance Insights

Limites de tamanho de texto para mecanismos Amazon RDS

Quando você baixa um texto SQL, o mecanismo de banco de dados determina o tamanho máximo dele. Você pode baixar texto SQL até os seguintes limites por mecanismo.

Mecanismo de banco de dados	Tamanho máximo do texto para download
Amazon RDS para MySQL e MariaDB	O comprimento é fixado em 4.096 bytes quando o Performance Schema está habilitado. Se o Performance Schema não estiver habilitado, o comprimento será fixado em 65.535 bytes.
Amazon RDS para Microsoft SQL Server	4,096 caracteres
Amazon RDS para Oracle	1.000 bytes

A seção SQL text (Texto SQL) do console do Performance Insights mostra até o máximo que o mecanismo retorna. Por exemplo, se o MySQL retorna no máximo 1 KB para o Performance Insights, ele só pode coletar e mostrar 1 KB, mesmo que a consulta original seja maior. Assim, quando você

visualiza ou baixa a consulta em SQL text (Texto SQL), o Performance Insights retorna o mesmo número de bytes.

Se você usar a AWS CLI ou a API, o Insights de Performance não terá o limite de 4 KB aplicado pelo console. `DescribeDimensionKeys` e `GetResourceMetrics` exibem no máximo 500 bytes.

> **Note**
>
> `GetDimensionKeyDetails` exibe a consulta completa, mas o tamanho está sujeito ao limite do mecanismo.

Definir o limite do texto SQL para instâncias de banco de dados do Amazon RDS para PostgreSQL

Amazon RDS para PostgreSQL lida com texto de forma diferente. Você pode definir o limite de tamanho de texto com o parâmetro de instância de banco de dados `track_activity_query_size`. Esse parâmetro tem as seguintes características:

Tamanho de texto padrão

No Amazon RDS para PostgreSQL versão 9.6, a configuração padrão do parâmetro `track_activity_query_size` é 1.024 bytes. No Amazon RDS para PostgreSQL versão 10 ou superior, o padrão é 4.096 bytes.

Tamanho máximo do texto

O limite para `track_activity_query_size` é de 102.400 bytes para Amazon RDS para PostgreSQL versão 12 e inferiores. O máximo é de 1 MB para a versão 13 e posterior.

Se o mecanismo retornar 1 MB ao Performance Insights, o console exibirá somente os primeiros 4 KB. Se você baixar a consulta, você obtém o total de 1 MB. Nesse caso, a visualização e o download retornam números diferentes de bytes. Para obter mais informações sobre o parâmetro de instância de banco de dados `track_activity_query_size`, consulte Estatísticas de tempo de execução na documentação do PostgreSQL.

Para aumentar o tamanho do texto SQL, aumente o limite `track_activity_query_size`. Para modificar o parâmetro, altere a configuração dele no grupo de parâmetros que está associado à instância de banco de dados do Amazon RDS para PostgreSQL.

Para alterar a configuração quando a instância usa o grupo de parâmetros padrão

1. Crie um novo grupo de parâmetros de instância de banco de dados para o mecanismo de banco de dados apropriado e sua respectiva versão.
2. Defina o parâmetro no novo grupo de parâmetros.
3. Associe o novo grupo de parâmetros à instância de banco de dados.

Para obter informações sobre como configurar um parâmetro de instância de banco de dados, consulte Modificar parâmetros em um grupo de parâmetros de banco de dados no Amazon RDS.

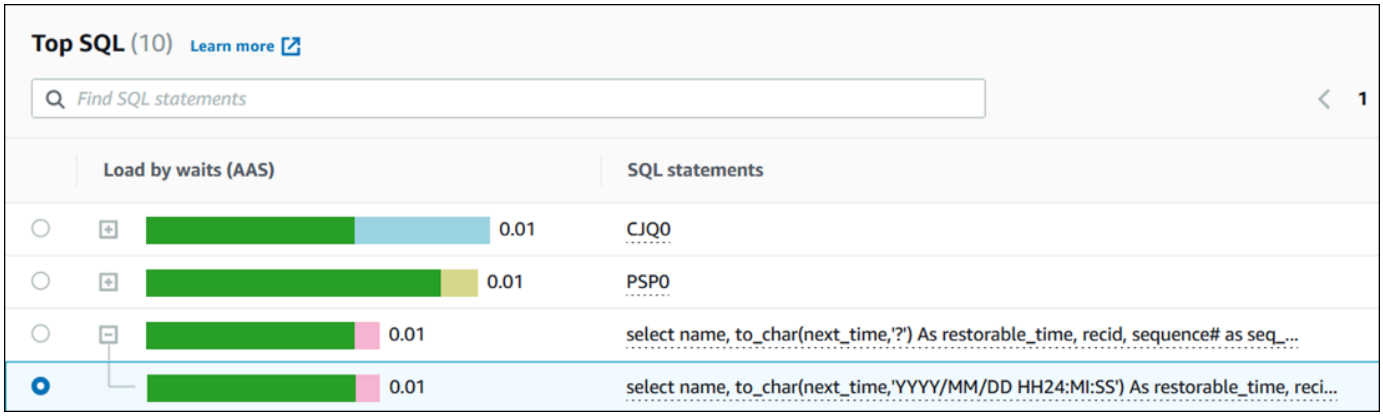
Visualizar e baixar texto SQL no painel do Performance Insights

No painel do Performance Insights, é possível visualizar ou baixar o texto SQL.

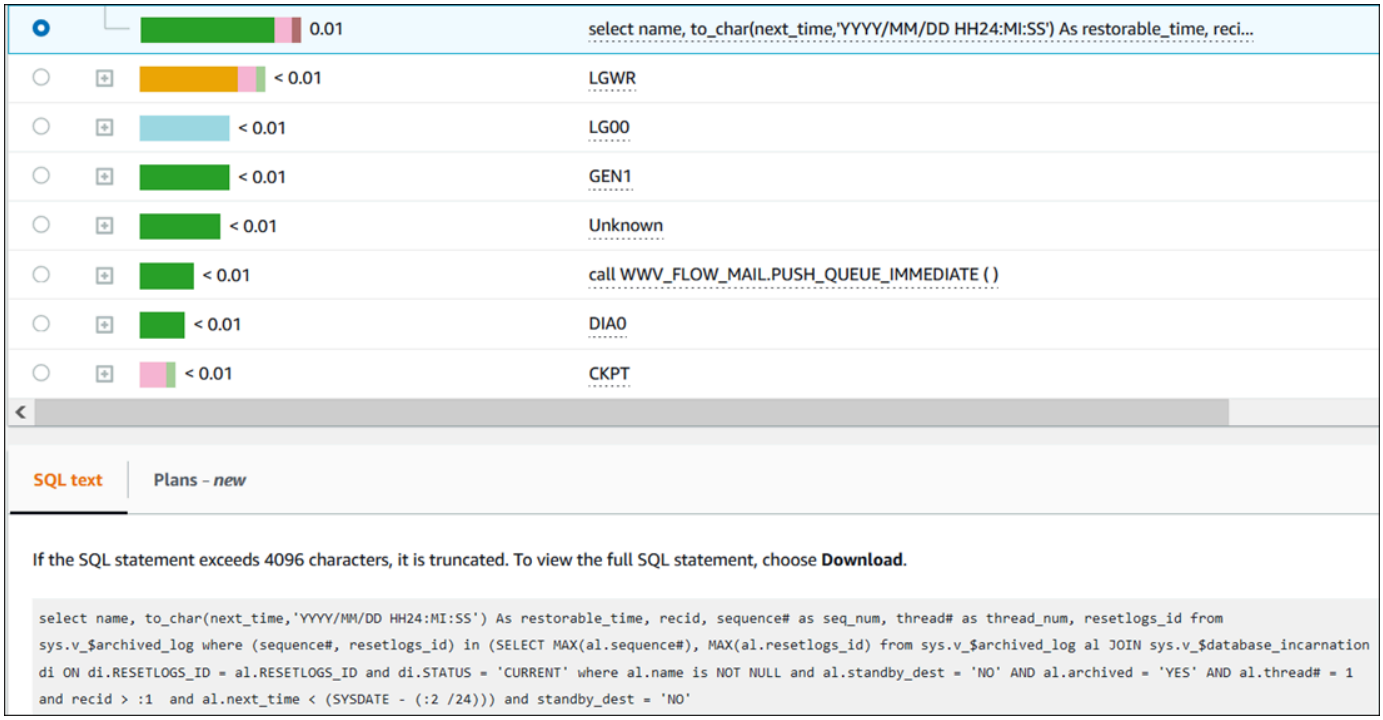
Para visualizar mais texto SQL no painel do Performance Insights

1. Abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Performance Insights.
3. Escolha uma instância de banco de dados.
4. Role para baixo até a guia SQL principal no painel do Insights de Performance.
5. Escolha o sinal de adição para expandir um resumo SQL e selecione uma das consultas secundárias do resumo.

   Instruções SQL com texto maior que 500 bytes são semelhantes à imagem a seguir.



6. Role para baixo até a guia SQL text (texto SQL).



O painel do Performance Insights pode exibir até 4.096 bytes para cada instrução SQL.

7. (Opcional) Escolha Copy (Copiar) para copiar a instrução SQL exibida ou escolha Download (Baixar) para baixar a instrução SQL a fim de visualizar o texto SQL até o limite do mecanismo de banco de dados.

> **Note**
>
> Para copiar ou baixar a instrução SQL, desabilite bloqueadores de pop-up.

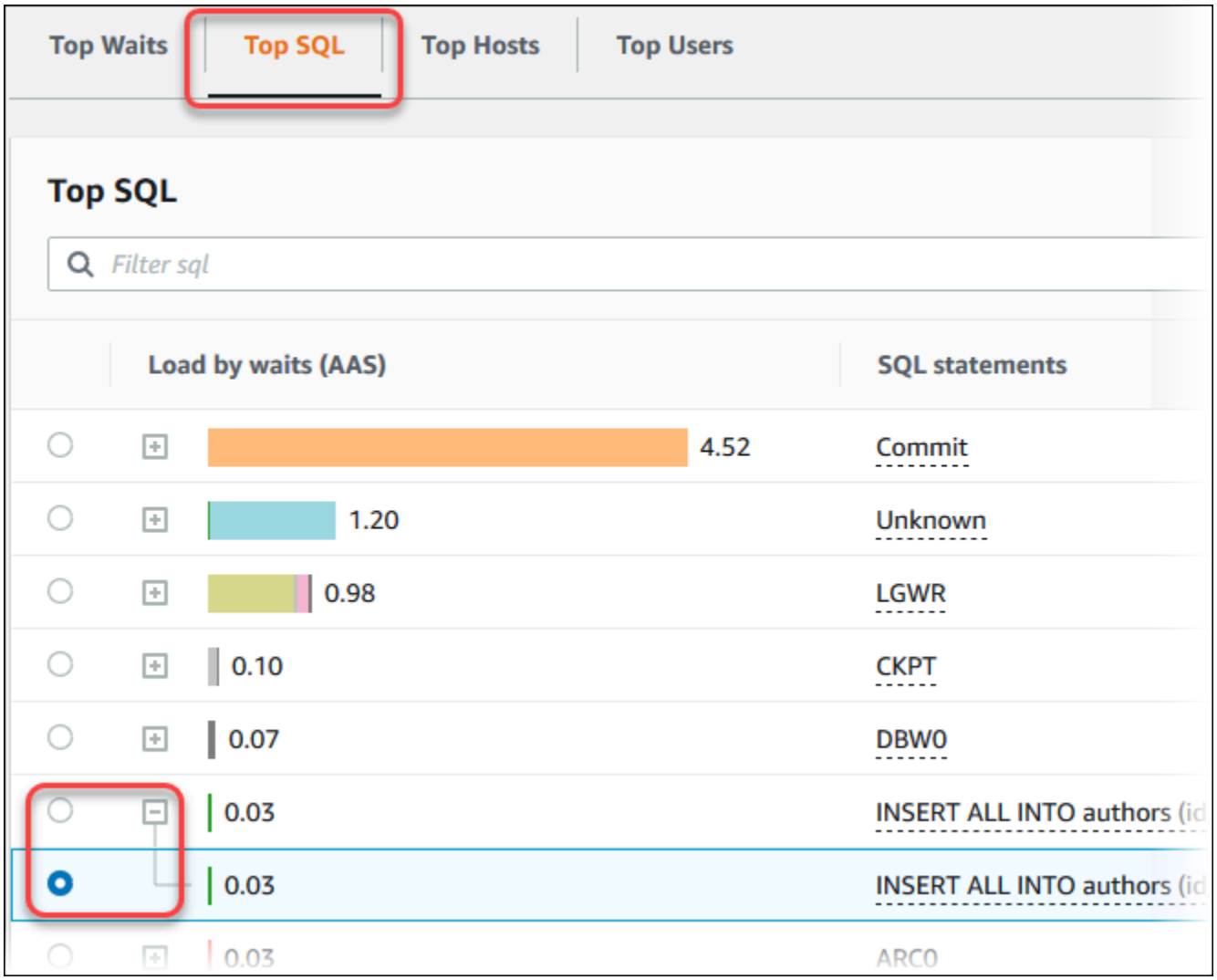
Visualizar estatísticas SQL no painel do Performance Insights

No painel do Performance Insights, as estatísticas SQL estão disponíveis na guia Top SQL (SQL principal) do grafo Database load (Carga do banco de dados).

Como visualizar estatísticas SQL

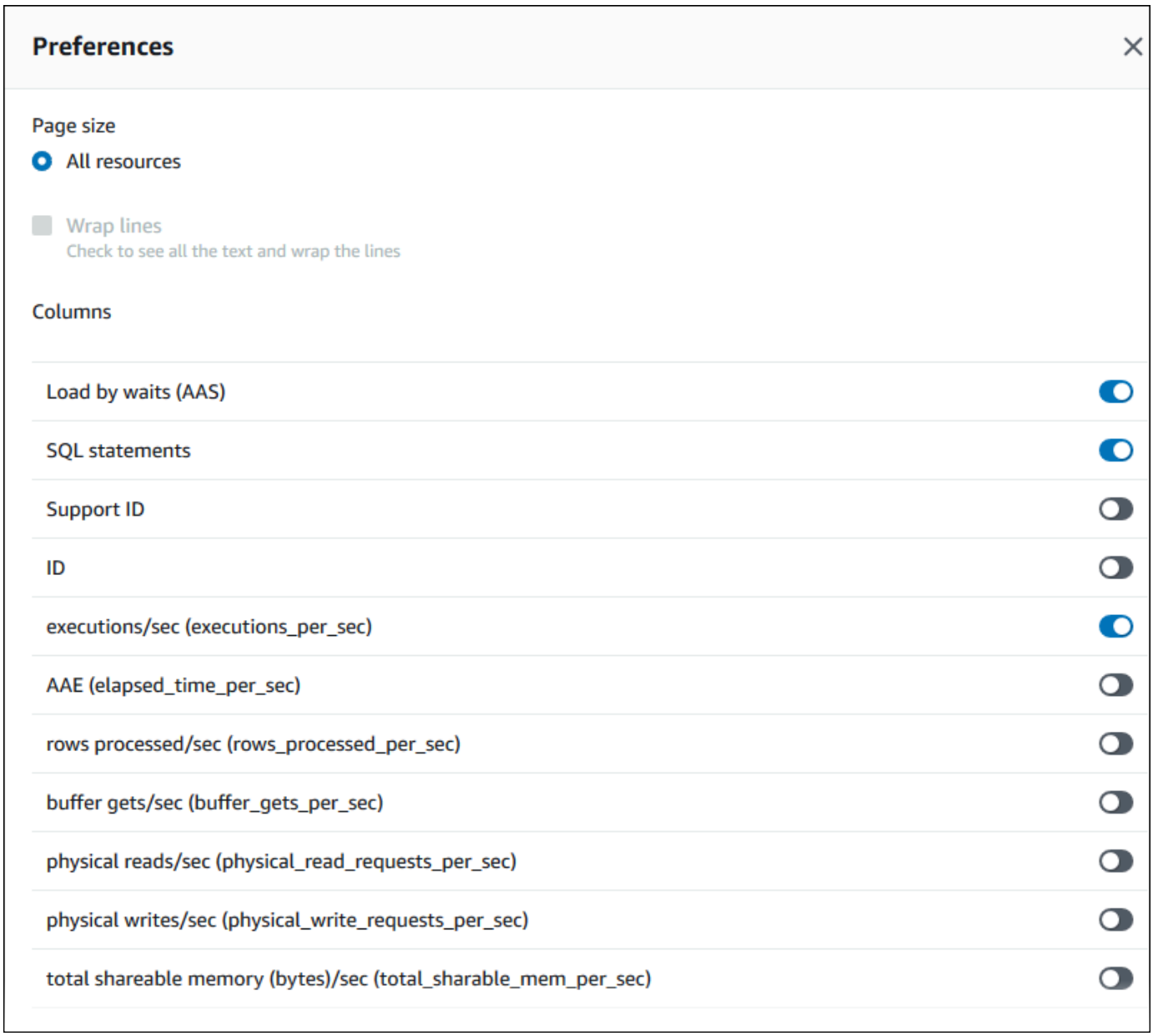
1. Abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Performance Insights.
3. Na parte posterior da página, escolha o banco de dados cujas estatísticas SQL você deseja ver.
4. Navegue até o final da página e escolha a guia Top SQL (SQL principal).

5. Escolha uma instrução individualou consulta de resumo.

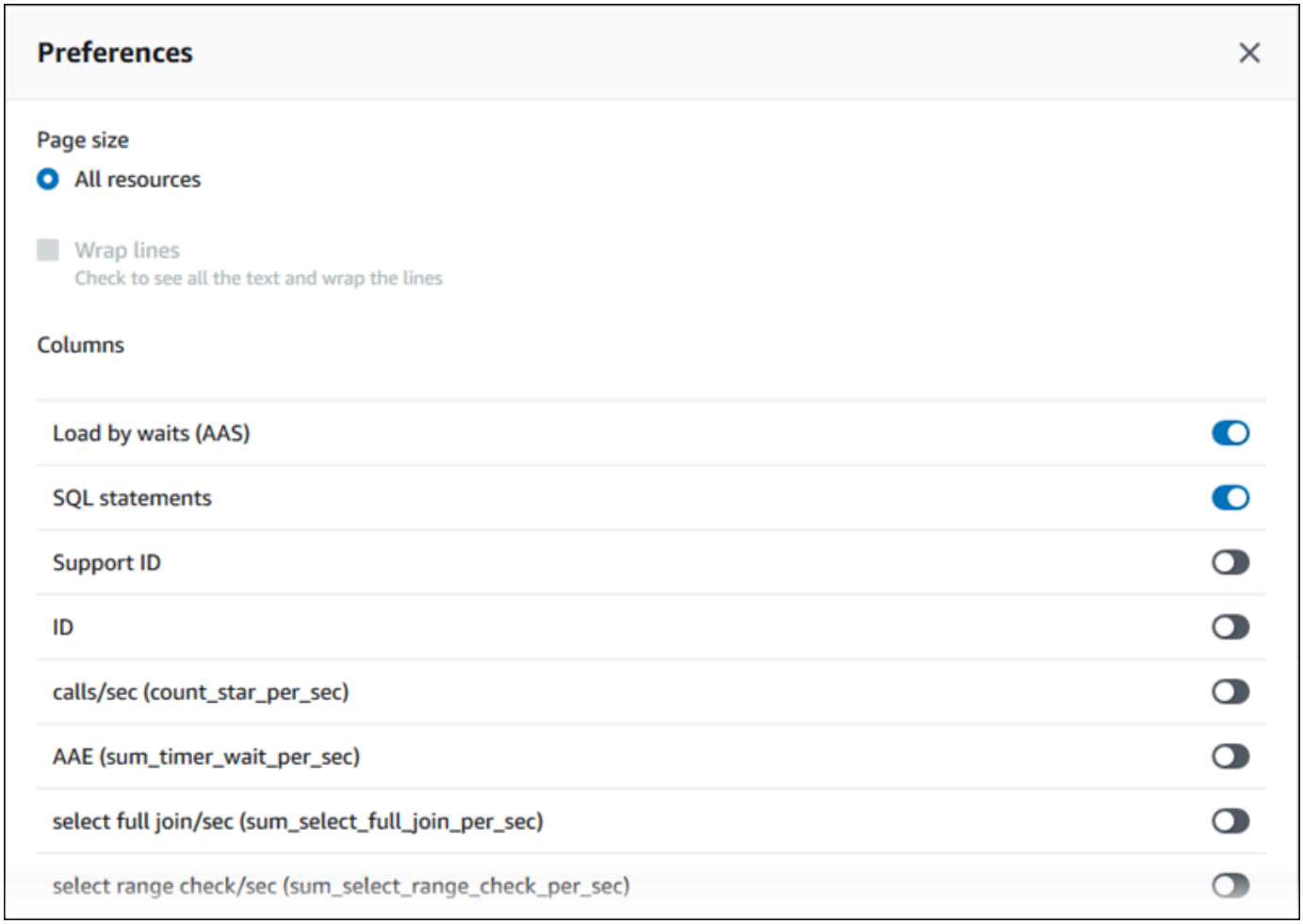


6. Escolha quais estatísticas exibir, escolhendo o ícone de engrenagem no canto superior direito do gráfico. Para obter as descrições das estatísticas SQL para o mecanismos do Amazon RDS, consulte Estatísticas SQL para Performance Insights.

   O exemplo a seguir mostra as preferências de estatísticas para instâncias de banco de dados Oracle.



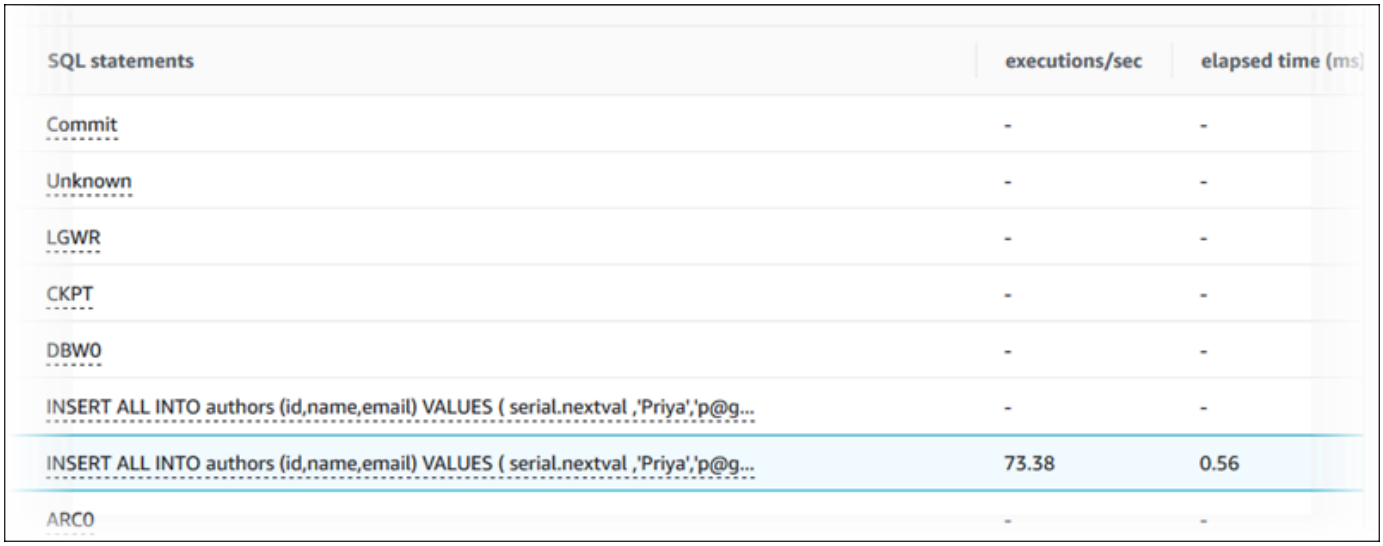
O exemplo a seguir mostra as preferências para instâncias de banco de dados do MariaDB e MySQL.



7. Escolha Save (Salvar) para salvar suas preferências.

   A tabela Top SQL (SQL principal) é atualizada.

   O exemplo a seguir mostra estatísticas de uma consulta SQL do Oracle.



Analisar a carga superior de PDBs da Oracle

Ao analisar a carga em um banco de dados de contêiner (CDB) da Oracle, convém identificar quais bancos de dados conectáveis (PDBs) contribuem mais para a carga do banco de dados. Além disso, é recomendável comparar a performance de PDBs individuais que estão executando consultas semelhantes para ajustar a performance. Para ter mais informações sobre CDBs da Oracle, consulte Arquitetura de banco de dados RDS para Oracle.

No painel do Insights de Performance do Amazon RDS, é possível encontrar informações sobre bancos de dados conectáveis (PDBs) na guia Principal PDB, na guia Dimensões.

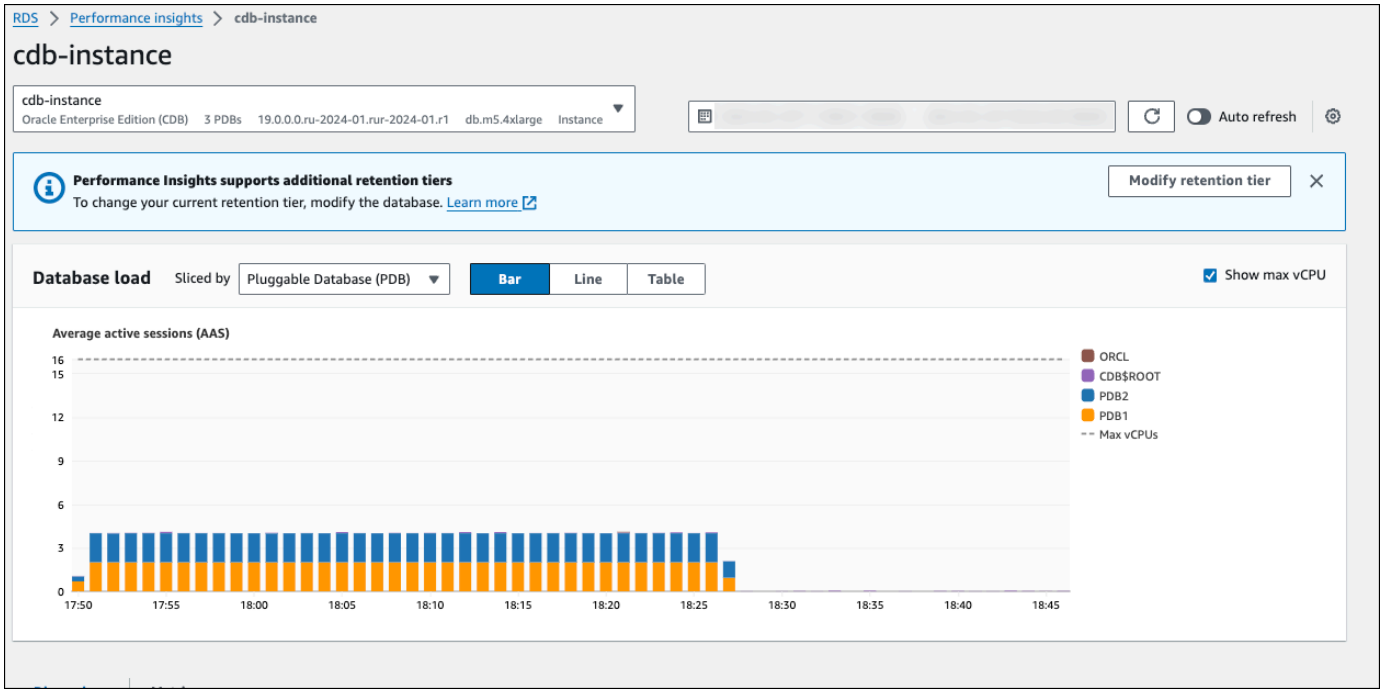
Para ter informações sobre compatibilidade de regiões, mecanismos de banco de dados e classes de instância com esse recurso, consulte O mecanismo de banco de dados do Amazon RDS, a região e a classe de instância são compatíveis com atributos do Insights de Performance..

Como analisar a carga do principal PDB em um CDB da Oracle

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação à esquerda, escolha Insights de Performance.
3. Escolha uma instância de banco de dados Oracle.

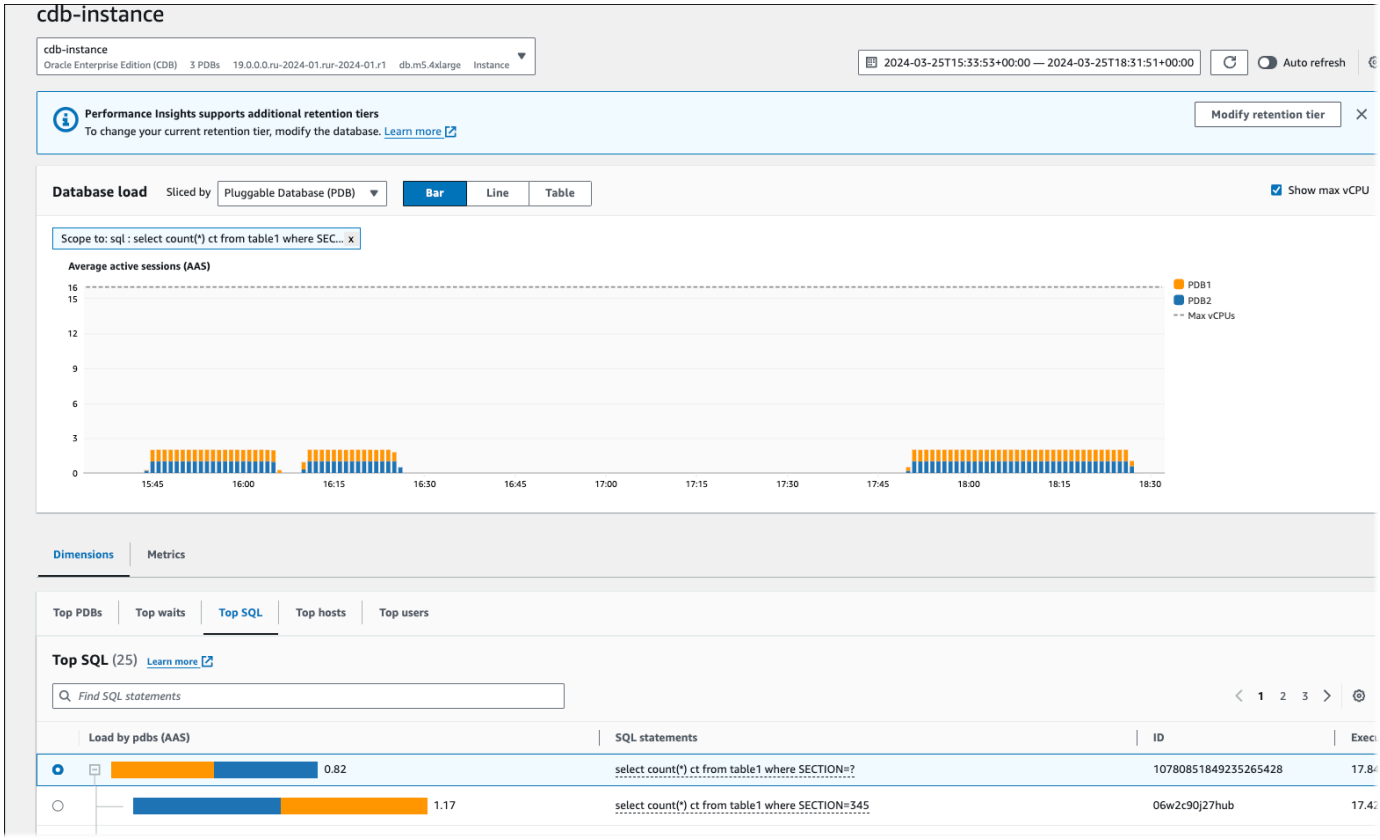
   O painel do Insights de Performance é exibido para a instância de banco de dados.
4. Na seção Carga do banco de dados, selecione Banco de dados conectável (PDB) o lado de Separar por.

O grafo Média de sessões ativas mostra o PDB com a carga mais alta. Os identificadores de PDBs aparecem à direita dos quadrados codificados por cores. Cada identificador identifica exclusivamente um PDB.

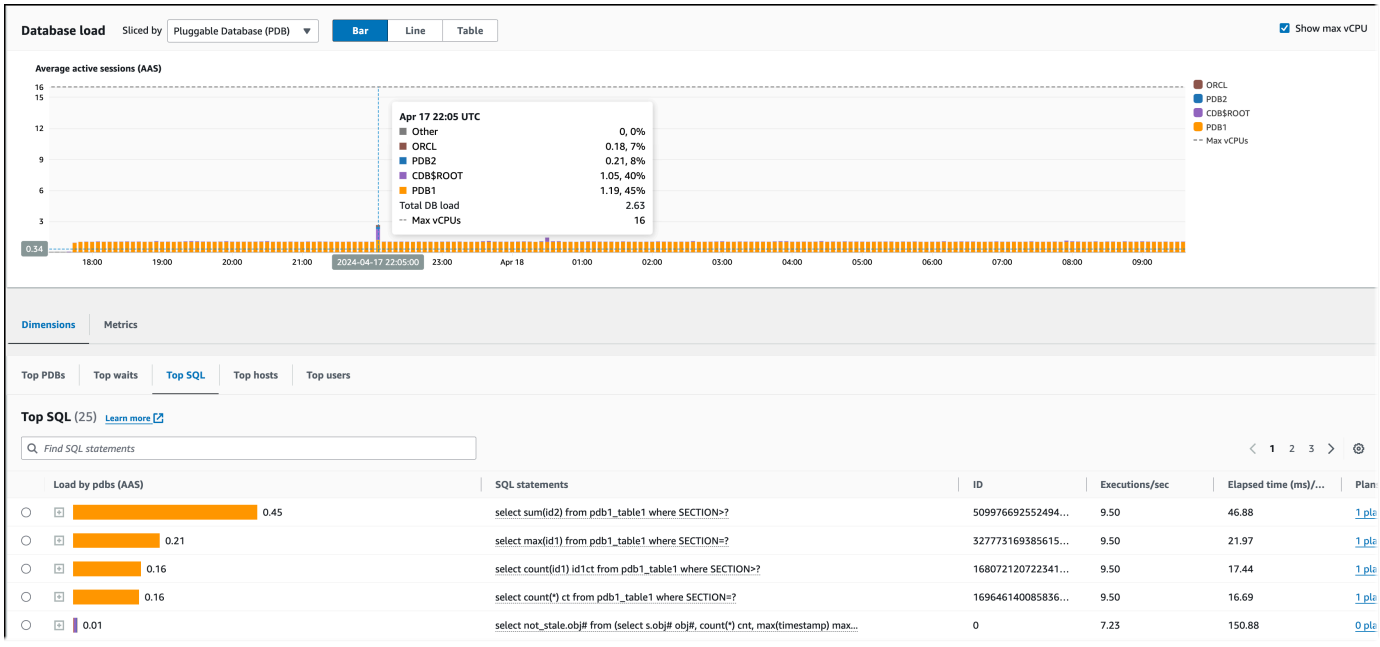


5. Role para baixo até a guia Top SQL (SQL principal).

   No exemplo a seguir, é possível ver a mesma consulta SQL e a carga que ela direciona para vários PDBs.



No exemplo a seguir, um único PDB está lidando com uma carga maior do que outros PDBs no CDB.



Para ter mais informações sobre CDBs da Oracle, consulte CDBs and PDBs.

Analisar planos de execução usando o painel do Insights de Performance para o Amazon RDS

No painel do Insights de Performance do Amazon RDS, é possível encontrar informações sobre planos de execução para instâncias de banco de dados do Oracle e do SQL Server. É possível usar essas informações para saber quais planos contribuem mais para a carga do banco de dados.

Para analisar os planos de execução do Oracle ou do SQL Server, consulte os tópicos a seguir.

Analisar planos de execução

- Analisar planos de execução do Oracle usando o painel do Insights de Performance para o Amazon RDS
- Analisar planos de execução do SQL Server usando o painel do Insights de Performance para o Amazon RDS

Visão geral da análise de planos de execução para o Amazon RDS

É possível usar o painel do Insights de Performance do Amazon RDS para saber quais planos contribuem mais para a carga de banco de dados das instâncias de banco de dados do Oracle e do SQL Server.

Por exemplo, as principais instruções SQL em um determinado momento podem estar usando os planos mostrados na tabela a seguir.

SQL principal	Planejamento
SELECT SUM(amount_sold) FROM sales WHERE prod_id = 10	Plano A
SELECT SUM(amount_sold) FROM sales WHERE prod_id = 521	Plano B
SELECT SUM(s_total) FROM sales WHERE region = 10	Plano A
SELECT * FROM emp WHERE emp_id = 1000	Plano C
SELECT SUM(amount_sold) FROM sales WHERE prod_id = 72	Plano A

Com o recurso de plano do Performance Insights, você pode fazer o seguinte:

- Descobrir quais planos são usados pelas principais consultas SQL.

  Por exemplo, você pode descobrir que a maior parte da carga de banco de dados é gerada por consultas usando o plano A e o plano B, com apenas uma pequena porcentagem usando o plano C.
- Comparar planos diferentes para a mesma consulta.

  No exemplo anterior, três consultas são idênticas, exceto o ID do produto. Duas consultas usam o plano A, mas uma consulta usa o plano B. Para ver a diferença nos dois planos, você pode usar o Performance Insights.
- Descobrir quando uma consulta mudou para um novo plano.

  Você pode ver que uma consulta usou o plano A e, em seguida, mudou para o plano B em um determinado momento. Houve uma alteração no banco de dados neste momento? Por exemplo, se uma tabela estiver vazia, o otimizador poderá escolher uma varredura de tabela completa. Se a tabela for carregada com um milhão de linhas, o otimizador poderá alternar para uma varredura de intervalo de índice.
- Verifique as etapas específicas de um plano com o maior custo.

  Por exemplo, uma consulta de longa execução pode mostrar uma condição de união ausente em uma equi-join. Essa condição faltante força uma união cartesiana, que une todas as linhas de duas tabelas.

Você pode executar as tarefas anteriores usando o recurso de captura de plano do Performance Insights. Assim como você pode dividir consultas por eventos de espera e SQL principal, é possível separá-las pela dimensão do plano.

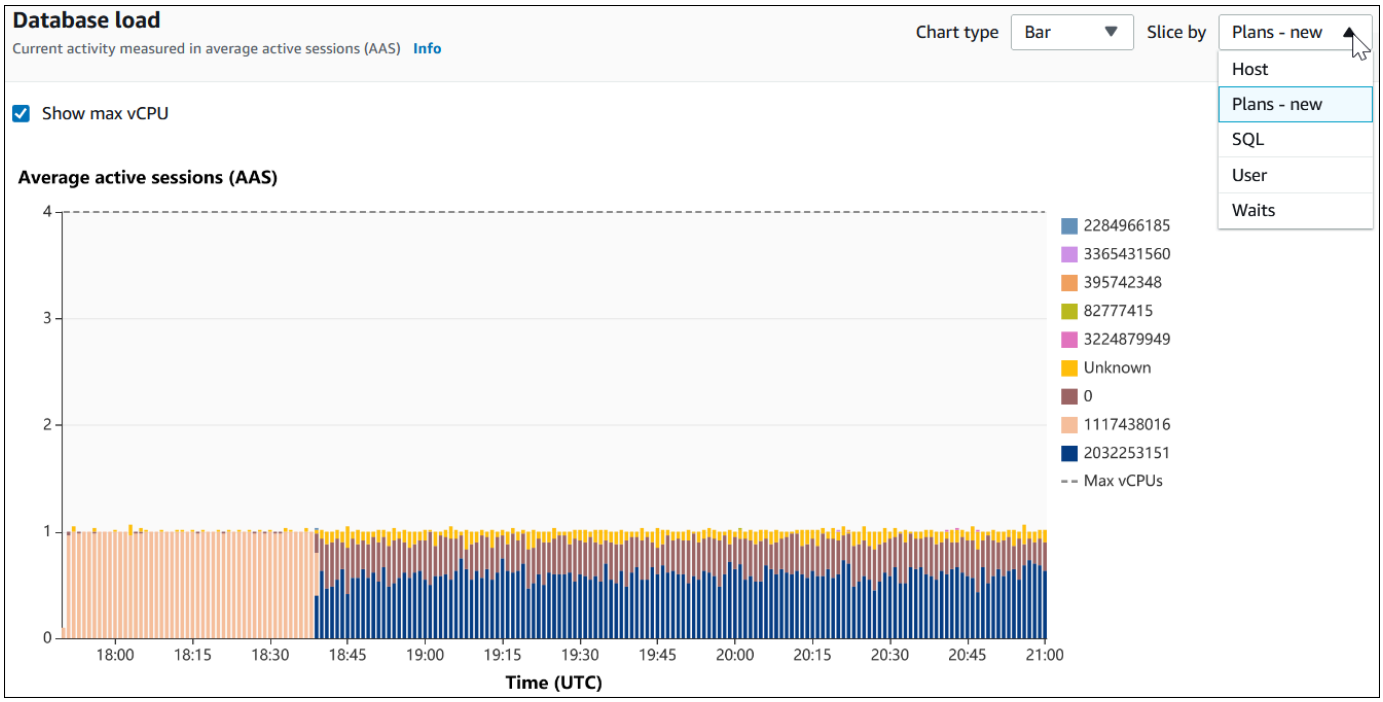
Analisar planos de execução do Oracle usando o painel do Insights de Performance para o Amazon RDS

Ao analisar a carga de banco de dados em um banco de dados Oracle, talvez você queira saber quais planos estão contribuindo mais para a carga do banco de dados. É possível determinar quais planos estão contribuindo mais para a carga de banco de dados usando o recurso de captura de plano do Insights de Performance.

Para analisar planos de execução Oracle usando o console

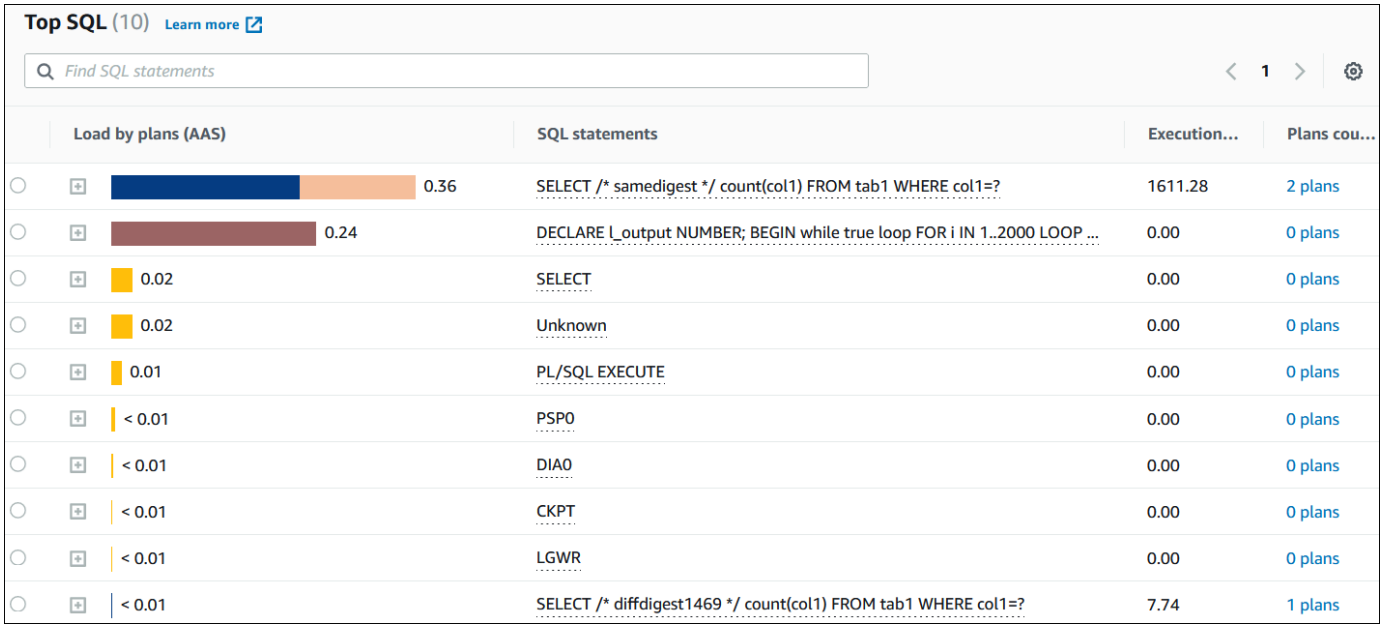
1. Abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Performance Insights.
3. Escolha uma instância de banco de dados do Oracle. O painel do Performance Insights será exibido nessa instância de banco de dados.
4. Na seção Database load (DB load) (Carga de banco de dados (carga de DB)), escolha Plans (Planos) o lado de Slice by (Separar por).

   O gráfico Média de sessões ativas mostra os planos usados pelas instruções SQL principais. Os valores de hash do plano aparecem à direita dos quadrados codificados por cores. Cada valor de hash identifica exclusivamente um plano.



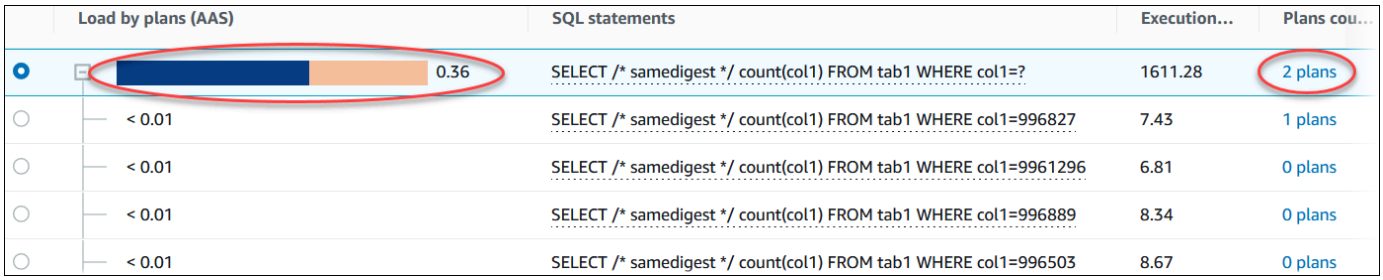
5. Role para baixo até a guia Top SQL (SQL principal).

   No exemplo a seguir, o resumo do SQL principal tem dois planos. Você pode dizer que é um resumo pelo ponto de interrogação na instrução.



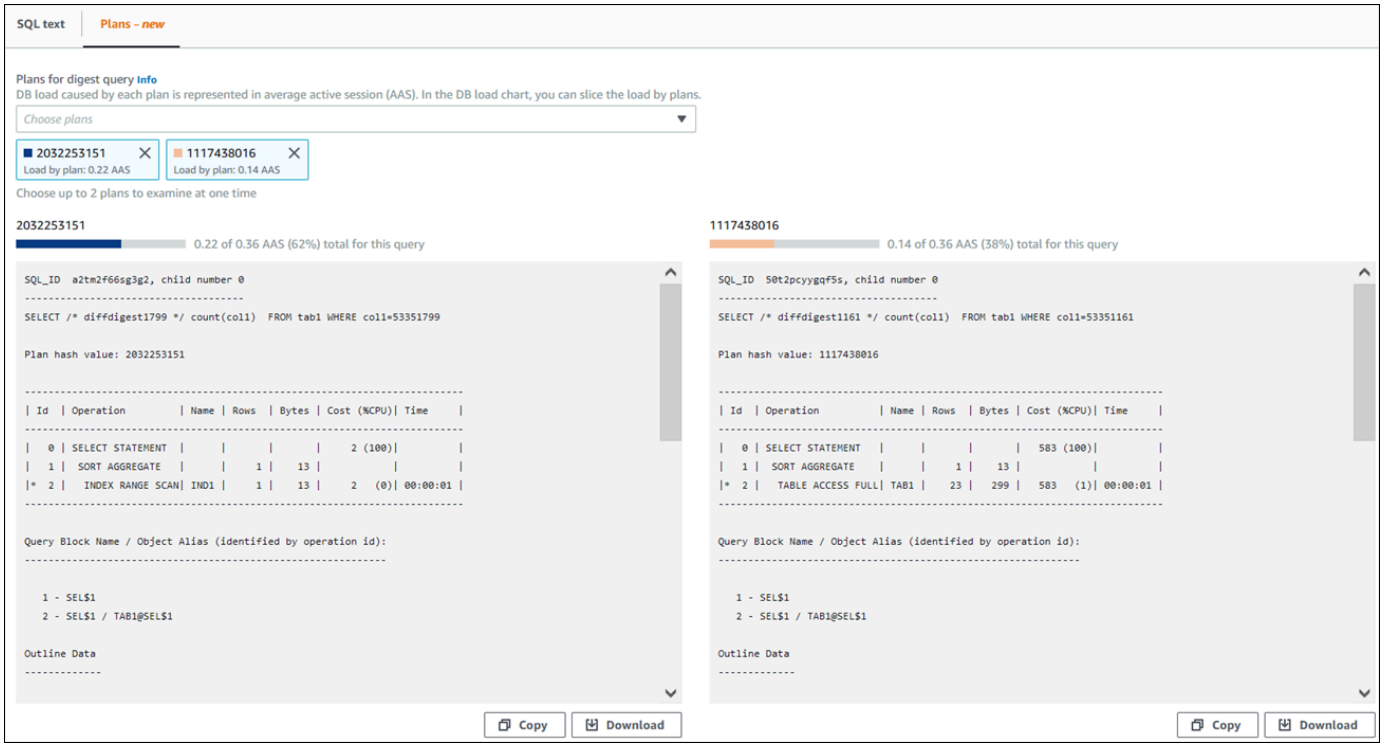
6. Escolha o resumo para expandi-lo em suas instruções de componente.

   No exemplo a seguir, a instrução SELECT é uma consulta de resumo. As consultas de componentes no resumo usam dois planos diferentes. As cores dos planos correspondem ao gráfico de carga do banco de dados. O número total de planos no resumo é mostrado na segunda coluna.

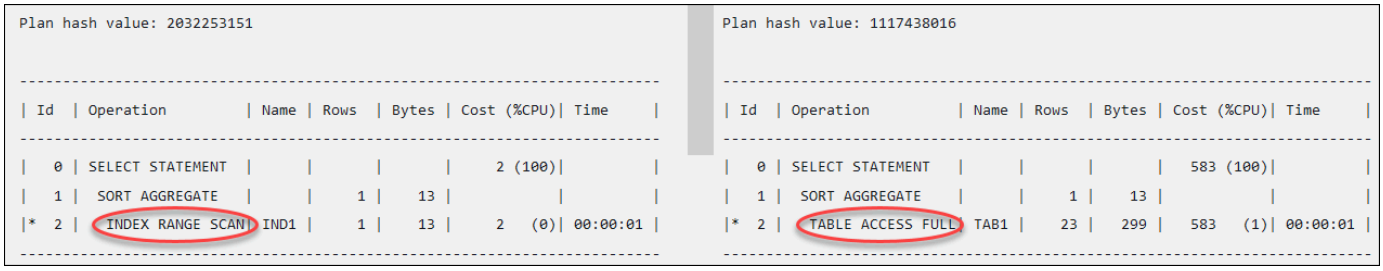


7. Role para baixo e escolha dois Planos para comparar na lista de Planos para consulta de resumo.

   Você pode visualizar um ou dois planos para uma consulta por vez. A captura de tela a seguir compara os dois planos no resumo, com hash 2032253151 e hash 1117438016. No exemplo a seguir, 62% das sessões ativas médias que executam essa consulta de resumo estão usando o plano à esquerda, enquanto 38% estão usando o plano à direita.



Neste exemplo, os planos diferem de forma importante. A Etapa 2 do plano 2032253151 usa uma verificação de índice, enquanto o plano 1117438016 usa uma varredura de tabela completa. Para uma tabela com um grande número de linhas, uma consulta de uma única linha é quase sempre mais rápida com uma varredura de índice.



8. (Opcional) Escolha Copy (Copiar) para copiar o plano para a área de transferência ou Download (Baixar) para salvar o plano no disco rígido.

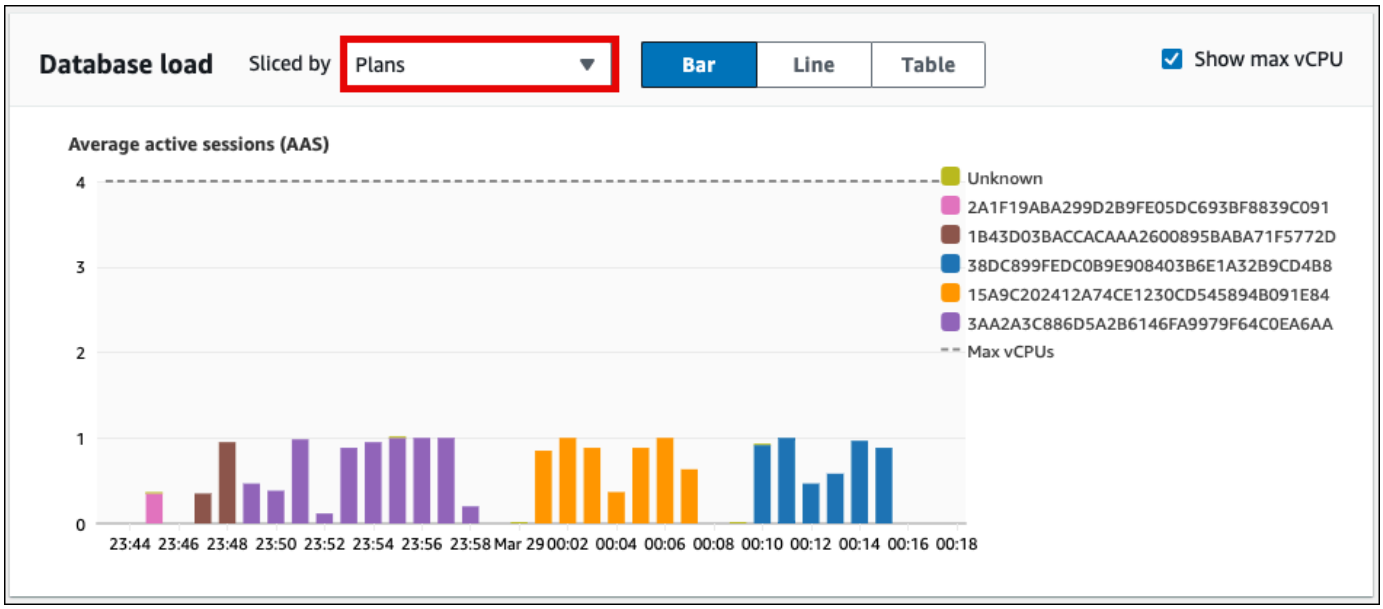
Analisar planos de execução do SQL Server usando o painel do Insights de Performance para o Amazon RDS

Ao analisar a carga do banco de dados em um SQL Server Database, convém saber quais planos estão contribuindo mais para a carga do banco de dados. É possível determinar quais planos estão contribuindo mais para a carga de banco de dados usando o recurso de captura de plano do Insights de Performance.

Como analisar planos de execução do SQL Server usando o console

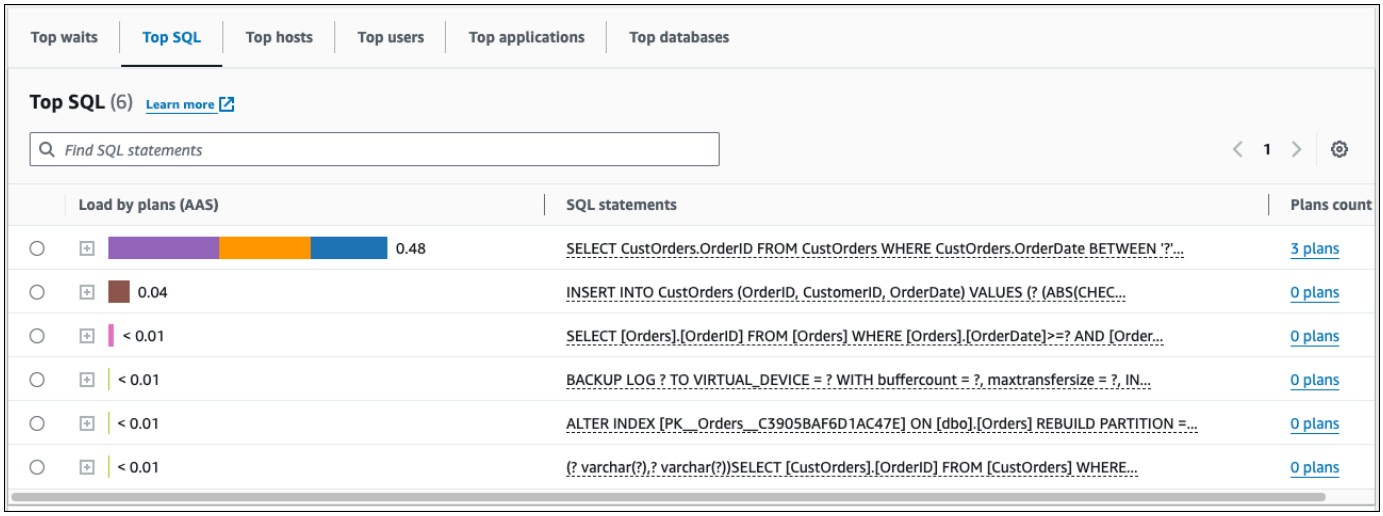
1. Abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Performance Insights.
3. Escolha uma instância de banco de dados do SQL Server. O painel do Performance Insights será exibido nessa instância de banco de dados.
4. Na seção Database load (DB load) (Carga de banco de dados (carga de DB)), escolha Plans (Planos) o lado de Slice by (Separar por).

   O gráfico Média de sessões ativas mostra os planos usados pelas instruções SQL principais. Os valores de hash do plano aparecem à direita dos quadrados codificados por cores. Cada valor de hash identifica exclusivamente um plano.



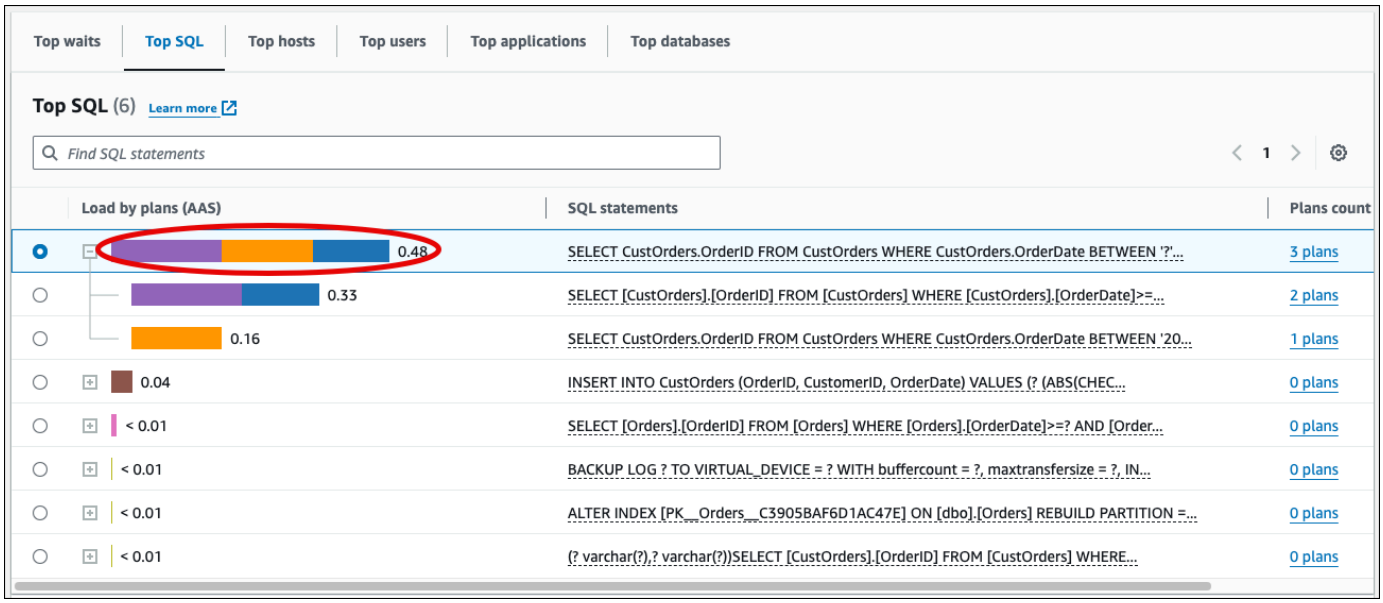
5. Role para baixo até a guia Top SQL (SQL principal).

   No exemplo a seguir, o resumo do SQL principal tem três planos. A presença de um ponto de interrogação na declaração SQL indica que a declaração é um resumo. Para visualizar a declaração SQL completa, escolha um valor na coluna Declarações SQL.



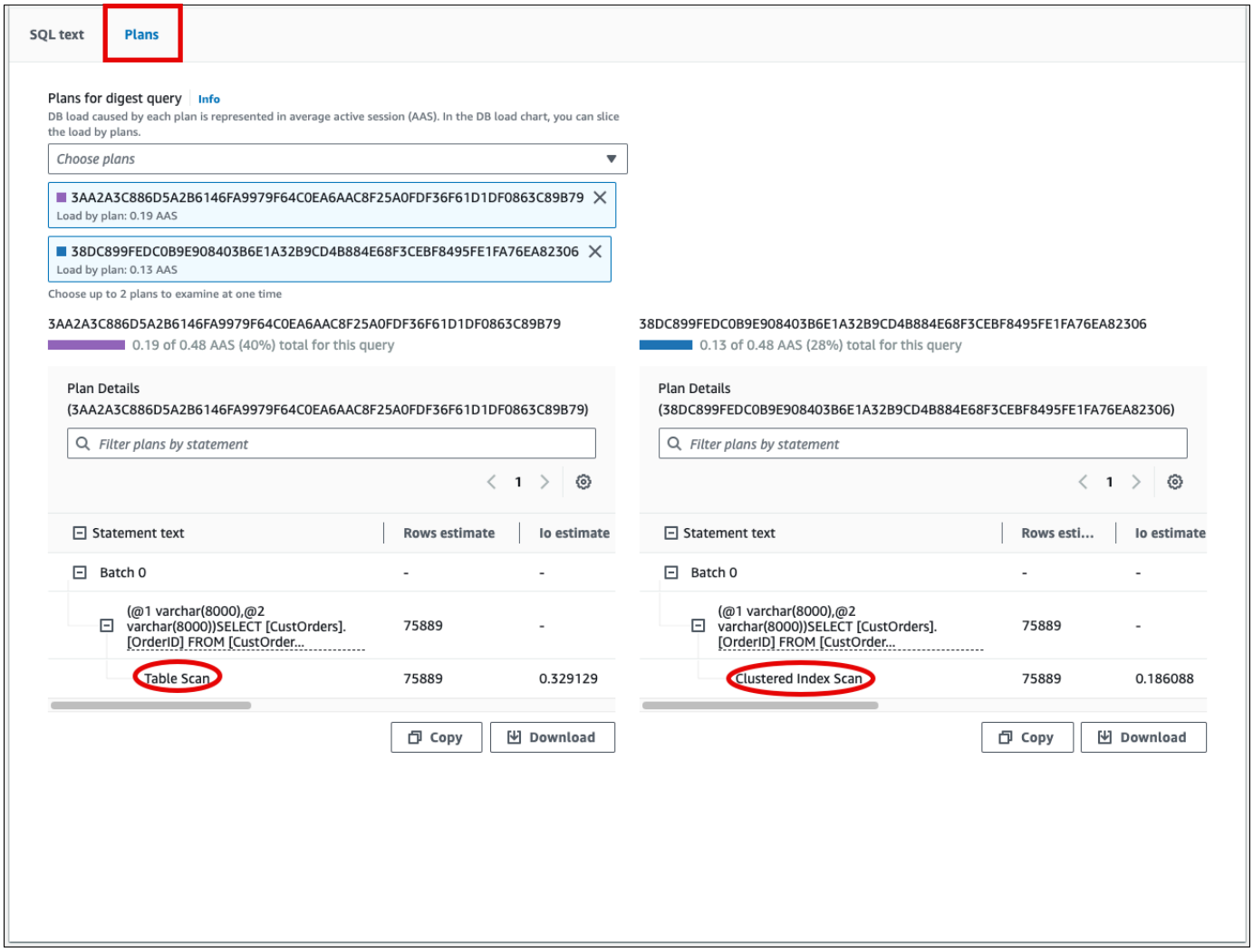
6. Escolha o resumo para expandi-lo em suas instruções de componente.

   No exemplo a seguir, a instrução SELECT é uma consulta de resumo. As consultas de componentes no resumo usam três planos diferentes. As cores atribuídas aos planos correspondem ao grafo de carga do banco de dados.



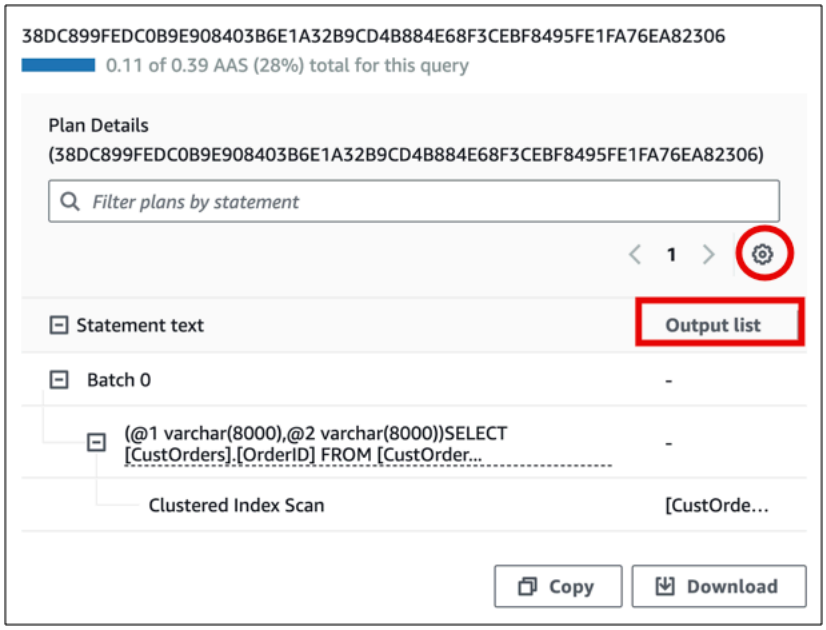
7. Role para baixo e escolha dois Planos para comparar na lista de Planos para consulta de resumo.

   Você pode visualizar um ou dois planos para uma consulta por vez. O screenshot a seguir compara os dois planos no resumo. No exemplo a seguir, 40% das sessões ativas médias que executam essa consulta de resumo estão usando o plano à esquerda, enquanto 28% estão usando o plano à direita.



No exemplo anterior, os planos apresentam diferenças importantes. A etapa 2 no plano à esquerda usa uma verificação de tabela, enquanto o plano à direita usa uma verificação de índice em cluster. Para uma tabela com um grande número de linhas, uma consulta que recupere uma única linha é quase sempre mais rápida com uma verificação de índice em cluster.

8. (Opcional) Selecione o ícone Configurações na tabela Detalhes do plano para personalizar a visibilidade e a ordem das colunas. O screenshot a seguir mostra a tabela Detalhes do plano com a coluna Lista de saídas como a segunda coluna.



9. (Opcional) Escolha Copy (Copiar) para copiar o plano para a área de transferência ou Download (Baixar) para salvar o plano no disco rígido.

> **Note**
>
> O Insights de Performance exibe planos de execução estimados usando uma tabela de árvore hierárquica. A tabela inclui as informações de execução parcial de cada declaração. Para ter mais informações sobre as colunas na tabela Detalhes do plano, consulte SET SHOWPLAN_ALL na documentação do SQL Server. Para exibir as informações de execução completa de um plano de execução estimado, escolha Fazer download para baixar o plano e, depois, carregue o plano no SQL Server Management Studio. Para ter mais informações sobre a exibição de um plano de execução estimado usando o SQL Server Management Studio, consulte Display an Estimated Execution Plan na documentação do SQL Server.

Visualizar as recomendações proativas do Performance Insights

O Insights de Performance do Amazon RDS monitora métricas específicas e cria limites automaticamente analisando quais níveis podem ser problemáticos para um recurso específico. Quando os novos valores de métricas ultrapassam um limite predefinido em um período específico, o Performance Insights gera uma recomendação proativa. Essa recomendação ajuda a evitar um impacto futuro na performance do banco de dados. Para receber essas recomendações proativas, é necessário ativar o Performance Insights com um período de retenção de nível pago.

Para obter mais informações sobre como ativar o Insights de Performance, consulte Ativar e desativar o Insights de Performance do Amazon RDS. Para ter informações sobre preços e retenção de dados para o Performance Insights, consulte Preços e retenção de dados para o Performance Insights.

Para descobrir as regiões, os mecanismos de banco de dados e as classes de instância compatíveis com as recomendações proativas, consulte O mecanismo de banco de dados do Amazon RDS, a região e a classe de instância são compatíveis com atributos do Insights de Performance..

É possível visualizar a análise detalhada e as investigações recomendadas de recomendações proativas na página de detalhes da recomendação.

Para obter mais informações e recomendações, consulte Recomendações do Amazon RDS.

Como visualizar a análise detalhada de uma recomendação proativa

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, execute qualquer uma das seguintes opções:
   - Selecione Recomendações.

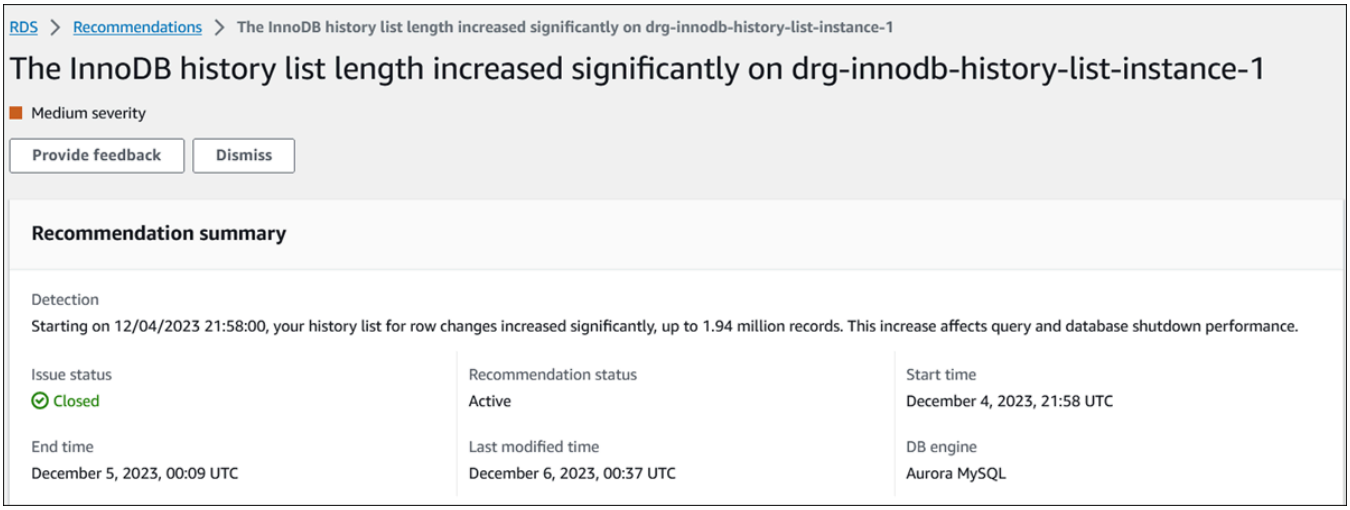
     A página Recomendações exibe uma lista de recomendações classificadas pela gravidade de todos os recursos da conta.
   - Selecione Bancos de dados e, depois, escolha Recomendações para um recurso na página de bancos de dados.

     A guia Recomendações exibe as recomendações e os detalhes do recurso selecionado.
3. Encontre uma recomendação proativa e escolha Visualizar detalhes.

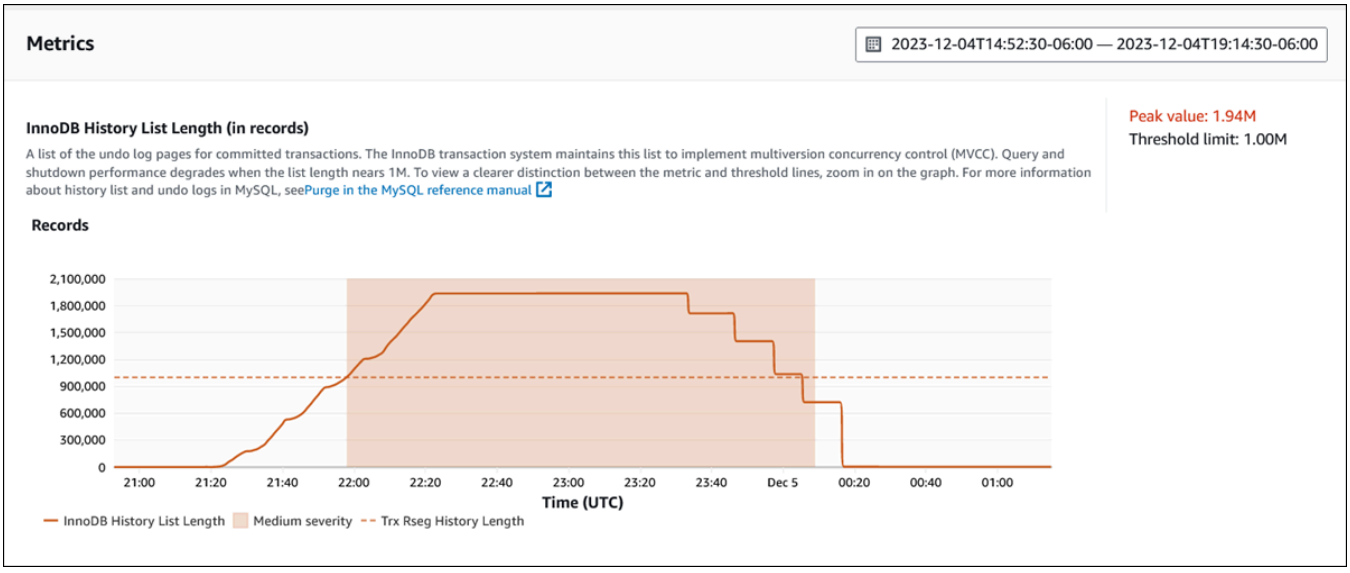
   A página de detalhes da recomendação é exibida. O título fornece o nome do recurso afetado com o problema detectado e a gravidade.

   Veja a seguir os componentes na página de detalhes da recomendação:

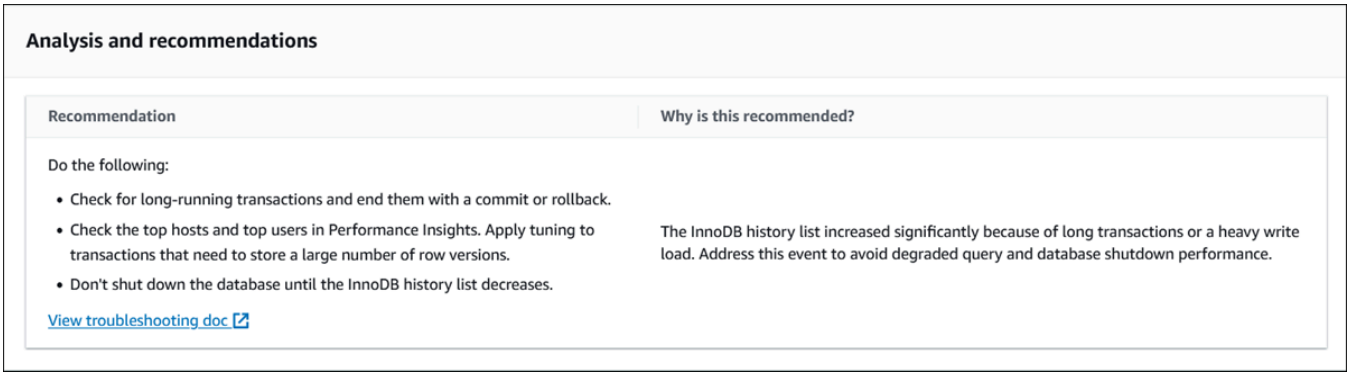
   - Resumo da recomendação: o problema detectado, a recomendação e o status do problema, a hora de início e término do problema, a hora de modificação da recomendação e o tipo de mecanismo.



- Métricas: os grafos do problema detectado. Cada grafo exibe um limite determinado pelo comportamento básico do recurso, bem como dados da métrica relatados a partir da hora de início do problema.



- Análise e recomendações: a recomendação e o motivo da recomendação sugerida.



É possível analisar a causa do problema e, depois, executar as ações recomendadas sugeridas para corrigir o problema ou escolher Dispensar no canto superior direito para dispensar a recomendação.

Recuperar métricas com a API do Insights de Performance do Amazon RDS

Quando o Insights de Performance está ativado, a API oferece visibilidade à performance da instância. O Amazon CloudWatch Logs fornece a fonte de autorização para métricas de monitoramento fornecidas para serviços da AWS.

O Performance Insights oferece uma visão específica do domínio da carga do banco de dados medida como sessões ativas médias (AAS). Essa métrica aparece para os consumidores de API como um conjunto de dados bidimensional de séries temporais. A dimensão de tempo dos dados fornece a carga do banco de dados para cada ponto de tempo no intervalo de tempo consultado. Cada ponto de tempo decompõe a carga geral em relação às dimensões solicitadas, como `SQL`, `Wait-event`, `User` ou `Host`, medidas naquele ponto de tempo.

O Amazon RDS Performance Insights monitora o cluster Amazon RDS da instância de banco de dados , para que você possa analisar e solucionar problemas relacionados à performance do banco de dados. Uma maneira de visualizar os dados do Performance Insights está no AWS Management Console. O Performance Insights também fornece uma API pública para que você possa consultar seus próprios dados. É possível usar a API para fazer o seguinte:

- Descarregar dados em um banco de dados
- Adicione dados do Performance Insights aos painéis de monitoramento existentes
- Criar ferramentas de monitoramento

Para usar a API do Performance Insights, habilite o Performance Insights em uma das suas instâncias de banco de dados do Amazon RDS. Para obter informações sobre como habilitar o Performance Insights, consulte Ativar e desativar o Insights de Performance do Amazon RDS. Para obter mais informações sobre a API do Performance Insights, consulte a Referência de API do Amazon RDS Performance Insights.

A API do Performance Insights fornece as operações a seguir.

Ação do Performance Insights	AWS CLI command	Descrição
CreatePerformanceAnalysisReport	aws pi create-performance-analysis-report	Cria um relatório de análise de performance referente a um período específico para a instância de banco de dados. O resultado é `AnalysisReportId`, que é o identificador exclusivo do relatório.
DeletePerformanceAnalysisReport	aws pi delete-performance-analysis-report	Exclui um relatório de análise de performance.
DescribeDimensionKeys	aws pi describe-dimension-keys	Recuperar as N principais chaves de dimensão de uma métrica por um período específico.
GetDimensionKeyDetails	aws pi get-dimension-key-details	Recupera os atributos do grupo de dimensões especificado para uma instância de banco de dados ou fonte de dados. Por exemplo, se você especificar um ID SQL e se os detalhes da dimensão estiverem disponíveis, `GetDimensionKeyDetails` recuperará o texto completo da dimensão `db.sql.statement` associada a esse ID. Essa operação é útil porque `GetResourceMetrics` e `DescribeDimensionKeys` não oferecem suporte à

Ação do Performance Insights	AWS CLI command	Descrição
		recuperação de texto grande de instrução SQL.
GetPerformanceAnalysisReport	aws pi get-performance-analysis-report	Recupera o relatório, incluindo os insights do relatório. O resultado inclui o status do relatório, o ID do relatório , os detalhes do horário do relatório, os insights e as recomendações.
GetResourceMetadata	aws pi get-resource-metadata	Recupere os metadados para diferentes recursos. Por exemplo, os metadados podem indicar que um recurso está ativado ou desativado em uma instância de banco de dados específica.
GetResourceMetrics	aws pi get-resource-metrics	Recupera as métricas do Performance Insights para um conjunto de fontes de dados, ao longo de um período. É possível fornecer grupos de dimensão e dimensões específicos e fornecer critérios de filtragem e agregação para cada grupo.
ListAvailableResourceDimensions	aws pi list-available-resource-dimensions	Recupere as dimensões que podem ser consultadas para cada tipo de métrica especificado em uma instância especificada.

Ação do Performance Insights	AWS CLI command	Descrição
ListAvailableResourceMetrics	aws pi list-available-resource-metrics	Recupere todas as métricas disponíveis dos tipos de métrica especificados que podem ser consultados para uma instância de banco de dados especificada.
ListPerformanceAnalysisReports	aws pi list-performance-analysis-reports	Recupera todos os relatórios de análise disponíveis para a instância de banco de dados. Os relatórios são listados com base na hora de início de cada relatório.
ListTagsForResource	aws pi list-tags-for-resource	Lista todas as tags de metadados adicionadas ao recurso. A lista inclui o nome e o valor da tag.
TagResource	aws pi tag-resource	Adiciona tags de metadados ao recurso do Amazon RDS. A tag inclui um nome e um valor.
UntagResource	aws pi untag-resource	Remove a tag de metadados do recurso.

Para ter mais informações sobre como recuperar métricas de séries temporais e exemplos da AWS CLI para o Insights de Performance, consulte os tópicos a seguir.

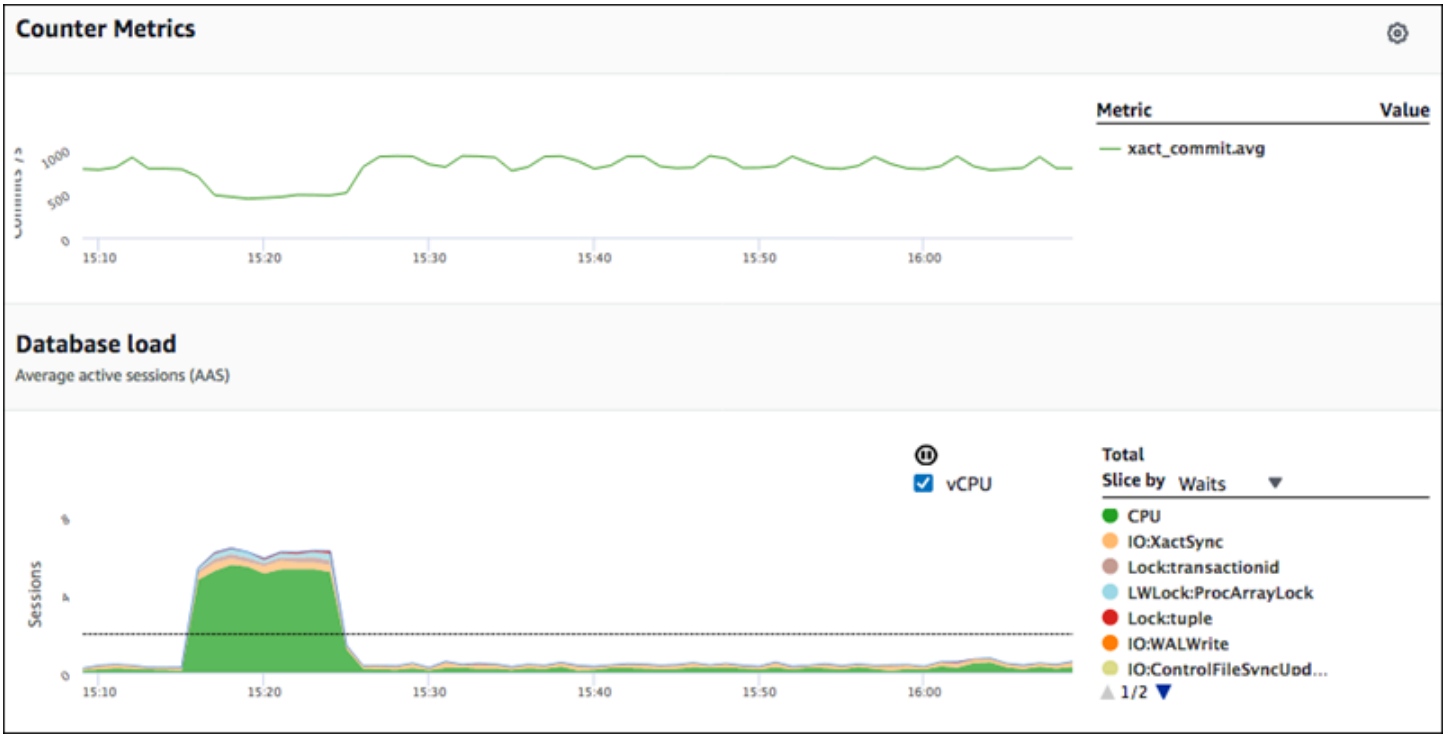
Tópicos

- Recuperar métricas de séries temporais do Insights de Performance
- AWS CLIExemplos da para o Performance Insights

Recuperar métricas de séries temporais do Insights de Performance

A operação `GetResourceMetrics` recupera uma ou mais métricas de séries temporais dos dados do Performance Insights. `GetResourceMetrics` requer uma métrica e um período de tempo e retorna uma resposta com uma lista de pontos de dados.

Por exemplo, o AWS Management Console usa `GetResourceMetrics` para preencher o gráfico Métricas de contador e o gráfico Carregamento do banco de dados, como visto na imagem a seguir.



Todas as métricas retornadas por `GetResourceMetrics` são métricas de séries temporais padrão, com exceção de `db.load`. Essa métrica é exibida no gráfico Carregamento do banco de dados. A métrica `db.load` é diferente das outras métricas da série temporal, pois você pode fragmentá-la em subcomponentes chamados de dimensões. Na imagem anterior, `db.load` é dividido e agrupado pelos estados de espera que compõem o `db.load`.

> **Note**
>
> `GetResourceMetrics` também pode retornar a métrica `db.sampleload`, mas a métrica `db.load` é apropriada na maioria dos casos.

Para obter informações sobre as métricas de contador retornadas pelo `GetResourceMetrics`, consulte Métricas de contadores do Performance Insights.

Os cálculos a seguir são compatíveis com as métricas:

- Média: o valor médio para a métrica por um período. Adicione `.avg` ao nome da métrica.
- Mínimo: o valor mínimo para a métrica por um período. Adicione `.min` ao nome da métrica.
- Máximo: o valor máximo para a métrica por um período. Adicione `.max` ao nome da métrica.
- Soma: a soma dos valores da métrica por um período. Adicione `.sum` ao nome da métrica.
- Contagem de amostra: o número de vezes que a métrica foi coletada por um período. Adicione `.sample_count` ao nome da métrica.

Por exemplo, considere que uma métrica é coletada por 300 segundos (5 minutos) e que a métrica seja coletada uma vez por minuto. Os valores de cada minuto são 1, 2, 3, 4 e 5. Nesse caso, os seguintes cálculos são retornados:

- Média: 3
- Mínimo: 1
- Máximo: 5
- Soma: 15
- Contagem de amostras: 5

Para obter informações sobre como usar o comando `get-resource-metrics` AWS CLI, consulte `get-resource-metrics`.

Para a opção `--metric-queries`, especifique uma ou mais consultas para as quais deseja obter resultados. Cada consulta consiste em um parâmetro obrigatório `Metric` e opcional `GroupBy` e em parâmetros `Filter`. Veja a seguir um exemplo de uma especificação de opção `--metric-queries`.

```
{
   "Metric": "string",
   "GroupBy": {
     "Group": "string",
     "Dimensions": ["string", ...],
     "Limit": integer
   },
   "Filter": {"string": "string"
     ...}
```

AWS CLIExemplos da para o Performance Insights

Nas seções a seguir, saiba mais sobre a AWS Command Line Interface (AWS CLI) para o Insights de Performance e use exemplos da AWS CLI.

Tópicos

- Ajuda integrada da AWS CLI para o Insights de Performance
- Recuperar métricas de contador
- Recuperar a média de carga de banco de dados para eventos de espera superior
- Recuperar a média de carga de banco de dados para SQL principal
- Recuperação da média de carga de banco de dados filtrada por SQL
- Recuperar o texto completo de uma instrução SQL
- Criar um relatório de análise de performance para um período
- Recuperar um relatório de análise de performance
- Listar todos os relatórios de análise de performance da instância de banco de dados
- Excluir um relatório de análise de performance
- Adicionar tags a um relatório de análise de performance
- Listar todas as tags para um relatório de análise de performance
- Excluir tags de um relatório de análise de performance

Ajuda integrada da AWS CLI para o Insights de Performance

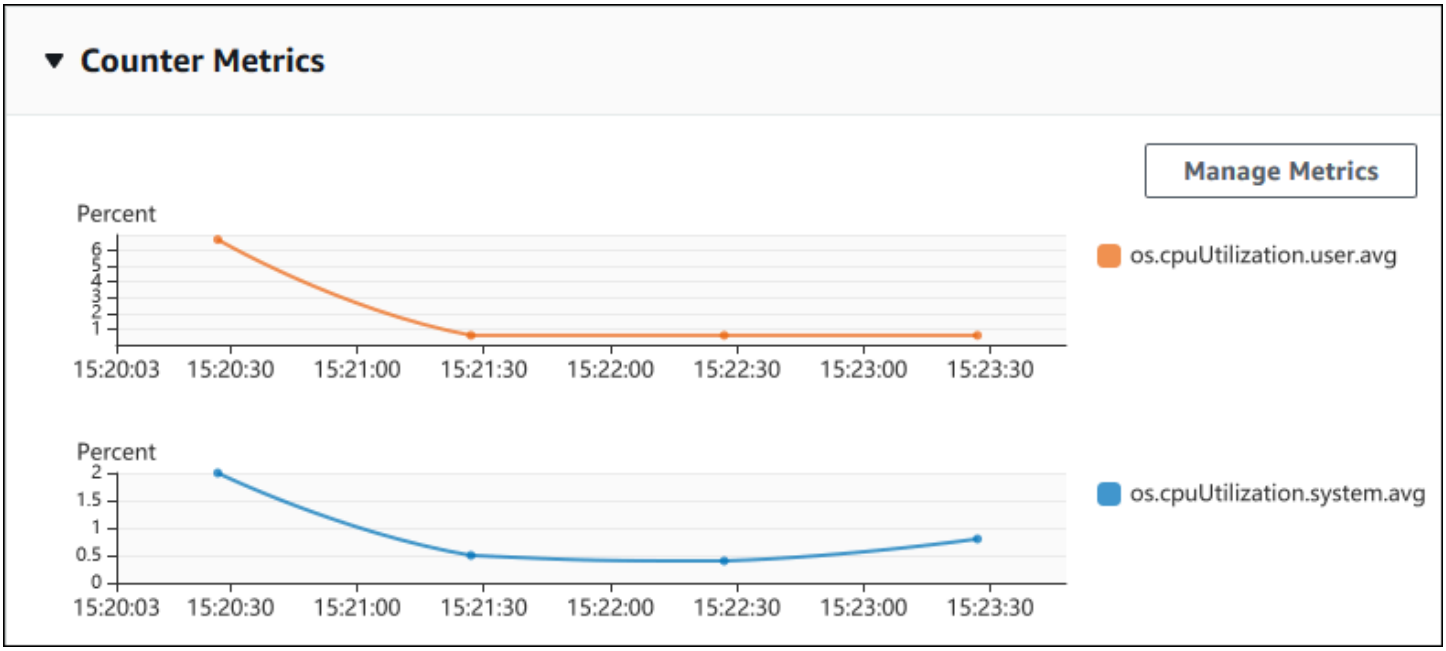
É possível visualizar dados do Performance Insights usando o AWS CLI. Você pode visualizar a ajuda dos comandos da AWS CLI para o Performance Insights, inserindo o seguinte na linha de comando.

```
aws pi help
```

Se você não tiver a AWS CLI instalada, consulte informações sobre como instalá-la em Installing the AWS CLI no Guia do usuário do AWS CLI.

Recuperar métricas de contador

A captura de tela a seguir mostra dois gráficos de métricas de contador no AWS Management Console.



O exemplo a seguir mostra como reunir os mesmos dados que o AWS Management Console usa para gerar os dois gráficos de métricas de contador.

Para Linux, macOS ou Unix:

```
aws pi get-resource-metrics \
   --service-type RDS \
   --identifier db-ID \
   --start-time 2018-10-30T00:00:00Z \
   --end-time   2018-10-30T01:00:00Z \
   --period-in-seconds 60 \
   --metric-queries '[{"Metric": "os.cpuUtilization.user.avg"  },
                      {"Metric": "os.cpuUtilization.idle.avg"}]'
```

Para Windows:

```
aws pi get-resource-metrics ^
   --service-type RDS ^
   --identifier db-ID ^
   --start-time 2018-10-30T00:00:00Z ^
   --end-time   2018-10-30T01:00:00Z ^
   --period-in-seconds 60 ^
   --metric-queries '[{"Metric": "os.cpuUtilization.user.avg"  },
                      {"Metric": "os.cpuUtilization.idle.avg"}]'
```

Você também pode tornar um comando mais fácil de ler, especificando um arquivo para a opção `--metrics-query`. O exemplo a seguir usa um arquivo chamado query.json para a opção. O arquivo tem o seguinte conteúdo.

```
[
    {
        "Metric": "os.cpuUtilization.user.avg"
    },
    {
        "Metric": "os.cpuUtilization.idle.avg"
    }
]
```

Execute o seguinte comando para usar o arquivo.

Para Linux, macOS ou Unix:

```
aws pi get-resource-metrics \
   --service-type RDS \
   --identifier db-ID \
   --start-time 2018-10-30T00:00:00Z \
   --end-time   2018-10-30T01:00:00Z \
   --period-in-seconds 60 \
   --metric-queries file://query.json
```

Para Windows:

```
aws pi get-resource-metrics ^
   --service-type RDS ^
   --identifier db-ID ^
   --start-time 2018-10-30T00:00:00Z ^
   --end-time   2018-10-30T01:00:00Z ^
   --period-in-seconds 60 ^
   --metric-queries file://query.json
```

O exemplo anterior especifica os seguintes valores para as opções:

- `--service-type`: RDS para Amazon RDS
- `--identifier` – O ID do recurso para a instância do banco de dados
- `--start-time` e `--end-time` – Os valores ISO 8601 de `DateTime` para o período a consultar, com vários formatos compatíveis

Ele consulta um intervalo de tempo de uma hora:

- `--period-in-seconds` – `60` para uma consulta por minuto
- `--metric-queries` – uma matriz de duas consultas, cada uma apenas para uma métrica.

  O nome da métrica usa pontos para classificar a métrica em uma categoria útil, com o elemento final sendo uma função. No exemplo, a função é `avg` para cada consulta. Como no Amazon CloudWatch, as funções com suporte são `min`, `max`, `total` e `avg`.

A resposta é semelhante à seguinte.

```
{
    "Identifier": "db-XXX",
    "AlignedStartTime": 1540857600.0,
    "AlignedEndTime": 1540861200.0,
    "MetricList": [
        { //A list of key/datapoints
            "Key": {
                "Metric": "os.cpuUtilization.user.avg" //Metric1
            },
            "DataPoints": [
                //Each list of datapoints has the same timestamps and same number of
 items
                {
                    "Timestamp": 1540857660.0, //Minute1
                    "Value": 4.0
                },
                {
                    "Timestamp": 1540857720.0, //Minute2
                    "Value": 4.0
                },
                {
                    "Timestamp": 1540857780.0, //Minute 3
                    "Value": 10.0
                }
                //... 60 datapoints for the os.cpuUtilization.user.avg metric
            ]
        },
        {
            "Key": {
                "Metric": "os.cpuUtilization.idle.avg" //Metric2
            },
```

```
            "DataPoints": [
                {
                    "Timestamp": 1540857660.0, //Minute1
                    "Value": 12.0
                },
                {
                    "Timestamp": 1540857720.0, //Minute2
                    "Value": 13.5
                },
                //... 60 datapoints for the os.cpuUtilization.idle.avg metric
            ]
        }
    ] //end of MetricList
} //end of response
```

A resposta tem `Identifier`, `AlignedStartTime` e `AlignedEndTime`. Se o valor de `--period-in-seconds` fosse `60`, as horas de início e término seriam alinhadas ao minuto. Se `--period-in-seconds` fosse `3600`, as horas de início e término teriam sido alinhadas à hora.

O `MetricList` na resposta tem um número de entradas, cada uma com uma entrada `Key` e `DataPoints`. Cada `DataPoint` tem um `Timestamp` e um `Value`. Cada lista `Datapoints` tem 60 pontos de dados, pois as consultas são para dados por minuto ao longo de uma hora, com `Timestamp1/Minute1`, `Timestamp2/Minute2` e assim por diante, até `Timestamp60/Minute60`.

Como a consulta é para duas métricas de contador diferentes, há dois elementos na resposta `MetricList`.

Recuperar a média de carga de banco de dados para eventos de espera superior

O exemplo a seguir é a mesma consulta que o AWS Management Console usa para gerar um gráfico de linha de área empilhada. Este exemplo recupera o `db.load.avg` para a última hora com carga dividida de acordo com os sete principais eventos de espera. O comando é o mesmo que o comando em Recuperar métricas de contador. No entanto, o arquivo query.json tem o seguinte conteúdo.

```
[
    {
        "Metric": "db.load.avg",
        "GroupBy": { "Group": "db.wait_event", "Limit": 7 }
    }
]
```

Execute o seguinte comando:

Para Linux, macOS ou Unix:

```
aws pi get-resource-metrics \
   --service-type RDS \
   --identifier db-ID \
   --start-time 2018-10-30T00:00:00Z \
   --end-time   2018-10-30T01:00:00Z \
   --period-in-seconds 60 \
   --metric-queries file://query.json
```

Para Windows:

```
aws pi get-resource-metrics ^
   --service-type RDS ^
   --identifier db-ID ^
   --start-time 2018-10-30T00:00:00Z ^
   --end-time   2018-10-30T01:00:00Z ^
   --period-in-seconds 60 ^
   --metric-queries file://query.json
```

O exemplo especifica a métrica de `db.load.avg` e um `GroupBy` dos sete principais eventos de espera. Para obter detalhes sobre valores válidos para esse exemplo, consulte DimensionGroup na Referência de API do Performance Insights.

A resposta é semelhante à seguinte.

```
{
    "Identifier": "db-XXX",
    "AlignedStartTime": 1540857600.0,
    "AlignedEndTime": 1540861200.0,
    "MetricList": [
        { //A list of key/datapoints
            "Key": {
                //A Metric with no dimensions. This is the total db.load.avg
                "Metric": "db.load.avg"
            },
            "DataPoints": [
                //Each list of datapoints has the same timestamps and same number of
 items
                {
                    "Timestamp": 1540857660.0, //Minute1
```

```
                    "Value": 0.5166666666666667
                },
                {
                    "Timestamp": 1540857720.0, //Minute2
                    "Value": 0.38333333333333336
                },
                {
                    "Timestamp": 1540857780.0, //Minute 3
                    "Value": 0.26666666666666666
                }
                //... 60 datapoints for the total db.load.avg key
            ]
        },
        {
            "Key": {
                //Another key. This is db.load.avg broken down by CPU
                "Metric": "db.load.avg",
                "Dimensions": {
                    "db.wait_event.name": "CPU",
                    "db.wait_event.type": "CPU"
                }
            },
            "DataPoints": [
                {
                    "Timestamp": 1540857660.0, //Minute1
                    "Value": 0.35
                },
                {
                    "Timestamp": 1540857720.0, //Minute2
                    "Value": 0.15
                },
                //... 60 datapoints for the CPU key
            ]
        },
        //... In total we have 8 key/datapoints entries, 1) total, 2-8) Top Wait Events
    ] //end of MetricList
} //end of response
```

Nessa resposta, há oito entradas no `MetricList`. Há uma entrada para o total `db.load.avg`, e sete entradas cada para o `db.load.avg`, divididas de acordo com um dos sete principais eventos de espera. Ao contrário do primeiro exemplo, como havia uma dimensão de agrupamento, deve haver uma chave para cada agrupamento da métrica. Não pode haver apenas uma chave para cada métrica, como no caso de uso de métricas de contador.

Recuperar a média de carga de banco de dados para SQL principal

O exemplo a seguir agrupa `db.wait_events` pelas 10 principais instruções SQL. Existem dois grupos diferentes para instruções SQL:

- `db.sql` – a instrução SQL completa, como `select * from customers where customer_id = 123`
- `db.sql_tokenized` – a instrução SQL tokenizada, como `select * from customers where customer_id = ?`

Ao analisar a performance do banco de dados, pode ser útil considerar instruções SQL que diferem apenas por seus parâmetros como um item lógico. Então, você pode usar `db.sql_tokenized` ao consultar. No entanto, especialmente quando você está interessado em explicar planos, às vezes é mais útil examinar instruções SQL completas com parâmetros e agrupamentos de consulta por `db.sql`. Existe um relacionamento pai-filho entre o SQL tokenizado e o SQL completo, com vários SQL completos (filhos) agrupados sob o mesmo SQL (pai) tokenizado.

O comando neste exemplo é semelhante ao comando em Recuperar a média de carga de banco de dados para eventos de espera superior. No entanto, o arquivo query.json tem o seguinte conteúdo.

```
[
    {
        "Metric": "db.load.avg",
        "GroupBy": { "Group": "db.sql_tokenized", "Limit": 10 }
    }
]
```

O exemplo a seguir usa `db.sql_tokenized`.

Para Linux, macOS ou Unix:

```
aws pi get-resource-metrics \
   --service-type RDS \
   --identifier db-ID \
   --start-time 2018-10-29T00:00:00Z \
   --end-time   2018-10-30T00:00:00Z \
   --period-in-seconds 3600 \
   --metric-queries file://query.json
```

Para Windows:

```
aws pi get-resource-metrics ^
   --service-type RDS ^
   --identifier db-ID ^
   --start-time 2018-10-29T00:00:00Z ^
   --end-time   2018-10-30T00:00:00Z  ^
   --period-in-seconds 3600 ^
   --metric-queries file://query.json
```

Este exemplo consulta mais de 24 horas, com um período de uma hora em segundos.

O exemplo especifica a métrica de `db.load.avg` e um `GroupBy` dos sete principais eventos de espera. Para obter detalhes sobre valores válidos para esse exemplo, consulte DimensionGroup na Referência de API do Performance Insights.

A resposta é semelhante à seguinte.

```
{
    "AlignedStartTime": 1540771200.0,
    "AlignedEndTime": 1540857600.0,
    "Identifier": "db-XXX",

    "MetricList": [ //11 entries in the MetricList
        {
            "Key": { //First key is total
                "Metric": "db.load.avg"
            }
            "DataPoints": [ //Each DataPoints list has 24 per-hour Timestamps and a
 value
                {
                    "Value": 1.6964980544747081,
                    "Timestamp": 1540774800.0
                },
                //... 24 datapoints
            ]
        },
        {
            "Key": { //Next key is the top tokenized SQL
                "Dimensions": {
                    "db.sql_tokenized.statement": "INSERT INTO authors (id,name,email)
 VALUES\n( nextval(?)  ,?,?)",
                    "db.sql_tokenized.db_id": "pi-2372568224",
                    "db.sql_tokenized.id": "AKIAIOSFODNN7EXAMPLE"
                },
```

```
                "Metric": "db.load.avg"
            },
            "DataPoints": [ //... 24 datapoints
            ]
        },
        // In total 11 entries, 10 Keys of top tokenized SQL, 1 total key
    ] //End of MetricList
} //End of response
```

Essa resposta tem 11 entradas no `MetricList` (1 total, 10 principais SQLs tokenizados), com cada entrada com 24 `DataPoints` por hora.

Para o SQL tokenizado, existem três entradas em cada lista de dimensões:

- `db.sql_tokenized.statement` – a instrução SQL tokenizada.
- `db.sql_tokenized.db_id` – o ID do banco de dados nativo usado para referência ao SQL ou um ID sintético que o Performance Insights gera para você quando o ID do banco de dados nativo não está disponível. Este exemplo retorna o ID sintético `pi-2372568224`.
- `db.sql_tokenized.id` – o ID da consulta dentro do Performance Insights.

  No AWS Management Console, esse ID é chamado de ID de suporte. Ele é chamado assim porque o ID consiste em dados que o AWS Support pode examinar para ajudar você a solucionar um problema no banco de dados. A AWS leva a segurança e privacidade de seus dados extremamente a sério, e quase todos os dados são armazenados criptografados com a chave do AWS KMS. Portanto, ninguém dentro da AWS pode examinar esses dados. No exemplo precedente, `tokenized.statement` e `tokenized.db_id` são armazenados em formato criptografado. Se você tiver um problema com o banco de dados, o Suporte da AWS poderá ajudar você consultando o ID de suporte.

Ao consultar, pode ser conveniente especificar `Group` em `GroupBy`. No entanto, para um controle mais refinado sobre os dados retornados, especifique a lista de dimensões. Por exemplo, se tudo o que for necessário for o `db.sql_tokenized.statement`, um atributo `Dimensions` poderá ser adicionado ao arquivo query.json.

```
[
    {
        "Metric": "db.load.avg",
        "GroupBy": {
            "Group": "db.sql_tokenized",
```

```
            "Dimensions":["db.sql_tokenized.statement"],
            "Limit": 10
        }
    }
]
```

Recuperação da média de carga de banco de dados filtrada por SQL



A imagem anterior mostra que uma consulta específica está selecionada e que o gráfico de linhas da área empilhada das sessões ativas da média superior tem o escopo para essa consulta. Embora a consulta ainda seja para os sete principais eventos de espera geral, o valor da resposta é filtrado. O filtro faz com que ele leve em consideração apenas as sessões correspondentes ao filtro específico.

A consulta da API correspondente neste exemplo é semelhante ao comando em Recuperar a média de carga de banco de dados para SQL principal. No entanto, o arquivo query.json tem o seguinte conteúdo.

```
[
 {
        "Metric": "db.load.avg",
        "GroupBy": { "Group": "db.wait_event", "Limit": 5  },
        "Filter": { "db.sql_tokenized.id": "AKIAIOSFODNN7EXAMPLE" }
```

```
    }
]
```

Para Linux, macOS ou Unix:

```
aws pi get-resource-metrics \
   --service-type RDS \
   --identifier db-ID \
   --start-time 2018-10-30T00:00:00Z \
   --end-time   2018-10-30T01:00:00Z \
   --period-in-seconds 60 \
   --metric-queries file://query.json
```

Para Windows:

```
aws pi get-resource-metrics ^
   --service-type RDS ^
   --identifier db-ID ^
   --start-time 2018-10-30T00:00:00Z ^
   --end-time   2018-10-30T01:00:00Z ^
   --period-in-seconds 60 ^
   --metric-queries file://query.json
```

A resposta é semelhante à seguinte.

```
{
    "Identifier": "db-XXX",
    "AlignedStartTime": 1556215200.0,
    "MetricList": [
        {
            "Key": {
                "Metric": "db.load.avg"
            },
            "DataPoints": [
                {
                    "Timestamp": 1556218800.0,
                    "Value": 1.4878117913832196
                },
                {
                    "Timestamp": 1556222400.0,
                    "Value": 1.192823803967328
                }
```

```
            ]
        },
        {
            "Key": {
                "Metric": "db.load.avg",
                "Dimensions": {
                    "db.wait_event.type": "io",
                    "db.wait_event.name": "wait/io/aurora_redo_log_flush"
                }
            },
            "DataPoints": [
                {
                    "Timestamp": 1556218800.0,
                    "Value": 1.1360544217687074
                },
                {
                    "Timestamp": 1556222400.0,
                    "Value": 1.058051341890315
                }
            ]
        },
        {
            "Key": {
                "Metric": "db.load.avg",
                "Dimensions": {
                    "db.wait_event.type": "io",
                    "db.wait_event.name": "wait/io/table/sql/handler"
                }
            },
            "DataPoints": [
                {
                    "Timestamp": 1556218800.0,
                    "Value": 0.16241496598639457
                },
                {
                    "Timestamp": 1556222400.0,
                    "Value": 0.05163360560093349
                }
            ]
        },
        {
            "Key": {
                "Metric": "db.load.avg",
                "Dimensions": {
```

```
                    "db.wait_event.type": "synch",
                    "db.wait_event.name": "wait/synch/mutex/innodb/
aurora_lock_thread_slot_futex"
                }
            },
            "DataPoints": [
                {
                    "Timestamp": 1556218800.0,
                    "Value": 0.11479591836734694
                },
                {
                    "Timestamp": 1556222400.0,
                    "Value": 0.013127187864644107
                }
            ]
        },
        {
            "Key": {
                "Metric": "db.load.avg",
                "Dimensions": {
                    "db.wait_event.type": "CPU",
                    "db.wait_event.name": "CPU"
                }
            },
            "DataPoints": [
                {
                    "Timestamp": 1556218800.0,
                    "Value": 0.05215419501133787
                },
                {
                    "Timestamp": 1556222400.0,
                    "Value": 0.05805134189031505
                }
            ]
        },
        {
            "Key": {
                "Metric": "db.load.avg",
                "Dimensions": {
                    "db.wait_event.type": "synch",
                    "db.wait_event.name": "wait/synch/mutex/innodb/lock_wait_mutex"
                }
            },
            "DataPoints": [
```

```
                {
                    "Timestamp": 1556218800.0,
                    "Value": 0.017573696145124718
                },
                {
                    "Timestamp": 1556222400.0,
                    "Value": 0.002333722287047841
                }
            ]
        }
    ],
    "AlignedEndTime": 1556222400.0
} //end of response
```

Nessa resposta, todos os valores são filtrados de acordo com a contribuição de SQL tokenizado AKIAIOSFODNN7EXAMPLE especificado no arquivo query.json. As chaves também podem seguir uma ordem diferente de uma consulta sem um filtro, porque são os cinco principais eventos de espera que afetaram o SQL filtrado.

Recuperar o texto completo de uma instrução SQL

O exemplo a seguir recupera o texto completo de uma instrução SQL para a instância de banco de dados `db-10BCD2EFGHIJ3KL4M5NO6PQRS5`. O `--group` é `db.sql`, e o `--group-identifier` é `db.sql.id`. Nesse exemplo, *`my-sql-id`* representa um ID SQL recuperado que invoca `pi get-resource-metrics` ou `pi describe-dimension-keys`.

Execute o seguinte comando:

Para Linux, macOS ou Unix:

```
aws pi get-dimension-key-details \
   --service-type RDS \
   --identifier db-10BCD2EFGHIJ3KL4M5NO6PQRS5 \
   --group db.sql \
   --group-identifier my-sql-id \
   --requested-dimensions statement
```

Para Windows:

```
aws pi get-dimension-key-details ^
   --service-type RDS ^
   --identifier db-10BCD2EFGHIJ3KL4M5NO6PQRS5 ^
```

```
   --group db.sql ^
   --group-identifier my-sql-id ^
   --requested-dimensions statement
```

Nesse exemplo, os detalhes das dimensões estão disponíveis. Assim, o Performance Insights recupera o texto completo da instrução SQL, sem truncá-lo.

```
{
    "Dimensions":[
    {
        "Value": "SELECT e.last_name, d.department_name FROM employees e, departments d
 WHERE e.department_id=d.department_id",
        "Dimension": "db.sql.statement",
        "Status": "AVAILABLE"
    },
    ...
    ]
}
```

Criar um relatório de análise de performance para um período

O exemplo a seguir cria um relatório de análise de performance com o horário de início `1682969503` e o horário de término `1682979503` do banco de dados `db-loadtest-0`.

```
aws pi create-performance-analysis-report \
        --service-type RDS \
        --identifier db-loadtest-0 \
        --start-time 1682969503 \
        --end-time 1682979503 \
        --region us-west-2
```

A resposta é o identificador exclusivo `report-0234d3ed98e28fb17` do relatório.

```
{
   "AnalysisReportId": "report-0234d3ed98e28fb17"
}
```

Recuperar um relatório de análise de performance

O exemplo a seguir recupera os detalhes do relatório de análise do relatório `report-0d99cc91c4422ee61`.

```
aws pi get-performance-analysis-report \
--service-type RDS \
--identifier db-loadtest-0 \
--analysis-report-id report-0d99cc91c4422ee61 \
--region us-west-2
```

A resposta fornece o status, o ID, os detalhes do horário e os insights do relatório.

```
        {
    "AnalysisReport": {
        "Status": "Succeeded",
        "ServiceType": "RDS",
        "Identifier": "db-loadtest-0",
        "StartTime": 1680583486.584,
        "AnalysisReportId": "report-0d99cc91c4422ee61",
        "EndTime": 1680587086.584,
        "CreateTime": 1680587087.139,
        "Insights": [
           ... (Condensed for space)
        ]
    }
}
```

Listar todos os relatórios de análise de performance da instância de banco de dados

O exemplo a seguir lista todos os relatórios de análise de performance disponíveis para o banco de dados `db-loadtest-0`.

```
aws pi list-performance-analysis-reports \
--service-type RDS \
--identifier db-loadtest-0 \
--region us-west-2
```

A resposta lista todos os relatórios com o ID do relatório, o status e os detalhes do período.

```
{
    "AnalysisReports": [
        {
            "Status": "Succeeded",
            "EndTime": 1680587086.584,
            "CreationTime": 1680587087.139,
```

```
            "StartTime": 1680583486.584,
            "AnalysisReportId": "report-0d99cc91c4422ee61"
        },
        {
            "Status": "Succeeded",
            "EndTime": 1681491137.914,
            "CreationTime": 1681491145.973,
            "StartTime": 1681487537.914,
            "AnalysisReportId": "report-002633115cc002233"
        },
        {
            "Status": "Succeeded",
            "EndTime": 1681493499.849,
            "CreationTime": 1681493507.762,
            "StartTime": 1681489899.849,
            "AnalysisReportId": "report-043b1e006b47246f9"
        },
        {
            "Status": "InProgress",
            "EndTime": 1682979503.0,
            "CreationTime": 1682979618.994,
            "StartTime": 1682969503.0,
            "AnalysisReportId": "report-01ad15f9b88bcbd56"
        }
    ]
}
```

Excluir um relatório de análise de performance

O exemplo a seguir exclui o relatório de análise do banco de dados `db-loadtest-0`.

```
aws pi delete-performance-analysis-report \
--service-type RDS \
--identifier db-loadtest-0 \
--analysis-report-id report-0d99cc91c4422ee61 \
--region us-west-2
```

Adicionar tags a um relatório de análise de performance

O exemplo a seguir adiciona uma tag com uma chave `name` e um valor `test-tag` ao relatório `report-01ad15f9b88bcbd56`.

```
aws pi tag-resource \
```

```
--service-type RDS \
--resource-arn arn:aws:pi:us-west-2:356798100956:perf-reports/RDS/db-loadtest-0/
report-01ad15f9b88bcbd56 \
--tags Key=name,Value=test-tag \
--region us-west-2
```

Listar todas as tags para um relatório de análise de performance

O exemplo a seguir lista todas as tags do relatório `report-01ad15f9b88bcbd56`.

```
aws pi list-tags-for-resource \
--service-type RDS \
--resource-arn arn:aws:pi:us-west-2:356798100956:perf-reports/RDS/db-loadtest-0/
report-01ad15f9b88bcbd56 \
--region us-west-2
```

A resposta lista o valor e a chave de todas as tags adicionadas ao relatório:

```
{
    "Tags": [
        {
            "Value": "test-tag",
            "Key": "name"
        }
    ]
}
```

Excluir tags de um relatório de análise de performance

O exemplo a seguir exclui a tag `name` do relatório `report-01ad15f9b88bcbd56`.

```
aws pi untag-resource \
--service-type RDS \
--resource-arn arn:aws:pi:us-west-2:356798100956:perf-reports/RDS/db-loadtest-0/
report-01ad15f9b88bcbd56 \
--tag-keys name \
--region us-west-2
```

Depois que a tag for excluída, chamar a API `list-tags-for-resource` não listará essa tag.

Registrar em log as chamadas do Performance Insights usando o AWS CloudTrail

O Performance Insights é executado com o AWS CloudTrail, um serviço que fornece um registro das ações realizadas por um usuário, uma função ou um serviço da AWS no Performance Insights. O CloudTrail captura todas as chamadas de API para o Performance Insights como eventos. Essa captura inclui chamadas do console do Amazon RDS e chamadas de código para as operações de API do Performance Insights.

Se você criar uma trilha, será possível ativar a entrega contínua de eventos do CloudTrail para um bucket do Amazon S3, incluindo eventos do Performance Insights. Se você não configurar uma trilha, ainda poderá visualizar os eventos mais recentes no console do CloudTrail no Histórico de eventos. Ao usar os dados coletados pelo CloudTrail, é possível determinar certas informações. Essas informações incluem a solicitação que foi feita ao Performance Insights, o endereço IP do qual a solicitação foi feita, quem fez a solicitação e quando ela foi feita. Também inclui detalhes adicionais.

Para saber mais sobre o CloudTrail, consulte o Guia do usuário do AWS CloudTrail.

Trabalhar com informações do Performance Insights no CloudTrail

O CloudTrail é habilitado em sua conta AWS ao criá-la. Quando ocorre uma atividade no Performance Insights, ela é registrada em um evento do CloudTrail com outros eventos de serviços da AWS no console do CloudTrail em Event history (Histórico de eventos). Você pode visualizar, pesquisar e baixar eventos recentes em sua conta da AWS. Para obter mais informações, consulte Viewing Events with CloudTrail Event History (Visualizar eventos com o histórico de eventos CloudTrail) no Guia do Usuário do AWS CloudTrail.

Para obter um registro contínuo de eventos na conta da AWS, incluindo eventos para o Performance Insights, crie uma trilha. Uma trilha permite que o CloudTrail entregue arquivos de log a um bucket do Amazon S3. Por padrão, quando uma trilha é criada no console, a mesma é aplicada a todas as regiões da AWS. A trilha registra em log eventos de todas as regiões da AWS na partição da AWS e entrega os arquivos de log para o bucket do Amazon S3 especificado por você. Além disso, é possível configurar outros AWS serviços para melhor analisar e agir de acordo com dados coletados do evento nos logs CloudTrail. Para obter mais informações, consulte os seguintes tópicos no Guia do usuário do AWS CloudTrail:

- Visão geral da criação de uma trilha
- Serviços e integrações com suporte no CloudTrail

- Configurando Notificações Amazon SNS para CloudTrail
- Receber arquivos de log do CloudTrail de várias regiões e receber arquivos de log do CloudTrail de várias contas

Todas as operações do Performance Insights são registradas em log pelo CloudTrail e documentadas na Referência de API do Performance Insights. Por exemplo, as chamadas para as operações `DescribeDimensionKeys` e `GetResourceMetrics` geram entradas nos arquivos de log do CloudTrail.

Cada entrada de log ou evento contém informações sobre quem gerou a solicitação. As informações de identidade ajudam a determinar:

- Se a solicitação foi feita com credenciais de usuário-raiz ou usuário do IAM.
- Se a solicitação foi feita com credenciais de segurança temporárias de uma função ou de um usuário federado.
- Se a solicitação foi feita por outro AWS serviço.

Para obter mais informações, consulte o Elemento userIdentity do CloudTrail.

Entradas de arquivo de log do Performance Insights

Uma trilha é uma configuração que permite a entrega de eventos como registros de log a um bucket do Amazon S3 especificado. Os arquivos de log do CloudTrail contêm uma ou mais entradas de log. Um evento representa uma solicitação única de qualquer fonte. Cada evento inclui informações sobre a operação solicitada, a data e a hora da operação, os parâmetros de solicitação e assim por diante. Os arquivos de log do CloudTrail não são um rastreamento de pilha ordenada de chamadas de API pública. Dessa forma, eles não são exibidos em uma ordem específica.

O exemplo a seguir mostra uma entrada de log do CloudTrail que demonstra a operação `GetResourceMetrics`.

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
         "principalId": "AKIAIOSFODNN7EXAMPLE",
        "arn": "arn:aws:iam::123456789012:user/johndoe",
        "accountId": "123456789012",
```

```
        "accessKeyId": "AKIAI44QH8DHBEXAMPLE",
        "userName": "johndoe"
    },
    "eventTime": "2019-12-18T19:28:46Z",
    "eventSource": "pi.amazonaws.com",
    "eventName": "GetResourceMetrics",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "72.21.198.67",
    "userAgent": "aws-cli/1.16.240 Python/3.7.4 Darwin/18.7.0 botocore/1.12.230",
    "requestParameters": {
        "identifier": "db-YTDU5J5V66X7CXSCVDFD2V3SZM",
        "metricQueries": [
            {
                "metric": "os.cpuUtilization.user.avg"
            },
            {
                "metric": "os.cpuUtilization.idle.avg"
            }
        ],
        "startTime": "Dec 18, 2019 5:28:46 PM",
        "periodInSeconds": 60,
        "endTime": "Dec 18, 2019 7:28:46 PM",
        "serviceType": "RDS"
    },
    "responseElements": null,
    "requestID": "9ffbe15c-96b5-4fe6-bed9-9fccff1a0525",
    "eventID": "08908de0-2431-4e2e-ba7b-f5424f908433",
    "eventType": "AwsApiCall",
    "recipientAccountId": "123456789012"
}
```


API do Performance Insights e endpoints da VPC de interface (AWS PrivateLink)

É possível usar o AWS PrivateLink para criar uma conexão privada entre a VPC e o Insights de Performance do Amazon RDS. É possível acessar o Performance Insights como se estivesse em sua VPC, sem usar um gateway da Internet, um dispositivo NAT, uma conexão VPN ou uma conexão do AWS Direct Connect. As instâncias na VPC não precisam de endereços IP públicos para acessar o Performance Insights.

Você estabelece essa conectividade privada criando um endpoint de interface, desenvolvido pelo AWS PrivateLink. Criaremos um endpoint de interface de rede em cada sub-rede que você habilitar

para o endpoint de interface. Estas são interfaces de rede gerenciadas pelo solicitante que servem como ponto de entrada para o tráfego destinado ao Performance Insights.

Para obter mais informações, consulte Acessar os Serviços da AWS pelo AWS PrivateLink no Guia do AWS PrivateLink.

Considerações sobre o Performance Insights

Antes de configurar um endpoint de interface para o Performance Insights, leia Considerations no Guia do AWS PrivateLink.

O Performance Insights aceita fazer chamadas para todas as suas ações de API por meio do endpoint da interface.

Por padrão, o acesso total ao Performance Insights é permitido no endpoint da interface. Para controlar o tráfego pra o Performance Insights por meio do endpoint da interface, associe um grupo de segurança às interfaces de rede do endpoint.

Disponibilidade

Atualmente, a API do Performance Insights comporta endpoints da VPC em Regiões da AWS que aceitam o Performance Insights. Para ter informações sobre a disponibilidade do Performance Insights, consulte Regiões e mecanismos de banco de dados compatíveis com o Insights de Performance no Amazon RDS .

Criar um endpoint de interface para o Performance Insights

É possível criar um endpoint de interface para o Performance Insights usando o console da Amazon VPC ou a AWS Command Line Interface (AWS CLI). Para obter mais informações, consulte Criar um endpoint de interface no Guia do usuário do AWS PrivateLink.

Crie um endpoint de interface para o Performance Insights usando o seguinte nome de serviço:

Se você habilitar o DNS privado para o endpoint da interface, poderá fazer solicitações de API ao Performance Insights usando seu nome do DNS regional padrão. Por exemplo, `pi.us-east-1.amazonaws.com`.

Criar uma política de endpoint da VPC para a API do Performance Insights

Política de endpoint é um recurso do IAM que você pode anexar ao endpoint de interface. A política de endpoint padrão permite acesso total ao Performance Insights por meio do endpoint de interface.

Para controlar o acesso permitido ao Performance Insights pela VPC anexe uma política de endpoint personalizada ao endpoint de interface.

Uma política de endpoint especifica as seguintes informações:

- As entidades principais que podem realizar ações (Contas da AWS, usuários do IAM e perfis do IAM).
- As ações que podem ser executadas.
- Os recursos nos quais as ações podem ser executadas.

Para obter mais informações, consulte Controlar o Acesso a Serviços Usando Políticas de Endpoint no AWS PrivateLink Guia.

Exemplo: Política de endpoint da VPC para ações do Performance Insights

O exemplo a seguir refere-se a uma política de endpoint personalizada. Quando anexada ao endpoint de interface, essa política concede acesso às ações do Performance Insights listadas para todas as entidades principais em todos os recursos.

```
{
   "Statement":[
      {
         "Principal":"*",
         "Effect":"Allow",
         "Action":[
            "rds:CreatePerformanceAnalysisReport",
            "rds:DeletePerformanceAnalysisReport",
            "rds:GetPerformanceAnalysisReport"
         ],
         "Resource":"*"
      }
   ]
}
```

Exemplo: política de endpoint da VPC que nega todo o acesso de uma conta da AWS especificada

A política de endpoint da VPC a seguir nega à conta da AWS `123456789012` todo o acesso aos recursos que usam o limite. A política permite todas as ações de outras contas.

```
{
```

```
  "Statement": [
    {
      "Action": "*",
      "Effect": "Allow",
      "Resource": "*",
      "Principal": "*"
    },
    {
      "Action": "*",
      "Effect": "Deny",
      "Resource": "*",
      "Principal": { "AWS": [ "123456789012" ] }
     }
   ]
}
```

Endereçamento IP para Performance Insights

Os endereços IP habilitam recursos na sua VPC para se comunicar com outros e com recursos na Internet. O Performance Insights comporta protocolos de endereçamento IPv4 e IPv6. Por padrão, o Performance Insights e a Amazon VPC usam o protocolo de endereçamento IPv4. Você não pode desativar esse comportamento. Ao criar uma VPC, especifique um bloco CIDR IPv4 (um intervalo de endereços IPv4 privados).

Também é possível atribuir um bloco CIDR IPv6 à VPC e às sub-redes e atribuir endereços IPv6 desse bloco a recursos do RDS na sub-rede. A compatibilidade com o protocolo IPv6 expande o número de endereços IP compatíveis. Ao usar o protocolo IPv6, você tem a garantia de que terá endereços disponíveis suficientes para o crescimento futuro da Internet. Os recursos do RDS novos e existentes podem usar endereços IPv4 e IPv6 na VPC. Configurar, proteger e converter o tráfego de rede entre os dois protocolos usados em diferentes partes de uma aplicação pode causar sobrecarga operacional. Você pode padronizar o protocolo IPv6 para recursos do Amazon RDS a fim de simplificar sua configuração de rede. Para ter mais informações sobre os endpoints e cotas de serviço, consulte Endpoints e cotas do Amazon Relational Database Service.

Para ter mais informações sobre o endereçamento IP do Amazon RDS, consulte Endereçamento IP do Amazon RDS.

Analisar anomalias de performance com o DevOps Guru para Amazon RDS

O Amazon DevOps Guru é um serviço de operações totalmente gerenciado que ajuda desenvolvedores e operadores a melhorar a performance e a disponibilidade de suas aplicações. O DevOps Guru dispensa as tarefas associadas à identificação de problemas operacionais, para que você possa implementar rapidamente recomendações para melhorar sua aplicação. Para ter mais informações, consulte O que é Configurar o Amazon DevOps Guru? no Guia do usuário do Amazon DevOps Guru.

O DevOps Guru detecta, analisa e faz recomendações de problemas operacionais para todos os mecanismos de banco de dados do Amazon RDS. O DevOps Guru para RDS amplia essa capacidade, aplicando machine learning a métricas do Performance Insights para bancos de dados do RDS para PostgreSQL. Esses recursos de monitoramento permitem que o DevOps Guru para RDS detecte e diagnostique gargalos de performance e recomende ações corretivas específicas. O DevOps Guru para RDS também pode detectar condições problemáticas no banco de dados RDS para PostgreSQL antes que elas ocorram.

Agora é possível ver essas recomendações no console do RDS. Para ter mais informações, consulte Recomendações do Amazon RDS.

O vídeo a seguir apresenta uma visão geral do DevOps Guru para RDS.

Para se aprofundar no assunto, consulte Bastidores do Amazon DevOps Guru para RDS.

Tópicos

- Benefícios do DevOps Guru para RDS
- Como funciona o DevOps Guru para RDS
- Configurar o DevOps Guru para RDS

Benefícios do DevOps Guru para RDS

Se você é responsável por um banco de dados RDS para PostgreSQL, talvez não esteja ciente da existência de um evento ou regressão que está afetando esse banco de dados. Quando você ficar sabendo do problema, talvez não saiba por que ele está ocorrendo ou o que fazer a respeito. Em vez de recorrer a um administrador de banco de dados (DBA) para obter ajuda ou depender de ferramentas de terceiros, você pode seguir as recomendações do DevOps Guru para RDS.

Estas são as vantagens das análises detalhadas do DevOps Guru para RDS:

Diagnóstico rápido

O DevOps Guru para RDS monitora e analisa continuamente a telemetria do banco de dados. O Performance Insights, o Enhanced Monitoring e o Amazon CloudWatch coletam dados de telemetria para sua instância de banco de dados. O DevOps Guru para RDS usa técnicas estatísticas e de machine learning para explorar esses dados e detectar anomalias. Para saber mais sobre dados de telemetria, consulte Monitoramento da carga do banco de dados com Performance Insights no Amazon RDS e Monitoramento de métricas do sistema operacional com monitoramento aprimorado no Guia do usuário do Amazon RDS.

Resolução rápida

Cada anomalia identifica o problema de performance e sugere rotas de investigação ou ações corretivas. Por exemplo, o DevOps Guru para RDS pode recomendar que você investigue eventos de espera específicos. Ou ele pode recomendar que você ajuste as configurações do seu grupo de aplicações para limitar o número de conexões de banco de dados. Com base nessas recomendações, é possível resolver problemas de performance mais rapidamente do que solucionando problemas manualmente.

Insights proativos

O DevOps Guru para RDS utiliza métricas de seus recursos para detectar um comportamento possivelmente problemático antes que ele se torne um problema maior. Por exemplo, ele pode detectar quando seu banco de dados está utilizando um número crescente de tabelas temporárias em disco, o que pode começar a afetar a performance. Depois, o DevOps Guru fornece recomendações para ajudar você a resolver problemas antes que eles se tornem problemas maiores.

Conhecimento profundo dos engenheiros da Amazon e de "machine learning"

Para detectar problemas de performance e ajudar você a resolver gargalos, o DevOps Guru para RDS conta com machine learning (ML) e fórmulas matemáticas avançadas. Os engenheiros de banco de dados da Amazon contribuíram para o desenvolvimento das descobertas do DevOps Guru para RDS, que englobam muitos anos gerenciando centenas de milhares de bancos de dados. Com base nesse conhecimento coletivo, o DevOps Guru para RDS pode ensinar práticas recomendadas para você.

Como funciona o DevOps Guru para RDS

O DevOps Guru para RDS coleta dados sobre seus bancos de dados RDS para PostgreSQL por meio do Amazon RDS Performance Insights. A métrica mais importante é `DBLoad`. O DevOps Guru for RDS consome as métricas do Performance Insights, analisa-as com machine learning e publica insights no painel.

Um insight é uma coleção de anomalias relacionadas que foram detectadas pelo DevOps Guru.

No DevOps Guru para RDS, uma anomalia é um padrão que se desvia do que é considerada a performance normal do seu banco de dados RDS para PostgreSQL.

Insights proativos

Um insight proativo informa você sobre um comportamento problemático antes que ele ocorra. Contém anomalias com recomendações e métricas relacionadas para ajudar você a resolver problemas em seus bancos de dados RDS para PostgreSQL antes que se tornem problemas maiores. Esses insights são publicados no painel do DevOps Guru.

Por exemplo, o DevOps Guru pode detectar que seu banco de dados do RDS para PostgreSQL está criando muitas tabelas temporárias em disco. Se não for tratada, essa tendência poderá gerar problemas de performance. Cada insight proativo inclui recomendações para comportamento corretivo e links para tópicos relevantes em Ajustar o RDS para PostgreSQL com insights proativos do Amazon DevOps Guru. Para ter mais informações, consulte Trabalhar com insights no DevOps Guru no Guia do usuário do Amazon DevOps Guru.

Insights reativos

Um insight reativo identifica um comportamento anômalo quando ele ocorre. Se o DevOps Guru para RDS encontrar problemas de performance nas suas instâncias de banco de dados do RDS para PostgreSQL, ele publicará um insight reativo no painel do DevOps Guru. Para ter mais informações, consulte Trabalhar com insights no DevOps Guru no Guia do usuário do Amazon DevOps Guru.

Anomalias causais

Uma anomalia causal é uma anomalia de nível superior dentro de um insight reativo. Carga do banco de dados é a anomalia causal do DevOps Guru para RDS.

Uma anomalia mede o impacto na performance, atribuindo um nível de gravidade de Alto, Médio ou Baixo. Para saber mais, consulte os Principais conceitos do DevOps Guru para RDS, no Guia do usuário do Amazon DevOps Guru.

Se o DevOps Guru detectar uma anomalia em sua instância de banco de dados, você será alertado na página Databases (Bancos de dados) do console do RDS. O console também alerta você sobre anomalias que ocorreram nas últimas 24 horas. Para acessar a página de anomalias no console do RDS, escolha o link na mensagem de alerta. O console do RDS também alerta você na página da instância de banco de dados do RDS para PostgreSQL.

Anomalias contextuais

Uma anomalia contextual é uma descoberta em Carga do banco de dados (carga do BD) que é relatada a um insight reativo. Cada anomalia contextual descreve um problema de performance específico do RDS para PostgreSQL que requer investigação. Por exemplo, o DevOps Guru para RDS pode recomendar que você aumente a capacidade da CPU ou investigue eventos de espera que estão contribuindo para a carga do banco de dados.

> **Important**
>
> Convém testar todas as alterações na instância de teste antes de modificar a instância de produção. Dessa forma, você pode compreender o impacto da alteração.

Para saber mais, consulte Analisar anomalias no Amazon RDS no Guia do usuário do Amazon DevOps Guru.

Configurar o DevOps Guru para RDS

Para permitir que o DevOps Guru para Amazon RDS publique insights de um banco de dados do RDS para PostgreSQL, conclua as tarefas a seguir.

Tópicos

- Configurar políticas de acesso do IAM para DevOps Guru para RDS
- Ativar o Performance Insights para suas instâncias de banco de dados do RDS para PostgreSQL
- Ativar o DevOps Guru e especificar a cobertura de recursos

Configurar políticas de acesso do IAM para DevOps Guru para RDS

Para visualizar alertas do DevOps Guru no console do RDS, seu usuário ou perfil do AWS Identity and Access Management (IAM) deve ter uma das seguintes políticas:

- A política `AmazonDevOpsGuruConsoleFullAccess` gerenciada pelo AWS
- A política gerenciada `AmazonDevOpsGuruConsoleReadOnlyAccess` da AWS e uma das seguintes políticas:
  - A política `AmazonRDSFullAccess` gerenciada pelo AWS
  - Uma política gerenciada pelo cliente que inclua `pi:GetResourceMetrics` e `pi:DescribeDimensionKeys`

Para ter mais informações, consulte Configurar políticas de acesso para o Performance Insights.

Ativar o Performance Insights para suas instâncias de banco de dados do RDS para PostgreSQL

O DevOps Guru para RDS depende do Performance Insights para seus dados. Sem o Performance Insights, o DevOps Guru publica anomalias, mas não inclui a análise e as recomendações detalhadas.

Ao criar ou modificar uma instância de banco de dados do RDS para PostgreSQL, você pode ativar o Performance Insights. Para ter mais informações, consulte Ativar e desativar o Insights de Performance do Amazon RDS.

Ativar o DevOps Guru e especificar a cobertura de recursos

Você pode ativar o DevOps Guru para que ele monitore seus bancos de dados do RDS para PostgreSQL de uma das maneiras a seguir.

Tópicos

- Ativar o DevOps Guru no console do RDS
- Adicionar recursos do RDS para PostgreSQL no console do DevOps Guru
- Adicionar recursos do RDS para PostgreSQL usando AWS CloudFormation

Ativar o DevOps Guru no console do RDS

Você pode seguir vários caminhos no console do Amazon RDS para ativar o DevOps Guru.

Tópicos

- Ativar o DevOps Guru ao criar um banco de dados do RDS para PostgreSQL

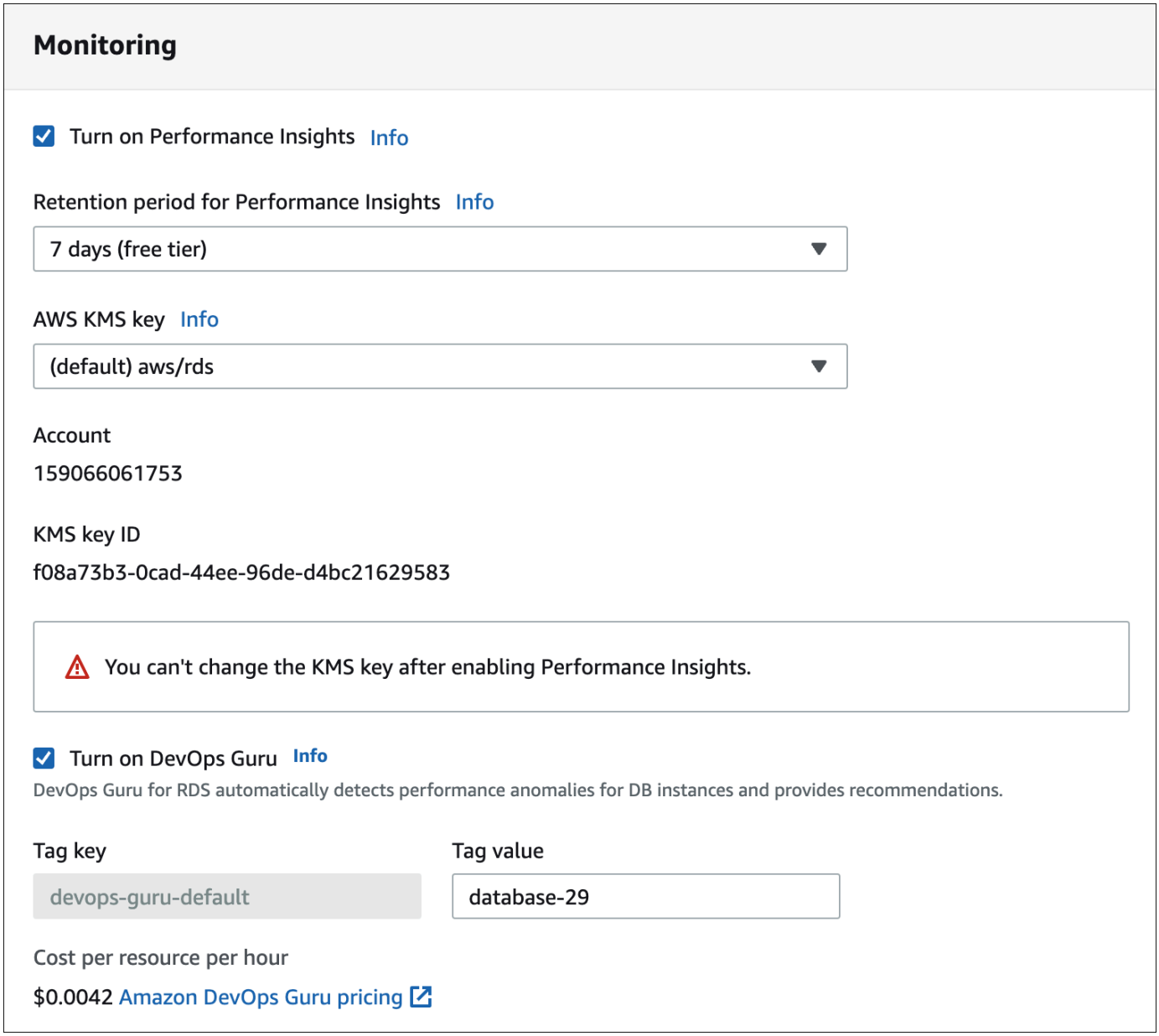
- Ativar o DevOps Guru a partir do banner de notificação
- Responder a um erro de permissões quando você ativa o DevOps Guru

Ativar o DevOps Guru ao criar um banco de dados do RDS para PostgreSQL

O fluxo de trabalho de criação inclui uma configuração que ativa a cobertura do DevOps Guru para seu banco de dados. Essa configuração é ativada por padrão quando você escolhe o modelo Production (Produção).

Como ativar o DevOps Guru ao criar um banco de dados do RDS para PostgreSQL

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. Siga as etapas em Criar uma instância de banco de dados exceto a etapa em que você escolhe as configurações de monitoramento.
3. Em Monitoring (Monitoramento), escolha Turn on Performance Insights (Ativar Performance Insights). Para que o DevOps Guru para RDS forneça uma análise detalhada de anomalias de performance, o Performance Insights deve estar ativado.
4. Escolha Turn on DevOps Guru (Ativar DevOps Guru).



5. Crie uma tag para seu banco de dados para que o DevOps Guru possa monitorá-lo. Faça o seguinte:

   - No campo de texto para Tag key (Chave de tag), insira um nome que comece com **`Devops-Guru-`**.
   - No campo de texto para Tag value (Valor da tag), insira qualquer valor. Por exemplo, se você inserir **`rds-database-1`** para obter o nome do banco de dados do RDS para PostgreSQL, também é possível inserir **`rds-database-1`** como o valor da tag.

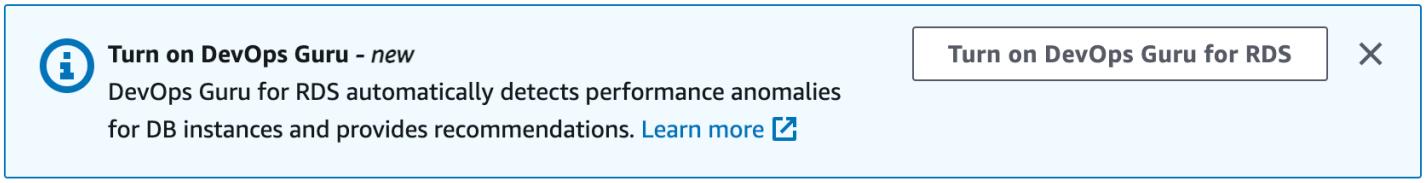
Para ter mais informações sobre tags, consulte "Usar tags para identificar recursos em suas aplicações do DevOps Guru" no Guia do usuário do Amazon DevOps Guru.

6. Conclua as etapas restantes em Criar uma instância de banco de dados.

Ativar o DevOps Guru a partir do banner de notificação

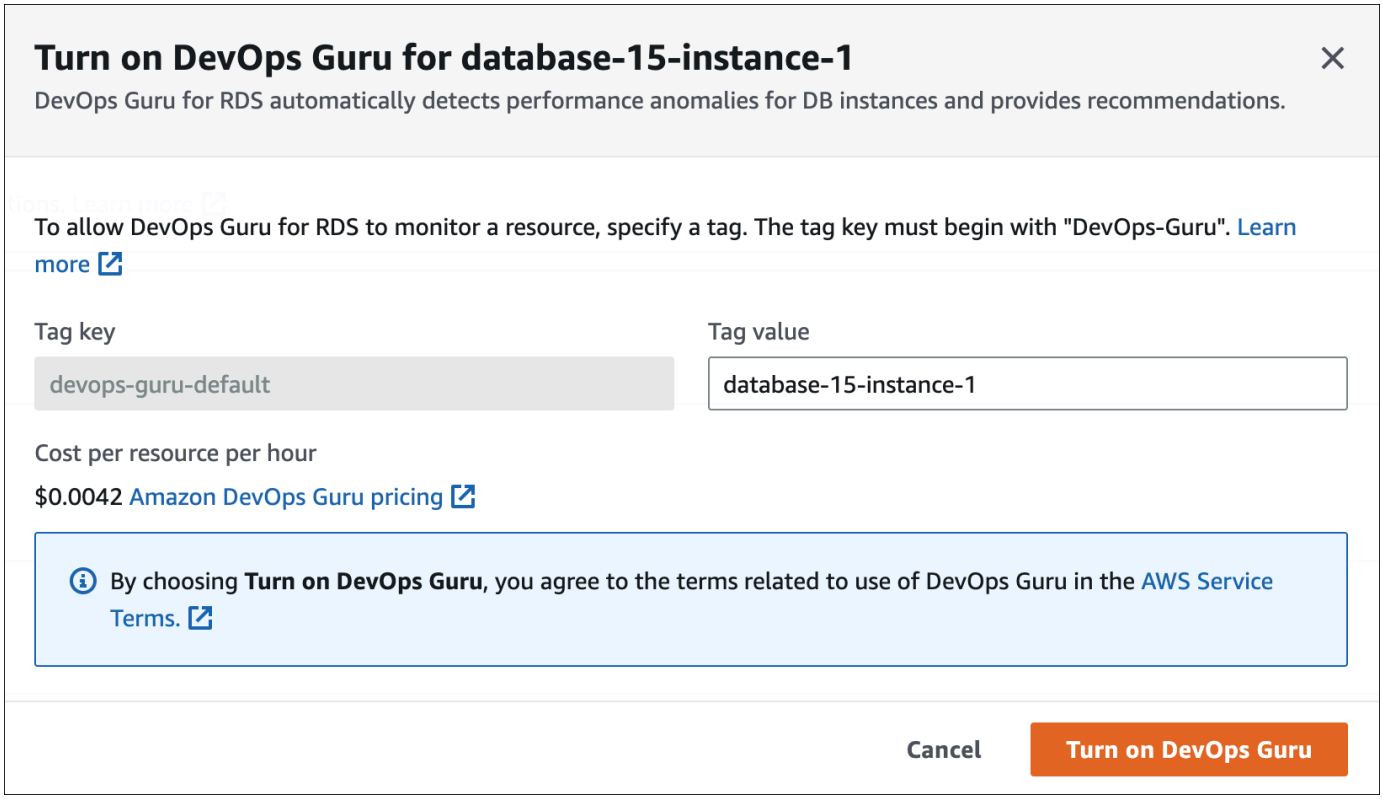
Se seus recursos não forem cobertos pelo DevOps Guru, o Amazon RDS notificará você com um banner nos seguintes locais:

- A guia Monitoring (Monitoramento) de uma instância de cluster de banco de dados
- O painel do Performance Insights



Como ativar o DevOps Guru para seu banco de dados do RDS para PostgreSQL

1. No banner, escolha Turn on DevOps Guru for RDS (Ativar DevOps Guru para RDS).
2. Insira um nome de chave e um valor para a tag. Para ter mais informações sobre tags, consulte "Usar tags para identificar recursos em suas aplicações do DevOps Guru" no Guia do usuário do Amazon DevOps Guru.



3. Escolha Turn on DevOps Guru (Ativar DevOps Guru).

Responder a um erro de permissões quando você ativa o DevOps Guru

Se você ativar o DevOps Guru no console do RDS ao criar um banco de dados, o RDS poderá exibir o banner a seguir sobre permissões ausentes.

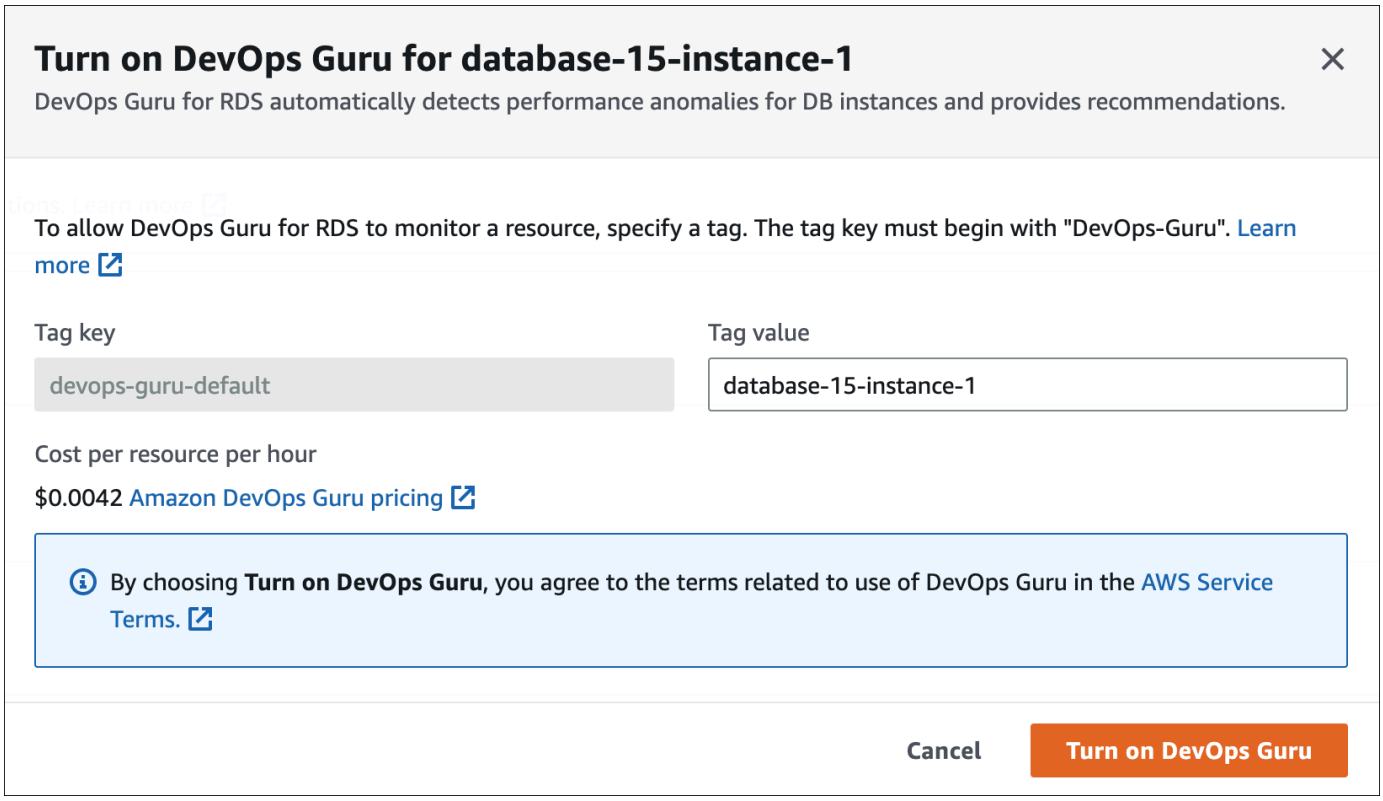


Como responder a um erro de permissões

1. Conceda ao usuário ou ao perfil do IAM a função gerenciada pelo usuário `AmazonDevOpsGuruConsoleFullAccess`. Para ter mais informações, consulte Configurar políticas de acesso do IAM para DevOps Guru para RDS.
2. Abra o console do RDS.
3. No painel de navegação, escolha Performance Insights.
4. Escola uma instância de banco de dados no cluster que você acabou de criar.
5. Selecione o switch para ativar o DevOps Guru para RDS.

DevOps Guru for RDS

6. Escolha um valor de tag. Para ter mais informações, consulte "Usar tags para identificar recursos em suas aplicações do DevOps Guru" no Guia do usuário do Amazon DevOps Guru.



7. Escolha Turn on DevOps Guru (Ativar DevOps Guru).

Adicionar recursos do RDS para PostgreSQL no console do DevOps Guru

Você pode especificar a cobertura de recursos do DevOps Guru no console do DevOps Guru. Siga a etapa descrita em Especificar a cobertura de recursos do DevOps Guru no Guia do usuário do Amazon DevOps Guru. Ao editar os recursos analisados, escolha uma das seguintes opções:

- Selecione Todos os recursos da conta para analisar todos os recursos compatíveis, inclusive os bancos de dados do RDS para PostgreSQL, em sua Conta da AWS e região.
- Selecione Pilhas do CloudFormation para analisar os bancos de dados do RDS para PostgreSQL que estão nas pilhas escolhidas. Para ter mais informações, consulte Usar pilhas do AWS CloudFormation para identificar recursos em suas aplicações do DevOps Guru no Guia do usuário do Amazon DevOps Guru.

- Selecione Tags para analisar os bancos de dados do RDS para PostgreSQL marcados. Para ter mais informações, consulte Usar tags para identificar recursos em suas aplicações do DevOps Guru no Guia do usuário do Amazon DevOps Guru.

Para ter mais informações, consulte Enable DevOps Guru (Ativar o DevOps Guru), no Guia do usuário do Amazon DevOps Guru.

Adicionar recursos do RDS para PostgreSQL usando AWS CloudFormation

Você pode usar tags para adicionar cobertura dos recursos do RDS para PostgreSQL aos modelos do CloudFormation. O procedimento a seguir pressupõe que você tenha um modelo do CloudFormation para sua instância de banco de dados do RDS para PostgreSQL e para a pilha do DevOps Guru.

Como especificar uma instância de banco de dados do RDS para PostgreSQL usando uma tag do CloudFormation

1. No modelo do CloudFormation para sua instância de banco de dados, defina uma tag usando um par de chave/valor.

   O exemplo a seguir atribui o valor `my-db-instance1` a `Devops-guru-cfn-default` para uma instância de banco de dados do RDS para PostgreSQL.

   ```
   MyDBInstance1:
     Type: "AWS::RDS::DBInstance"
     Properties:
       DBInstanceIdentifier: my-db-instance1
       Tags:
         - Key: Devops-guru-cfn-default
           Value: devopsguru-my-db-instance1
   ```

2. No modelo do CloudFormation para sua pilha do DevOps Guru, especifique a mesma tag em seu filtro de coleta de recursos.

   O exemplo a seguir configura o DevOps Guru para fornecer cobertura para o recurso com o valor da tag `my-db-instance1`.

   ```
   DevOpsGuruResourceCollection:
     Type: AWS::DevOpsGuru::ResourceCollection
     Properties:
       ResourceCollectionFilter:
   ```

```
      Tags:
        - AppBoundaryKey: "Devops-guru-cfn-default"
          TagValues:
          - "devopsguru-my-db-instance1"
```

O exemplo a seguir fornece cobertura para todos os recursos dentro dos limites da aplicação `Devops-guru-cfn-default`.

```
DevOpsGuruResourceCollection:
  Type: AWS::DevOpsGuru::ResourceCollection
  Properties:
    ResourceCollectionFilter:
      Tags:
        - AppBoundaryKey: "Devops-guru-cfn-default"
          TagValues:
          - "*"
```

Para ter mais informações, consulte AWS::DevOpsGuru::ResourceCollection e AWS::RDS::DBInstance no Guia do usuário do AWS CloudFormation.

Monitorar métricas do SO com o monitoramento avançado

Com o monitoramento avançado, você pode monitorar o sistema operacional da sua instância de banco de dados em tempo real. As métricas de monitoramento avançado são úteis quando você quiser ver como os diferentes processos ou threads usam a CPU.

Tópicos

- Visão geral do monitoramento avançado
- Configurar e habilitar o monitoramento avançado
- Como visualizar métricas do SO no console do RDS
- Visualizar métricas do SO usando CloudWatch Logs

Visão geral do monitoramento avançado

O Amazon RDS dispõe de métricas em tempo real para o sistema operacional (SO) no qual sua instância de banco de dados é executada. Você pode visualizar todas as métricas e informações de processo do sistema das suas instâncias de banco de dados do RDS no console. Você pode gerenciar quais métricas deseja monitorar para cada instância e personalizar o painel de acordo com os seus requisitos. Para ver as descrições das métricas do monitoramento avançado, consulte Métricas do sistema operacional em monitoramento avançado.

O RDS fornece as métricas do monitoramento avançado à sua conta do Amazon CloudWatch Logs. Você pode criar filtros de métricas no CloudWatch com o CloudWatch Logs e exibir os gráficos no painel do CloudWatch. É possível consumir o resultado do JSON de monitoramento avançado do CloudWatch Logs em um sistema de monitoramento de sua escolha. Para obter mais informações, consulte Monitoramento avançado nas Perguntas frequentes do Amazon RDS.

Tópicos

- Disponibilidade do monitoramento avançado
- Diferenças entre o CloudWatch e métricas de monitoramento avançado
- Retenção de métricas do monitoramento avançado
- Custo do monitoramento avançado

Disponibilidade do monitoramento avançado

O monitoramento avançado está disponível para os seguintes mecanismos de banco de dados:

- Db2
- MariaDB
- Microsoft SQL Server
- MySQL
- Oracle
- PostgreSQL

O monitoramento avançado está disponível para todas as classes de instâncias de banco de dados, exceto para a classe de instância db.m1.small.

Diferenças entre o CloudWatch e métricas de monitoramento avançado

Um hipervisor cria e executa as máquinas virtuais (VMs). Usando um hipervisor, a instância pode oferecer suporte a várias VMs convidadas compartilhando memória e CPU virtualmente. O CloudWatch coleta métricas sobre a utilização da CPU do hipervisor para uma instância de banco de dados. Em contraste, o monitoramento avançado reúne as métricas de um agente na instância de banco de dados.

Você poderá encontrar diferenças entre as CloudWatch medidas do monitoramento avançado, pois a camada do hipervisor realiza uma pequena quantidade de trabalho. As diferenças podem ser maiores se as instâncias de banco de dados usarem classes de instância menores. Nesse cenário, mais máquinas virtuais (VMs) provavelmente são gerenciadas pela camada de hipervisor em uma única instância física.

Para ver as descrições das métricas do monitoramento avançado, consulte Métricas do sistema operacional em monitoramento avançado. Para obter mais informações sobre as métricas do CloudWatch, consulte o Guia do usuário do Amazon CloudWatch.

Retenção de métricas do monitoramento avançado

Por padrão, as métricas do monitoramento avançado são armazenadas por 30 dias no CloudWatch Logs. Esse período de retenção é diferente das métricas típicas do CloudWatch.

Para modificar o tempo em que as métricas são armazenadas nos logs do CloudWatch Logs, altere a retenção do grupo de logs do `RDSOSMetrics` no console do CloudWatch. Para obter mais informações, consulte Alterar retenção de dados de log em logs do CloudWatch no Amazon CloudWatch Logs User Guide.

Custo do monitoramento avançado

As métricas de monitoramento avançado são armazenadas no CloudWatch Logs e não métricas do CloudWatch. O custo do monitoramento avançado depende dos seguintes fatores:

- Só haverá cobrança pelo monitoramento avançado se você exceder o nível gratuito fornecido pelo Amazon CloudWatch Logs. As cobranças são baseadas nas taxas de transferência de dados e armazenamento do CloudWatch Logs.
- A quantidade de informações transferidas para uma instância do RDS é diretamente proporcional ao detalhamento definido para o recurso de monitoramento avançado. Um menor intervalo de monitoramento resulta em relatórios mais frequentes das métricas do sistema operacional e aumenta seu custo de monitoramento. Para gerenciar custos, defina diferentes detalhamentos para diferentes instâncias em suas contas.
- Os custos de uso do Monitoramento avançado são aplicados a cada instância de banco de dados para a qual o Monitoramento avançado esteja habilitado. O monitoramento de um grande número de instâncias de banco de dados é mais caro do que o monitoramento de apenas algumas.
- Instâncias de bancos de dados que oferecem suporte a uma workload que exige computação intensiva têm mais atividade de processos de SO para reportar e custos mais altos para Monitoramento avançado.

Para obter mais informações sobre a definição de preço, consulte Definição de preço do Amazon CloudWatch.

Configurar e habilitar o monitoramento avançado

Para usar o monitoramento avançado, você deve criar uma função do IAM e habilitar o monitoramento avançado.

Tópicos

- Criar uma função do IAM para o monitoramento avançado
- Ativar e desativar o monitoramento aprimorado
- Proteção contra o problema do substituto confuso

Criar uma função do IAM para o monitoramento avançado

O Monitoramento avançado requer permissão para agir em seu nome para enviar informações de métricas do SO ao CloudWatch Logs. Você concede as permissões ao monitoramento avançado usando uma função do AWS Identity and Access Management (IAM). Você pode criar essa função ao ativar o monitoramento aprimorado ou criá-la de antemão.

Tópicos

- Criar a função do IAM ao habilitar o Monitoramento Avançado
- Criar a função do IAM antes de ativar o Monitoramento Avançado

Criar a função do IAM ao habilitar o Monitoramento Avançado

Quando você habilita o Monitoramento Avançado no console do RDS, o Amazon RDS pode criar a função do IAM necessária para você. A função é chamada `rds-monitoring-role`. O RDS usa essa função para a instância de banco de dados especificada, réplica de leitura ou cluster de banco de dados multi-AZ.

Para criar a função do IAM ao habilitar o Monitoramento Avançado

1. Siga as etapas em Ativar e desativar o monitoramento aprimorado.
2. Defina a Monitoring Role (Função de monitoramento) como Default (Padrão) na etapa em que você escolhe uma função.

Criar a função do IAM antes de ativar o Monitoramento Avançado

Você pode criar a função necessária antes de habilitar o Monitoramento Avançado. Ao habilitar o Monitoramento Avançado, especifique o nome da nova função. Você deverá criar essa função necessária se habilitar o Monitoramento avançado usando a AWS CLI ou a API do RDS.

O usuário que habilita o monitoramento aprimorado precisa receber a permissão `PassRole`. Para obter mais informações, consulte o Exemplo 2 em Conceder permissões ao usuário para transmitir uma função para um serviço da AWS no Guia do usuário do IAM.

Como criar uma função do IAM para o monitoramento avançado do Amazon RDS

1. Abra o Console do IAM em https://console.aws.amazon.com.
2. No painel de navegação, selecione Perfis.

3. Escolha Criar Perfil.
4. Escolha a guia Serviço da AWS e, em seguida, RDS na lista de serviços.
5. Escolha RDS - Enhanced Monitoring (RDS: monitoramento aprimorado) e Next (Próximo).
6. A página Permissions policies (Políticas de permissões) deve mostrar AmazonRDSEnhancedMonitoringRole. Escolha Next (Próximo).
7. Em Role name (Nome da função), digite um nome para sua função. Por exemplo, digite **`emaccess`**.

   A entidade confiável para sua função é o serviço da AWS monitoring.rds.amazonaws.com.
8. Selecione Create role (Criar função).

Ativar e desativar o monitoramento aprimorado

É possível gerenciar o monitoramento avançado usando o AWS Management Console, a AWS CLI ou a API do RDS. Você pode definir detalhamentos diferentes para a coleta de métricas em cada instância de banco de dados.

Console

É possível ativar o monitoramento avançado ao criar uma instância de banco de dados, um cluster de banco de dados multi-AZ ou uma réplica de leitura, ou ao modificar uma instância de banco de dados ou cluster de banco de dados multi-AZ. Se você modificar uma instância de banco de dados para ativar o Monitoramento avançado, não será necessário reinicializar a instância de banco de dados para que a alteração entre em vigor.

Você pode ativar o monitoramento aprimorado no console do RDS ao realizar uma das seguintes ações na página Databases (Banco de dados):

- “Create a DB instance or Multi-AZ DB cluster” (Criar uma instância de banco de dados ou cluster de banco de dados multi-AZ): escolha Create database (Criar banco de dados).
- Create a read replica (Criar uma réplica de leitura) — Escolha Actions (Ações) e depois Create read replica (Criar réplica de leitura).
- Modificar uma instância de banco de dados ou cluster de banco de dados multi-AZ: escolha Modificar.

Para ativar ou desativar o monitoramento aprimorado no console do RDS

1. Role até Additional configuration (Configuração adicional).
2. Em Monitoramento, escolha Habilitar o Monitoramento avançado para a instância de banco de dados ou a réplica de leitura. Desmarque a opção para desabilitar o Monitoramento avançado.
3. Defina a propriedade Monitoring Role (Função de monitoramento) como a função do IAM que você criou para permitir que o Amazon RDS se comunique com o Amazon CloudWatch Logs por você. Ou escolha Default (Padrão) para que o RDS crie uma função para você chamada `rds-monitoring-role`.
4. Defina a propriedade Granularidade como o intervalo, em segundos, entre pontos quando as métricas são coletadas para a instância de banco de dados ou a réplica de leitura. A propriedade Granularity (Granularidade) pode ser definida como um dos seguintes valores: `1`, `5`, `10`, `15`, `30` ou `60`.

   A frequência de atualização mais rápida do console do RDS é a cada 5 segundos. Se você definir a granularidade como 1 segundo no console do RDS, ainda verá as métricas atualizadas apenas a cada 5 segundos. É possível recuperar atualizações de métricas de 1 segundo usando a CloudWatch Logs.

AWS CLI

Para ativar o monitoramento avançado usando a AWS CLI, defina a opção `--monitoring-interval` nos comandos a seguir como um valor diferente de `0` e defina a opção `--monitoring-role-arn` para a função criada em Criar uma função do IAM para o monitoramento avançado.

- create-db-instance
- create-db-instance-read-replica
- modify-db-instance
- create-db-cluster (cluster de banco de dados multi-AZ)
- modify-db-cluster (cluster de banco de dados multi-AZ)

A opção `--monitoring-interval` especifica o intervalo, em segundos, entre pontos quando as métricas de monitoramento avançado são coletadas. Os valores válidos para a opção são `0`, `1`, `5`, `10`, `15`, `30` e `60`.

Para desativar o monitoramento aprimorado usando a AWS CLI, defina a opção `--monitoring-interval` para `0` nestes comandos.

Example

O seguinte exemplo ativa o monitoramento avançado para uma instância de banco de dados:

Para Linux, macOS ou Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --monitoring-interval 30 \
    --monitoring-role-arn arn:aws:iam::123456789012:role/emaccess
```

Para Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --monitoring-interval 30 ^
    --monitoring-role-arn arn:aws:iam::123456789012:role/emaccess
```

Example

O seguinte exemplo ativa o monitoramento avançado para um cluster de banco de dados multi-AZ:

Para Linux, macOS ou Unix:

```
aws rds modify-db-cluster \
    --db-cluster-identifier mydbcluster \
    --monitoring-interval 30 \
    --monitoring-role-arn arn:aws:iam::123456789012:role/emaccess
```

Para Windows:

```
aws rds modify-db-cluster ^
    --db-cluster-identifier mydbcluster ^
    --monitoring-interval 30 ^
    --monitoring-role-arn arn:aws:iam::123456789012:role/emaccess
```

API do RDS

Para ativar o monitoramento aprimorado usando a API do RDS, defina o parâmetro `MonitoringInterval` como um valor diferente de `0` e defina o parâmetro `MonitoringRoleArn` para a função criada em Criar uma função do IAM para o monitoramento avançado. Defina esses parâmetros nas seguintes ações:

- CreateDBInstance
- CreateDBInstanceReadReplica
- ModifyDBInstance
- CreateDBCluster (cluster de banco de dados multi-AZ)
- ModifyDBCluster (cluster de banco de dados multi-AZ)

O parâmetro `MonitoringInterval` especifica o intervalo, em segundos, entre pontos quando as métricas de monitoramento avançado são coletadas. Os valores válidos são `0`, `1`, `5`, `10`, `15`, `30` e `60`.

Para desativar o monitoramento aprimorado usando a API do RDS, defina `MonitoringInterval` como `0`.

Proteção contra o problema do substituto confuso

“Confused deputy” é um problema de segurança no qual uma entidade sem permissão para executar uma ação pode coagir uma entidade mais privilegiada a executá-la. Na AWS, a personificação entre serviços pode resultar no problema do ‘confused deputy’. A personificação entre serviços pode ocorrer quando um serviço (o serviço de chamada) chama outro serviço (o serviço chamado). O serviço de chamada pode ser manipulado de modo a usar suas permissões para atuar nos recursos de outro cliente de uma forma na qual ele não deveria ter permissão para acessar. Para evitar isso, a AWS fornece ferramentas que ajudam você a proteger seus dados para todos os serviços com entidades principais de serviço que receberam acesso aos recursos em sua conta. Para obter mais informações, consulte O problema do substituto confuso.

Para limitar as permissões ao recurso que o Amazon RDS pode fornecer a outro serviço, recomendamos usar as chaves de contexto de condição global `aws:SourceArn` e `aws:SourceAccount` em uma política de confiança para sua função de monitoramento aprimorado. Se você usar as duas chaves de contexto de condição global, elas deverão usar o mesmo ID de conta.

A maneira mais eficaz de se proteger contra o problema do substituto confuso é usar a chave de contexto de condição global `aws:SourceArn` com o ARN completo do recurso. Para o Amazon RDS, defina `aws:SourceArn` como `arn:aws:rds:`*`Region`*`:`*`my-account-id`*`:db:`*`dbname`*.

O exemplo a seguir usa as chaves de contexto de condição global `aws:SourceArn` e `aws:SourceAccount` em uma política de confiança para evitar o problema de substituto confuso.

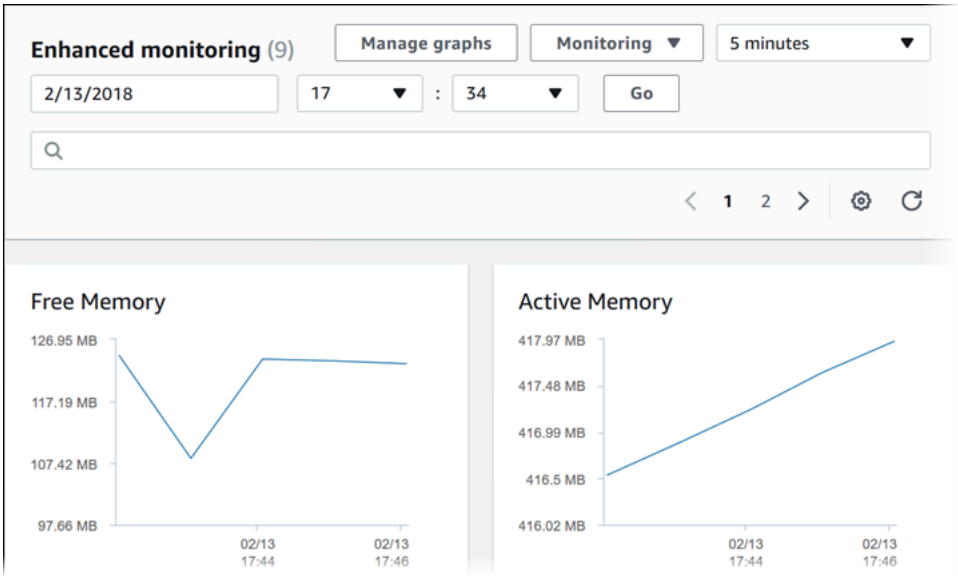
JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "monitoring.rds.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringLike": {
          "aws:SourceArn": "arn:aws:rds:Region:my-account-id:db:dbname"
        },
        "StringEquals": {
          "aws:SourceAccount": "my-account-id"
        }
      }
    }
  ]
}
```

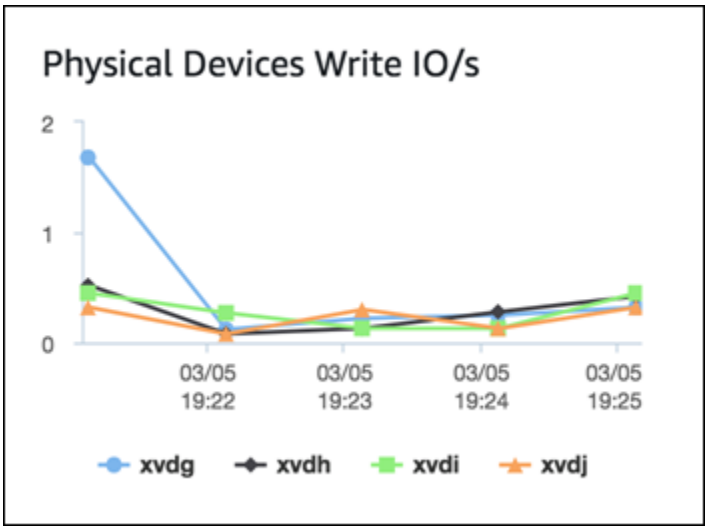
Como visualizar métricas do SO no console do RDS

Você pode visualizar as métricas do sistema operacional informadas pelo Monitoramento avançado no console do RDS escolhendo a exibição Enhanced monitoring (Monitoramento avançado) para Monitoring (Monitoramento).

O exemplo a seguir mostra a página Monitoramento avançado. Para ver as descrições das métricas do monitoramento avançado, consulte Métricas do sistema operacional em monitoramento avançado.



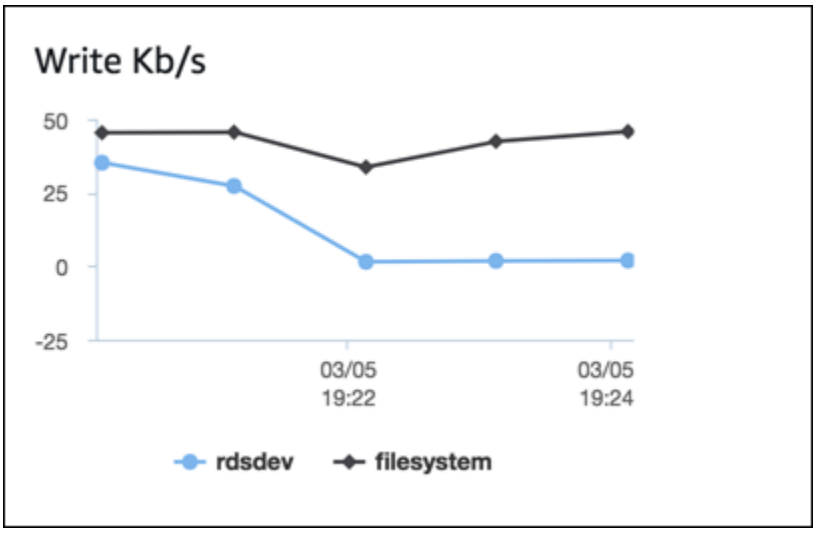
Algumas instâncias de banco de dados usam mais de um disco para o volume de armazenamento de dados da instância de banco de dados. Nessas instâncias de banco de dados, os gráficos Physical Devices (Dispositivos físicos) mostram métricas para cada um dos discos. Por exemplo, o gráfico a seguir mostra métricas para quatro discos.



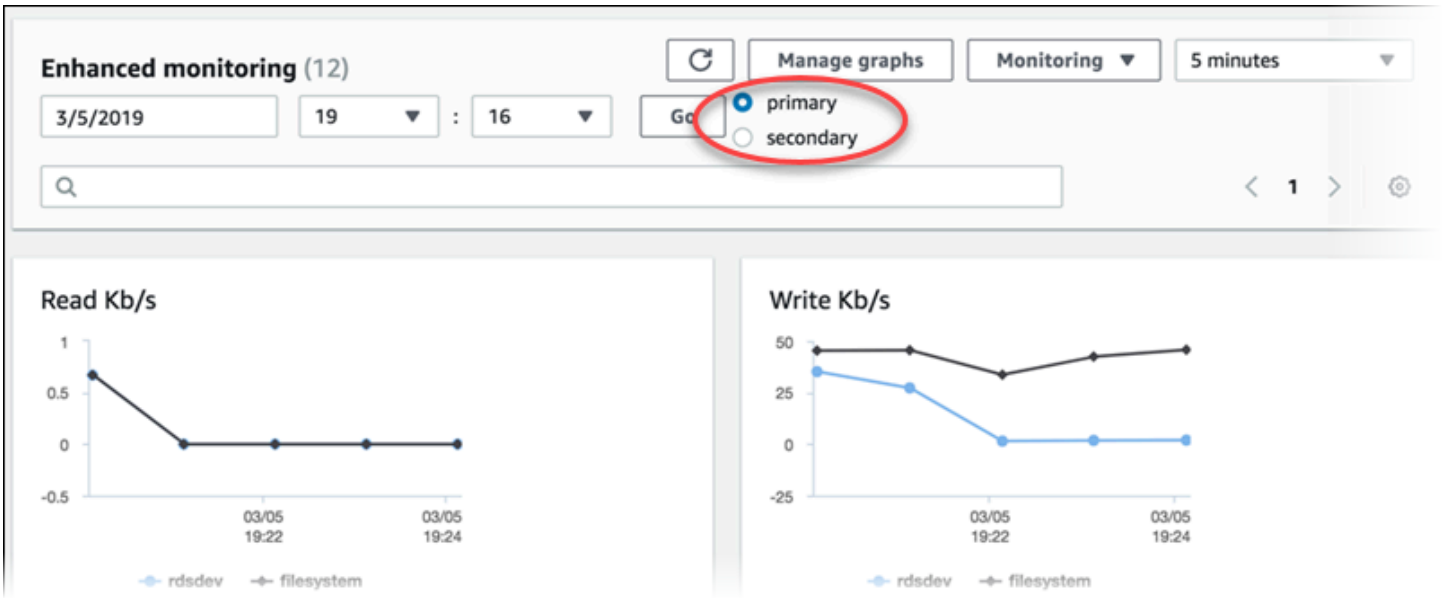
> **Note**
>
> Atualmente, gráficos Physical Devices (Dispositivos físicos) não estão disponíveis para instâncias de banco de dados do Microsoft SQL Server.

Nos gráficos Disk I/O (E/S de disco) e File system (Sistema de arquivos), o dispositivo rdsdev está relacionado ao sistema de arquivos `/rdsdbdata`, em que os arquivos e logs de bancos de dados

são armazenados. O dispositivo filesystem está relacionado ao sistema de arquivos / (também conhecido como root), no qual os arquivos relacionados ao sistema operacional são armazenados.



Se a instância de banco de dados for uma implantação Multi-AZ, você poderá visualizar as métricas do sistema operacional para a instância de banco de dados primária e sua réplica em espera Multi-AZ. Na visualização Enhanced monitoring (Monitoramento avançado), escolha primary (primária) para ver as métricas do sistema operacional para a instância de banco de dados primária ou escolha secondary (secundária) para ver as métricas do sistema operacional para a réplica de espera.

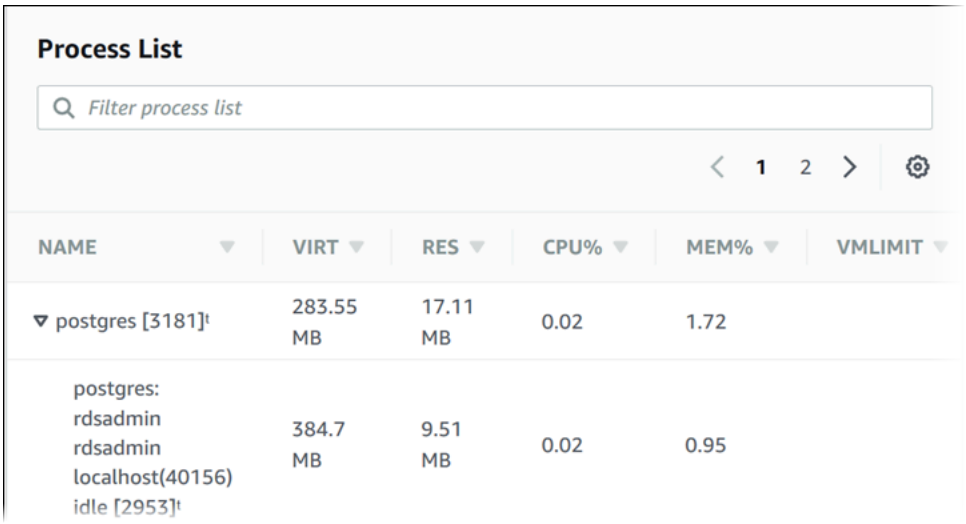


Para obter mais informações sobre implantações Multi-AZ, consulte Configurar e gerenciar uma implantação multi-AZ para o Amazon RDS.

> **Note**
>
> No momento, a visualização de métricas do sistema operacional para uma réplica em espera multi-AZ não é compatível com instâncias de banco de dados do MariaDB.

Se quiser ver detalhes dos processos em execução na sua instância de banco de dados, escolha OS process list (Lista de processos do SO) para Monitoring (Monitoramento).

A visualização Process List (Lista de processos) é mostrada a seguir.



As métricas de Monitoramento avançado mostradas na visualização Process list (Lista de processos) estão organizadas da seguinte forma:

- RDS child processes (Processos filho do RDS) – mostra um resumo dos processos do RDS que oferecem suporte à instância de banco de dados, por exemplo, `mysqld` para instâncias de banco de dados MySQL. Os segmentos de processos aparecem aninhados abaixo do processo pai. Os threads de processos mostram a utilização da CPU apenas enquanto as outras métricas forem as mesmas para todos os threads do processo. O console exibe um máximo de 100 processos e threads. Os resultados são uma combinação dos principais processos e threads que consomem CPU e memória. Se houver mais de 50 processos e mais de 50 threads, o console exibirá os 50 melhores consumidores em cada categoria. Essa exibição ajuda a identificar quais processos estão tendo o maior impacto sobre a performance.
- RDS processes (Processos do RDS): mostra um resumo dos recursos utilizados pelo agente de gerenciamento do RDS, processos de monitoramento de diagnóstico e outros processos da AWS necessários para dar suporte a instâncias de bancos de dados do RDS.

- OS processes (Processos do SO) – Mostra um resumo dos processos de kernel e do sistema, que geralmente têm um impacto mínimo sobre a performance.

Os itens listados para cada processo são:

- VIRT – Exibe o tamanho virtual do processo.
- RES – Exibe a memória física real que está sendo usada pelo processo.
- CPU% – exibe a porcentagem da largura de banda total da CPU que está sendo usada pelo processo.
- MEM% – exibe a porcentagem da memória total que está sendo consumida pelo processo.

Os dados de monitoramento que são mostrados no console do RDS são recuperados do Amazon CloudWatch Logs. Você também pode recuperar as métricas para uma instância de banco de dados como um stream de log do CloudWatch Logs. Para obter mais informações, consulte Visualizar métricas do SO usando CloudWatch Logs.

Métricas de Monitoramento avançado não são retornadas durante o seguinte:

- Um failover da instância de banco de dados.
- Alteração da classe da instância de banco de dados (computação de escala).

As métricas de Monitoramento avançado são retornadas durante uma reinicialização de uma instância de banco de dados, pois somente o mecanismo do banco de dados é reinicializado. Métricas para o sistema operacional ainda serão informadas.

Visualizar métricas do SO usando CloudWatch Logs

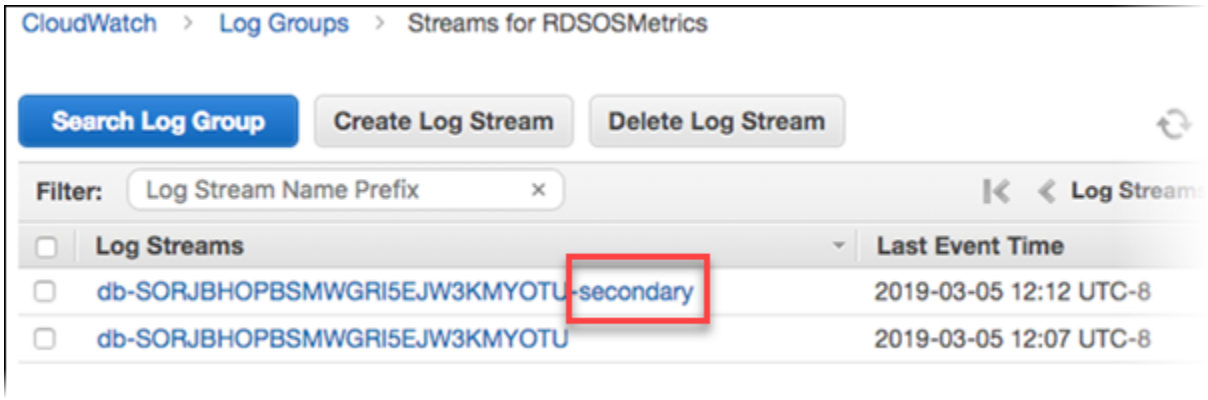
Depois de ativar o monitoramento avançado em sua instância de banco de dados ou cluster de banco de dados multi-AZ, você poderá exibir as respectivas métricas usando o CloudWatch Logs, com cada transmissão de log representando uma única instância ou cluster de banco de dados em monitoramento. O identificador da transmissão de log é o identificador de recurso (`DbiResourceId`) da instância ou cluster de banco de dados.

Para visualizar os dados de log de Monitoramento avançado

1. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/.

2. Se necessário, selecione a Região da AWS em que sua instância de banco de dados ou cluster de banco de dados multi-AZ está. Para obter mais informações, consulte Regiões e endpoints do na Referência geral do Amazon Web Services.
3. Escolha Logs no painel de navegação.
4. Escolha RDSOSMetrics na lista de grupos de log.

   Em uma implantação de instância de banco de dados multi-AZ, arquivos de log com `-secondary` anexado ao nome se referem à réplica em espera multi-AZ.



5. Escolha o stream de log que você deseja visualizar na lista de streams de log.

Referência de métricas do Amazon RDS

Nesta referência, você pode encontrar descrições de métricas do Amazon RDS referentes ao Amazon CloudWatch, Performance Insights e monitoramento avançado.

Tópicos

- Métricas do Amazon CloudWatch para o Amazon RDS
- Dimensões do Amazon CloudWatch para o Amazon RDS
- Métricas do Amazon CloudWatch para Insights de Performance do Amazon RDS
- Métricas de contadores do Performance Insights
- Estatísticas SQL para Performance Insights
- Métricas do sistema operacional em monitoramento avançado

Métricas do Amazon CloudWatch para o Amazon RDS

As métricas do Amazon CloudWatch fornecem insights sobre o desempenho e a integridade de instâncias e clusters do Amazon RDS, permitindo monitorar o comportamento do sistema e tomar decisões baseadas em dados. Essas métricas ajudam a monitorar a utilização de recursos, a atividade do banco de dados e a eficiência operacional, oferecendo visibilidade sobre o desempenho das instâncias.

Essa referência descreve as métricas específicas disponíveis para o Amazon RDS e explica como interpretá-las e usá-las para otimizar o desempenho do banco de dados, solucionar problemas e garantir alta disponibilidade.

O Amazon RDS publica métricas para o Amazon CloudWatch nos namespaces `AWS/RDS` e `AWS/Usage`.

Tópicos

- Métricas específicas da instância do Amazon CloudWatch para Amazon RDS
- Métricas de uso do Amazon CloudWatch para Amazon RDS

Métricas específicas da instância do Amazon CloudWatch para Amazon RDS

O namespace `AWS/RDS` no Amazon CloudWatch inclui as métricas específicas de instância a seguir.

> **Note**
>
> O console do Amazon RDS pode exibir métricas em unidades diferentes das unidades enviadas para o Amazon CloudWatch. Por exemplo, o console do Amazon RDS pode exibir uma métrica em megabytes (MB), enquanto a métrica é enviada para o Amazon CloudWatch em bytes.

Métrica	Descrição	Aplica-se a	Unidades
`BinLogDiskUsage`	A quantidade de espaço em disco ocupada por logs binários. Se backups automáticos estiverem habilitados para instâncias MySQL e MariaDB, incluindo réplicas de leitura, serão criados logs binários.	MariaDB MySQL	Bytes
`BurstBalance`	A porcentagem de créditos de E/S em burst-bucket SSD de finalidade geral (gp2) disponível.	Todos	Percentual
`CheckpointLag`	A quantidade de tempo desde o ponto de verificação mais recente.		Segundos
`ConnectionAttempts`	O número de tentativas de conexão com uma instância, sejam elas bem-sucedidas ou não.	MySQL	Contagem
`CPUUtilization`	O percentual de utilização da CPU.	Todos	Percentual
`CPUCreditUsage`	O número de créditos de CPU gastos pela instância por utilização de CPU. Um crédito de CPU é igual a um vCPU em execução com 100% de utilização por um minuto ou uma combinação equivalente de vCPUs, utilização e tempo. Por exemplo, talvez você		Créditos (minutos de vCPU)

Métrica	Descrição	Aplica-se a	Unidades
	tenha uma vCPU em execução com 50% de utilização por dois minutos ou duas vCPUs em execução com 25% de utilização por dois minutos. Essa métrica se aplica apenas às instâncias `db.t2`, `db.t3` e `db.t4g`. **Note** Recomendamos usar as classes de instância de banco de dados T somente para servidores de desenvolvimento e teste, ou outros servidores que não sejam de produção. Para ter mais detalhes sobre as classes de instâncias T, consulte Tipos de classe de instância de banco de dados. As métricas de crédito de CPU estão disponíveis a uma frequência de apenas 5 minutos. Se você especificar um período de mais cinco minutos, use a estatística `Sum` em vez da estatística `Average`.		

Métrica	Descrição	Aplica-se a	Unidades
`CPUCreditBalance`	O número de créditos ganhos de CPU que uma instância acumulou desde que foi executada ou iniciada. Para a T2 Padrão, o `CPUCreditBalance` também inclui o número de créditos de execução que foram acumulados. Os créditos são acumulados no saldo de créditos após terem sido ganhos e são removidos do saldo de créditos quando são gastos. O saldo de crédito tem um limite máximo, determinado pelo tamanho da instância. Depois que o limite for atingido, todos os novos créditos ganhos serão descartados. Para a T2 Padrão, os créditos de execução não são contabilizados para o limite. Os créditos do `CPUCreditBalance` são disponibilizados para que a instância gaste e apresente intermitê ncia com uma utilização de CPU acima da linha de base. Quando uma instância está em execução, os créditos do `CPUCredit Balance` não expiram. Quando a instância para, o `CPUCredit Balance` não persiste, e todos os créditos acumulados são perdidos. As métricas de crédito de CPU estão disponíveis a uma frequência de apenas 5 minutos.		Créditos (minutos de vCPU)

Métrica	Descrição	Aplica-se a	Unidades
	Essa métrica se aplica apenas às instâncias `db.t2`, `db.t3` e `db.t4g`. **Note** Recomendamos usar as classes de instância de banco de dados T somente para servidores de desenvolvimento e teste, ou outros servidores que não sejam de produção. Para ter mais detalhes sobre as classes de instâncias T, consulte Tipos de classe de instância de banco de dados. Os créditos de execução funcionam da mesma forma no Amazon RDS e no Amazon EC2. Para ter mais informaçõ es, consulte Créditos de execução no Guia do usuário do Amazon Elastic Compute Cloud para instâncias do Linux.		

Métrica	Descrição	Aplica-se a	Unidades
`CPUSurplusCreditBalance`	O número de créditos excedentes gastos por uma instância ilimitada quando seu valor `CPUCreditBalance` é zero. O valor `CPUSurplusCreditBalance` é pago pelos créditos de CPU ganhos. Se o número de créditos excedentes ultrapassar o número máximo de créditos que a instância pode ganhar em um período de 24 horas, os créditos excedentes gastos acima do limite máximo incorrerão em uma taxa adicional. As métricas de crédito de CPU estão disponíveis apenas a uma frequência de 5 minutos.	Todos	Créditos (minutos de vCPU)

Métrica	Descrição	Aplica-se a	Unidades
`CPUSurplusCreditsCharged`	O número de créditos excedentes gastos que não são pagos pelos créditos de CPU ganhos e que, portanto, incorrem em uma cobrança adicional. Os créditos excedentes gastos são cobrados quando uma das seguintes situações ocorre: • Os créditos excedentes ultrapassaram o número máximo de créditos que a instância pode obter em um período de 24 horas. Os créditos excedentes gastos acima do limite máximo são cobrados no final da hora. • A instância é interrompida ou encerrada. • A instância é alterada de `unlimited` para `standard`. As métricas de crédito de CPU estão disponíveis apenas a uma frequência de 5 minutos.	Todos	Créditos (minutos de vCPU)

Métrica	Descrição	Aplica-se a	Unidades
`DatabaseConnections`	O número de conexões de rede cliente com a instância do banco de dados. O número de sessões de banco de dados pode ser maior que o valor da métrica porque o valor da métrica não inclui o seguinte: • Sessões que não têm mais uma conexão de rede, mas que o banco de dados não limpou • Sessões criadas pelo mecanismo de banco de dados para seus próprios propósitos • Sessões criadas pelos recursos de execução paralela do mecanismo de banco de dados • Sessões criadas pelo programador de trabalhos do mecanismo de banco de dados • Conexões do Amazon RDS	Todos	Contagem
`DiskQueueDepth`	O número de E/Ss (solicitações de leitura/gravação) pendentes aguardando para acessar o disco.	Todos	Contagem
`DiskQueueDepthLogVolume`	O número de E/Ss (solicitações de leitura/gravação) pendentes aguardando para acessar o disco.	Instâncias de banco de dados com volume de log dedicado habilitado	Contagem

Métrica	Descrição	Aplica-se a	Unidades
`EBSByteBalance%`	A porcentagem de créditos de taxa de transferência restantes no bucket de intermitência do seu banco de dados do RDS. Essa métrica está disponível somente para monitoramento básico. O valor da métrica é baseado no throughput e nas IOPS de todos os volumes, inclusive o volume raiz, em vez de apenas nos volumes que contêm arquivos de banco de dados. Para localizar os tamanhos de instância compatíveis com essa métrica, consulte os tamanhos de instância com um asterisco (*) na tabela Otimizadas para EBS por padrão no Guia do usuário do Amazon EC2. A estatística `Sum` não é aplicável a essa métrica.	Todos	Percentual

Métrica	Descrição	Aplica-se a	Unidades
`EBSIOBalance%`	A porcentagem de créditos de E/S restantes no bucket de intermitência do seu banco de dados do RDS. Essa métrica está disponível somente para monitoramento básico. O valor da métrica é baseado no throughput e nas IOPS de todos os volumes, inclusive o volume raiz, em vez de apenas nos volumes que contêm arquivos de banco de dados. Para localizar os tamanhos de instância que comportam essa métrica, consulte Amazon EBS–optimized instance types no Guia do usuário do Amazon EC2. A estatística `Sum` não é aplicável a essa métrica. Esta métrica é diferente de `BurstBalance` . Para saber como usar essa métrica, consulte Melhorar a performance das aplicações e reduzir custos com o recurso de intermitência de instâncias otimizadas para Amazon EBS.	Todos	Percentual
`FailedSQLServerAgentJobsCount`	O número de trabalhos do Microsoft SQL Server Agent durante o último minuto.	Microsoft SQL Server	Contagem a cada minuto

Métrica	Descrição	Aplica-se a	Unidades
`FreeableMemory`	A quantidade de memória de acesso aleatório disponível. Para instâncias de banco de dados MariaDB, MySQL, Oracle e PostgreSQL, essa métrica reporta o valor do campo `MemAvailable` de `/proc/meminfo` .	Todos	Bytes
`FreeLocalStorage`	A quantidade de espaço de armazenamento local disponível. Essa métrica só se aplica a classes de instância de banco de dados com volumes de armazenamento de instância SSD de NVMe. Para obter informações sobre instâncias do Amazon EC2 com volumes de armazenamento de instância SSD de NVMe, consulte Volumes de armazenamento de instância. As classes de instância de banco de dados RDS equivalentes têm os mesmos volumes de armazenamento de instância. Por exemplo, as classes de instância de banco de dados db.m6gd e db.r6gd têm volumes de armazenamento de instância SSD de NVMe.		Bytes

Métrica	Descrição	Aplica-se a	Unidades
`FreeLocalStoragePercent`	A porcentagem de espaço de armazenamento local disponível. Essa métrica só se aplica a classes de instância de banco de dados com volumes de armazenamento de instância SSD de NVMe. Para obter informações sobre instâncias do Amazon EC2 com volumes de armazenamento de instância SSD de NVMe, consulte Volumes de armazenamento de instância. As classes de instância de banco de dados RDS equivalentes têm os mesmos volumes de armazenamento de instância. Por exemplo, as classes de instância de banco de dados db.m6gd e db.r6gd têm volumes de armazenamento de instância SSD de NVMe.		Percentual
`FreeStorageSpace`	A quantidade de espaço de armazenamento disponível.	Todos	Bytes
`FreeStorageSpaceLogVolume`	A quantidade de espaço de armazenamento disponível.	Instâncias de banco de dados com volume de log dedicado habilitado	Bytes
`IamDbAuthConnectionRequests`	O número de solicitações de conexão usando autenticação do IAM para a instância de banco de dados.	Todos	Contagem
`MaximumUsedTransactionIDs`	Os IDs máximos de transação que foram usados.	PostgreSQL	Contagem

Métrica	Descrição	Aplica-se a	Unidades
`NetworkReceiveThroughput`	O tráfego de rede de entrada (recebido) na instância de banco de dados, incluindo o tráfego de banco de dados do cliente e o tráfego do Amazon RDS usado para monitoramento e replicação.	Todos	Bytes por segundo
`NetworkTransmitThroughput`	O tráfego de rede de saída (transmitido) na instância de banco de dados, incluindo o tráfego de banco de dados do cliente e o tráfego do Amazon RDS usado para monitoramento e replicação.	Todos	Bytes por segundo
`OldestLogicalReplicationSlotLag`	O tamanho do atraso do Amazon RDS é a diferença entre o momento em que uma transação é confirmada no banco de dados de origem e o momento em que o RDS aplica a transação no banco de dados de réplica.	PostgreSQL	Bytes
`OldestReplicationSlotLag`	O tamanho do atraso da réplica que demora mais em termos de dados de log com gravação antecipada (WAL) recebidos.	PostgreSQL	Bytes
`ReadIOPS`	O número médio de operações E/S de leitura de disco por segundo.	Todos	Contagem por segundo

Métrica	Descrição	Aplica-se a	Unidades
`ReadIOPSLocalStorage`	O número médio de operações de E/S de leitura de disco no armazenamento local por segundo. Essa métrica só se aplica a classes de instância de banco de dados com volumes de armazenamento de instância SSD de NVMe. Para obter informações sobre instâncias do Amazon EC2 com volumes de armazenamento de instância SSD de NVMe, consulte Volumes de armazenamento de instância. As classes de instância de banco de dados RDS equivalentes têm os mesmos volumes de armazenamento de instância. Por exemplo, as classes de instância de banco de dados db.m6gd e db.r6gd têm volumes de armazenamento de instância SSD de NVMe.		Contagem por segundo
`ReadIOPSLogVolume`	O número médio de operações E/S de leitura de disco por segundo.	Instâncias de banco de dados com volume de log dedicado habilitado	Contagem por segundo
`ReadLatency`	O tempo médio necessário por operação de E/S de disco.	Todos	Segundos

Métrica	Descrição	Aplica-se a	Unidades
`ReadLatencyLocalStorage`	O tempo médio necessário por operação de I/O de disco para armazenamento local. Essa métrica só se aplica a classes de instância de banco de dados com volumes de armazenamento de instância SSD de NVMe. Para obter informações sobre instâncias do Amazon EC2 com volumes de armazenamento de instância SSD de NVMe, consulte Volumes de armazenamento de instância. As classes de instância de banco de dados RDS equivalentes têm os mesmos volumes de armazenamento de instância. Por exemplo, as classes de instância de banco de dados db.m6gd e db.r6gd têm volumes de armazenamento de instância SSD de NVMe.		Segundos
`ReadLatencyLogVolume`	O tempo médio necessário por operação de E/S de disco.	Instâncias de banco de dados com volume de log dedicado habilitado	Segundos
`ReadThroughput`	O número médio de bytes lidos do disco por segundo.	Todos	Bytes por segundo

Métrica	Descrição	Aplica-se a	Unidades
`ReadThroughputLocalStorage`	O número médio de bytes lidos do disco por segundo para armazenamento local. Essa métrica só se aplica a classes de instância de banco de dados com volumes de armazenamento de instância SSD de NVMe. Para obter informações sobre instâncias do Amazon EC2 com volumes de armazenamento de instância SSD de NVMe, consulte Volumes de armazenamento de instância. As classes de instância de banco de dados RDS equivalentes têm os mesmos volumes de armazenamento de instância. Por exemplo, as classes de instância de banco de dados db.m6gd e db.r6gd têm volumes de armazenamento de instância SSD de NVMe.		Bytes por segundo
`ReadThroughputLogVolume`	O número médio de bytes lidos do disco por segundo.	Instâncias de banco de dados com volume de log dedicado habilitado	Bytes por segundo

Métrica	Descrição	Aplica-se a	Unidades
`ReplicaLag`	Em relação às configurações de réplicas de leitura, a quantidade de tempo que uma instância de banco de dados de réplica de leitura atrasa em relação à instância de banco de dados de origem. Aplica-se a réplicas de leitura do MySQL, MariaDB, Oracle, PostgreSQL e SQL Server. Em relação aos clusters de banco de dados multi-AZ, a diferença de tempo entre a transação mais recente na instância de banco de dados de gravador e a transação aplicada mais recente em uma instância de banco de dados de leitor.		Segundos
`ReplicationChannelLag`	Em relação às configurações de réplicas de várias origens, a quantidade de tempo que um canal específico na réplica de várias origens atrasa em relação à origem da instância de banco de dados. Para obter mais informações, consulte the section called “Monitorar canais de replicação de várias fontes”.	MySQL	Segundos
`ReplicationSlotDiskUsage`	O espaço em disco usado por arquivos de slot de replicação.	PostgreSQL	Bytes

Métrica	Descrição	Aplica-se a	Unidades
`SwapUsage`	A quantidade de troca usada na instância de banco de dados.	MariaDB MySQL Oracle PostgreSQL	Bytes
`TempDbAvailableDataSpace`	A quantidade de espaço de dados disponível no tempdb e no volume em que o tempdb está localizado. Use essa métrica para monitorar a disponibilidade de espaço de dados do tempdb e planejar a capacidade adequadamente. Valores baixos podem indicar a necessidade de aumentar o armazenamento ou otimizar consultas que usam muito o tempdb.	SQL Server	Bytes
`TempDbAvailableLogSpace`	A quantidade de espaço de log disponível no tempdb e no volume em que o tempdb está localizado. Use essa métrica para monitorar a disponibilidade de espaço de log do tempdb e evitar condições de log de transações cheio. Essencial para workloads com transações grandes ou alta simultaneidade que geram atividade de log significativa.	SQL Server	Bytes

Métrica	Descrição	Aplica-se a	Unidades
`TempDbDataFileUsage`	A porcentagem de arquivos de dados usados no tempdb. Essa métrica não leva em conta a possibilidade de crescimento de arquivos. Use essa métrica para monitorar a utilização do arquivo de dados do tempdb e identificar possíveis gargalos de desempenho. Valores altos podem indicar a necessidade de otimizar consultas que criam grandes objetos temporários ou de aumentar o tamanho do tempdb.	SQL Server	Percentual
`TempDbLogFileUsage`	A porcentagem de arquivos de log usados no tempdb. Essa métrica não leva em conta a possibilidade de crescimento de arquivos. Use essa métrica para monitorar a utilização do arquivo de log do tempdb e evitar problemas de desempenho. Valores altos podem indicar transações de longa duração ou atividade excessiva de registro em log que podem afetar o desempenho geral do banco de dados.	SQL Server	Percentual
`TransactionLogsDiskUsage`	O espaço em disco usado por logs de transação.	PostgreSQL	Bytes
`TransactionLogsGeneration`	O tamanho dos logs de transação gerados por segundo.	PostgreSQL	Bytes por segundo

Métrica	Descrição	Aplica-se a	Unidades
`WriteIOPS`	O número médio de operações de E/S de gravação de disco por segundo.	Todos	Contagem por segundo
`WriteIOPS LocalStorage`	O número médio de operações de E/S de gravação de disco por segundo no armazenamento local. Essa métrica só se aplica a classes de instância de banco de dados com volumes de armazenamento de instância SSD de NVMe. Para obter informações sobre instâncias do Amazon EC2 com volumes de armazenamento de instância SSD de NVMe, consulte Volumes de armazenamento de instância. As classes de instância de banco de dados RDS equivalentes têm os mesmos volumes de armazenamento de instância. Por exemplo, as classes de instância de banco de dados db.m6gd e db.r6gd têm volumes de armazenamento de instância SSD de NVMe.		Contagem por segundo
`WriteIOPS LogVolume`	O número médio de operações de E/S de gravação de disco por segundo.	Instâncias de banco de dados com volume de log dedicado habilitado	Contagem por segundo
`WriteLatency`	O tempo médio necessário por operação de E/S de disco.	Todos	Segundos

Métrica	Descrição	Aplica-se a	Unidades
`WriteLatencyLocalStorage`	O tempo médio necessário por operação de E/S de disco no armazenamento local. Essa métrica só se aplica a classes de instância de banco de dados com volumes de armazenamento de instância SSD de NVMe. Para obter informações sobre instâncias do Amazon EC2 com volumes de armazenamento de instância SSD de NVMe, consulte Volumes de armazenamento de instância. As classes de instância de banco de dados RDS equivalentes têm os mesmos volumes de armazenamento de instância. Por exemplo, as classes de instância de banco de dados db.m6gd e db.r6gd têm volumes de armazenamento de instância SSD de NVMe.		Segundos
`WriteLatencyLogVolume`	O tempo médio necessário por operação de E/S de disco.	Instâncias de banco de dados com volume de log dedicado habilitado	Segundos
`WriteThroughput`	O número médio de bytes gravados no disco por segundo.	Todos	Bytes por segundo
`WriteThroughputLogVolume`	O número médio de bytes gravados no disco por segundo.	Instâncias de banco de dados com volume de log dedicado habilitado	Bytes por segundo

Métrica	Descrição	Aplica-se a	Unidades
`WriteThroughputLocalStorage`	O número médio de bytes gravados no disco por segundo para o armazenamento local. Essa métrica só se aplica a classes de instância de banco de dados com volumes de armazenamento de instância SSD de NVMe. Para obter informações sobre instâncias do Amazon EC2 com volumes de armazenamento de instância SSD de NVMe, consulte Volumes de armazenamento de instância. As classes de instância de banco de dados RDS equivalentes têm os mesmos volumes de armazenamento de instância. Por exemplo, as classes de instância de banco de dados db.m6gd e db.r6gd têm volumes de armazenamento de instância SSD de NVMe.		Bytes por segundo

Métricas de uso do Amazon CloudWatch para Amazon RDS

O namespace `AWS/Usage` no Amazon CloudWatch inclui métricas de uso específico da conta para suas cotas de serviço do Amazon RDS. O CloudWatch coleta métricas de uso automaticamente para todas as Regiões da AWS.

Para ter mais informações, consulte Métricas de uso do CloudWatch no Guia do usuário do Amazon CloudWatch. Para ter mais informações sobre cotas, consulte Cotas e restrições do Amazon RDS e Solicitar um aumento da cota no Guia do usuário do Service Quotas.

Métrica	Descrição	Unidades*
`AllocatedStorage`	O armazenamento total para todas as instâncias de banco de dados A soma exclui instâncias de migração temporári a.	Gigabytes
`Authoriza tionsPerD BSecurityGroup`	O número de regras de entrada por grupo de segurança do banco de dados na sua Conta da AWS. O valor usado é o maior número de regras de entrada em um grupo de segurança de banco de dados na conta. Outros grupos de segurança de banco de dados na conta podem ter um número menor de regras de entrada.	Contagem
`CustomEnd pointsPer DBCluster`	O número de endpoints personalizados por cluster de banco de dados na sua Conta da AWS. O valor usado é o maior número de endpoints personalizados em um cluster de banco de dados na conta. Outros clusters de banco de dados na conta podem ter um número menor de endpoints personalizados.	Contagem
`CustomEng ineVersions`	O número de versões personalizadas do mecanismo (CEVs) para o Amazon RDS Custom na sua Conta da AWS.	Contagem
`DBCluster ParameterGroups`	O número máximo de grupos de parâmetros de cluster de banco de dados em sua Conta da AWS. A contagem exclui grupos de parâmetros padrão.	Contagem
`DBClusterRoles`	O número de perfis do AWS Identity and Access Management (IAM) associados por cluster de banco de dados na sua Conta da AWS. O valor usado é o maior número de perfis do IAM associadas a um cluster de banco de dados na conta. Outros clusters de banco de dados na conta podem ter um número menor de perfis do IAM associados.	Contagem
`DBClusters`	O número de clusters de banco de dados do Amazon Aurora em sua Conta da AWS.	Contagem

Métrica	Descrição	Unidades*
DBInstanceRoles	O número de perfis do AWS Identity and Access Management (IAM) associados por instância de banco de dados na sua Conta da AWS. O valor usado é o maior número de perfis do IAM associadas a uma instância de banco de dados na conta. Outras instâncias de banco de dados na conta podem ter um número menor de perfis do IAM associados.	Contagem
DBInstances	O número de instâncias de banco de dados em sua Conta da AWS.	Contagem
DBParameterGroups	O número de grupos de parâmetros de banco de dados em sua Conta da AWS. A contagem exclui grupos de parâmetros de banco de dados padrão.	Contagem
DBSecurityGroups	O número de grupos de segurança em sua Conta da AWS. A contagem exclui o grupo de segurança padrão e o da VPC padrão.	Contagem
DBSubnetGroups	O número de grupos de sub-redes de banco de dados em sua Conta da AWS. A contagem exclui o grupo de sub-redes padrão.	Contagem
EventSubscriptions	O número de assinaturas de notificação de eventos em sua Conta da AWS.	Contagem
Integrations	O número de Integrações ETL zero com o Amazon Redshift em sua Conta da AWS.	Contagem
ManualClusterSnapshots	O número de snapshots de cluster de banco de dados criados manualmente em sua Conta da AWS. A contagem exclui snapshots inválidos.	Contagem
ManualSnapshots	O número de snapshots de banco de dados criados manualmente em sua Conta da AWS. A contagem exclui snapshots inválidos.	Contagem

Métrica	Descrição	Unidades*
`OptionGroups`	O número de grupos de opções em sua Conta da AWS. A contagem exclui os grupo de opções padrão.	Contagem
`Proxies`	O número de proxies do RDS na sua conta da AWS.	Contagem
`ReadReplicasPerMaster`	O número de réplicas de leitura por instância de banco de dados na sua conta. O valor usado é o maior número de réplicas de leitura para uma instância de banco de dados na conta. Outras instâncias de banco de dados na conta podem ter um número menor de réplicas de leitura.	Contagem
`ReservedDBInstances`	O número de instâncias de banco de dados reservadas em sua Conta da AWS. A contagem exclui instâncias desativadas ou recusadas.	Contagem
`SubnetsPerDBSubnetGroup`	O número de sub-redes para por grupo de sub-redes de banco de dados na sua Conta da AWS. O número máximo de sub-redes para um grupo de sub-redes de banco de dados na conta. Outros grupos de sub-redes de banco de dados na conta podem ter um número menor de sub-redes.	Contagem

> **Note**
>
> O Amazon RDS não publica unidades para métricas de uso no CloudWatch. As unidades só aparecem na documentação.

Dimensões do Amazon CloudWatch para o Amazon RDS

Você pode filtrar dados de métricas do Amazon RDSmétricas usando qualquer dimensão na tabela a seguir.

Dimensão	Filtra os dados solicitados para . . .
`DBInstanceIdentifier`	Uma instância específica de banco de dados.
`DatabaseClass`	Todas as instâncias em uma classe de banco de dados. Por exemplo, você pode agregar métricas para todas as instâncias que pertencem à classe do banco de dado `db.r5.large` .
`EngineName`	Apenas o nome do motor identificado. Por exemplo, você pode agrupar métricas para todas as instâncias que têm o nome de mecanismo `postgres`.
`SourceRegion`	Apenas a região especificada. Por exemplo, você pode agregar métricas de todas as instâncias de Bancos de Dados na região `us-east-1` .

Métricas do Amazon CloudWatch para Insights de Performance do Amazon RDS

O Insights de Performance publica automaticamente algumas métricas no Amazon CloudWatch. Os mesmos dados podem ser consultados do Performance Insights, mas ter as métricas no CloudWatch facilita a adição de alarmes do CloudWatch. Também facilita a adição de métricas aos painéis do CloudWatch existentes.

Métrica	Descrição
DBLoad	O número de sessões ativas para o banco de dados. Normalmente, você deseja os dados para o número médio de sessões ativas. No Performance Insights, esses dados são consultados como `db.load.avg` .
DBLoadCPU	O número de sessões ativas em que o tipo do evento de espera é CPU. No Performance Insights, esses dados são consultados como

Métrica	Descrição
	`db.load.avg` , filtrados pelo tipo de evento de espera `CPU`.
DBLoadNonCPU	O número de sessões ativas em que o tipo do evento de espera não é CPU.
DBLoadRelativeToNumVCPUs	A proporção entre a carga de banco de dados e o número de CPUs virtuais para o banco de dados.

> **Note**
>
> Essas métricas serão publicadas no CloudWatch somente se houver carga na instância de banco de dados.

Você pode examinar essas métricas usando o console do CloudWatch, a AWS CLI ou a API do CloudWatch. Você também pode examinar outras métricas de contador do Insights de Performance usando uma função matemática de métrica especial. Para obter mais informações, consulte Consultar outras métricas de contadores do Insights de Performance no CloudWatch.

Por exemplo, você pode obter as estatísticas da métrica `DBLoad` executando o comando get-metric-statistics.

```
aws cloudwatch get-metric-statistics \
    --region us-west-2 \
    --namespace AWS/RDS \
    --metric-name DBLoad  \
    --period 60 \
    --statistics Average \
    --start-time 1532035185 \
    --end-time 1532036185 \
    --dimensions Name=DBInstanceIdentifier,Value=db-loadtest-0
```

Este exemplo gera uma saída semelhante à seguinte.

```
{
```

```
"Datapoints": [
{
"Timestamp": "2021-07-19T21:30:00Z",
"Unit": "None",
"Average": 2.1
},
{
"Timestamp": "2021-07-19T21:34:00Z",
"Unit": "None",
"Average": 1.7
},
{
"Timestamp": "2021-07-19T21:35:00Z",
"Unit": "None",
"Average": 2.8
},
{
"Timestamp": "2021-07-19T21:31:00Z",
"Unit": "None",
"Average": 1.5
},
{
"Timestamp": "2021-07-19T21:32:00Z",
"Unit": "None",
"Average": 1.8
},
{
"Timestamp": "2021-07-19T21:29:00Z",
"Unit": "None",
"Average": 3.0
},
{
"Timestamp": "2021-07-19T21:33:00Z",
"Unit": "None",
"Average": 2.4
}
],
"Label": "DBLoad"
}
```


Para obter mais informações sobre o CloudWatch, consulte O que é o Amazon CloudWatch? no Guia do usuário do Amazon CloudWatch.

Consultar outras métricas de contadores do Insights de Performance no CloudWatch

> **Note**
>
> Se você habilitar o modo Avançado do Database Insights, o Amazon RDS publicará métricas de contador do Insights de Performance no Amazon CloudWatch. Com o Database Insights, você não precisa usar a função de matemática métrica `DB_PERF_INSIGHTS`. Você pode usar o painel do CloudWatch Database Insights para pesquisar, consultar e definir alarmes para métricas de contador do Insights de Performance.

É possível consultar, emitir alarmes e criar grafos sobre as métricas do Insights de Performance do RDS pelo CloudWatch. Você pode acessar informações sobre a instância de banco de dados usando a função matemática de métrica `DB_PERF_INSIGHTS` do CloudWatch. Essa função permite que você use as métricas do Insights de Performance que não são diretamente informadas ao CloudWatch para criar uma série temporal.

É possível usar a nova função matemática de métrica clicando no menu suspenso Adicionar matemática na tela Selecionar métrica no console do CloudWatch. Você pode usá-lo para criar alarmes e grafos sobre as métricas do Insights de Performance ou sobre combinações das métricas do CloudWatch e do Insights de Performance, incluindo alarmes de alta resolução para métricas abaixo de um minuto. Também é possível usar a função programaticamente incluindo a expressão matemática de métrica em uma solicitação `get-metric-data`. Consulte mais informações em Metric math syntax and functions e Create an alarm on Performance Insights counter metrics from an AWS database.

Métricas de contadores do Performance Insights

Métricas de contador são métricas de performance do sistema operacional e do banco de dados no painel do Performance Insights. Para ajudar a identificar e analisar problemas de performance, é possível correlacionar métricas de contadores com a carga de banco de dados. Você deve adicionar uma função estatística à métrica para obter os valores métricos. Por exemplo, as funções compatíveis com a métrica `os.memory.active` são `.avg`, `.min`, `.max`, `.sum` e `.sample_count`.

As métricas do contador são coletadas uma vez por minuto. A coleta de métricas do sistema operacional depende do status do recurso Monitoramento Avançado. Se o recurso estiver desativado, as métricas do sistema operacional serão coletadas uma vez por minuto. Se o recurso estiver ativado, as métricas do sistema operacional serão coletadas para o período selecionado.

Para obter mais informações sobre como ativar ou desativar o recurso Monitoramento Avançado, consulte Ativar e desativar o monitoramento aprimorado.

Tópicos

- Contadores de sistema operacional do Performance Insights
- Contadores do Performance Insights para o Amazon RDS para MariaDB e MySQL
- Contadores do Performance Insights para o Amazon RDS para Microsoft SQL Server
- Contadores do Performance Insights para o Amazon RDS para Oracle
- Contadores do Performance Insights para o Amazon RDS para PostgreSQL

Contadores de sistema operacional do Performance Insights

Os contadores de sistema operacional a seguir, que são prefixados com `os`, estão disponíveis para o recurso Insights de Performance em todos os mecanismos do RDS, exceto o RDS para SQL Server.

Você pode usar a API `ListAvailableResourceMetrics` para obter a lista de métricas de contador disponíveis para sua instância de banco de dados. Para obter mais informações, consulte ListAvailableResourceMetrics no Guia de referência de API do Insights de Performance do Amazon RDS.

Contador	Type	Unidade	Métrica	Descrição
Ativo	Memória	Kilobytes	os.memory.active	A quantidade de memória atribuída, em kilobytes.
Buffers	Memória	Kilobytes	os.memory.buffers	A quantidade de memória usada para o buffer de solicitações de E/S antes de gravar no dispositivo de armazenamento, em kilobytes.

Contador	Type	Unidade	Métrica	Descrição
Em cache	Memória	Kilobytes	os.memory.cached	A quantidade de memória utilizada para o armazenamento em cache da E/S baseada em sistema de arquivos, em quilobytes.
Cache de banco de dados	Memória	Bytes	os.memory.db.cache	A quantidade de memória usada para o cache de páginas por processo de banco de dados, incluindo tmpfs (shmem), em bytes.
Tamanho do conjunto residente de banco de dados	Memória	Bytes	os.memory.db.residentSetSize	A quantidade de memória usada para o cache anônimo e swap por processo de banco de dados, sem incluir tmpfs (shmem), em bytes.

Contador	Type	Unidade	Métrica	Descrição
Swap de banco de dados	Memória	Bytes	os.memory .db.swap	A quantidade de memória usada para swap por processo de banco de dados, em bytes.
Sujeira	Memória	Kilobytes	os.memory.dirty	A quantidad e de páginas de memória na RAM que foram modificad as, mas não gravadas nos blocos de dados relacionados no armazenamento, em kilobytes.
Gratuito	Memória	Kilobytes	os.memory.free	A quantidade de memória não atribuída, em kilobytes.
Páginas enormes livres	Memória	Páginas	os.memory .hugePagesFree	O número de páginas enormes livres. Páginas enormes são um recurso do kernel do Linux.
Páginas enormes reservadas	Memória	Páginas	os.memory .hugePagesRsvd	O número de páginas enormes confirmadas.

Contador	Type	Unidade	Métrica	Descrição
Tamanho de páginas enormes	Memória	Kilobytes	os.memory .hugePagesSize	O tamanho de cada unidade de páginas enormes, em kilobytes.
Páginas enormes surp	Memória	Páginas	os.memory .hugePagesSurp	O número de páginas enormes excedentes disponíveis em comparação com o total.
Total de páginas enormes	Memória	Páginas	os.memory .hugePagesTotal	O número total de páginas enormes.
Inactive	Memória	Kilobytes	os.memory .inactive	A quantidade de páginas de memória usadas com menos frequência, em kilobytes.
Mapeamento	Memória	Kilobytes	os.memory .mapped	A quantidade total de conteúdo do sistema de arquivos que é mapeada na memória dentro de um espaço de endereçamento de processos, em kilobytes.

Contador	Type	Unidade	Métrica	Descrição
Contagem de eliminações por memória insuficiente	Memória	Eliminações	os.memory.outOfMemoryKillCount	O número de encerramentos OOM que aconteceram durante o último intervalo de coleta.
Tabelas de página	Memória	Kilobytes	os.memory.pageTables	A quantidade de memória usada por tabelas de página, em kilobytes.
Slab	Memória	Kilobytes	os.memory.slab	A quantidade de estruturas de dados reutilizáveis do kernel, em kilobytes.
Total	Memória	Kilobytes	os.memory.total	A quantidade total de memória, em kilobytes.
Writeback	Memória	Kilobytes	os.memory.writeback	A quantidade de páginas sujas na RAM que ainda estão sendo gravadas no armazenamento de suporte, em kilobytes.

Contador	Type	Unidade	Métrica	Descrição
Visitas	Utilização da CPU	Porcentagem	os.cpuUtilization.guest	A porcentagem de CPU em uso por programas de convidado.
Ocioso	Utilização da CPU	Porcentagem	os.cpuUtilization.idle	A porcentagem de CPU que está ociosa.
Irq	Utilização da CPU	Porcentagem	os.cpuUtilization.irq	A porcentagem de CPU em uso por interrupções de software.
Amigável	Utilização da CPU	Porcentagem	os.cpuUtilization.nice	A porcentagem de CPU em uso por programas em execução com a prioridade mais baixa.
Roubo	Utilização da CPU	Porcentagem	os.cpuUtilization.steal	A porcentag em de CPU em uso por outras máquinas virtuais.
Sistema	Utilização da CPU	Porcentagem	os.cpuUtilization.system	A porcentagem de CPU em uso pelo kernel.
Total	Utilização da CPU	Porcentagem	os.cpuUtilization.total	A porcentagem total da CPU em uso. Esse valor inclui o valor amigável.

Contador	Type	Unidade	Métrica	Descrição
Usuário	Utilização da CPU	Porcentagem	os.cpuUtilization.user	A porcentagem de CPU em uso por programas do usuário.
Aguardar	Utilização da CPU	Porcentagem	os.cpuUtilization.wait	A porcentagem de CPU fora de uso ao aguardar o acesso de E/S.
PS de E/S de leitura	E/S de disco	Solicitações por segundo	os.diskIO.<devicename>.readIOsPS	O número de operações de leitura por segundo.
PS de E/S de gravação	E/S de disco	Solicitações por segundo	os.diskIO.<devicename>.writeIOsPS	O número de operações de gravação por segundo.
Comprimento médio da fila	E/S de disco	Solicitações	os.diskIO.<devicename>.avgQueueLen	O número de solicitações que aguardam na fila do dispositivo de E/S.
Tamanho médio de solicitação	E/S de disco	Solicitações	os.diskIO.<devicename>.avgReqSz	O número de solicitações que aguardam na fila do dispositivo de E/S.

Contador	Type	Unidade	Métrica	Descrição
Espera	E/S de disco	Milissegundos	os.diskIO.<devicename>.await	O número de milissegundos necessários para responder a solicitações, incluindo o tempo na fila e o tempo de serviço.
PS de E/S de leitura	E/S de disco	Solicitações	os.diskIO.<devicename>.readIOsPS	O número de operações de leitura por segundo.
KB de leitura	E/S de disco	Kilobytes	os.diskIO.<devicename>.readKb	O número total de kilobytes lidos.
PS de KB de leitura	E/S de disco	Quilobytes/segundo	os.diskIO.<devicename>.readKbPS	O número de kilobytes lidos por segundo.
PS de rrqm	E/S de disco	Solicitações por segundo	os.diskIO.<devicename>.rrqmPS	O número de solicitações de leitura mescladas enfileiradas por segundo.
TPS	E/S de disco	Transações por segundo	os.diskIO.<devicename>.tps	O número de transações de E/S por segundo.

Contador	Type	Unidade	Métrica	Descrição
Util	E/S de disco	Porcentagem	os.diskIO.<devicename>.util	A porcentagem de tempo de CPU durante o qual as solicitações foram emitidas.
KB de gravação	E/S de disco	Kilobytes	os.diskIO.<devicename>.writeKb	O número total de kilobytes gravados.
PS de KB de gravação	E/S de disco	Quilobytes/segundo	os.diskIO.<devicename>.writeKbPS	O número de kilobytes gravados por segundo.
PS de wrqm	E/S de disco	Solicitações por segundo	os.diskIO.<devicename>.wrqmPS	O número de solicitações de gravação mescladas enfileiradas por segundo.
Bloqueado	Tarefas	Tarefas	os.tasks.blocked	O número de tarefas que estão bloqueadas.
Em execução	Tarefas	Tarefas	os.tasks.running	O número de tarefas que estão sendo executadas.

Contador	Type	Unidade	Métrica	Descrição
Sleeping	Tarefas	Tarefas	os.tasks.sleeping	O número de tarefas que estão em suspensão.
Interrompido	Tarefas	Tarefas	os.tasks.stopped	O número de tarefas que estão interrompidas.
Total	Tarefas	Tarefas	os.tasks.total	O número total de tarefas.
Zumbi	Tarefas	Tarefas	os.tasks.zombie	O número de tarefas filho que estão inativas com uma tarefa pai ativa.
Um	Carga média por minuto	Processos	os.loadAverageMinute.one	O número de processos que estão solicitando tempo de CPU no último minuto.
Quinze	Carga média por minuto	Processos	os.loadAverageMinute.fifteen	O número de processos que estão solicitando tempo de CPU nos últimos 15 minutos.

Contador	Type	Unidade	Métrica	Descrição
Cinco	Carga média por minuto	Processos	os.loadAverageMinute.five	O número de processos que estão solicitando tempo de CPU nos últimos 5 minutos.
Em cache	Troca	Kilobytes	os.swap.cached	A quantidade de memória de permuta, em kilobytes, usada como a memória cache.
Gratuito	Troca	Kilobytes	os.swap.free	A quantidade de memória de troca livre, em kilobytes.
Em	Troca	Kilobytes	os.swap.in	A quantidade de memória, em kilobytes, transferida temporariamente do disco.
Saída	Troca	Kilobytes	os.swap.out	A quantidade de memória, em kilobytes, transferida temporariamente para o disco.

Contador	Type	Unidade	Métrica	Descrição
Total	Troca	Kilobytes	os.swap.total	A quantidade de memória swap disponível, em kilobytes.
Máximo de arquivos	Sistema de arquivos	Arquivos	os.fileSys.maxFiles	O número máximo de arquivos que podem ser criados para o sistema de arquivos.
Arquivos usados	Sistema de arquivos	Arquivos	os.fileSys.usedFiles	O número de arquivos no sistema de arquivos.
Porcentagem de arquivos usados	Sistema de arquivos	Arquivos	os.fileSys.usedFilePercent	A porcentagem de arquivos disponíveis em uso.
Porcentagem usada	Sistema de arquivos	Porcentagem	os.fileSys.usedPercent	A porcentagem do espaço em disco do sistema de arquivos em uso.
Usado	Sistema de arquivos	Kilobytes	os.fileSys.used	A quantidade de espaço em disco usada pelos arquivos no sistema de arquivos, em kilobytes.

Contador	Type	Unidade	Métrica	Descrição
Total	Sistema de arquivos	Kilobytes	os.fileSys.total	O número total de espaço disponível em disco para o sistema de arquivos, em kilobytes.
Rx	Rede	Bytes por segundo	os.network.rx	O número de bytes recebidos por segundo.
Tx	Rede	Bytes por segundo	os.network.tx	O número de bytes carregados por segundo.
Utilização de acu	Geral	Porcentagem	os.general.acuUtilization	A porcentagem de capacidade atual da capacidade máxima configurada.
Configuração máxima de acu	Geral	ACUs	os.general.maxConfiguredAcu	A capacidade máxima configurada pelo usuário, em unidades de capacidade do Aurora (ACUs).
Configuração mínima de acu	Geral	ACUs	os.general.minConfiguredAcu	A capacidade mínima configurada pelo usuário, em ACUs.

Contador	Type	Unidade	Métrica	Descrição
Número de vCPUs	Geral	vCPUs	os.general.numVCPUs	O número de CPUs virtuais (vCPUs) para a instância de banco de dados.
Capacidade de banco de dados sem servidor	Geral	ACUs	os.general.serverlessDatabaseCapacity	A capacidade atual da instância, em ACUs.

Contadores do Performance Insights para o Amazon RDS para MariaDB e MySQL

Os seguintes contadores de banco de dados estão disponíveis com o Performance Insights para Amazon RDS para MariaDB e MySQL.

Tópicos

- Contadores nativos para o RDS para MariaDB e o RDS para MySQL
- Contadores não nativos para o Amazon RDS para MariaDB e MySQL

Contadores nativos para o RDS para MariaDB e o RDS para MySQL

Métricas nativas são definidas pelo mecanismo de banco de dados e não pelo Amazon RDS. Para obter definições dessas métricas nativas, consulte Variáveis de status do servidor (para 8.0) e Variáveis de status do servidor (para 8.4) na documentação do MySQL.

Contador	Type	Unidade	Métrica
Com_analyze	SQL	Consultas por segundo	db.SQL.Com_analyze
Com_optimize	SQL	Consultas por segundo	db.SQL.Com_optimize

Contador	Type	Unidade	Métrica
Com_select	SQL	Consultas por segundo	db.SQL.Com_select
Conexões	SQL	O número de tentativas de conexão por minuto (bem-sucedidas ou não) para o servidor MySQL	db.Users.Connections
Innodb_rows_deleted	SQL	Linhas por segundo	db.SQL.Innodb_rows_deleted
Innodb_rows_inserted	SQL	Linhas por segundo	db.SQL.Innodb_rows_inserted
Innodb_rows_read	SQL	Linhas por segundo	db.SQL.Innodb_rows_read
Innodb_rows_updated	SQL	Linhas por segundo	db.SQL.Innodb_rows_updated
Select_full_join	SQL	Consultas por segundo	db.SQL.Select_full_join
Select_full_range_join	SQL	Consultas por segundo	db.SQL.Select_full_range_join
Select_range	SQL	Consultas por segundo	db.SQL.Select_range
Select_range_check	SQL	Consultas por segundo	db.SQL.Select_range_check

Contador	Type	Unidade	Métrica
Select_scan	SQL	Consultas por segundo	db.SQL.Select_scan
Slow_queries	SQL	Consultas por segundo	db.SQL.Slow_queries
Sort_merge_passes	SQL	Consultas por segundo	db.SQL.Sort_merge_passes
Sort_range	SQL	Consultas por segundo	db.SQL.Sort_range
Sort_rows	SQL	Consultas por segundo	db.SQL.Sort_rows
Sort_scan	SQL	Consultas por segundo	db.SQL.Sort_scan
Perguntas	SQL	Consultas por segundo	db.SQL.Questions
Innodb_row_lock_time	Travas	Milissegundos (média)	db.Locks.Innodb_row_lock_time
Table_locks_immediate	Travas	Solicitações por segundo	db.Locks.Table_locks_immediate
Table_locks_waited	Travas	Solicitações por segundo	db.Locks.Table_locks_waited
Aborted_clients	Usuários	Conexões	db.Users.Aborted_clients
Aborted_connects	Usuários	Conexões	db.Users.Aborted_connects
max_connections	Usuários	Conexões	db.User.max_connections
Threads_created	Usuários	Conexões	db.Users.Threads_created

Contador	Type	Unidade	Métrica
Threads_running	Usuários	Conexões	db.Users.Threads_running
Innodb_data_writes	E/S	Operações por segundo	db.IO.Innodb_data_writes
Innodb_dblwr_writes	I/O	Operações por segundo	db.IO.Innodb_dblwr_writes
Innodb_log_write_requests	I/O	Operações por segundo	db.IO.Innodb_log_write_requests
Innodb_log_writes	I/O	Operações por segundo	db.IO.Innodb_log_writes
Innodb_pages_written	I/O	Páginas por segundo	db.IO.Innodb_pages_written
Created_tmp_disk_tables	Temporário	Tabelas por segundo	db.Temp.Created_tmp_disk_tables
Created_tmp_tables	Temporário	Tabelas por segundo	db.Temp.Created_tmp_tables
Innodb_buffer_pool_pages_data	Cache	Páginas	db.Cache.Innodb_buffer_pool_pages_data
Innodb_buffer_pool_pages_total	Cache	Páginas	db.Cache.Innodb_buffer_pool_pages_total
Innodb_buffer_pool_read_requests	Cache	Páginas por segundo	db.Cache.Innodb_buffer_pool_read_requests
Innodb_buffer_pool_reads	Cache	Páginas por segundo	db.Cache.Innodb_buffer_pool_reads
Opened_tables	Cache	Tabelas	db.Cache.Opened_tables

Contador	Type	Unidade	Métrica
Opened_table_definitions	Cache	Tabelas	db.Cache.Opened_table_definitions
Qcache_hits	Cache	Consultas	db.Cache.Qcache_hits

Contadores não nativos para o Amazon RDS para MariaDB e MySQL

Métricas de contador não nativas são contadores definidos pelo Amazon RDS. Uma métrica não nativa pode ser uma métrica obtida com uma consulta específica. Uma métrica não nativa também pode ser uma métrica derivada, em que dois ou mais contadores nativos são usados em cálculos para proporções, taxas de ocorrência ou latências.

Contador	Type	Unidade	Métrica	Descrição	Definição
innodb_buffer_pool_hits	Cache	Leituras	db.Cache.innoDB_buffer_pool_hits	O número de leituras que o InnoDB pode atender no pool de buffer.	`innodb_buffer_pool_read_requests - innodb_buffer_pool_reads`
innodb_buffer_pool_hit_rate	Cache	Porcentagem	db.Cache.innoDB_buffer_pool_hit_rate	A porcentagem de leituras que o InnoDB pode atender no pool de buffer.	`100 * innodb_buffer_pool_read_requests / (innodb_buffer_pool_read_requests + innodb_buffer_pool_reads)`

Contador	Type	Unidade	Métrica	Descrição	Definição
innodb_buffer_pool_usage	Cache	Porcentagem	db.Cache.innoDB_buffer_pool_usage	A porcentagem do pool de buffers do InnoDB que contém dados (páginas). **Note** Ao usar tabelas compactas, esse valor pode variar. Para obter mais informações, consulte as informações sobre Innodb_buffer_	`Innodb_buffer_pool_pages_data / Innodb_buffer_pool_pages_total * 100.0`

Contador	Type	Unidade	Métrica	Descrição	Definição
				page ta e Innod ffer _page tal em Variáv de status do servid (para 8.0) e Variáv de status do servid (para 8.4) na docum ção do MySQ	

Contador	Type	Unidade	Métrica	Descrição	Definição
query_cache_hit_rate	Cache	Porcentagem	db.Cache.query_cache_hit_rate	Índice de acertos de cache do conjunto de resultados do MySQL (cache de consulta).	`Qcache_hits / (QCache_hits + Com_select) * 100`
innodb_datafile_writes_to_disk	E/S	Gravaçõe	db.IO.innoDB_datafile_writes_to_disk	O número de gravações de arquivos de dados do InnoDB em disco, excluindo operações de gravação de duplas e de gravação em logs redo.	`Innodb_data_writes - Innodb_log_writes - Innodb_dblwr_writes`

Contador	Type	Unidade	Métrica	Descrição	Definição
innodb_rows_changed	SQL	Linhas	db.SQL.innodb_rows_changed	O total de operações de linhas do InnoDB.	`db.SQL.Innodb_rows_inserted + db.SQL.Innodb_rows_deleted + db.SQL.Innodb_rows_updated`
active_transactions	Transaçõ s	Transaçõ s	db.Transactions.active_transactions	O total de transaçõe s ativas.	`SELECT COUNT(1) AS active_transactions FROM INFORMATION_SCHEMA.INNODB_TRX`

Contador	Type	Unidade	Métrica	Descrição	Definição
trx_rseg_history_len	Transaçõ s	Nenhum	db.Transactions.trx_rseg_history_len	Uma lista das páginas de undo log de transaçõe s confirmad as que é mantida pelo sistema de transaçõe s InnoDB para implement ar o controle de simultane idade de várias versões. Para obter mais informaçõ es sobre os detalhes para desfazer os registros	`SELECT COUNT AS trx_rseg_ history_len FROM INFORMATI ON_SCHEMA .INNODB_M ETRICS WHERE NAME='trx _rseg_his tory_len'`

Contador	Type	Unidade	Métrica	Descrição	Definição
				de log, consulte InnoDB Multi-Versioning (para 8.0) e InnoDB Multi-Versioning (para 8.4) na documentação do MySQL.	
innodb_deadlocks	Travas	Travas	db.Locks.innodb_deadlocks	O número total de deadlocks.	`SELECT COUNT AS innodb_deadlocks FROM INFORMATION_SCHEMA.INNODB_METRICS WHERE NAME='lock_deadlocks'`
innodb_lock_timeouts	Travas	Travas	db.Locks.innodb_lock_timeouts	O número total de bloqueios que expiraram.	`SELECT COUNT AS innodb_lock_timeouts FROM INFORMATION_SCHEMA.INNODB_METRICS WHERE NAME='lock_timeouts'`

Contador	Type	Unidade	Métrica	Descrição	Definição
innodb_row_lock_waits	Travas	Travas	db.Locks.innodb_row_lock_waits	O número total de bloqueios de linha que resultaram em uma espera.	`SELECT COUNT AS innodb_row_lock_waits FROM INFORMATION_SCHEMA.INNODB_METRICS WHERE NAME='lock_row_lock_waits'`

Contadores do Performance Insights para o Amazon RDS para Microsoft SQL Server

Os contadores de banco de dados a seguir estão disponíveis para o Performance Insights para o RDS for Microsoft SQL Server.

Contadores nativos do RDS for Microsoft SQL Server

Métricas nativas são definidas pelo mecanismo de banco de dados e não pelo Amazon RDS. Você pode encontrar definições para essas métricas nativas em Usar objetos do SQL Server, na documentação do Microsoft SQL Server.

Contador	Type	Unidade	Métrica
Registros encaminhados	Métodos de acesso	Registros por segundo	db.Access Methods.Forwarded Records
Divisões de página	Métodos de acesso	Divisões por segundo	db.Access Methods.Page Splits
Proporção de acertos de cache de buffer	Gerenciador de buffer	Proporção	Proporção de acertos do cache de db.Buffer Manager.Buffer

Contador	Type	Unidade	Métrica
Expectativa de duração de páginas	Gerenciador de buffer	Expectativa em segundos	Expectativa de vida de db.Buffer Manager.Page
Consultas de página	Gerenciador de buffer	Consultas por segundo	db.Buffer Manager.Page lookups
Leituras de página	Gerenciador de buffer	Leituras por segundo	db.Buffer Manager.Page reads
Gravações de página	Gerenciador de buffer	Gravações por segundo	db.Buffer Manager.Page writes
Transações ativas	Bancos de dados	Transações	db.Databases.Active Transactions (_Total)
Bytes de log liberados	Bancos de dados	Bytes liberados por segundo	db.Databases.Log Bytes Flushed (_Total)
Esperas de liberação de log	Bancos de dados	Esperas por segundo	db.Databases.Log Flush Waits (_Total)
Liberações de log	Bancos de dados	Liberações por segundo	db.Databases.Log Flushes (_Total)
Transações de gravação	Bancos de dados	Transações por segundo	db.Databases.Write Transactions (_Total)
Processos bloqueados	Estatísticas gerais	Processos bloqueados	db.General Statistics.Processes blocked
Conexões de usuários	Estatísticas gerais	Conexões	db.General Statistics.User Connections
Esperas de travamento	Travamentos	Esperas por segundo	db.Latches.Latch Waits

Contador	Type	Unidade	Métrica
Número de deadlocks	Travas	Deadlocks por segundo	db.Locks.Number of Deadlocks (_Total)
Concessões de memória pendentes	Gerenciador de memória	Concessões de memória	Concessões pendentes para db.Memory Manager.Memory
Solicitações em lote	Estatísticas SQL	Solicitações por segundo	db.SQL Statistic s.Batch Requests
Compilações SQL	Estatísticas SQL	Compilações por segundo	db.SQL Statistics.SQL Compilations
Recompilações SQL	Estatísticas SQL	Recompilações por segundo	db.SQL Statistics.SQL Re-Compilations

Contadores do Performance Insights para o Amazon RDS para Oracle

Os contadores de banco de dados a seguir estão disponíveis para o Performance Insights para o RDS for Oracle.

Contadores nativos para o RDS for Oracle

Métricas nativas são definidas pelo mecanismo de banco de dados e não pelo Amazon RDS. Você pode encontrar definições para essas métricas nativas em Statistics Descriptions (Descrições de estatísticas) na documentação do Oracle.

> **Note**
>
> Para a métrica de contador `CPU used by this session`, a unidade foi transformada dos centésimos de segundo nativos para sessões ativas a fim de facilitar o uso do valor. Por exemplo, o envio de CPU no gráfico de carga do banco de dados representa a demanda por CPU. A métrica de contador `CPU used by this session` representa a quantidade de CPU usada por sessões do Oracle. É possível comparar o envio de CPU à métrica de

contador `CPU used by this session`. Quando a demanda por CPU for maior que a CPU usada, as sessões aguardarão o tempo da CPU.

Contador	Type	Unidade	Métrica
CPU usada por esta sessão	Usuário	Sessões ativas	db.User.CPU usada por esta sessão
Rotas de ida e volta do SQL*Net para/do cliente	Usuário	Idas e voltas por segundo	Rotas de ida e volta do db.User.SQL*Net para/do cliente
Bytes recebidos via SQL*Net do cliente	Usuário	Bytes por segundo	db.User.bytes recebidos pelo SQL*Net do cliente
Confirmações de usuários	Usuário	Confirmações por segundo	Confirmações de db.User.user
Logons cumulativos	Usuário	Logons por segundo	db.User.logons cumulativos
Chamadas de usuários	Usuário	Chamadas por segundo	Chamadas de db.User.user
Bytes enviados via SQL*Net ao cliente	Usuário	Bytes por segundo	db.User.bytes enviados pelo SQL*Net ao cliente
Reversões de usuários	Usuário	Reversões por segundo	Reversões de db.User.user
Tamanho do redo	Redo	Bytes por segundo	Tamanho de db.Redo.redo
Contagem de análises (total)	SQL	Análises por segundo	Contagem de db.SQL.parse (total)

Contador	Type	Unidade	Métrica
Contagem de análises (fixa)	SQL	Análises por segundo	Contagem de db.SQL.parse (fixa)
Linhas de varredura de tabela obtidas	SQL	Linhas por segundo	Linhas de varredura de db.SQL.table obtidas
Classificações (memória)	SQL	Classificações por segundo	db.SQL.sorts (memória)
Classificações (disco)	SQL	Classificações por segundo	db.SQL.sorts (disco)
Classificações (linhas)	SQL	Classificações por segundo	db.SQL.sorts (linhas)
Bytes de leitura física	Cache	Bytes por segundo	Bytes de leitura de db.Cache.physical
Obtenções de bloqueios de banco de dados	Cache	Blocos por segundo	Obtenções de blocos de db.Cache.db
Pontos de verificação DBWR	Cache	Pontos de verificação por minuto	Pontos de verificação de db.Cache.DBWR
Leituras físicas	Cache	Leituras por segundo	Leituras de db.Cache. physical
Obtenções consisten tes do cache	Cache	Obtenções por segundo	Obtenções consisten tes de db.Cache. consistent do cache
Obtenções de bloqueios de banco de dados do cache	Cache	Obtenções por segundo	Obtenções de blocos de db.Cache.db do cache

Contador	Type	Unidade	Métrica
Obtenções consistentes	Cache	Obtenções por segundo	Obtenções consistentes de db.Cache.consistent

Contadores do Performance Insights para o Amazon RDS para PostgreSQL

Os seguintes contadores de banco de dados estão disponíveis com o Performance Insights para Amazon RDS para PostgreSQL.

Tópicos

- Contadores nativos para o Amazon RDS para PostgreSQL
- Contadores não nativos para o Amazon RDS para PostgreSQL

Contadores nativos para o Amazon RDS para PostgreSQL

Métricas nativas são definidas pelo mecanismo de banco de dados e não pelo Amazon RDS. É possível encontrar definições dessas métricas em Visualizar estatísticas (em inglês) na documentação do PostgreSQL.

Contador	Type	Unidade	Métrica
blks_hit	Cache	Blocos por segundo	db.Cache.blks_hit
buffers_alloc	Cache	Blocos por segundo	db.Cache.buffers_alloc
buffers_checkpoint	Ponto de verificação	Blocos por segundo	db.Checkpoint.buffers_checkpoint
checkpoint_sync_time	Ponto de verificação	Milissegundos por ponto de verificação	db.Checkpoint.checkpoint_sync_time

Contador	Type	Unidade	Métrica
checkpoint_write_time	Ponto de verificação	Milissegundos por ponto de verificação	db.Checkpoint.checkpoint_write_time
checkpoints_req	Ponto de verificação	Pontos de verificação por minuto	db.Checkpoint.checkpoints_req
checkpoints_timed	Ponto de verificação	Pontos de verificação por minuto	db.Checkpoint.checkpoints_timed
maxwritten_clean	Ponto de verificação	Paradas de limpeza de Bgwriter por minuto	db.Checkpoint.maxwritten_clean
deadlocks	Simultaneidade	Deadlocks por minuto	db.Concurrency.deadlocks
blk_read_time	I/O	Milissegundos	db.IO.blk_read_time
blks_read	I/O	Blocos por segundo	db.IO.blks_read
buffers_backend	I/O	Blocos por segundo	db.IO.buffers_backend
buffers_backend_fsync	I/O	Blocos por segundo	db.IO.buffers_backend_fsync
buffers_clean	I/O	Blocos por segundo	db.IO.buffers_clean
tup_deleted	SQL	Tuplas por segundo	db.SQL.tup_deleted
tup_fetched	SQL	Tuplas por segundo	db.SQL.tup_fetched

Contador	Type	Unidade	Métrica
tup_inserted	SQL	Tuplas por segundo	db.SQL.tup_inserted
tup_returned	SQL	Tuplas por segundo	db.SQL.tup_returned
tup_updated	SQL	Tuplas por segundo	db.SQL.tup_updated
temp_bytes	Temporário	Bytes por segundo	db.Temp.temp_bytes
temp_files	Temporário	Arquivos por minuto	db.Temp.temp_files
xact_commit	Transações	Confirmações por segundo	db.Transactions.xact_commit
xact_rollback	Transações	Reversões por segundo	db.Transactions.xact_rollback
numbackends	Usuário	Conexões	db.User.numbackends
archived_count	Log de gravação antecipada (WAL)	Arquivos por minuto	db.WAL.archived_count

Contadores não nativos para o Amazon RDS para PostgreSQL

Métricas de contador não nativas são contadores definidos pelo Amazon RDS. Uma métrica não nativa pode ser uma métrica obtida com uma consulta específica. Uma métrica não nativa também pode ser uma métrica derivada, em que dois ou mais contadores nativos são usados em cálculos para proporções, taxas de ocorrência ou latências.

Contador	Type	Unidad	Métrica	Descrição	Definição
checkpoint_sync_latency	Ponto de verificação	db.Checkpoint.checkpoint_sync_latency		O tempo que foi gasto na parte do processamento de ponto de verificação em que os arquivos são sincronizados no disco.	`checkpoint_sync_time / (checkpoints_timed + checkpoints_req)`
checkpoint_write_latency	Ponto de verificação	db.Checkpoint.checkpoint_write_latency	O tempo que foi gasto na parte do processamento de ponto de verificação em que os arquivos são gravados no disco.	`checkpoint_write_time / (checkpoints_timed + checkpoints_req)`	
read_latency	I/O	db.IO.rd_later	O tempo gasto lendo blocos de arquivos de dados pelos backends nesta instância.	`blk_read_time / blks_read`	
idle_in_transaction_aborted_count	Estado	Sessõ	db.state.idle_in_transaction_aborted_count	O número de sessões no estado `idle in transaction (aborted)` .	Não aplicável
idle_in_transaction_count	Estado	Sessõ	db.state.idle_in_transaction_count	O número de sessões no estado `idle in transaction` .	Não aplicável

Contador	Type	Unidad	Métrica	Descrição	Definição
idle_in_transaction_max_time	Estado	Segun	db.state.idle_in_transaction_max_time	A duração da transação mais longa no estado `idle in transaction` , em segundos.	Não aplicável
active_transactions	Transa s	Transa s	db.Transactions.active_transactions	O número de transações ativas.	Não aplicável
blocked_transactions	Transa s	Transa s	db.Transactions.blocked_transactions	O número de transações bloqueadas.	Não aplicável
oldest_active_logical_replication_slot_xid_age	Transa s	db.Transactions.dest_active_logl_replication_sl_xid_a	A idade da transação mais antiga em um slot de replicação lógica ativo. Para obter mais informações, consulte Slot de replicação lógica.	–	
oldest_inactive_logical_replication_slot_xid_age	Transa s	db.Transactions.dest_inactive_local_replication_ot_xid_e	A idade da transação mais antiga em um slot de replicação lógica inativo. Para obter mais informações, consulte Slot de replicação lógica.	–	

Contador	Type	Unidad	Métrica	Descrição	Definição
oldest_prepared_transaction_xid_age	Transas	db.transactions.oldest_prepared_transaction_id_age	A idade da transação preparada mais antiga. Para obter mais informações, consulte Transação preparada.	–	
oldest_running_transaction_xid_age	Transas	db.Transactions.oldest_running_transaction_id_age	A idade da transação em execução mais antiga. Para obter mais informações, consulte Instrução ativa para a transação ativa em execução mais antiga e Ocioso na transação para a transação ociosa em execução mais antiga.	–	
oldest_hot_standby_feedback_xid_age	Transas	db.Transactions.oldest_hot_standby_feedback_id_age	A idade da transação em execução mais antiga em uma réplica de leitura com `hot_standby_feedback` habilitado. Para obter mais informações, consulte Réplicas de leitura.	–	

Contador	Type	Unidad	Métrica	Descrição	Definição
max_used_xact_ids	Transa s	Transa s	db.Transactions.max_used_xact_ids	O número de transações que não foram apagadas.	Não aplicável
max_connections	Usuári	Conex	db.User.max_connections	O número máximo de conexões permitido para uma instância de banco de dados, conforme configurado no parâmetro `max_connections` .	Não aplicável
archive_failed_count	WAL	Arquiv por minuto	db.WAL.archive_failed_count	O número de tentativas malsucedidas de arquivamento de arquivos WAL, em arquivos por minuto.	Não aplicável

Estatísticas SQL para Performance Insights

Estatísticas SQL são métricas de performance sobre consultas SQL coletadas pelo Performance Insights. O Performance Insights reúne estatísticas para cada segundo em que uma consulta está sendo executada e para cada chamada SQL. As estatísticas do SQL são uma média para o intervalo de tempo selecionado.

Um resumo SQL é um composto de todas as consultas com determinado padrão, mas não necessariamente com os mesmos valores literais. O resumo substitui valores literais por um ponto de interrogação. Por exemplo, `SELECT * FROM emp WHERE lname= ?`. Esse resumo pode consistir nas seguintes consultas subordinadas:

```
SELECT * FROM emp WHERE lname = 'Sanchez'
SELECT * FROM emp WHERE lname = 'Olagappan'
SELECT * FROM emp WHERE lname = 'Wu'
```

Todos os mecanismos são compatíveis com estatísticas de consultas de resumo.

Para receber informações sobre a compatibilidade da região, do mecanismo de banco de dados e da classe de instância com esse atributo, consulte O mecanismo de banco de dados do Amazon RDS, a região e a classe de instância são compatíveis com atributos do Insights de Performance..

Tópicos

- Estatísticas SQL para o MariaDB e o MySQL
- Estatísticas de SQL para o Amazon RDS para Oracle
- Estatísticas de SQL para o Amazon RDS para SQL Server
- Estatísticas SQL do RDS PostgreSQL

Estatísticas SQL para o MariaDB e o MySQL

O MariaDB e o MySQL coletam estatísticas SQL somente em nível de resumo. Nenhuma estatística é mostrada em nível de instrução.

Tópicos

- Estatísticas de resumo para MariaDB e MySQL
- Estatísticas por segundo para o MariaDB e o MySQL
- Estatísticas por chamada para o MariaDB e o MySQL
- Principais estatísticas para MariaDB e MySQL

Estatísticas de resumo para MariaDB e MySQL

O Performance Insights coleta estatísticas de resumo do SQL da tabela `events_statements_summary_by_digest`. A tabela `events_statements_summary_by_digest` é gerenciada pelo seu banco de dados.

Ela não tem uma política de evicção. Quando a tabela estiver cheia, o AWS Management Console mostrará a seguinte mensagem:

```
Performance Insights is unable to collect SQL Digest statistics on new queries because
 the table events_statements_summary_by_digest is full.
Please truncate events_statements_summary_by_digest table to clear the issue. Check the
 User Guide for more details.
```

Nessa situação, o MariaDB e o MySQL não rastreiam consultas SQL. Para resolver esse problema, o Performance Insights trunca automaticamente a tabela de resumo quando ambas as condições são satisfeitas:

- A tabela está cheia.
- O Performance Insights gerencia o Performance Schema automaticamente.

  Para gerenciamento automático, o parâmetro `performance_schema` deve ser definido como `0` e Source (Origem) não deve ser definido como `user`. Se o Performance Insights não estiver gerenciando o Performance Schema automaticamente, consulte Visão geral do Performance Schema para o Insights de Performance no Amazon RDS para MariaDB ou MySQL.

Na AWS CLI, verifique a origem de um valor de parâmetro executando o comando describe-db-parameters.

Estatísticas por segundo para o MariaDB e o MySQL

As seguintes estatísticas do SQL estão disponíveis para instâncias de banco de dados do MariaDB e MySQL

Métrica	Unidade
db.sql_tokenized.stats.count_star_per_sec	Chamadas por segundo
db.sql_tokenized.stats.sum_timer_wait_per_sec	Latência média por segundo (em ms)
db.sql_tokenized.stats.sum_select_full_join_per_sec	Selecionar junção completa por segundo
db.sql_tokenized.stats.sum_select_range_check_per_sec	Selecionar verificação de intervalo por segundo
db.sql_tokenized.stats.sum_select_scan_per_sec	Selecionar verificação por segundo

Métrica	Unidade
db.sql_tokenized.stats.sum_sort_merge_passes_per_sec	Classificar passagens de mesclagem por segundo
db.sql_tokenized.stats.sum_sort_scan_per_sec	Classificar verificações por segundo
db.sql_tokenized.stats.sum_sort_range_per_sec	Classificar intervalos por segundo
db.sql_tokenized.stats.sum_sort_rows_per_sec	Classificar linhas por segundo
db.sql_tokenized.stats.sum_rows_affected_per_sec	Linhas afetadas por segundo
db.sql_tokenized.stats.sum_rows_examined_per_sec	Linhas examinadas por segundo
db.sql_tokenized.stats.sum_rows_sent_per_sec	Linhas enviadas por segundo
db.sql_tokenized.stats.sum_created_tmp_disk_tables_per_sec	Tabelas de disco temporárias criadas por segundo
db.sql_tokenized.stats.sum_created_tmp_tables_per_sec	Tabelas temporárias criadas por segundo
db.sql_tokenized.stats.sum_lock_time_per_sec	Tempo de bloqueio por segundo (em ms)

Estatísticas por chamada para o MariaDB e o MySQL

As métricas a seguir fornecem estatísticas por chamada para uma instrução SQL.

Métrica	Unidade
db.sql_tokenized.stats.sum_timer_wait_per_call	Latência média por chamada (em ms)
db.sql_tokenized.stats.sum_select_full_join_per_call	Selecionar junções completas por chamada

Métrica	Unidade
db.sql_tokenized.stats.sum_select_range_check_per_call	Selecionar verificação de intervalo por chamada
db.sql_tokenized.stats.sum_select_scan_per_call	Selecionar verificações por chamada
db.sql_tokenized.stats.sum_sort_merge_passes_per_call	Classificar passagens de mesclagem por chamada
db.sql_tokenized.stats.sum_sort_scan_per_call	Classificar verificações por chamada
db.sql_tokenized.stats.sum_sort_range_per_call	Classificar intervalos por chamada
db.sql_tokenized.stats.sum_sort_rows_per_call	Classificar linhas por chamada
db.sql_tokenized.stats.sum_rows_affected_per_call	Linhas afetadas por chamada
db.sql_tokenized.stats.sum_rows_examined_per_call	Linhas examinadas por chamada
db.sql_tokenized.stats.sum_rows_sent_per_call	Linhas enviadas por chamada
db.sql_tokenized.stats.sum_created_tmp_disk_tables_per_call	Tabelas de disco temporárias criadas por chamada
db.sql_tokenized.stats.sum_created_tmp_tables_per_call	Tabelas temporárias criadas por chamada
db.sql_tokenized.stats.sum_lock_time_per_call	Tempo de leitura por chamada (em ms)

Principais estatísticas para MariaDB e MySQL

As seguintes estatísticas do SQL estão disponíveis para instâncias de banco de dados do MariaDB e MySQL

Métrica	Unidade
db.sql_tokenized.stats.count_star	Calls
db.sql_tokenized.stats.sum_timer_wait	Tempo de espera (em ms)
db.sql_tokenized.stats.sum_select_full_join	Selecionar junção completa
db.sql_tokenized.stats.sum_select_range_check	Selecionar verificações de intervalo
db.sql_tokenized.stats.sum_select_scan	Selecionar verificações
db.sql_tokenized.stats.sum_sort_merge_passes	Classificar passagens de mesclagem
db.sql_tokenized.stats.sum_sort_scan	Classificar verificações
db.sql_tokenized.stats.sum_sort_range	Classificar intervalos
db.sql_tokenized.stats.sum_sort_rows	Classificar linhas
db.sql_tokenized.stats.sum_rows_affected	Linhas afetadas
db.sql_tokenized.stats.sum_rows_examined	Linhas examinadas
db.sql_tokenized.stats.sum_rows_sent	Linhas enviadas
db.sql_tokenized.stats.sum_created_tmp_disk_tables	Tabelas de disco temporárias criadas
db.sql_tokenized.stats.sum_created_tmp_tables	Tabelas temporárias criadas
db.sql_tokenized.stats.sum_lock_time	Tempo de bloqueio (em ms)

Estatísticas de SQL para o Amazon RDS para Oracle

O Amazon RDS para Oracle coleta estatísticas SQL tanto em nível de instrução quanto em nível de resumo. Em nível de instrução, a coluna ID representa o valor de `V$SQL.SQL_ID`. Em nível de resumo, a coluna ID mostra o valor de `V$SQL.FORCE_MATCHING_SIGNATURE`.

Se o ID for `0` em nível de resumo, significa que o Oracle Database determinou que essa instrução não é apropriada para reutilização. Nesse caso, as instruções SQL subordinadas podem pertencer a resumos diferentes. Entretanto, as instruções são agrupadas sob `digest_text` para a primeira instrução SQL coletada.

Tópicos

- Estatísticas por segundo para o Oracle
- Estatísticas por chamada para o Oracle
- Principais estatísticas do Oracle

Estatísticas por segundo para o Oracle

As métricas a seguir fornecem estatísticas por segundo referentes a uma consulta SQL do Oracle.

Métrica	Unidade
db.sql.stats.executions_per_sec	Número de execuções por segundo
db.sql.stats.elapsed_time_per_sec	Média de execuções ativas (AAE)
db.sql.stats.rows_processed_per_sec	Linhas processadas por segundo
db.sql.stats.buffer_gets_per_sec	Obtenções de buffer por segundo
db.sql.stats.physical_read_requests_per_sec	Leituras físicas por segundo
db.sql.stats.physical_write_requests_per_sec	Gravações físicas por segundo
db.sql.stats.total_sharable_mem_per_sec	Memória compartilhável total por segundo (em bytes)
db.sql.stats.cpu_time_per_sec	Tempo de CPU por segundo (em ms)

As métricas a seguir fornecem estatísticas por segundo referentes a uma consulta de SQL resumo do Oracle.

Métrica	Unidade
db.sql_tokenized.stats.executions_per_sec	Número de execuções por segundo
db.sql_tokenized.stats.elapsed_time_per_sec	Média de execuções ativas (AAE)
db.sql_tokenized.stats.rows_processed_per_sec	Linhas processadas por segundo
db.sql_tokenized.stats.buffer_gets_per_sec	Obtenções de buffer por segundo
db.sql_tokenized.stats.physical_read_requests_per_sec	Leituras físicas por segundo
db.sql_tokenized.stats.physical_write_requests_per_sec	Gravações físicas por segundo
db.sql_tokenized.stats.total_sharable_mem_per_sec	Memória compartilhável total por segundo (em bytes)
db.sql_tokenized.stats.cpu_time_per_sec	Tempo de CPU por segundo (em ms)

Estatísticas por chamada para o Oracle

As métricas a seguir fornecem estatísticas por chamada referentes a uma instrução SQL do Oracle.

Métrica	Unidade
db.sql.stats.elapsed_time_per_exec	Tempo decorrido por execuções (em ms)
db.sql.stats.rows_processed_per_exec	Linhas processadas por execução
db.sql.stats.buffer_gets_per_exec	Obtenções de buffer por execução
db.sql.stats.physical_read_requests_per_exec	Leituras físicas por execução
db.sql.stats.physical_write_requests_per_exec	Gravações físicas por execução

Métrica	Unidade
db.sql.stats.total_sharable_mem_per_exec	Memória compartilhável total por execução (em bytes)
db.sql.stats.cpu_time_per_exec	Tempo de CPU por execução (em ms)

As métricas a seguir fornecem estatísticas por chamada referentes a uma consulta de resumo SQL do Oracle.

Métrica	Unidade
db.sql_tokenized.stats.elapsed_time_per_exec	Tempo decorrido por execuções (em ms)
db.sql_tokenized.stats.rows_processed_per_exec	Linhas processadas por execução
db.sql_tokenized.stats.buffer_gets_per_exec	Obtenções de buffer por execução
db.sql_tokenized.stats.physical_read_requests_per_exec	Leituras físicas por execução
db.sql_tokenized.stats.physical_write_requests_per_exec	Gravações físicas por execução
db.sql_tokenized.stats.total_sharable_mem_per_exec	Memória compartilhável total por execução (em bytes)
db.sql_tokenized.stats.cpu_time_per_exec	Tempo de CPU por execução (em ms)

Principais estatísticas do Oracle

As métricas a seguir fornecem as principais estatísticas referentes a uma consulta SQL do Oracle.

Métrica	Unidade
db.sql.stats.executions	Número de execuções

Métrica	Unidade
db.sql.stats.elapsed_time	Tempo decorrido (em ms)
db.sql.stats.rows_processed	Linhas processadas
db.sql.stats.buffer_gets	Obtenções de buffer
db.sql.stats.physical_read_requests	Leituras físicas
db.sql.stats.physical_write_requests	Gravações físicas
db.sql.stats.total_sharable_mem	Memória compartilhável total (em bytes)
db.sql.stats.cpu_time	Tempo de CPU (em ms)

As métricas a seguir fornecem as principais estatísticas referentes a uma consulta de resumo SQL do Oracle.

Métrica	Unidade
db.sql_tokenized.stats.executions	Número de execuções
db.sql_tokenized.stats.elapsed_time	Tempo decorrido (em ms)
db.sql_tokenized.stats.rows_processed	Linhas processadas
db.sql_tokenized.stats.buffer_gets	Obtenções de buffer
db.sql_tokenized.stats.physical_read_requests	Leituras físicas
db.sql_tokenized.stats.physical_write_requests	Gravações físicas
db.sql_tokenized.stats.total_sharable_mem	Memória compartilhável total (em bytes)
db.sql_tokenized.stats.cpu_time	Tempo de CPU (em ms)

Estatísticas de SQL para o Amazon RDS para SQL Server

O Amazon RDS para SQL Server coleta estatísticas SQL tanto em nível de instrução quanto em nível de resumo. Em nível de instrução, a coluna ID representa o valor de `sql_handle`. Em nível de resumo, a coluna ID mostra o valor de `query_hash`.

O SQL Server retorna valores NULL para `query_hash` para algumas instruções. Por exemplo, ALTER INDEX, CHECKPOINT, UPDATE STATISTICS, COMMIT TRANSACTION, FETCH NEXT FROM Cursor e algumas instruções INSERT, SELECT @<variable>, instruções condicionais e procedimentos armazenados executáveis. Nesse caso, o valor `sql_handle` é exibido como o ID em nível de resumo dessa instrução.

Tópicos

- Estatísticas por segundo para o SQL Server
- Estatísticas por chamada para o SQL Server
- Principais estatísticas para o SQL Server

Estatísticas por segundo para o SQL Server

As métricas a seguir fornecem estatísticas por segundo referentes a uma consulta SQL do SQL Server.

Métrica	Unidade
db.sql.stats.execution_count_per_sec	Número de execuções por segundo
db.sql.stats.total_elapsed_time_per_sec	Tempo total decorrido por segundo
db.sql.stats.total_rows_per_sec	Total de linhas processadas por segundo
db.sql.stats.total_logical_reads_per_sec	Total de leituras lógicas por segundo
db.sql.stats.total_logical_writes_per_sec	Total de gravações lógicas por segundo
db.sql.stats.total_physical_reads_per_sec	Total de leituras físicas por segundo
db.sql.stats.total_worker_time_per_sec	Tempo total de CPU (em ms)

As métricas a seguir fornecem estatísticas por segundo referentes a uma consulta SQL de resumo do SQL Server.

Métrica	Unidade
db.sql_tokenized.stats.execution_count_per_sec	Número de execuções por segundo
db.sql_tokenized.stats.total_elapsed_time_per_sec	Tempo total decorrido por segundo
db.sql_tokenized.stats.total_rows_per_sec	Total de linhas processadas por segundo
db.sql_tokenized.stats.total_logical_reads_per_sec	Total de leituras lógicas por segundo
db.sql_tokenized.stats.total_logical_writes_per_sec	Total de gravações lógicas por segundo
db.sql_tokenized.stats.total_physical_reads_per_sec	Total de leituras físicas por segundo
db.sql_tokenized.stats.total_worker_time_per_sec	Tempo total de CPU (em ms)

Estatísticas por chamada para o SQL Server

As métricas a seguir fornecem estatísticas por chamada para uma instrução SQL do SQL Server.

Métrica	Unidade
db.sql.stats.total_elapsed_time_per_call	Tempo total decorrido por execução (em ms)
db.sql.stats.total_rows_per_call	Total de linhas processadas por execução
db.sql.stats.total_logical_reads_per_call	Total de leituras lógicas por execução
db.sql.stats.total_logical_writes_per_call	Total de gravações lógicas por execução

Métrica	Unidade
db.sql.stats.total_physical_reads_per_call	Total de leituras físicas por execução
db.sql.stats.total_worker_time_per_call	Total de tempo de CPU por execução (em ms)

As métricas a seguir fornecem estatísticas por chamada referentes a uma consulta SQL de resumo do SQL Server.

Métrica	Unidade
db.sql_tokenized.stats.total_elapsed_time_per_call	Tempo total decorrido por execução
db.sql_tokenized.stats.total_rows_per_call	Total de linhas processadas por execução
db.sql_tokenized.stats.total_logical_reads_por_call	Total de leituras lógicas por execução
db.sql_tokenized.stats.total_logical_writes_per_call	Total de gravações lógicas por execução
db.sql_tokenized.stats.total_physical_reads_por_call	Total de leituras físicas por execução
db.sql_tokenized.stats.total_worker_time_per_call	Total de tempo de CPU por execução (em ms)

Principais estatísticas para o SQL Server

As métricas a seguir fornecem as principais estatísticas referentes a uma consulta SQL do SQL Server.

Métrica	Unidade
db.sql.stats.execution_count	Número de execuções
db.sql.stats.total_elapsed_time	Tempo total decorrido (em ms)

Métrica	Unidade
db.sql.stats.total_rows	Total de linhas processadas
db.sql.stats.total_logical_reads	Total de leituras lógicas
db.sql.stats.total_logical_writes	Total de gravações lógicas
db.sql.stats.total_physical_reads	Total de leituras lógicas
db.sql.stats.total_worker_time	Tempo total de CPU (em ms)

As métricas a seguir fornecem as principais estatísticas referentes a uma consulta de resumo SQL do SQL Server.

Métrica	Unidade
db.sql_tokenized.stats.execution_count	Número de execução
db.sql_tokenized.stats.total_elapsed_time	Tempo total decorrido (em ms)
db.sql_tokenized.stats.total_rows	Total de linhas processadas
db.sql_tokenized.stats.total_logical_reads	Total de leituras lógicas
db.sql_tokenized.stats.total_logical_writes	Total de gravações lógicas
db.sql_tokenized.stats.total_physical_reads	Total de leituras lógicas
db.sql_tokenized.stats.total_worker_time	Tempo total de CPU (em ms)

Estatísticas SQL do RDS PostgreSQL

Para cada chamada SQL e para cada segundo em que uma consulta é executada, o Performance Insights coleta estatísticas SQL. O Aurora PostgreSQL coleta estatísticas SQL somente em nível de resumo. Nenhuma estatística é mostrada em nível de instrução.

A seguir, você pode encontrar informações sobre estatísticas no nível do resumo para RDS para PostgreSQL.

Tópicos

- Estatísticas de resumo do RDSPostgreSQL
- Estatísticas de resumo por segundo para o RDS PostgreSQL
- Estatísticas de resumo por chamada para o RDS PostgreSQL
- Principais estatísticas do RDS PostgreSQL

Estatísticas de resumo do RDSPostgreSQL

Para visualizar estatísticas de resumo SQL, o RDS PostgreSQL deve carregar a biblioteca `pg_stat_statements`. Para instâncias de banco de dados do PostgreSQL compatíveis com o PostgreSQL 11 ou versões posteriores, o banco de dados carrega essa biblioteca por padrão. Para instâncias de banco de dados PostgreSQL compatíveis com PostgreSQL 10 ou versões anteriores, essa biblioteca é habilitada manualmente. Para ativá-la manualmente, adicione `pg_stat_statements` a `shared_preload_libraries` no grupo de parâmetros de banco de dados associado à instância de banco de dados. Em seguida, reinicialize sua instância de banco de dados. Para obter mais informações, consulte Grupos de parâmetros para Amazon RDS.

> **Note**
>
> O Performance Insights só pode coletar estatísticas em relação a consultas não truncadas em `pg_stat_activity`. Por padrão, os bancos de dados PostgreSQL truncam consultas com mais de 1.024 bytes. Para aumentar o tamanho das consultas, altere o parâmetro `track_activity_query_size` no grupo de parâmetros de banco de dados associado à sua instância de banco de dados. Ao alterar esse parâmetro, uma reinicialização da instância de banco de dados é necessária.

Estatísticas de resumo por segundo para o RDS PostgreSQL

As seguintes estatísticas de resumo SQL estão disponíveis para instância de banco de dados PostgreSQL.

Métrica	Unidade
db.sql_tokenized.stats.calls_per_sec	Chamadas por segundo
db.sql_tokenized.stats.rows_per_sec	Linhas por segundo

Métrica	Unidade
db.sql_tokenized.stats.total_time_per_sec	Média de execuções ativas por segundo (AAE)
db.sql_tokenized.stats.shared_blks_hit_per_sec	Acertos de bloco por segundo
db.sql_tokenized.stats.shared_blks_read_per_sec	Leituras de bloco por segundo
db.sql_tokenized.stats.shared_blks_dirtied_per_sec	Blocos sujos por segundo
db.sql_tokenized.stats.shared_blks_written_per_sec	Gravações de bloco por segundo
db.sql_tokenized.stats.local_blks_hit_per_sec	Acertos de bloco local por segundo
db.sql_tokenized.stats.local_blks_read_per_sec	Leituras de bloco local por segundo
db.sql_tokenized.stats.local_blks_dirtied_per_sec	Bloco local sujo por segundo
db.sql_tokenized.stats.local_blks_written_per_sec	Gravações de bloco local por segundo
db.sql_tokenized.stats.temp_blks_written_per_sec	Gravações temporárias por segundo
db.sql_tokenized.stats.temp_blks_read_per_sec	Leituras temporárias por segundo
db.sql_tokenized.stats.blk_read_time_per_sec	Leituras simultâneas médias por segundo
db.sql_tokenized.stats.blk_write_time_per_sec	Gravações simultâneas médias por segundo

Estatísticas de resumo por chamada para o RDS PostgreSQL

As métricas a seguir fornecem estatísticas por chamada para uma instrução SQL.

Métrica	Unidade
db.sql_tokenized.stats.rows_per_call	Linhas por chamada
db.sql_tokenized.stats.avg_latency_per_call	Latência média por chamada (em ms)
db.sql_tokenized.stats.shared_blks_hit_per_call	Acertos de bloco por chamada
db.sql_tokenized.stats.shared_blks_read_per_call	Leituras de bloco por chamada
db.sql_tokenized.stats.shared_blks_written_per_call	Gravações de bloco por chamada
db.sql_tokenized.stats.shared_blks_dirtied_per_call	Blocos sujos por chamada
db.sql_tokenized.stats.local_blks_hit_per_call	Acertos de bloco local por chamada
db.sql_tokenized.stats.local_blks_read_per_call	Leituras de bloco local por chamada
db.sql_tokenized.stats.local_blks_dirtied_per_call	Bloco local sujo por chamada
db.sql_tokenized.stats.local_blks_written_per_call	Gravações de bloco local por chamada
db.sql_tokenized.stats.temp_blks_written_per_call	Gravações temporárias de bloco por chamada
db.sql_tokenized.stats.temp_blks_read_per_call	Leituras temporárias de bloco por chamada
db.sql_tokenized.stats.blk_read_time_per_call	Tempo de leitura por chamada (em ms)
db.sql_tokenized.stats.blk_write_time_per_call	Tempo de gravação por chamada (em ms)

Principais estatísticas do RDS PostgreSQL

As estatísticas de SQL a seguir estão disponíveis para instâncias de banco de dados do PostgreSQL.

Métrica	Unidade
db.sql_tokenized.stats.calls	Calls
db.sql_tokenized.stats.rows	Linhas
db.sql_tokenized.stats.total_time	Tempo total (em ms)
db.sql_tokenized.stats.shared_blks_hit	Acertos de bloco
db.sql_tokenized.stats.shared_blks_read	Leituras de bloco
db.sql_tokenized.stats.shared_blks_dirtied	Blocos sujos
db.sql_tokenized.stats.shared_blks_written	Gravações de bloco
db.sql_tokenized.stats.local_blks_hit	Acertos de bloco local
db.sql_tokenized.stats.local_blks_read	Leituras de bloco local
db.sql_tokenized.stats.local_blks_dirtied	Blocos locais sujos
db.sql_tokenized.stats.local_blks_written	Gravações de bloco local
db.sql_tokenized.stats.temp_blks_written	Gravações temporárias
db.sql_tokenized.stats.temp_blks_read	Leituras temporárias
db.sql_tokenized.stats.blk_read_time	Média de leituras simultâneas (em ms)
db.sql_tokenized.stats.blk_write_time	Média de gravações simultâneas (em ms)

Para obter mais informações sobre essas métricas, consulte pg_stat_statements na documentação do PostgreSQL.

Métricas do sistema operacional em monitoramento avançado

O Amazon RDS fornece métricas em tempo real para o sistema operacional (SO) no qual seu de instância de banco de dados é executado. O RDS fornece as métricas do monitoramento avançado

à sua conta do Amazon CloudWatch Logs. As tabelas a seguir listam métricas de SO disponíveis usando o Amazon CloudWatch Logs.

Tópicos

- Métricas do sistema operacional Db2, MariaDB, MySQL, Oracle e PostgreSQL
- Métricas de SO para Microsoft SQL Server

Métricas do sistema operacional Db2, MariaDB, MySQL, Oracle e PostgreSQL

Grupo	Métrica	Nome do console	Descrição
`General`	`engine`	Não aplicável	O mecanismo de banco de dados para a instância de banco de dados.
	`instanceID`	Não aplicável	O DB instance identifier.
	`instanceResourceID`	Não aplicável	Um identificador imutável para a instância de Bancos de Dados que é exclusivo para uma região da AWS, também usado como o identificador de stream de logs.
	`numVCPUs`	Não aplicável	O número de CPUs virtuais para a instância de banco de dados.
	`timestamp`	Não aplicável	A hora em que as métricas foram obtidas.
	`uptime`	Não aplicável	Por quanto tempo a instância de banco de dados esteve ativa.
	`version`	Não aplicável	A versão do formato JSON do stream de métricas do sistema operacional.
`cpuUtilization`	`guest`	Convidado da CPU	A porcentagem de CPU em uso por programas de convidado.

Grupo	Métrica	Nome do console	Descrição
	`idle`	CPU ociosa	A porcentagem de CPU que está ociosa.
	`irq`	IRQ da CPU	A porcentagem de CPU em uso por interrupções de software.
	`nice`	CPU boa	A porcentagem de CPU em uso por programas em execução com a prioridade mais baixa.
	`steal`	Roubo de CPU	A porcentagem de CPU em uso por outras máquinas virtuais.
	`system`	Sistema de CPU	A porcentagem de CPU em uso pelo kernel.
	`total`	Total da CPU	A porcentagem total da CPU em uso. Esse valor inclui o valor de `nice`.
	`user`	Usuário da CPU	A porcentagem de CPU em uso por programas do usuário.
	`wait`	Espera da CPU	A porcentagem de CPU fora de uso ao aguardar o acesso de E/S.
`diskIO`	`avgQueueLen`	Tamanho médio da fila	O número de solicitações que aguardam na fila do dispositivo de E/S.
	`avgReqSz`	Tamanho médio da solicitação	O tamanho médio da solicitação, em kilobytes.
	`await`	Espera de E/S de disco	O número de milissegundos necessários para responder a solicitações, incluindo o tempo na fila e o tempo de serviço.

Grupo	Métrica	Nome do console	Descrição
	`device`	Não aplicável	O identificador do dispositivo de disco em uso.
	`readIOsPS`	E/Ss de leitura	O número de operações de leitura por segundo.
	`readKb`	Total de leitura	O número total de kilobytes lidos.
	`readKbPS`	Kb/s de leitura	O número de kilobytes lidos por segundo.
	`readLatency`	Latência de leitura	O tempo decorrido entre o envio de uma solicitação de E/S de leitura e sua conclusão, em milissegundos. Essa métrica só está disponível para o Amazon Aurora.
	`readThroughput`	Taxa de transferência de leitura	A quantidade de taxa de transferência da rede usada por solicitações para o cluster de banco de dados, em bytes por segundo. Essa métrica só está disponível para o Amazon Aurora.
	`rrqmPS`	Rrqms	O número de solicitações de leitura mescladas enfileiradas por segundo.
	`tps`	TPS	O número de transações de E/S por segundo.
	`util`	Utilização de E/S de disco	A porcentagem de tempo de CPU durante o qual as solicitações foram emitidas.
	`writeIOsPS`	E/Ss de gravação	O número de operações de gravação por segundo.

Grupo	Métrica	Nome do console	Descrição
	writeKb	Total de gravação	O número total de kilobytes gravados.
	writeKbPS	Kb/s de gravação	O número de kilobytes gravados por segundo.
	writeLatency	Latência de gravação	O tempo médio decorrido entre o envio de uma solicitação de E/S de gravação e sua conclusão, em milissegundos. Essa métrica só está disponível para o Amazon Aurora.
	writeThroughput	Taxa de transferência de gravação	A quantidade de taxa de transferência da rede usada por respostas do cluster de banco de dados, em bytes por segundo. Essa métrica só está disponível para o Amazon Aurora.
	wrqmPS	Wrqms	O número de solicitações de gravação mescladas enfileiradas por segundo.
physicalDeviceIO	avgQueueLen	Tamanho médio da fila de dispositivos físicos	O número de solicitações que aguardam na fila do dispositivo de E/S.
	avgReqSz	Tamanho médio da solicitação de dispositivos físicos	O tamanho médio da solicitação, em kilobytes.

Grupo	Métrica	Nome do console	Descrição
	`await`	E/S de disco de dispositivos físicos em espera	O número de milissegundos necessários para responder a solicitações, incluindo o tempo na fila e o tempo de serviço.
	`device`	Não aplicável	O identificador do dispositivo de disco em uso.
	`readIOsPS`	E/S de leitura de dispositivos físicos	O número de operações de leitura por segundo.
	`readKb`	Total de leitura de dispositivos físicos	O número total de kilobytes lidos.
	`readKbPS`	Kb/s de leitura de dispositivos físicos	O número de kilobytes lidos por segundo.
	`rrqmPS`	Rrqms de dispositivos físicos	O número de solicitações de leitura mescladas enfileiradas por segundo.
	`tps`	TPS de dispositivos físicos	O número de transações de E/S por segundo.

Grupo	Métrica	Nome do console	Descrição
	`util`	Utilização de E/S de disco de dispositivos físicos	A porcentagem de tempo de CPU durante o qual as solicitações foram emitidas.
	`writeIOsPS`	E/S de gravação de dispositivos físicos	O número de operações de gravação por segundo.
	`writeKb`	Total de gravação de dispositivos físicos	O número total de kilobytes gravados.
	`writeKbPS`	Kb/s de gravação de dispositivos físicos	O número de kilobytes gravados por segundo.
	`wrqmPS`	Wrqms de dispositivos físicos	O número de solicitações de gravação mescladas enfileiradas por segundo.
`fileSys`	`maxFiles`	Máximo de Inodes	O número máximo de arquivos que podem ser criados para o sistema de arquivos.
	`mountPoint`	Não aplicável	O caminho para o sistema de arquivos.
	`name`	Não aplicável	O nome do sistema de arquivos.

Grupo	Métrica	Nome do console	Descrição
	`total`	Total do sistema de arquivos	O número total de espaço disponível em disco para o sistema de arquivos, em kilobytes.
	`used`	Sistema de arquivos usado	A quantidade de espaço em disco usada pelos arquivos no sistema de arquivos, em kilobytes.
	`usedFilePercent`	Inodes usados	A porcentagem de arquivos disponíveis em uso.
	`usedFiles`	% de utilização	O número de arquivos no sistema de arquivos.
	`usedPercent`	Sistema de arquivos usado	A porcentagem do espaço em disco do sistema de arquivos em uso.
`loadAverageMinute`	`fifteen`	Carga média de 15 min	O número de processos que estão solicitando tempo de CPU nos últimos 15 minutos.
	`five`	Carga média de 5 min	O número de processos que estão solicitando tempo de CPU nos últimos 5 minutos.
	`one`	Carga média de 1 min	O número de processos que estão solicitando tempo de CPU no último minuto.
`memory`	`active`	Memória ativa	A quantidade de memória atribuída, em kilobytes.
	`buffers`	Memória armazenada em buffer	A quantidade de memória usada para o buffer de solicitações de E/S antes de gravar no dispositivo de armazenamento, em kilobytes.

Grupo	Métrica	Nome do console	Descrição
	`cached`	Memória em cache	A quantidade de memória utilizada para o armazenamento em cache da E/S baseada em sistema de arquivos.
	`dirty`	Memória suja	A quantidade de páginas de memória na RAM que foram modificadas, mas não gravadas nos blocos de dados relacionados no armazenamento, em kilobytes.
	`free`	Memória livre	A quantidade de memória não atribuída, em kilobytes.
	`hugePagesFree`	Páginas enormes livres	O número de páginas enormes livres. Páginas enormes são um recurso do kernel do Linux.
	`hugePagesRsvd`	Páginas enormes reservadas	O número de páginas enormes confirmadas.
	`hugePagesSize`	Tamanho de páginas enormes	O tamanho de cada unidade de páginas enormes, em kilobytes.
	`hugePagesSurp`	Páginas enormes surp	O número de páginas enormes excedentes disponíveis em comparação com o total.
	`hugePagesTotal`	Total de páginas enormes	O número total de páginas enormes.
	`inactive`	Memória inativa	A quantidade de páginas de memória usadas com menos frequência, em kilobytes.

Grupo	Métrica	Nome do console	Descrição
	`mapped`	Memória mapeada	A quantidade total de conteúdo do sistema de arquivos que é mapeada na memória dentro de um espaço de endereçamento de processos, em kilobytes.
	`pageTables`	Tabelas de página	A quantidade de memória usada por tabelas de página, em kilobytes.
	`slab`	Memória dividida	A quantidade de estruturas de dados reutilizáveis do kernel, em kilobytes.
	`total`	Memória total	A quantidade total de memória, em kilobytes.
	`writeback`	Memória de gravação	A quantidade de páginas sujas na RAM que ainda estão sendo gravadas no armazenamento de suporte, em kilobytes.
`network`	`interface`	Não aplicável	O identificador da interface de rede que está sendo usada para a instância de banco de dados.
	`rx`	RX	O número de bytes recebidos por segundo.
	`tx`	TX	O número de bytes carregados por segundo.
`processList`	`cpuUsedPc`	% da CPU	A porcentagem de CPU usada pelo processo.
	`id`	Não aplicável	O identificador do processo.
	`memoryUsedPc`	MEM%	A porcentagem da memória usada pelo processo.
	`name`	Não aplicável	O nome do processo.

Grupo	Métrica	Nome do console	Descrição
	`parentID`	Não aplicável	O identificador do processo para o processo pai do processo.
	`rss`	RES	A quantidade de RAM alocada ao processo, em kilobytes.
	`tgid`	Não aplicável	O identificador do grupo de threads, que é um número que representa o ID do processo ao qual um thread pertence. Esse identificador é usado para agrupar threads do mesmo processo.
	`vss`	VIRT	A quantidade de memória virtual alocada ao processo, em kilobytes.
`swap`	`total`	Troca	A quantidade de memória de permuta disponível, em kilobytes.
	`in`	Trocas feitas	A quantidade de memória, em kilobytes, transferida temporariamente do disco.
	`out`	Trocas recebidas	A quantidade de memória, em kilobytes, transferida temporariamente para o disco.
	`free`	Troca livre	A quantidade de memória de troca livre, em kilobytes.
	`cached`	Troca confirmada	A quantidade de memória de permuta, em kilobytes, usada como a memória cache.
`tasks`	`blocked`	Tarefas bloqueadas	O número de tarefas que estão bloqueadas.
	`running`	Tarefas em execução	O número de tarefas que estão sendo executadas.

Grupo	Métrica	Nome do console	Descrição
	`sleeping`	Tarefas em espera	O número de tarefas que estão em suspensão.
	`stopped`	Tarefas paradas	O número de tarefas que estão interrompidas.
	`total`	Total de tarefas	O número total de tarefas.
	`zombie`	Tarefas Zombie	O número de tarefas filho que estão inativas com uma tarefa pai ativa.

Métricas de SO para Microsoft SQL Server

Grupo	Métrica	Nome do console	Descrição
`General`	`engine`	Não aplicável	O mecanismo de banco de dados para a instância de banco de dados.
	`instanceID`	Não aplicável	O DB instance identifier.
	`instanceResourceID`	Não aplicável	Um identificador imutável para a instância de Bancos de Dados que é exclusivo para uma região da AWS, também usado como o identificador de stream de logs.
	`numVCPUs`	Não aplicável	O número de CPUs virtuais para a instância de banco de dados.
	`timestamp`	Não aplicável	A hora em que as métricas foram obtidas.
	`uptime`	Não aplicável	Por quanto tempo a instância de banco de dados esteve ativa.

Grupo	Métrica	Nome do console	Descrição
	`version`	Não aplicável	A versão do formato JSON do stream de métricas do sistema operacional.
`cpuUtilization`	`idle`	CPU ociosa	A porcentagem de CPU que está ociosa.
	`kern`	Kernel da CPU	A porcentagem de CPU em uso pelo kernel.
	`user`	Usuário da CPU	A porcentagem de CPU em uso por programas do usuário.
`disks`	`name`	Não aplicável	O identificador do disco.
	`totalKb`	Espaço total em disco	O espaço total do disco, em kilobytes.
	`usedKb`	Espaço em disco usado	A quantidade de espaço usado no disco, em kilobytes.
	`usedPc`	% de espaço em disco usado	A porcentagem de espaço em uso no disco.
	`availKb`	Espaço em disco disponível	O espaço disponível no disco, em kilobytes.
	`availPc`	% de espaço em disco disponível	A porcentagem de espaço disponível no disco.
	`rdCountPS`	Leituras/s	O número de operações de leitura por segundo
	`rdBytesPS`	Kb/s de leitura	O número de bytes lidos por segundo.

Grupo	Métrica	Nome do console	Descrição
	`wrCountPS`	E/Ss de gravação	O número de operações de gravação por segundo.
	`wrBytesPS`	Kb/s de gravação	A quantidade de bytes escritos por segundo.
`memory`	`commitTotKb`	Total de confirmação	A quantidade de espaço de endereços virtual com suporte pelo arquivo de paginação em uso, ou seja, a taxa de confirmação atual. Esse valor é composto por memória principal (RAM) e disco (arquivos de paginação).
	`commitLimitKb`	Confirmação máxima	O valor máximo possível para a métrica `commitTotKb` . Esse valor é a soma do tamanho do arquivo de paginação atual, mais a memória física disponível para o conteúdo paginável, excluindo a RAM que é atribuída a áreas não paginável.
	`commitPeakKb`	Confirmar pico	O maior valor da métrica `commitTotKb` desde que o sistema operacional foi iniciado pela última vez.
	`kernTotKb`	Memória total do kernel	A soma da memória nos pools de kernel paginados e não paginados, em kilobytes.
	`kernPagedKb`	Memória do kernel paginada	A quantidade de memória no pool do kernel paginado, em kilobytes.
	`kernNonpagedKb`	Memória kernel não paginada	A quantidade de memória no pool do kernel não paginado, em kilobytes.
	`pageSize`	Tamanho da página	O tamanho de uma página, em bytes.

Grupo	Métrica	Nome do console	Descrição
	`physTotKb`	Memória total	A quantidade de memória física, em kilobytes.
	`physAvailKb`	Memória disponível	A quantidade de memória física disponível, em kilobytes.
	`sqlServerTotKb`	Memória total do SQL Server	A quantidade de memória confirmada com o SQL Server, em kilobytes.
	`sysCacheKb`	Cache do sistema	A quantidade de memória cache do sistema, em kilobytes.
`network`	`interface`	Não aplicável	O identificador da interface de rede que está sendo usada para a instância de banco de dados.
	`rdBytesPS`	Kb/s de leitura de rede	O número de bytes recebidos por segundo.
	`wrBytesPS`	Kb/s de gravação de rede	O número de bytes enviados por segundo.
`processList`	`cpuUsedPc`	% de utilização	A porcentagem de CPU usada pelo processo.
	`memUsedPc`	MEM%	A porcentagem da memória total usada pelo processo.
	`name`	Não aplicável	O nome do processo.
	`pid`	Não aplicável	O identificador do processo. Esse valor não está presente para processos que são de propriedade do Amazon RDS.

Grupo	Métrica	Nome do console	Descrição
	`ppid`	Não aplicável	O identificador do processo para o pai desse processo. Esse valor só está presente para processos filho.
	`tid`	Não aplicável	O identificador de thread. Esse valor só está presente para os threads. O processo de propriedade pode ser identificado usando o valor `pid`.
	`workingSetKb`	Não aplicável	A quantidade de memória no conjunto de trabalho particular mais a quantidade de memória que está sendo usada pelo processo e pode ser compartilhada com outros processos, em kilobytes.
	`workingSetPrivKb`	Não aplicável	A quantidade de memória que está sendo usada por um processo, mas que não pode ser compartilhada com outros processos, em kilobytes.
	`workingSetShareableKb`	Não aplicável	A quantidade de memória que está em uso por um processo, mas que não pode ser compartilhada com outros processos, em kilobytes.
	`virtKb`	Não aplicável	A quantidade de espaço de endereço virtual que o processo está usando, em kilobytes. O uso do espaço de endereço virtual não implica necessariamente o uso correspondente de páginas de disco ou de memória principal.
`system`	`handles`	Manipulações	O número de identificadores que o sistema está usando.
	`processes`	Processos	O número de processos em execução no sistema.

Grupo	Métrica	Nome do console	Descrição
	`threads`	Threads	O número de threads em execução no sistema.

Monitorar eventos, logs e transmissões em uma instância de banco de dados Amazon RDS

Quando você monitora seus bancos de dados do Amazon RDSe outras soluções da AWS, seu objetivo é manter o seguinte:

- Confiabilidade
- Disponibilidade
- Performance
- Segurança

Métricas de monitoramento em uma instância do Amazon RDS explica como monitorar sua instância usando métricas. Uma solução completa também deve monitorar eventos de banco de dados, arquivos de log e transmissões de atividades. A AWS fornece as seguintes ferramentas de monitoramento:

- O Amazon EventBridge é um serviço de barramento de eventos sem servidor que facilita a conexão de aplicações a dados de diversas origens. O EventBridge fornece uma fluxo de dados em tempo real de suas próprias aplicações, de aplicações de software como serviço (SaaS) e de serviços da AWS. O EventBridge encaminha esses dados para destinos como AWS Lambda. Dessa forma, você pode monitorar eventos que ocorrem em serviços e criar arquiteturas orientadas a eventos. Para obter mais informações, consulte o Guia do Usuário do Amazon EventBridge.
- O Amazon CloudWatch Logs oferece uma forma de monitorar, armazenar e acessar os arquivos de log de instâncias do Amazon RDS, do AWS CloudTrail e de outras fontes. O Amazon CloudWatch Logs pode monitorar informações nos arquivos de log e notificar você quando determinados limites forem atingidos. É possível também arquivar seus dados de log em armazenamento resiliente. Para obter mais informações, consulte o Guia do usuário do Amazon CloudWatch Logs.
- AWS CloudTrail captura chamadas de API e eventos relacionados realizados pela conta da Conta da AWS ou em nome dela. O CloudTrail fornece os arquivos de log para um bucket do Amazon S3 especificado por você. Você pode identificar quais usuários e contas chamaram a AWS, o endereço IP de origem no qual as chamadas foram feitas e quando elas ocorreram. Para mais informações, consulte o Guia do usuário do AWS CloudTrail.

- O Database Activity Streams é um recurso do Amazon RDS que fornece um fluxo quase em tempo real da atividade em sua instância de banco de dados do Oracle. O Amazon RDS envia atividades para um fluxo de dados do Amazon Kinesis. O fluxo do Kinesis é criado automaticamente. No Kinesis, é possível configurar serviços da AWS, como o Amazon Data Firehose e o AWS Lambda, para consumir o fluxo e armazenar os dados.

Tópicos

- Visualizar logs, eventos e transmissões no console do Amazon RDS
- Monitorar eventos do Amazon RDS
- Monitorar arquivos de log do Amazon RDS
- Monitorar chamadas de API do Amazon RDSno AWS CloudTrail
- Monitorar o Amazon RDS com o recurso Database Activity Streams
- Monitorar ameaças com o Amazon GuardDuty RDS Protection

Visualizar logs, eventos e transmissões no console do Amazon RDS

O Amazon RDS se integra aos Serviços da AWS para mostrar informações sobre logs, eventos e transmissões de atividades de banco de dados no console do RDS.

A guia Logs & events (Logs e eventos) de sua instância de banco de dados do RDS mostra estas informações:

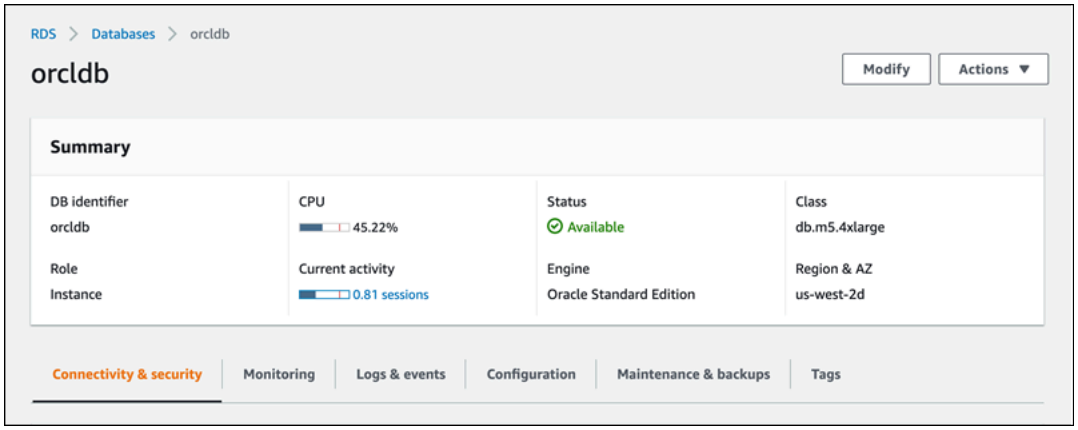
- Alertas do Amazon CloudWatch: mostra todos os alarmes de métricas configurados para a instância de banco de dados. Se você ainda não configurou alarmes, poderá criá-los no console do RDS. Para obter mais informações, consulte Monitorar métricas do Amazon RDS com o Amazon CloudWatch.
- Eventos recentes: mostra um resumo dos eventos (alterações no ambiente) relacionados à sua instância de banco de dados do RDS. Para obter mais informações, consulte Visualizar eventos do Amazon RDS.
- Logs: mostra os arquivos de log do banco de dados gerados por uma instância de banco de dados. Para obter mais informações, consulte Monitorar arquivos de log do Amazon RDS.

A guia Configuration (Configuração) exibe informações sobre transmissões de atividades de banco de dados.

Para visualizar registros, eventos e transmissões de sua instância de banco de dados no console do RDS

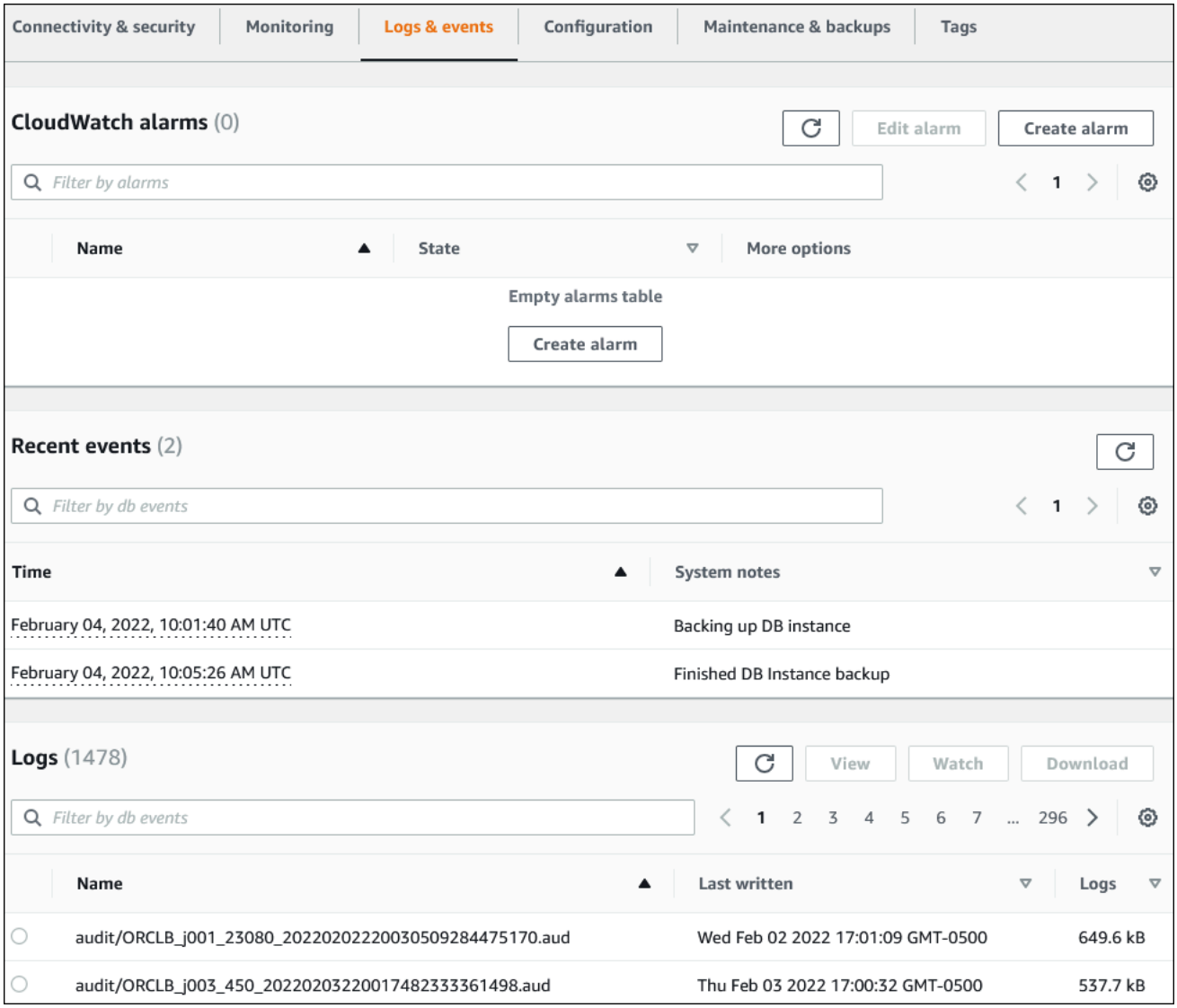
1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Bancos de dados.
3. Escolha o nome da instância de banco de dados que deseja monitorar.

   A página “Databases” (Bancos de dados) é exibida. O exemplo a seguir mostra um banco de dados Oracle chamado `orclb`.



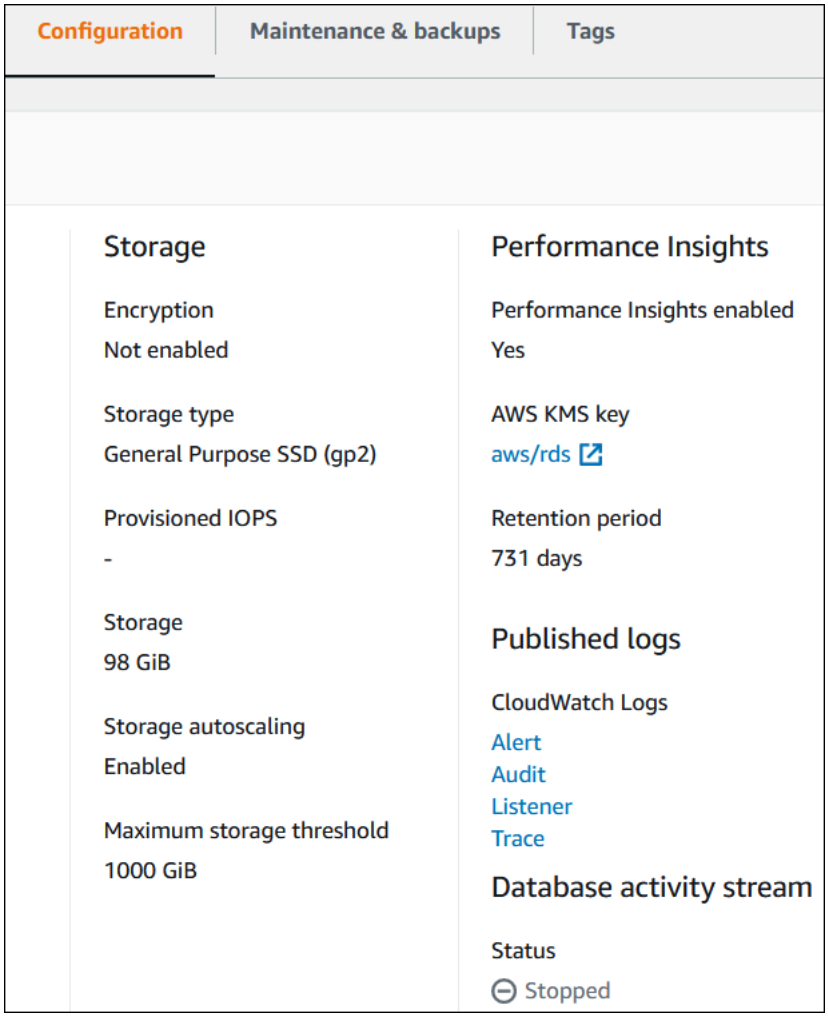
4. Escolha Logs & events (Logs e eventos).

   A seção “Logs & events” (Logs e eventos) é exibida.



5. Escolher configuração.

   O exemplo a seguir mostra o status dos fluxos de atividades de bancos de dados para sua instância de banco de dados.


Configuration
Maintenance & backups
Tags
Storage
Encryption
Not enabled
Storage type
General Purpose SSD (gp2)
Provisioned IOPS
-
Storage
98 GiB
Storage autoscaling
Enabled
Maximum storage threshold
1000 GiB
Performance Insights
Performance Insights enabled
Yes
AWS KMS key
aws/rds
Retention period
731 days
Published logs
CloudWatch Logs
Alert
Audit
Listener
Trace
Database activity stream
Status
Stopped

Monitorar eventos do Amazon RDS

Um evento indica uma alteração em um ambiente. Isso pode ser um ambiente da AWS, um serviço ou uma aplicação de parceiro de SaaS ou uma das suas próprios aplicações ou serviços personalizados. Para obter descrições de eventos do RDS, consulte Categorias de eventos e mensagens de eventos do Amazon RDS.

Tópicos

- Visão geral dos eventos para Amazon RDS
- Visualizar eventos do Amazon RDS
- Trabalhar com a notificação de eventos do Amazon RDS
- Criar uma regra que é acionada em um evento do Amazon RDS
- Categorias de eventos e mensagens de eventos do Amazon RDS

Visão geral dos eventos para Amazon RDS

Um evento do RDS indica uma alteração no ambiente do Amazon RDS. Por exemplo, o Amazon RDS gera um evento quando o estado de uma instância de banco de dados muda de pendente para em execução. O Amazon RDS entrega eventos ao EventBridge quase em tempo real.

> **Note**
>
> O Amazon RDS emite eventos em uma base de melhor esforço. Recomendamos que você evite gravar programas que dependam da ordem ou da existência de eventos de notificação, pois eles podem estar fora de sequência ou ausentes.

O Amazon RDS registra eventos relacionados aos seguintes recursos:

- Instâncias de banco de dados

  Para obter uma lista de eventos de instância de banco de dados, consulte Eventos de instância de banco de dados.
- Grupos de parâmetros do banco de dados

  Para obter uma lista de eventos de grupo de parâmetros de banco de dados, consulte Eventos de grupo de parâmetros de banco de dados.

- Grupos de segurança de banco de dados

  Para obter uma lista de eventos de grupo de segurança do banco de dados, consulte Eventos de grupos de segurança de banco de dados.
- Snapshots de banco de dados

  Para obter uma lista de eventos de snapshot do banco de dados, consulte Eventos de snapshot de banco de dados.
- Eventos do RDS Proxy

  Para obter uma lista de eventos do proxy do RDS, consulte Eventos do RDS Proxy.
- Eventos de implantação azul/verde

  Para obter uma lista de eventos de implantação azul/verde, consulte Eventos de implantação azul/verde.

Essas informações incluem:

- A data e a hora do evento.
- O nome da origem e o tipo de origem do evento
- A mensagem associada ao evento
- As notificações de eventos incluem tags de quando a mensagem foi enviada e podem não refletir as tags no momento em que o evento ocorreu

Visualizar eventos do Amazon RDS

É possível recuperar as seguintes informações do evento para seus recursos do Amazon RDS:

- Nome do recurso
- Tipo de recurso
- Horário do evento
- Resumo de mensagens do evento

É possível acessar eventos nas seguintes partes do AWS Management Console:

- A guia Eventos, que mostra eventos das últimas 24 horas.
- A tabela Eventos recentes na seção Logs e eventos na guia Bancos de dados, que pode mostrar eventos das últimas duas semanas.

Também é possível recuperar eventos usando o comando describe-events da AWS CLI ou a operação DescribeEvents da API do RDS. Se você usar a AWS CLI ou a API do RDS para visualizar eventos, poderá recuperar eventos até os últimos 14 dias.

> **Note**
>
> Se precisar armazenar eventos por períodos mais longos, você poderá enviar eventos do Amazon RDS ao EventBridge. Para ter mais informações, consulte Criar uma regra que é acionada em um evento do Amazon RDS.

Para obter descrições de eventos do Amazon RDS, consulte Categorias de eventos e mensagens de eventos do Amazon RDS.

Para acessar informações detalhadas sobre eventos usando o AWS CloudTrail, incluindo parâmetros de solicitação, consulte Eventos do CloudTrail.

Console

Como visualizar todos os eventos do Amazon RDS nas últimas 24 horas

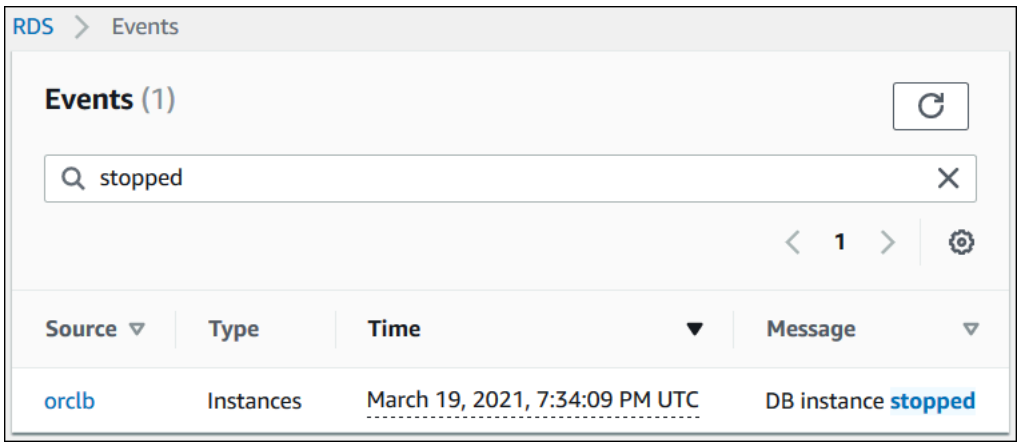
1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.

2. No painel de navegação, escolha Events.

   Os eventos disponíveis aparecem em uma lista.

3. (Opcional) Insira um termo de pesquisa para filtrar seus resultados.

   O exemplo a seguir mostra uma lista de eventos filtrados pelos caracteres **`stopped`**.



AWS CLI

Para visualizar todos os eventos gerados na última hora, chame describe-events sem parâmetros.

```
aws rds describe-events
```

O exemplo de saída a seguir mostra que uma instância de banco de dados foi interrompida.

```
{
    "Events": [
        {
            "EventCategories": [
                "notification"
            ],
            "SourceType": "db-instance",
            "SourceArn": "arn:aws:rds:us-east-1:123456789012:db:testinst",
            "Date": "2022-04-22T21:31:00.681Z",
            "Message": "DB instance stopped",
            "SourceIdentifier": "testinst"
        }
    ]
}
```

Para visualizar todos os eventos do Amazon RDS dos últimos 10.080 minutos (sete dias), chame o comando describe-events AWS CLI e defina o parâmetro `--duration` como `10080`.

```
aws rds describe-events --duration 10080
```

O exemplo a seguir mostra os eventos no intervalo de tempo especificado para a instância de banco de dados *`test-instance`*.

```
aws rds describe-events \
    --source-identifier test-instance \
    --source-type db-instance \
    --start-time 2022-03-13T22:00Z \
    --end-time 2022-03-13T23:59Z
```

O exemplo de saída a seguir mostra o status de um backup.

```
{
    "Events": [
        {
            "SourceType": "db-instance",
            "SourceIdentifier": "test-instance",
            "EventCategories": [
                "backup"
            ],
            "Message": "Backing up DB instance",
            "Date": "2022-03-13T23:09:23.983Z",
            "SourceArn": "arn:aws:rds:us-east-1:123456789012:db:test-instance"
        },
        {
            "SourceType": "db-instance",
            "SourceIdentifier": "test-instance",
            "EventCategories": [
                "backup"
            ],
            "Message": "Finished DB Instance backup",
            "Date": "2022-03-13T23:15:13.049Z",
            "SourceArn": "arn:aws:rds:us-east-1:123456789012:db:test-instance"
        }
    ]
}
```

API

É possível visualizar todos os eventos de instâncias do Amazon RDS dos últimos 14 dias, chamando a operação da API do RDS DescribeEvents e definindo o parâmetro `Duration` como `20160`.

Trabalhar com a notificação de eventos do Amazon RDS

O Amazon RDS usa o Amazon Simple Notification Service (Amazon SNS) para fornecer uma notificação quando um evento do Amazon RDS ocorre. Essas notificações podem estar em qualquer formato de notificação compatível com o Amazon SNS para uma região da AWS, como um e-mail, uma mensagem de texto ou uma chamada para um endpoint HTTP.

Tópicos

- Visão geral das notificações de eventos do Amazon RDS
- Conceder permissões para publicar notificações em um tópico do Amazon SNS
- Inscrever-se em notificações de eventos do Amazon RDS
- Tags e atributos de notificação de eventos do Amazon RDS
- Listar assinaturas de notificação de evento do Amazon RDS
- Modificar uma assinatura de notificação de evento do Amazon RDS
- Adicionar um identificador de origem a uma assinatura de notificação de evento do Amazon RDS
- Remover um identificador de origem de uma assinatura de notificação de eventos do Amazon RDS
- Listar as categorias de notificação de evento do Amazon RDS
- Excluir uma assinatura de notificação de evento do Amazon RDS

Visão geral das notificações de eventos do Amazon RDS

O Amazon RDS agrupa eventos em categorias em que você pode se inscrever para receber notificações quando um evento ocorrer na categoria.

Tópicos

- Recursos do RDS elegíveis para assinatura de eventos
- Processo básico de assinatura de notificações de eventos do Amazon RDS
- Entrega de notificações de evento do RDS
- Faturamento de notificações de eventos do Amazon RDS
- Exemplos de eventos do Amazon RDS usando o Amazon EventBridge

Recursos do RDS elegíveis para assinatura de eventos

Você pode se inscrever em uma categoria de evento para os seguintes recursos:

- DB instance (Instância de banco de dados)
- DB snapshot
- DB parameter group (grupo de parâmetros de banco de dados)
- DB security group (grupo de segurança de banco de dados)
- RDS Proxy
- Versões de mecanismo personalizadas

Por exemplo, se você se inscrever na categoria de backup para uma determinada instância de banco de dados, receberá notificações sempre que houver um evento relacionado ao backup que afete a instância de banco de dados. Se você se inscrever em uma categoria de alteração de configuração para uma instância de banco de dados, será notificado quando a instância de banco de dados for alterada. Você também recebe uma notificação quando uma assinatura de notificação de evento é alterada.

Você pode querer criar várias assinaturas diferentes. Por exemplo, você pode criar uma assinatura que receba todas as notificações de eventos de todas as instâncias de banco de dados e outra que inclua somente eventos críticos de um subconjunto das instâncias de bancos de dados. Para a segunda assinatura, especifique uma ou mais instâncias de banco de dados no filtro.

Processo básico de assinatura de notificações de eventos do Amazon RDS

O processo de inscrição na notificação de evento do Amazon RDS é o seguinte:

1. Você cria uma assinatura de notificação de eventos do Amazon RDS usando o console do Amazon RDS, a AWS CLI ou a API.

   O Amazon RDS usa o ARN de um tópico do Amazon SNS para identificar cada assinatura. O console do Amazon RDS cria o ARN para você quando cria a assinatura. Crie o ARN usando o console do Amazon SNS, a AWS CLI ou a API do Amazon SNS.
2. O Amazon RDS envia um e-mail ou mensagem SMS de aprovação para os endereços que você submeteu com a assinatura.
3. Você confirma a assinatura escolhendo o link na notificação que recebeu.
4. O console do Amazon RDS atualiza a seção My Event Subscriptions (Minhas assinaturas de eventos) com o status de sua assinatura.
5. O Amazon RDS começa a enviar notificações aos endereços que você forneceu ao criar a assinatura.

Para saber mais sobre o Gerenciamento de Identidade e Acesso ao usar o Amazon SNS, consulte Gerenciamento de Identidade e Acesso no Amazon SNS no Guia do desenvolvedor do Amazon Simple Notification Service.

É possível usar o AWS Lambda para processar notificações de eventos de uma instância de banco de dados. Para obter mais informações, consulte Uso do AWS Lambda com o Amazon RDS no Guia do desenvolvedor do AWS Lambda.

Entrega de notificações de evento do RDS

O Amazon RDS envia notificações aos endereços que você fornece ao criar a assinatura. A notificação pode incluir atributos de mensagem que fornecem metadados estruturados sobre a mensagem. Para obter mais informações sobre os atributos de mensagem, consulte Categorias de eventos e mensagens de eventos do Amazon RDS.

As notificações de eventos podem levar até cinco minutos para serem entregues.

> **Important**
>
> O Amazon RDS não garante a ordem dos eventos enviados em um fluxo de eventos. A ordem do evento está sujeita a alterações.

Quando o Amazon SNS envia uma notificação para um endpoint HTTP ou HTTPS inscrito, a mensagem POST enviada ao endpoint tem um corpo de mensagem que contém um documento JSON. Para obter mais informações, consulte Mensagens do Amazon SNS e formatos JSON no Guia do desenvolvedor do Amazon Simple Notification Service.

É possível configurar o SNS para notificar você com mensagens de texto. Para obter mais informações, consulte Mensagens de texto móveis (SMS) no Guia do desenvolvedor do Amazon Simple Notification Service.

Para desativar as notificações sem excluir uma assinatura, escolha No (Não) para Enabled (Habilitado) no console do Amazon RDS. Ou você pode definir o parâmetro `Enabled` para `false` usando a AWS CLI ou a API do Amazon RDS.

Faturamento de notificações de eventos do Amazon RDS

O faturamento da notificação de eventos do Amazon RDS é feito por meio do Amazon SNS. As taxas do Amazon SNS se aplicam durante o uso da notificação de eventos. Para obter mais informações sobre o faturamento do Amazon SNS, consulte Preço do Amazon Simple Notification Service.

Exemplos de eventos do Amazon RDS usando o Amazon EventBridge

Os exemplos a seguir mostram diferentes tipos de eventos do Amazon RDS no formato JSON. Para acessar um tutorial que mostre como capturar e visualizar eventos no formato JSON, consulte Tutorial: Registro de alterações de estado de uma instância de banco de dados usando o Amazon EventBridge.

Tópicos

- Exemplo de um evento de instância de banco de dados
- Exemplo de um evento de grupo de parâmetros de banco de dados
- Exemplo de um evento de snapshot do banco de dados

Exemplo de um evento de instância de banco de dados

Veja a seguir um exemplo de um evento de instância de banco de dados no formato JSON. O evento mostra que o RDS executou um failover multi-AZ para a instância chamada `my-db-instance`. O ID do evento é RDS-EVENT-0049.

```
{
  "version": "0",
  "id": "68f6e973-1a0c-d37b-f2f2-94a7f62ffd4e",
  "detail-type": "RDS DB Instance Event",
  "source": "aws.rds",
  "account": "123456789012",
  "time": "2018-09-27T22:36:43Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:rds:us-east-1:123456789012:db:my-db-instance"
  ],
  "detail": {
    "EventCategories": [
      "failover"
    ],
    "SourceType": "DB_INSTANCE",
    "SourceArn": "arn:aws:rds:us-east-1:123456789012:db:my-db-instance",
    "Date": "2018-09-27T22:36:43.292Z",
    "Message": "A Multi-AZ failover has completed.",
    "SourceIdentifier": "my-db-instance",
    "EventID": "RDS-EVENT-0049"
  }
```

```
}
```

Exemplo de um evento de grupo de parâmetros de banco de dados

Veja a seguir um exemplo de um evento de grupo de parâmetros de banco de dados no formato JSON. O evento mostra que o parâmetro `time_zone` foi atualizado no grupo de parâmetros `my-db-param-group`. O ID do evento é RDS-EVENT-0037.

```
{
  "version": "0",
  "id": "844e2571-85d4-695f-b930-0153b71dcb42",
  "detail-type": "RDS DB Parameter Group Event",
  "source": "aws.rds",
  "account": "123456789012",
  "time": "2018-10-06T12:26:13Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:rds:us-east-1:123456789012:pg:my-db-param-group"
  ],
  "detail": {
    "EventCategories": [
      "configuration change"
    ],
    "SourceType": "DB_PARAM",
    "SourceArn": "arn:aws:rds:us-east-1:123456789012:pg:my-db-param-group",
    "Date": "2018-10-06T12:26:13.882Z",
    "Message": "Updated parameter time_zone to UTC with apply method immediate",
    "SourceIdentifier": "my-db-param-group",
    "EventID": "RDS-EVENT-0037"
  }
}
```

Exemplo de um evento de snapshot do banco de dados

Veja a seguir um exemplo de um evento de snapshot de banco de dados no formato JSON. O evento mostra a exclusão do snapshot chamado `my-db-snapshot`. O ID do evento é RDS-EVENT-0041.

```
{
  "version": "0",
  "id": "844e2571-85d4-695f-b930-0153b71dcb42",
  "detail-type": "RDS DB Snapshot Event",
  "source": "aws.rds",
```

```
  "account": "123456789012",
  "time": "2018-10-06T12:26:13Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:rds:us-east-1:123456789012:snapshot:rds:my-db-snapshot"
  ],
  "detail": {
    "EventCategories": [
      "deletion"
    ],
    "SourceType": "SNAPSHOT",
    "SourceArn": "arn:aws:rds:us-east-1:123456789012:snapshot:rds:my-db-snapshot",
    "Date": "2018-10-06T12:26:13.882Z",
    "Message": "Deleted manual snapshot",
    "SourceIdentifier": "my-db-snapshot",
    "EventID": "RDS-EVENT-0041"
  }
}
```

Conceder permissões para publicar notificações em um tópico do Amazon SNS

Para conceder permissões do Amazon RDS a fim de publicar notificações em um tópico do Amazon Simple Notification Service (Amazon SNS), anexe uma política do AWS Identity and Access Management (IAM) ao tópico de destino. Para obter mais informações sobre permissões, consulte Exemplos de casos de controle de acesso do Amazon Simple Notification Service no Guia do desenvolvedor do Amazon Simple Notification Service.

Por padrão, um tópico do Amazon SNS tem uma política que permite que todos os recursos do Amazon RDS na mesma conta publiquem notificações nele. Você pode anexar uma política personalizada para possibilitar notificações entre contas ou restringir o acesso a determinados recursos.

Veja a seguir um exemplo de uma política do IAM anexada ao tópico de destino do Amazon SNS. Ela restringe o tópico a instâncias de banco de dados com nomes correspondentes ao prefixo especificado. Para usar essa política, especifique os seguintes valores:

- `Resource`: o nome do recurso da Amazon (ARN) do tópico do Amazon SNS
- `SourceARN`: o ARN de recursos do RDS
- `SourceAccount`: o ID de sua Conta da AWS

Para ver uma lista dos tipos de recursos e seus ARNs, consulte Recursos definidos pelo Amazon RDS na Referência de autorização do serviço.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "events.rds.amazonaws.com"
      },
      "Action": [
        "sns:Publish"
      ],
      "Resource": "arn:aws:sns:us-east-1:123456789012:topic_name",
      "Condition": {
```

```
        "ArnLike": {
          "aws:SourceArn": "arn:aws:rds:us-east-1:123456789012:db:prefix-*"
        },
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        }
      }
    }
  ]
}
```

Inscrever-se em notificações de eventos do Amazon RDS

A forma mais fácil de criar uma assinatura é com o console do RDS. Se você preferir criar assinaturas de notificações de eventos usando a CLI ou API, é necessário criar um tópico do Amazon Simple Notification Service e fazer a assinatura desse tópico com o console do Amazon SNS ou a API do Amazon SNS. Também será necessário reter o nome de recurso da Amazon (ARN) do tópico, pois ele é usado ao enviar comandos da CLI ou operações da API. Para obter informações sobre como criar um tópico do SNS e assiná-lo, consulte Conceitos básicos do Amazon SNS no Guia do desenvolvedor do Amazon Simple Notification Service.

Você pode especificar o tipo de origem sobre o qual deseja ser notificado e a origem do Amazon RDS que aciona o evento:

Source type (Tipo de origem)

O tipo de fonte. Por exemplo, Source type (Tipo de origem) pode ser Instances (Instâncias). Você deve escolher um tipo de origem.

Resources to include (Recursos para incluir)

Os recursos do Amazon RDS que estão gerando os eventos. Por exemplo, você pode escolher Select specific instances (Selecionar instâncias específicas), depois myDBInstance1.

A tabela a seguir explica o resultado quando você especifica ou não ***Resources*** to include (Recursos para incluir).

Recursos para incluir	Descrição	Exemplo
Especificado	O RDS notifica você sobre todos os eventos somente do recurso especificado.	Se o seu Source type (Tipo de origem) for Instances (Instâncias) e seu recurso for myDBInstance1, o RDS notificará você sobre todos os eventos somente para myDBInstance1.
Não especificado	O RDS notifica você sobre os eventos do tipo de fonte especificado para todos os recursos do Amazon RDS.	Se o Source type (Tipo de origem) for Instances (Instâncias), o RDS notificará você sobre

Recursos para incluir	Descrição	Exemplo
		todos os eventos relacionados à instância em sua conta.

Por padrão, um assinante de tópico do Amazon SNS recebe todas as mensagens publicadas no tópico. Para receber apenas um subconjunto das mensagens, o assinante deve atribuir uma política de filtro à assinatura do tópico. Para obter mais informações sobre filtragem de mensagens do SNS, consulte Filtragem de mensagens do Amazon SNS no Guia do desenvolvedor do Amazon Simple Notification Service.

Console

Para assinar a notificação de eventos do RDS

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Event subscriptions (Assinaturas de eventos).
3. No painel Event subscriptions (Assinaturas de eventos), escolha Create event subscription (Criar assinatura de evento).
4. Insira os detalhes da sua assinatura da seguinte forma:

    a. Em Name (Nome), insira um nome para a assinatura de notificação de eventos.

    b. Para Send notifications to (Enviar notificação para), utilize um dos seguintes procedimentos:

    - Escolha New email topic (Novo tópico de e-mail). Insira um nome para o tópico do seu e-mail e uma lista de destinatários. Recomendamos que você configure as assinaturas de eventos com o mesmo endereço de e-mail do contato primário da sua conta. As mensagens de recomendações, eventos de serviço e integridade pessoal são enviadas por meio de canais diferentes. As assinaturas com o mesmo endereço de e-mail garantem que todas as mensagens sejam consolidadas em um único local.
    - Escolha Amazon Resource Name (ARN) [Nome do recurso da Amazon (ARN)]. Depois, escolha o ARN do Amazon SNS existente para um tópico do Amazon SNS.

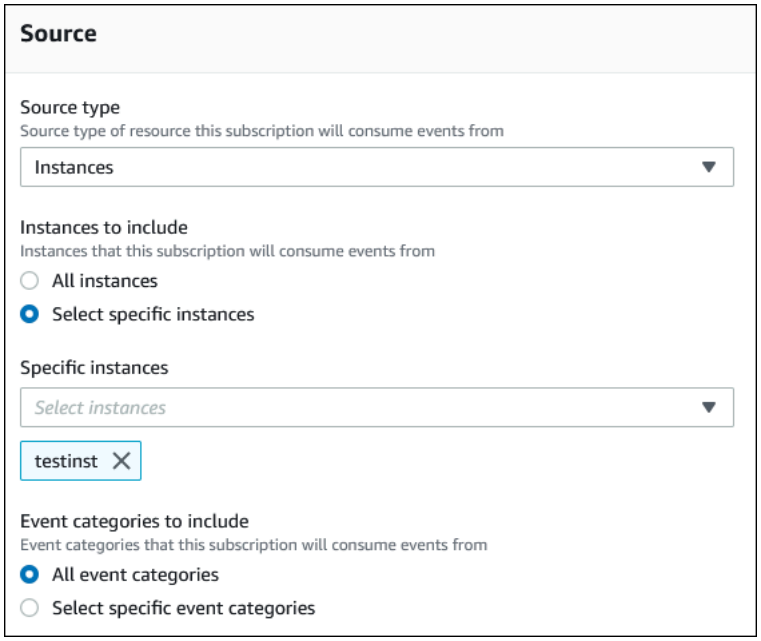
      Se você quiser usar um tópico que tenha sido ativado para criptografia do lado do servidor (SSE), conceda ao Amazon RDS as permissões necessárias para acessar a

AWS KMS key. Para obter mais informações, consulte Ativar a compatibilidade entre as fontes de eventos de tópicos criptografados e serviços da AWS no Guia do desenvolvedor do Amazon Simple Notification Service.

c. Em Source type (Tipo de origem), escolha um tipo de origem. Por exemplo, escolha Instances (Instâncias) ou Parameter groups (Grupos de parâmetros).

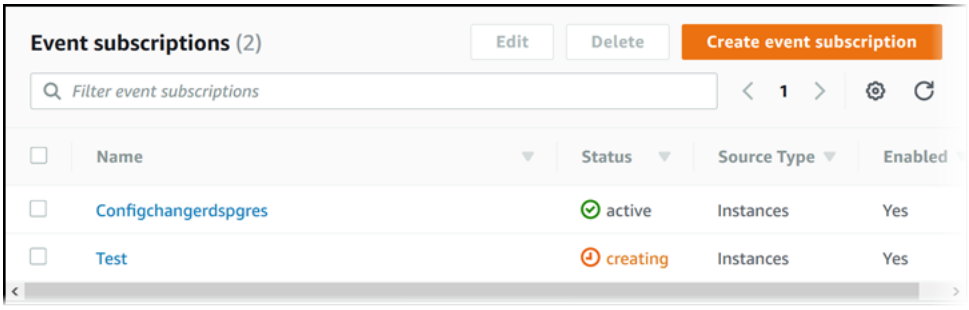
d. Escolha as categorias e recursos de eventos para os quais quer receber notificações de eventos.

O exemplo a seguir configura notificações de eventos para a instância de banco de dados denominada `testinst`.



e. Escolha Criar.

O console do Amazon RDS indica que a assinatura está sendo criada.



AWS CLI

Para assinar a Notificação de eventos do RDS, use o comando AWS CLI da `create-event-subscription`. Inclua os seguintes parâmetros necessários:

- `--subscription-name`
- `--sns-topic-arn`

Example

Para Linux, macOS ou Unix:

```
aws rds create-event-subscription \
    --subscription-name myeventsubscription \
    --sns-topic-arn arn:aws:sns:us-east-1:123456789012:myawsuser-RDS \
    --enabled
```

Para Windows:

```
aws rds create-event-subscription ^
    --subscription-name myeventsubscription ^
    --sns-topic-arn arn:aws:sns:us-east-1:123456789012:myawsuser-RDS ^
    --enabled
```

API

Para assinar a notificação de eventos do Amazon RDS, chame a função da API `CreateEventSubscription` do Amazon RDS. Inclua os seguintes parâmetros necessários:

- `SubscriptionName`
- `SnsTopicArn`

Tags e atributos de notificação de eventos do Amazon RDS

Quando o Amazon RDS envia uma notificação de evento ao Amazon Simple Notification Service (SNS) ou ao Amazon EventBridge, a notificação contém atributos de mensagem e tags de evento. O RDS envia os atributos da mensagem separadamente com a mensagem, enquanto as tags de evento estão no corpo da mensagem. Use os atributos de mensagem e as tags do Amazon RDS para adicionar metadados aos seus recursos. Você pode modificar essas tags com suas próprias notações sobre as instâncias de banco de dados Para ter mais informações sobre recursos de marcação do Amazon RDS, consulte Marcar recursos do Amazon RDS.

Por padrão, o Amazon SNS e o Amazon EventBridge recebem todas as mensagens enviadas a eles. O SNS e o EventBridge podem filtrar a mensagem e enviar as notificações ao modo de comunicação preferencial, como um e-mail, uma mensagem de texto ou uma chamada para um endpoint HTTP.

> **Note**
>
> A notificação enviada em um e-mail ou uma mensagem de texto não terá tags de evento.

A tabela a seguir mostra os atributos da mensagem de eventos do RDS enviada ao assinante do tópico.

Atributo de evento do Amazon RDS	Descrição
EventID	Identificador da mensagem de eventos do RDS, por exemplo, RDS-EVENT-0006.
Recurso	O identificador do ARN do recurso que emite o evento, por exemplo, `arn:aws:rds:ap-southeast-2:123456789012:db:database-1` .

As tags do RDS fornecem dados sobre o recurso que foi afetado pelo evento do serviço. O RDS adiciona o estado atual das tags ao corpo da mensagem quando a notificação é enviada ao SNS ou ao EventBridge.

Para ter mais informações sobre filtragem de atributos de mensagens do SNS, consulte Filtragem de mensagens do Amazon SNS no Guia do desenvolvedor do Amazon Simple Notification Service.

Para obter mais informações sobre a filtragem de tags de evento para o EventBridge, consulte Operadores de comparação para uso em padrões de eventos do Amazon EventBridge no Guia do usuário do Amazon EventBridge.

Para obter mais informações sobre a filtragem de tags baseadas em carga útil para o SNS, consulte Introdução à filtragem de mensagens baseadas em carga útil para o Amazon SNS

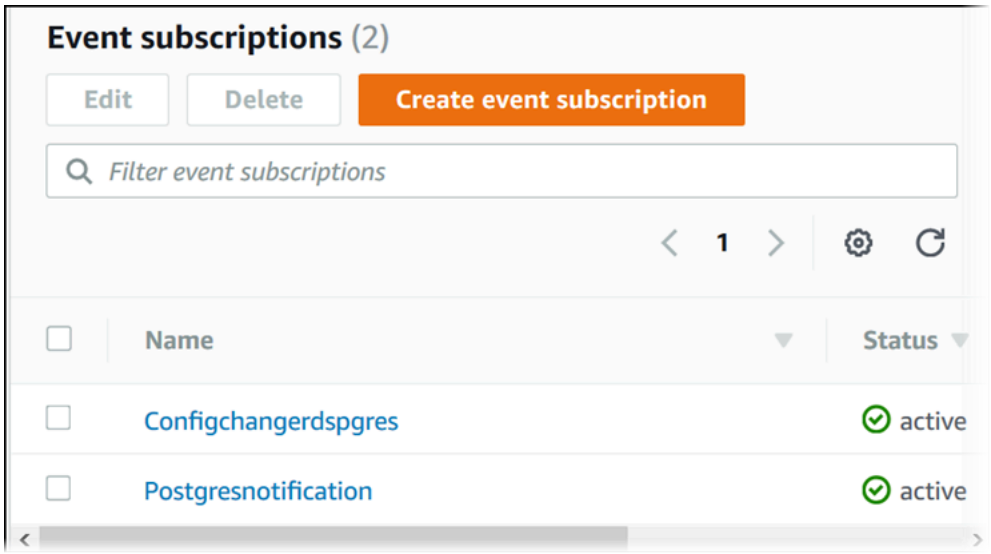
Listar assinaturas de notificação de evento do Amazon RDS

Você pode listar suas atuais assinaturas de notificações de eventos do Amazon RDS.

Console

Para listar suas atuais assinaturas de notificações de eventos do Amazon RDS

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Event subscriptions (Assinaturas de eventos). O painel Event subscriptions (Assinaturas de eventos) exibirá todas as suas assinaturas de notificação de eventos.



AWS CLI

Para listar suas atuais assinaturas de notificações de eventos do Amazon RDS, use o comando da AWS CLI `describe-event-subscriptions`l.

Example

O exemplo a seguir descreve todas as assinaturas de eventos.

```
aws rds describe-event-subscriptions
```

O exemplo a seguir descreve a pilha `myfirsteventsubscription`.

```
aws rds describe-event-subscriptions --subscription-name myfirsteventsubscription
```

API

Para listar suas atuais assinaturas de notificações de eventos do Amazon RDS, chame a ação da API do Amazon RDS `DescribeEventSubscriptions`.

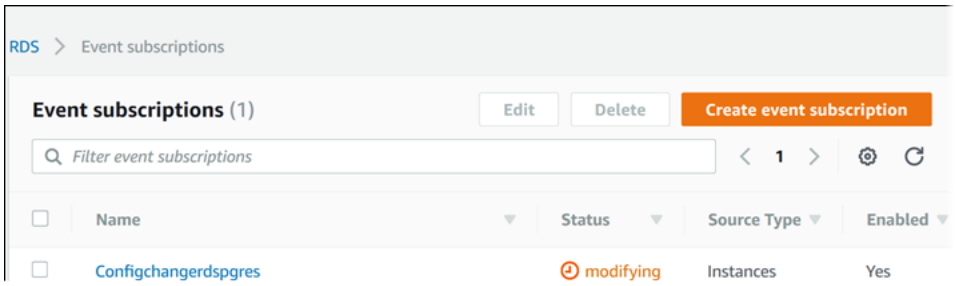
Modificar uma assinatura de notificação de evento do Amazon RDS

Depois que você criar uma assinatura, é possível alterar o nome, identificador de origem, categorias ou ARN do tópico da assinatura.

Console

Para modificar uma assinatura de notificação de evento do Amazon RDS

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Event subscriptions (Assinaturas de eventos).
3. No painel Event subscriptions (Assinaturas de eventos), escolha a assinatura que deseja modificar e escolha Edit (Editar).
4. Faça as alterações na assinatura usando as seções Target (Alvo) ou Source (Origem).
5. Selecione a opção Editar. O console do Amazon RDS indica que a assinatura está sendo modificada.



AWS CLI

Para modificar uma assinatura de notificação de evento do Amazon RDS, use o comando da AWS CLI `modify-event-subscription`. Inclua o seguinte parâmetro necessário:

- `--subscription-name`

Example

O código a seguir habilita `myeventsubscription`.

Para Linux, macOS ou Unix:

```
aws rds modify-event-subscription \
    --subscription-name myeventsubscription \
    --enabled
```

Para Windows:

```
aws rds modify-event-subscription ^
    --subscription-name myeventsubscription ^
    --enabled
```

API

Para modificar um evento do Amazon RDS, chame a operação da API do Amazon RDS `ModifyEventSubscription`. Inclua o seguinte parâmetro necessário:

- `SubscriptionName`

Adicionar um identificador de origem a uma assinatura de notificação de evento do Amazon RDS

Você pode adicionar um identificador de origem (a origem do Amazon RDS que gera o evento) à uma assinatura existente.

Console

Você pode facilmente adicionar ou remover identificadores de origem usando o console do Amazon RDS selecionando-os ou desmarcando-os ao modificar uma assinatura. Para obter mais informações, consulte Modificar uma assinatura de notificação de evento do Amazon RDS.

AWS CLI

Para adicionar um identificador de origem a uma assinatura de notificação de eventos do Amazon RDS, use o comando da AWS CLI `add-source-identifier-to-subscription`. Inclua os seguintes parâmetros necessários:

- `--subscription-name`
- `--source-identifier`

Example

O exemplo a seguir adiciona o identificador de origem `mysqldb` à assinatura `myrdseventsubscription`

Para Linux, macOS ou Unix:

```
aws rds add-source-identifier-to-subscription \
    --subscription-name myrdseventsubscription \
    --source-identifier mysqldb
```

Para Windows:

```
aws rds add-source-identifier-to-subscription ^
    --subscription-name myrdseventsubscription ^
    --source-identifier mysqldb
```

API

Para adicionar um identificador de origem a uma assinatura de notificação de evento do Amazon RDS, chame a API do Amazon RDS `AddSourceIdentifierToSubscription`. Inclua os seguintes parâmetros necessários:

- `SubscriptionName`
- `SourceIdentifier`

Remover um identificador de origem de uma assinatura de notificação de eventos do Amazon RDS

Você pode remover um identificador de origem (a origem do Amazon RDS que gera o evento) de uma assinatura se você não quiser mais ser notificado sobre eventos para aquela origem.

Console

Você pode facilmente adicionar ou remover identificadores de origem usando o console do Amazon RDS selecionando-os ou desmarcando-os ao modificar uma assinatura. Para obter mais informações, consulte Modificar uma assinatura de notificação de evento do Amazon RDS.

AWS CLI

Para remover um identificador de origem de uma assinatura de notificação de eventos do Amazon RDS, use o comando da AWS CLI `remove-source-identifier-from-subscription`. Inclua os seguintes parâmetros necessários:

- `--subscription-name`
- `--source-identifier`

Example

O exemplo a seguir remove o identificador de origem `mysqldb` da assinatura `myrdseventsubscription`.

Para Linux, macOS ou Unix:

```
aws rds remove-source-identifier-from-subscription \
    --subscription-name myrdseventsubscription \
    --source-identifier mysqldb
```

Para Windows:

```
aws rds remove-source-identifier-from-subscription ^
    --subscription-name myrdseventsubscription ^
    --source-identifier mysqldb
```

API

Para remover um identificador de origem de uma assinatura de notificação de eventos do Amazon RDS, use o comando `RemoveSourceIdentifierFromSubscription` da API do Amazon RDS. Inclua os seguintes parâmetros necessários:

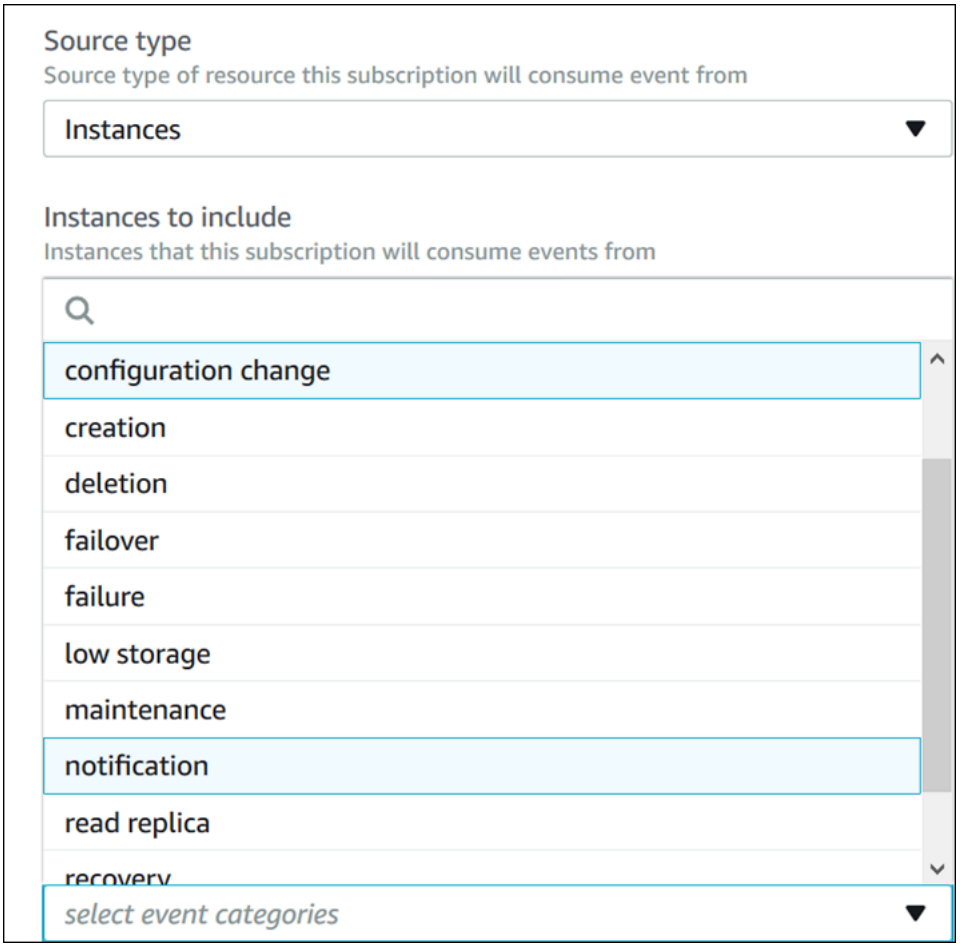
- `SubscriptionName`
- `SourceIdentifier`

Listar as categorias de notificação de evento do Amazon RDS

Todos os eventos para um tipo de recurso são agrupados em categorias. Para visualizar a lista de categorias disponíveis, use os seguintes procedimentos.

Console

Quando você cria ou modifica uma assinatura de notificação de evento, as categorias de eventos são exibidas no console do Amazon RDS. Para obter mais informações, consulte Modificar uma assinatura de notificação de evento do Amazon RDS.



AWS CLI

Para listar as categorias de notificação de evento do Amazon RDS, use o comando da AWS CLI `describe-event-categories`. Esse comando não possui parâmetros necessários.

Example

```
aws rds describe-event-categories
```

API

Para listar as categorias de notificação de evento do Amazon RDS, use o comando `DescribeEventCategories` da API do Amazon RDS. Esse comando não possui parâmetros necessários.

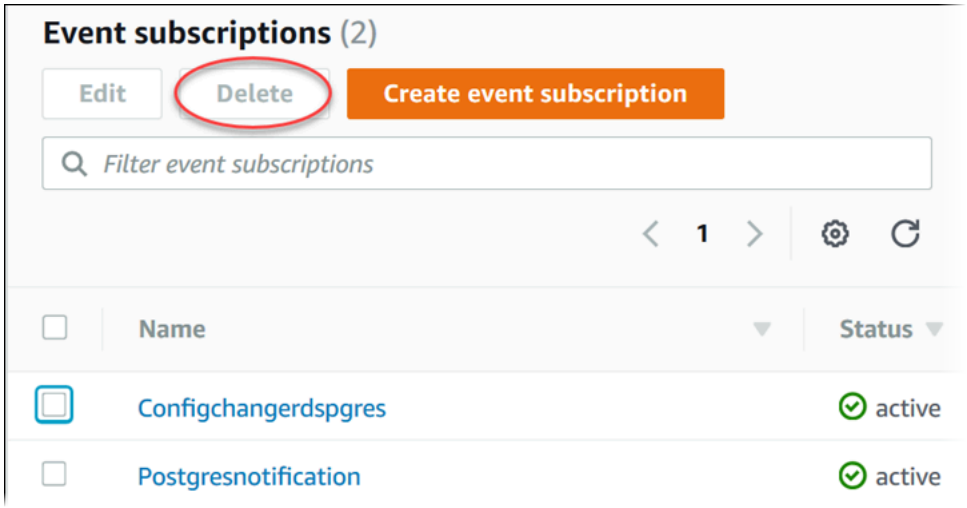
Excluir uma assinatura de notificação de evento do Amazon RDS

É possível excluir uma assinatura quando não precisar mais dela. Todos os assinantes do tópico não receberão mais notificações de evento especificadas pela assinatura.

Console

Para excluir uma assinatura de notificação de evento do Amazon RDS

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha DB Event Subscriptions (Assinaturas de eventos de banco de dados).
3. No painel My DB Event Subscriptions (Minhas assinaturas de eventos de banco de dados), escolha a assinatura que deseja excluir.
4. Escolha Excluir.
5. O console do Amazon RDS indica que a assinatura está sendo excluída.



AWS CLI

Para excluir uma assinatura de notificação de evento do Amazon RDS, use o comando da AWS CLI `delete-event-subscription`. Inclua o seguinte parâmetro necessário:

- `--subscription-name`

Example

O exemplo a seguir exclui a assinatura `myrdssubscription`.

```
aws rds delete-event-subscription --subscription-name myrdssubscription
```

API

Para excluir uma assinatura de notificação de evento do Amazon RDS, use o comando `DeleteEventSubscription` da API do RDS. Inclua o seguinte parâmetro necessário:

- `SubscriptionName`

Criar uma regra que é acionada em um evento do Amazon RDS

Usando o Amazon EventBridge, é possível automatizar serviços da AWS e responder aos eventos do sistema, como problemas de disponibilidade da aplicação ou alterações de recursos.

Tópicos

- Criar regras para enviar eventos do Amazon RDS ao Amazon EventBridge
- Tutorial: Registro de alterações de estado de uma instância de banco de dados usando o Amazon EventBridge

Criar regras para enviar eventos do Amazon RDS ao Amazon EventBridge

É possível gravar regras simples para indicar quais eventos do Amazon RDS são interessantes para você e quais ações automatizadas devem ser realizadas quando um evento corresponder a uma regra. É possível definir uma variedade de destinos, como uma função do AWS Lambda ou um tópico do Amazon SNS, que recebem eventos no formato JSON. Por exemplo, é possível configurar o Amazon RDS para enviar eventos ao Amazon EventBridge sempre que uma instância de banco de dados é criada ou excluída. Para obter mais informações, consulte o Guia do usuário de do Amazon CloudWatch Events e o Guia do usuário do Amazon EventBridge.

Para criar uma regra que é acionada em um evento do RDS:

1. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. Em Events (Eventos), no painel de navegação, escolha Rules (Regras).
3. Escolha Create rule (Criar regra).
4. Em Origem do evento, faça o seguinte:
    a. Escolha Event Pattern (Padrão de evento).
    b. Em Service Name (Nome do serviço), escolha Relational Database Service (RDS).
    c. Em Event Type (Tipo de evento), escolha o tipo de recurso do Amazon RDS que aciona o evento. Por exemplo, se uma instância de banco de dados acionar o evento, escolha RDS DB Instance Event (Evento de instância de banco de dados do RDS).
5. Em Targets (Destinos), escolha Add Target (Adicionar destino) e escolha o serviço da AWS que deve agir quando um evento do tipo selecionado é detectado.
6. Em outros campos nesta seção, insira informações específicas para esse tipo de destino, se necessário.

7. Para muitos tipos de destino, o Eventbridge precisa de permissões para enviar eventos ao destino. Nesses casos, o EventBridge pode criar o perfil do IAM necessário para o perfil ser executado:

   - Para criar um perfil do IAM automaticamente, escolha Create a new role for this specific resource.
   - Para usar um perfil do IAM que você criou antes, escolha Use existing role (Usar perfil existente).
8. Opcionalmente, repita as etapas de 5 a 7 para adicionar outro destino a esta regra.
9. Escolha Configure details (Configurar detalhes). Em Definição de regra, digite um nome e uma descrição para a regra.

   O nome da regra deve ser exclusivo dentro dessa região.
10. Escolha Create rule (Criar regra).

Para ter mais informações, consulte Creating an EventBridge Rule That Triggers on an Event no Guia do usuário do Amazon CloudWatch.

Tutorial: Registro de alterações de estado de uma instância de banco de dados usando o Amazon EventBridge

Neste tutorial, você pode criar uma função do AWS Lambda que registra as alterações de estado para uma instância do Amazon RDS. Depois, crie uma regra que execute a função sempre que houver uma alteração de estado de uma instância de banco de dados do RDS existente. O tutorial pressupõe que você tem uma pequena instância de teste em execução que você pode desligar temporariamente.

> **Important**
>
> Não execute este tutorial em uma instância de banco de dados de produção em execução.

Tópicos

- Etapa 1: Criar uma função do AWS Lambda
- Etapa 2: Criar uma regra
- Etapa 3: Testar a regra

Etapa 1: Criar uma função do AWS Lambda

Crie uma função Lambda para registrar em log os eventos de alteração de estado. Você especifica essa função quando cria sua regra.

Como criar uma função do Lambda

1. Abra o console AWS Lambda em https://console.aws.amazon.com/lambda/.
2. Se você estiver começando a usar o Lambda, verá uma página de boas-vindas. Escolha Get Started Now (Começar agora). Do contrário, escolha Create function (Criar função).
3. Escolha Author from scratch.
4. Na página Create function (Criar função), faça o seguinte:
   a. Digite um nome e uma descrição para a função Lambda. Por exemplo, atribua à função o nome **`RDSInstanceStateChange`**.
   b. Em Runtime (Tempo de execução), selecione Node.js 14x.
   c. Em Architecture (Arquitetura), escolha x86_64.
   d. Em Execution role (Perfil de execução), realize um dos seguintes procedimentos:
      - Escolha Create a new role with basic Lambda permissions (Criar uma nova função com permissões básicas do Lambda).
      - Em Existing role (Perfil existente), escolha Use an existing role (Usar um perfil existente). Escolha o perfil que deseja usar.
   e. Escolha Create function.
5. Na página RDSinstAncestateChange, faça o seguinte:
   a. Em Code source (Fonte do código), selecione index.js.
   b. No painel de index.js, exclua o código existente.
   c. Insira o seguinte código:

```
console.log('Loading function');

exports.handler = async (event, context) => {
    console.log('Received event:', JSON.stringify(event));
};
```

   d. Escolha Deploy (Implantar).

Etapa 2: Criar uma regra

Crie uma regra para executar sua função do Lambda sempre que executar uma instância do Amazon RDS.

Como criar a regra do EventBridge

1. Abra o console do Amazon EventBridge em https://console.aws.amazon.com/events/.
2. No painel de navegação, escolha Regras.
3. Escolha Create rule (Criar regra).
4. Insira um nome e uma descrição para a regra. Por exemplo, digite **`RDSInstanceStateChangeRule`**.
5. Escolha Rule with an event pattern (Regra com padrão de eventos), depois selecione Next (Próximo).
6. Em Event source (Origem do evento), selecione Eventos da AWS ou eventos de parceiro do EventBridge.
7. Role para baixo até a seção Event pattern (Padrão de eventos).
8. Em Event source, escolha Serviços da AWS.
9. Em Serviço da AWS, escolha Relational Database Service (RDS).
10. Em Event type (Tipo de evento), escolha RDS DB Instance Event (Evento de instância de banco de dados do RDS).
11. Mantenha o padrão de eventos predefinido. Em seguida, escolha Próximo.
12. Em Tipos de destino, escolha Serviço da AWS.
13. Em Select a target (Selecionar um destino), escolha Lambda function (Função do Lambda).
14. Em Function (Função), selecione a função do Lambda que você criou. Em seguida, escolha Próximo.
15. Em Configure tags (Configurar etiquetas), escolha Next (Próximo).
16. Revise as etapas da sua regra. Em seguida, escolha Create rule (Criar regra).

Etapa 3: Testar a regra

Para testar sua regra, desligue uma instância de banco de dados do RDS. Depois de esperar alguns minutos para a instância ser inicializada e executada, verifique se a sua função do Lambda foi chamada.

Como testar a regra ao interromper uma instância de banco de dados

1. Abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. Interrupção de uma instância de banco de dados do RDS.
3. Abra o console do Amazon EventBridge em https://console.aws.amazon.com/events/.
4. No painel de navegação, escolha Rules (Regras) e escolha o nome da regra criada por você.
5. Em Detalhes da regra, escolha Monitoramento.

   O sistema redireciona você para o console do Amazon CloudWatch. Se você não for redirecionado, clique em Visualizar as métricas no CloudWatch.
6. Em All metrics (Todas as métricas), escolha o nome da regra que você criou.

   O gráfico deve indicar que a regra foi invocada.
7. No painel de navegação, escolha Log groups (Grupos de logs).
8. Escolha o nome do grupo de logs para a sua função do Lambda (/aws/lambda/***nome-da-função***).
9. Escolha o nome do fluxo de logs para visualizar os dados fornecidos pela função para a instância que você iniciou. Será exibido um resultado semelhante ao seguinte:

```
{
    "version": "0",
    "id": "12a345b6-78c9-01d2-34e5-123f4ghi5j6k",
    "detail-type": "RDS DB Instance Event",
    "source": "aws.rds",
    "account": "111111111111",
    "time": "2021-03-19T19:34:09Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:rds:us-east-1:111111111111:db:testdb"
    ],
    "detail": {
        "EventCategories": [
            "notification"
        ],
        "SourceType": "DB_INSTANCE",
        "SourceArn": "arn:aws:rds:us-east-1:111111111111:db:testdb",
        "Date": "2021-03-19T19:34:09.293Z",
        "Message": "DB instance stopped",
        "SourceIdentifier": "testdb",
```

```
        "EventID": "RDS-EVENT-0087"
    }
}
```

Para obter mais exemplos de eventos do RDS no formato JSON, consulte Visão geral dos eventos para Amazon RDS.

10. (Opcional) Ao terminar, você poderá abrir o console do Amazon RDS e iniciar a instância interrompida.

Categorias de eventos e mensagens de eventos do Amazon RDS

O Amazon RDS gera um número significativo de eventos em categorias nas quais você pode fazer uma assinatura usando o console do Amazon RDS, a AWS CLI ou a API.

Tópicos

- Eventos de cluster de banco de dados
- Eventos de instância de banco de dados
- Eventos de grupo de parâmetros de banco de dados
- Eventos de grupos de segurança de banco de dados
- Eventos de snapshot de banco de dados
- Eventos de snapshot de cluster de banco de dados
- Eventos do RDS Proxy
- Eventos de implantação azul/verde
- Eventos de versão de mecanismos personalizados

Eventos de cluster de banco de dados

A tabela a seguir mostra a categoria de evento e uma lista de eventos quando um cluster de banco de dados é o tipo de origem.

Para ter mais informações sobre Implantações de cluster de banco de dados multi-AZ, consulte Implantações de cluster de banco de dados multi-AZ para o Amazon RDS.

Categoria	ID do evento do RDS	Message	Observações
alteração de configuração	RDS-EVENT-0016	Redefina as credenciais principais.	
criação	RDS-EVENT-0170	Cluster de banco de dados criado.	
failover	RDS-EVENT-0069	Falha no failover do cluster, verifique a integridade das	

Categoria	ID do evento do RDS	Message	Observações
		instâncias do cluster e tente novamente.	
failover	RDS-EVENT-0070	Promover novamente a primária anterior: *nome*.	
failover	RDS-EVENT-0071	Failover concluído para a instância de banco de dados: *nome*.	
failover	RDS-EVENT-0072	Iniciou o mesmo failover da AZ para a instância de banco de dados: *nome*.	
failover	RDS-EVENT-0073	Iniciou o failover cruzado da AZ para a instância de banco de dados: *nome*.	
falha	RDS-EVENT-0354	Não é possível criar o cluster de banco de dados devido a recursos incompatíveis. *mensagem*.	A *mensagem* inclui detalhes sobre a falha.
falha	RDS-EVENT-0355	O cluster de banco de dados não pode ser criado devido a limites insuficientes de recursos. *mensagem*.	A *mensagem* inclui detalhes sobre a falha.

Categoria	ID do evento do RDS	Message	Observações
Failover global	RDS-EVENT-0181	A transição global para o cluster de banco de dados *nome* na região *nome* foi iniciada.	Esse evento destina-se a uma operação de transição (anteriormente chamada de “failover planejado gerenciado”). O processo pode ser atrasado devido à execução de outras operações no cluster de banco de dados.
Failover global	RDS-EVENT-0182	O antigo cluster de banco de dados primário *nome* na região *nome* foi encerrado com sucesso.	Esse evento destina-se a uma operação de transição (anteriormente chamada de “failover planejado gerenciado”). A instância primária antiga no banco de dados global não aceita gravações. Todos os volumes são sincronizados.
Failover global	RDS-EVENT-0183	Aguardar a sincroniz ação de dados entre os membros do cluster global. Atrasos atuais em relação ao cluster de banco de dados primário: *motivo*.	Esse evento destina-se a uma operação de transição (anteriormente chamada de “failover planejado gerenciado”). Está ocorrendo um atraso de replicação durante a fase de sincronização do failover de banco de dados global.

Categoria	ID do evento do RDS	Message	Observações
Failover global	RDS-EVENT-0184	O novo do cluster de banco de dados primário *nome* na região *nome* foi promovido com sucesso.	Esse evento destina-se a uma operação de transição (anteriormente chamada de “failover planejado gerenciado”). A topologia de volumes do banco de dados global é restabelecida com o novo volume primário.
Failover global	RDS-EVENT-0185	A transição global para o cluster de banco de dados *nome* na região *nome* foi concluída.	Esse evento destina-se a uma operação de transição (anteriormente chamada de “failover planejado gerenciado”). A transição de banco de dados global está concluída no cluster de banco de dados primário. Réplicas podem demorar muito para ficarem on-line após a conclusão do failover.
Failover global	RDS-EVENT-0186	A transição global para o cluster de banco de dados *nome* na região *nome* foi cancelada.	Esse evento destina-se a uma operação de transição (anteriormente chamada de “failover planejado gerenciado”).

Categoria	ID do evento do RDS	Message	Observações
Failover global	RDS-EVENT-0187	A transição global para o cluster de banco de dados *nome* na região *nome* apresentou falha.	Esse evento destina-se a uma operação de transição (anteriormente chamada de “failover planejado gerenciado”).
Failover global	RDS-EVENT-0238	O failover global para o cluster de banco de dados *nome* na região *nome* foi concluído.	
Failover global	RDS-EVENT-0239	O failover global para o cluster de banco de dados *nome* na região *nome* falhou.	
Failover global	RDS-EVENT-0240	Começou a ressincronizar membros do cluster de banco de dados *nome* na região *nome* após o failover global.	
Failover global	RDS-EVENT-0241	Terminou de ressincronizar membros do cluster de banco de dados *nome* na região *nome* após o failover global.	
manutenção	RDS-EVENT-0156	O cluster de banco de dados tem uma atualizaç ão de versão secundária do mecanismo de banco de dados disponível.	

Categoria	ID do evento do RDS	Message	Observações
manutenção	RDS-EVENT-0176	A versão principal do mecanismo do cluster de banco de dados foi atualizada.	
manutenção	RDS-EVENT-0177	A atualização do cluster de banco de dados está em andamento.	
manutenção	RDS-EVENT-0286	A atualização da versão do mecanismo de cluster de banco de dados foi iniciada.	
manutenção	RDS-EVENT-0287	Requisito de atualização do sistema operacional detectado.	
manutenção	RDS-EVENT-0288	Início da atualização do sistema operacional do cluster.	
manutenção	RDS-EVENT-0289	Atualização do sistema operacional de cluster concluída.	
notificação	RDS-EVENT-0172	Cluster renomeado de *nome* para *nome*.	

Eventos de instância de banco de dados

As tabelas a seguir mostram a categoria de evento e uma lista de eventos quando uma instância de banco de dados é o tipo de origem.

Categoria	ID do evento do RDS	Message	Observações
disponibilidade	RDS-EVENT-0004	Desligamento da instância de banco de dados.	
disponibilidade	RDS-EVENT-0006	A instância de banco de dados foi reiniciada.	
disponibilidade	RDS-EVENT-0022	Erro ao reiniciar o mysql: *mensagem*.	Ocorreu um erro ao reiniciar o MySQL.
disponibilidade	RDS-EVENT-0221	A instância de banco de dados atingiu o limite total de armazenamento e o banco de dados foi encerrado. É possível aumentar o armazenamento alocado para resolver o problema.	
disponibilidade	RDS-EVENT-0222	A capacidade de armazenamento livre para a instância de banco de dados *nome* está baixa em *porcentagem* do armazenamento alocado [Armazenamento alocado: *quantidade* , Armazenamento livre: *quantidade*]. O banco de dados será encerrado para evitar corrupção se o armazenamento livre for menor que *valor*. É possível aumentar o	Para ter mais informações, consulte Armazenamento de instâncias de banco de dados do Amazon RDS.

Categoria	ID do evento do RDS	Message	Observações
		armazenamento alocado para resolver o problema.	
disponibilidade	RDS-EVENT-0330	A capacidade de armazenamento livre do volume dedicado de logs de transações está muito baixa para a instância de banco de dados *nome*. O armazenamento livre do volume de logs é a *porcentagem* do armazenamento alocado. [Armazenamento alocado: *quantidade*, Armazenamento livre: *quantidade*] O banco de dados será encerrado para evitar corrupção se o armazenamento livre for menor que *quantidade*. É possível desabilitar o volume dedicado de logs de transações para resolver esse problema.	Para ter mais informações, consulte Volume de log dedicado (DLV).

Categoria	ID do evento do RDS	Message	Observações
disponibilidade	RDS-EVENT-0331	A capacidade de armazenamento livre do volume dedicado de logs de transações está muito baixa para a instância de banco de dados *nome*. O armazenamento livre do volume de logs é a *porcentagem* do armazenamento provisionado. [Armazenamento provisionado: *quantidade* , Armazenamento livre: *quantidade*] Você pode desabilitar o volume dedicado de logs de transações para resolver esse problema.	Para ter mais informações, consulte Volume de log dedicado (DLV).
disponibilidade	RDS-EVENT-0396	O Amazon RDS programou uma reinicialização dessa réplica de leitura na próxima janela de manutenção dessa instância após a alternância interna da senha do usuário.	Não aplicável
backup	RDS-EVENT-0001	Fazendo backup da instância de banco de dados.	
backup	RDS-EVENT-0002	Backup da instância do banco de dados concluído.	

Categoria	ID do evento do RDS	Message	Observações
backup	RDS-EVENT-0086	Não foi possível associar o grupo de opções *nome* ao *nome* da instância de banco de dados. Confirme se o *nome* do grupo de opções é permitido na classe e na configuração da instância de banco de dados. Se for esse o caso, verifique todas as configurações do grupo de opções e tente novamente.	Para ter mais informações, consulte Trabalhar com grupos de opções.
alteração de configuração	RDS-EVENT-0011	Atualização para usar DBParameterGroup *name*.	
alteração de configuração	RDS-EVENT-0012	Aplicando modificação à classe de instância de banco de dados.	
alteração de configuração	RDS-EVENT-0014	Conclusão da aplicação de modificação à classe de instância de banco de dados.	
alteração de configuração	RDS-EVENT-0016	Redefina as credenciais principais.	
alteração de configuração	RDS-EVENT-0017	Concluída aplicação de modificação ao armazenamento alocado.	

Categoria	ID do evento do RDS	Message	Observações
alteração de configuração	RDS-EVENT-0018	Aplicação de modificação ao armazenamento alocado.	
alteração de configuração	RDS-EVENT-0024	Aplicação de modificação para converter em uma instância de banco de dados multi-AZ.	
alteração de configuração	RDS-EVENT-0025	Concluída a aplicação de modificação para converter em uma instância de banco de dados multi-AZ.	
alteração de configuração	RDS-EVENT-0028	Backups automáticos desabilitados.	
alteração de configuração	RDS-EVENT-0029	Concluída a aplicação da modificação para converter em uma instância de banco de dados padrão (single-AZ).	
alteração de configuração	RDS-EVENT-0030	Aplicação de modificação para converter em uma instância de banco de dados padrão (single-AZ).	
alteração de configuração	RDS-EVENT-0032	Backups automáticos habilitados.	

Categoria	ID do evento do RDS	Message	Observações
alteração de configuração	RDS-EVENT-0033	Há *número* usuários que correspondem ao nome de usuário principal; apenas redefinindo aquele que não está vinculado a um host específico.	
alteração de configuração	RDS-EVENT-0067	Não foi possível redefinir sua senha. Informações de erro: *mensagem*.	
alteração de configuração	RDS-EVENT-0078	Intervalo de monitoramento alterado para *número*.	A configuração de Monitoramento avançado foi alterada.
alteração de configuração	RDS-EVENT-0092	Conclusão da atualização do grupo de parâmetros do banco de dados.	
alteração de configuração	RDS-EVENT-0217	Aplicação de modificação iniciada em escalabilidade automática ao armazenamento alocado.	
alteração de configuração	RDS-EVENT-0218	Concluída aplicação de modificação iniciada em escalabilidade automática ao armazenamento alocado.	
alteração de configuração	RDS-EVENT-0237	Atualização da configuração de armazenamento iniciada.	

Categoria	ID do evento do RDS	Message	Observações
alteração de configuração	RDS-EVENT-0237	Atualização da configuração de armazenamento concluída.	
alteração de configuração	RDS-EVENT-0332	O volume de logs dedicado está desabilitado.	Para ter mais informações, consulte Volume de log dedicado (DLV).
alteração de configuração	RDS-EVENT-0333	A desabilitação do volume de logs dedicado foi iniciada.	Para ter mais informações, consulte Volume de log dedicado (DLV).
alteração de configuração	RDS-EVENT-0334	A habilitação do volume de logs dedicado foi iniciada.	Para ter mais informações, consulte Volume de log dedicado (DLV).
alteração de configuração	RDS-EVENT-0335	O volume de logs dedicado está habilitado.	Para ter mais informações, consulte Volume de log dedicado (DLV).
alteração de configuração	RDS-EVENT-0383	A *versão do mecanismo* não comporta o plug-in memcached. O RDS continuará atualizando sua instância de banco de dados e removerá esse plug-in.	A partir do MySQL 8.3.0, o plug-in memcached não é compatível. Para ter mais informações, consulte Changes in MySQL 8.3.0 (2024-01-16, Innovation Release).
criação	RDS-EVENT-0005	Instância de banco de dados criada.	
exclusão	RDS-EVENT-0003	A instância de banco de dados foi excluída.	

Categoria	ID do evento do RDS	Message	Observações
failover	RDS-EVENT-0013	Failover de instância multi-AZ iniciado.	Um failover de multi-AZ que resultou na promoção de uma instância de banco de dados em espera foi iniciado.
failover	RDS-EVENT-0015	Failover de multi-AZ para o modo de espera concluído : a propagação do DNS pode levar alguns minutos.	Um failover de multi-AZ que resultou na promoção de uma instância de banco de dados em espera está concluído. Pode levar vários minutos para o DNS ser transferido para a nova instância primária do banco de dados.
failover	RDS-EVENT-0034	Abandonando o failover solicitado pelo usuário, pois ocorreu um failover recentemente na instância do banco de dados.	O Amazon RDS não está tentando um failover solicitado porque ocorreu um failover recentemente na instância de banco de dados.
failover	RDS-EVENT-0049	O failover da instância multi-AZ foi concluído.	
failover	RDS-EVENT-0050	A ativação da instância multi-AZ foi iniciada.	Uma ativação multi-AZ foi iniciada após uma recuperação bem-sucedida da instância de banco de dados

Categoria	ID do evento do RDS	Message	Observações
failover	RDS-EVENT-0051	A ativação da instância multi-AZ foi concluída.	Uma ativação multi-AZ está completa. Seu banco de dados deve estar acessível agora.
failover	RDS-EVENT-0065	Recuperação de um failover parcial.	
falha	RDS-EVENT-0031	Instância de banco de dados colocada no estado *nome*. O RDS recomenda que você inicie uma restauração para um ponto no tempo.	Houve falha na instância de banco de dados devido a uma configuração incompatível ou a um problema do armazenam ento subjacente. Inicie um point-in-time-restore para a instância de banco de dados.
falha	RDS-EVENT-0035	Instância de banco de dados colocada no *estado*. *mensagem*.	A instância de banco de dados tem parâmetros inválidos. Por exemplo, se a instância de banco de dados não pôde ser iniciada porque um parâmetro relacionado à memória está definido como um valor muito alto para essa classe de instância, a ação seria modificar o parâmetro da memória e reinicializar a instância de banco de dados.

Categoria	ID do evento do RDS	Message	Observações
falha	RDS-EVENT-0036	Instância do banco de dados em *estado*. *mensagem*.	A instância de banco de dados está em uma rede incompatível. Alguns dos IDs de sub-rede especific ados são inválidos ou não existem.
falha	RDS-EVENT-0058	A instalação do Statspack falhou. *mensagem*.	Erro ao criar a conta de usuário PERFSTAT do Oracle Statspack. Descarte a conta antes de adicionar a opção STATSPACK .
falha	RDS-EVENT-0079	O Amazon RDS não conseguiu criar credencia is para o monitoram ento avançado e esse recurso foi desabilitado. Provavelmente, isso ocorre porque rds-monitoring-rol e não está presente e foi configurado corretamente em sua conta. Consulte a seção de solução de problemas na documenta ção do Amazon RDS para obter mais detalhes.	O Monitoramento avançado não pode ser habilitado sem o perfil do IAM de monitoramento avançado. Para obter informações sobre como criar o perfil do IAM, consulte Como criar uma função do IAM para o monitoramento avançado do Amazon RDS.

Categoria	ID do evento do RDS	Message	Observações
falha	RDS-EVENT-0080	O Amazon RDS não conseguiu configurar o monitoramento avançado em sua instância: *nome* e esse recurso foi desabilitado. Provavelmente, isso ocorre porque rds-monitoring-role não está presente e foi configurado corretamente em sua conta. Consulte a seção de solução de problemas na documentação do Amazon RDS para obter mais detalhes.	O Monitoramento avançado foi desabilitado porque ocorreu um erro durante a alteração da configuração. É provável que o perfil do IAM de monitoramento avançado esteja configurado incorretamente. Para obter informações sobre como criar o perfil do IAM de monitoramento avançado, consulte Como criar uma função do IAM para o monitoramento avançado do Amazon RDS.
falha	RDS-EVENT-0081	O Amazon RDS não conseguiu criar credenciais para a opção *nome*. Isso ocorre porque o perfil do IAM *nome* não está configurado corretamente em sua conta. Consulte a seção de solução de problemas na documentação do Amazon RDS para obter mais detalhes.	O perfil do IAM que você usa para acessar seu bucket do Amazon S3 para backup e restauração nativos do SQL Server está configurado incorretamente. Para ter mais informações, consulte Configurar o backup e a restauração nativos.

Categoria	ID do evento do RDS	Message	Observações
falha	RDS-EVENT-0165	A instância de banco de dados do RDS Custom está fora do perímetro de suporte.	É sua responsabilidade corrigir problemas de configuração que colocam sua instância de banco de dados do RDS Custom no estado `unsupported-configuration`. Se o problema for com a infraestrutura da AWS, será possível utilizar o console ou a AWS CLI para corrigi-lo. Se o problema for com o sistema operacional ou a configuração do banco de dados, será possível fazer login no host para corrigi-lo. Para ter mais informações, consulte Perímetro de suporte do RDS Custom.

Categoria	ID do evento do RDS	Message	Observações
falha	RDS-EVENT-0188	A instância de banco de dados está em um estado que não pode ser atualizado. *mensagem*	O Amazon RDS não conseguiu atualizar uma instância de banco de dados MySQL da versão 5.7 para a versão 8.0 devido a incompatibilidades relacionadas ao dicionário de dados. A instância de banco de dados foi revertida para o MySQL versão 5.7. Para ter mais informações, consulte Reversão após falha na atualização.
falha	RDS-EVENT-0219	A instância de banco de dados está em um estado inválido. Nenhuma ação é necessária. A escalabilidade automática tentará novamente mais tarde.	

Categoria	ID do evento do RDS	Message	Observações
falha	RDS-EVENT-0220	A instância de banco de dados está no período de desaquecimento de uma operação de armazenamento em escala anterior. Estamos otimizando a instância de banco de dados. Isso leva pelo menos seis horas. Nenhuma ação é necessária. A escalabil idade automática tentará novamente após o período de desaquecimento.	
falha	RDS-EVENT-0223	O ajuste de escala automático do armazenamento não pode ajudar a escala do armazenamento pelo seguinte motivo: *`motivo`*.	
falha	RDS-EVENT-0224	O ajuste de escala automático do armazenam ento acionou uma tarefa de armazenamento de escala pendente que atingirá ou excederá o limite máximo de armazenamento. Aumente o limite máximo do armazenamento.	

Categoria	ID do evento do RDS	Message	Observações
falha	RDS-EVENT-0237	A instância de banco de dados tem um tipo de armazenamento que no momento está indisponível na zona de disponibilidade. A escalabilidade automátic a tentará novamente mais tarde.	
falha	RDS-EVENT-0254	A cota de armazenam ento subjacente para essa conta de cliente excedeu o limite. Aumente a cota de armazenamento permitida para que o ajuste de escala continue na instância.	
falha	RDS-EVENT-0278	Falha na criação da instância de banco de dados. *mensagem*	A *mensagem* inclui detalhes sobre a falha.
falha	RDS-EVENT-0279	Falha na promoção da réplica de leitura do RDS Custom. *mensagem*	A *mensagem* inclui detalhes sobre a falha.
falha	RDS-EVENT-0280	O RDS Custom não conseguiu fazer upgrade da instância de banco de dados porque a pré-verif icação falhou. *mensagem*	A *mensagem* inclui detalhes sobre a falha.

Categoria	ID do evento do RDS	Message	Observações
falha	RDS-EVENT-0281	O RDS Custom não conseguiu modificar a instância de banco de dados porque a pré-verif icação falhou. *mensagem*	A *mensagem* inclui detalhes sobre a falha.
falha	RDS-EVENT-0282	O RDS Custom não conseguiu modificar a instância de banco de dados porque as permissõe s de IP elástico não estão corretas. Confirme se o endereço IP elástico está marcado com `AWSRDSCus tom` .	
falha	RDS-EVENT-0283	O RDS Custom não conseguiu modificar a instância de banco de dados porque o limite de IPs elásticos foi atingido em sua conta. Libere IPs elásticos não utilizados ou solicite um aumento de cota para seu limite de endereços IP elásticos.	
falha	RDS-EVENT-0284	O RDS Custom não conseguiu converter a instância para alta disponibi lidade porque a pré-verif icação falhou. *mensagem*	A *mensagem* inclui detalhes sobre a falha.

Categoria	ID do evento do RDS	Message	Observações
falha	RDS-EVENT-0285	O RDS Custom não conseguiu criar um snapshot final para a instância de banco de dados porque *mensagem*.	A *mensagem* inclui detalhes sobre a falha.
falha	RDS-EVENT-0237	Falha na atualização da configuração de armazenamento. Tente fazer o upgrade novamente .	
falha	RDS-EVENT-0237	Não é possível mover o banco de dados de rede incompatível,*nome*, para o status disponível: *mensagem*	A configuração de rede do banco de dados é inválida. Não foi possível mover o banco de dados da rede incompatível para a disponível.
falha	RDS-EVENT-0328	Falha ao ingressar um host a um domínio. O status de associação ao domínio, por exemplo, *nomedainstância* foi definido como “Com falha”.	

Categoria	ID do evento do RDS	Message	Observações
falha	RDS-EVENT-0329	Falha ao ingressar um host no domínio. Durante o processo de ingresso no domínio, o Microsoft Windows retornou a *mensagem* do código de erro. Verifique as configurações de rede e permissão e emita uma solicitação `modify-db-instance` para tentar ingressar no domínio novamente.	Ao usar um Active Directory autogerenciado, consulte Solução de problemas de Active Directory autogerenciado.
falha	RDS-EVENT-0353	A instância de banco de dados não pode ser criada devido a limites insuficientes de recursos. *mensagem*.	A *mensagem* inclui detalhes sobre a falha.
falha	RDS-EVENT-0356	O RDS não conseguiu configurar o endpoint do Kerberos no domínio. Isso pode impedir a autenticação do Kerberos para a instância de banco de dados. Verifique a configuração de rede entre a instância de banco de dados e os controladores de domínio.	

Categoria	ID do evento do RDS	Message	Observações
armazenamento baixo	RDS-EVENT-0007	O armazenamento alocado foi esgotado. Aloque armazenamento adicional para resolver.	O armazenamento alocado para a instância de banco de dados foi consumido. Para resolver esse problema, aloque armazenamento adicional para a instância de banco de dados. Para ter mais informações, consulte Perguntas frequentes do RDS. Você pode monitorar o espaço de armazenamento para uma instância de banco de dados usando a métrica Free Storage Space (Espaço de armazenamento livre).
armazenamento baixo	RDS-EVENT-0089	A capacidade de armazenamento livre para instância de banco de dados: *nome* está baixa em *porcentagem* do armazenamento provisionado [Armazenamento provisionado: *tamanho*, armazenamento livre: *tamanho*]. Talvez você queira aumentar o armazenamento provisionado para resolver esse problema.	A instância de banco de dados consumiu mais de 90% do armazenamento alocado. Você pode monitorar o espaço de armazenamento para uma instância de banco de dados usando a métrica Free Storage Space (Espaço de armazenamento livre).

Categoria	ID do evento do RDS	Message	Observações
armazenamento baixo	RDS-EVENT-0227	O armazenamento do cluster do Aurora está arriscadamente baixo, com apenas *quantidade* terabytes restantes. Tome medidas para reduzir a carga de armazenamento no cluster.	O subsistema de armazenamento do Aurora está com pouco espaço.
manutenção	RDS-EVENT-0026	Aplicação de patches off-line à instância de banco de dados.	Está ocorrendo a manutenção offline da instância de banco de dados. Atualmente, a instância do banco de dados está indisponível.
manutenção	RDS-EVENT-0027	Concluída a aplicação de patches off-line na instância de banco de dados.	A manutenção offline da instância de banco de dados está completa. A instância de banco de dados já está disponível.
manutenção	RDS-EVENT-0047	Instância de banco de dados corrigida.	
manutenção	RDS-EVENT-0155	A instância de banco de dados tem uma atualização de versão secundária do mecanismo de banco de dados disponível.	
manutenção	RDS-EVENT-0178	A atualização da instância do banco de dados está em andamento.	

Categoria	ID do evento do RDS	Message	Observações
manutenção	RDS-EVENT-0264	A pré-verificação foi iniciada para a atualização da versão do mecanismo de banco de dados.	
manutenção	RDS-EVENT-0265	A pré-verificação foi concluída para a atualização da versão do mecanismo de banco de dados.	
manutenção	RDS-EVENT-0266	O tempo de inatividade da instância de banco de dados começou.	
manutenção	RDS-EVENT-0267	A atualização da versão do mecanismo foi iniciada.	
manutenção	RDS-EVENT-0268	A atualização da versão do mecanismo foi concluída.	
manutenção	RDS-EVENT-0269	As tarefas pós-atualização estão em andamento.	
manutenção	RDS-EVENT-0270	A atualização da versão do mecanismo de banco de dados falhou. A reversão da atualização da versão do mecanismo foi bem-sucedida.	

Categoria	ID do evento do RDS	Message	Observações
manutenção, falha	RDS-EVENT-0195	*mensagem*	Falha na atualização do arquivo de fuso horário do Oracle. Para ter mais informações, consulte Atualização automática do arquivo de fuso horário do Oracle.
manutenção, notificação	RDS-EVENT-0191	Uma nova versão do arquivo de fuso horário está disponível para atualização.	Se você atualizar o mecanismo de banco de dados do RDS para Oracle, o Amazon RDS gerará esse evento se você não tiver escolhido uma atualização de arquivo de fuso horário e o banco de dados não usar o arquivo de fuso horário de verão mais recente disponíve l na instância. Para ter mais informações, consulte Atualização automática do arquivo de fuso horário do Oracle.
manutenção, notificação	RDS-EVENT-0192	A atualização do arquivo de fuso horário foi iniciada.	A atualização do arquivo de fuso horário do Oracle foi iniciada. Para ter mais informações, consulte Atualização automática do arquivo de fuso horário do Oracle.

Categoria	ID do evento do RDS	Message	Observações
manutenção, notificação	RDS-EVENT-0193	Nenhuma atualização está disponível para a versão atual do arquivo de fuso horário.	A instância de banco de dados Oracle está usando a versão mais recente do arquivo de fuso horário, e uma das seguintes afirmações é verdadeira: • Você adicionou recentemente a opção `TIMEZONE_FILE_AUTOUPGRADE` . • O mecanismo do banco de dados Oracle está sendo atualizado. Para ter mais informaçõ es, consulte Atualização automática do arquivo de fuso horário do Oracle.
manutenção, notificação	RDS-EVENT-0194	A atualização do arquivo de fuso horário foi concluída.	A atualização do arquivo de fuso horário do Oracle foi concluída. Para ter mais informações, consulte Atualização automática do arquivo de fuso horário do Oracle.
notificação	RDS-EVENT-0044	*mensagem*	Essa é uma notificação emitida pelo operador. Para ter mais informações, consulte a mensagem do evento.

Categoria	ID do evento do RDS	Message	Observações
notificação	RDS-EVENT-0048	Atraso na atualização do mecanismo de banco de dados, pois essa instância tem réplicas de leitura que precisam ser atualizadas primeiro.	O patch da instância de banco de dados foi atrasado.
notificação	RDS-EVENT-0054	*mensagem*	O mecanismo de armazenamento MySQL que você está usando não é o InnoDB, que é o mecanismo de armazenamento MySQL recomendado para o Amazon RDS. Para obter informações sobre os mecanismos de armazenamento do MySQL, consulte Suporte aos mecanismos de armazenamento no RDS para MySQL.

Categoria	ID do evento do RDS	Message	Observações
notificação	RDS-EVENT-0055	*mensagem*	O número de tabelas existentes para a sua instância de banco de dados excede as práticas recomendadas recomenda das para o Amazon RDS. Reduza o número de tabelas na sua instância de banco de dados. Para obter informações sobre as práticas recomenda das, consulte Diretrizes operacionais básicas do Amazon RDS.
notificação	RDS-EVENT-0056	*mensagem*	O número de bancos de dados existentes para a sua instância de banco de dados excede as práticas recomendadas recomendadas para o Amazon RDS. Reduza o número de bancos de dados na sua instância de banco de dados. Para obter informações sobre as práticas recomenda das, consulte Diretrizes operacionais básicas do Amazon RDS.

Categoria	ID do evento do RDS	Message	Observações
notificação	RDS-EVENT-0064	A chave de criptografia do TDE foi alternada com êxito.	Para obter informaçõ es sobre as práticas recomendadas, consulte Diretrizes operacionais básicas do Amazon RDS.
notificação	RDS-EVENT-0084	Não foi possível converter a instância de banco de dados em multi-AZ: *mensagem*.	Você tentou converter uma instância de banco de dados para multi-AZ, mas ela contém grupos de arquivos na memória que não têm suporte para multi-AZ. Para ter mais informações, consulte Implantações multi-AZ para o Amazon RDS for Microsoft SQL Server.
notificação	RDS-EVENT-0087	Instância de banco de dados foi interrompida.	
notificação	RDS-EVENT-0088	A instância de banco de dados iniciou.	
notificação	RDS-EVENT-0154	A instância de banco de dados está sendo iniciada porque excede o tempo máximo permitido para permanecer parada.	

Categoria	ID do evento do RDS	Message	Observações
notificação	RDS-EVENT-0157	Não foi possível modificar a classe da instância de banco de dados. *mensagem*.	O RDS não pode modificar a classe de instância de banco de dados porque a classe de instância de destino não comporta o número de bancos de dados existentes na instância de banco de dados de origem. A mensagem de erro é exibida como: "A instância possui N bancos de dados, mas, após a conversão, comportaria apenas N". Para ter mais informaçõ es, consulte Limitações das instâncias de banco de dados Microsoft SQL Server.
notificação	RDS-EVENT-0158	A instância de banco de dados está em um estado que não pode ser atualizad o: *mensagem*.	
notificação	RDS-EVENT-0167	*mensagem*	A configuração do perímetro de suporte do RDS Custom mudou.

Categoria	ID do evento do RDS	Message	Observações
notificação	RDS-EVENT-0189	Os créditos de saldo de intermitência gp2 para a instância de banco de dados do RDS estão baixos. Para resolver o problema, reduza o uso de IOPS ou modifique suas configurações de armazenamento para melhorar a performance.	Os créditos de saldo de intermitência gp2 para a instância de banco de dados do RDS estão baixos. Para resolver o problema, reduza o uso de IOPS ou modifique suas configurações de armazenamento para melhorar a performance. Para ter mais informaçõ es, consulte o tópico sobre créditos de E/S e performance de intermitê ncia, no Guia do usuário do Amazon Elastic Compute Cloud.
notificação	RDS-EVENT-0225	O tamanho do armazenam ento de *valor* GB está se aproximando do limite máximo de armazenam ento de *valor* GB. Aumente o limite máximo do armazenamento.	Esse evento é invocado quando o armazenam ento atinge 80% do limite máximo de armazenam ento. Para evitar o evento, aumente o limite máximo do armazenamento.

Categoria	ID do evento do RDS	Message	Observações
notificação	RDS-EVENT-0231	A modificação do armazenamento da instância de banco de dados encontrou um erro interno. A solicitação de modificação está pendente e será repetida posterior mente.	Ocorreu um erro no processo de replicação de leitura. Para ter mais informações, consulte a mensagem do evento. Além disso, consulte a seção de solução de problemas para réplicas de leitura para o mecanismo de banco de dados. • Solução de problemas da réplica de leitura do MariaDB • Solução de problemas de uma réplica de leitura do SQL Server • Solucionar problemas de uma réplica de leitura do MySQL • Solução de problemas de réplicas do RDS para Oracle

Categoria	ID do evento do RDS	Message	Observações
notificação	RDS-EVENT-0253	O banco de dados está usando o buffer de gravação dupla. *mensagem*. Para obter mais informações, consulte a documentação sobre gravações otimizadas do RDS para *nome*.	As gravações otimizadas pelo RDS são incompatíveis com a configuração de armazenamento da instância. Para ter mais informações, consulte Melhorar a performance das gravações com gravações otimizadas pelo RDS para MySQL e Melhorar a performance das gravações com gravações otimizadas pelo Amazon RDS para MariaDB. Você pode realizar a atualização da configuração de armazenamento para habilitar gravações otimizadas criando uma implantação azul/verde.

Categoria	ID do evento do RDS	Message	Observações
notificação	RDS-EVENT-0237	A configuração de armazenamento para o *nome* da instância de banco de dados suporta um tamanho máximo de 16384 GiB. Execute uma atualização da configuração de armazenamento para suportar tamanhos de armazenamento maiores que 16384 GiB.	Você não pode aumentar o tamanho de armazenamento alocado da instância de banco de dados além de 16384 GiB. Para superar essa limitação, realize um upgrade da configuração de armazenamento. Para ter mais informações, consulte Atualizando o sistema de arquivos de armazenamento para uma instância de banco de dados.
notificação	RDS-EVENT-0237	A configuração de armazenamento para o *nome* da instância de banco de dados suporta um tamanho máximo de tabela de 2048 GiB. Execute uma atualização da configuração de armazenamento para suportar tamanhos de tabela maiores que 2048 GiB.	As instâncias RDS MySQL e MariaDB com essa limitação não podem ter um tamanho de tabela superior a 2048 GiB. Para superar essa limitação, realize um upgrade da configuração de armazenamento. Para ter mais informações, consulte Atualizando o sistema de arquivos de armazenamento para uma instância de banco de dados.
notificação	RDS-EVENT-0327	*Mensagem*. O Amazon RDS não conseguiu encontrar o segredo *ARN DO SEGREDO*.	

Categoria	ID do evento do RDS	Message	Observações
réplica de leitura	RDS-EVENT-0045	A replicação foi interromp ida.	A replicação na instância de banco de dados foi interrompida devido ao armazenamento insuficie nte. Ajuste a escala do armazenamento ou reduza o tamanho máximo dos logs redo para permitir que a replicação continue. Para acomodar os logs de redo de tamanho *amount* MiB, é necessário ter pelo menos *amount* MiB de armazenamento livre.
réplica de leitura	RDS-EVENT-0046	A réplica de leitura foi retomada.	Essa mensagem aparece quando você cria uma réplica de leitura pela primeira vez, ou como uma mensagem de monitoram ento confirmando que a replicação está funcionan do corretamente. Se essa mensagem vem depois de uma notificação `RDS-EVENT-0045` , a replicaçã o foi retomada após um erro ou depois que a replicação foi interrompida.
réplica de leitura	RDS-EVENT-0057	O streaming de replicação foi encerrado.	

Categoria	ID do evento do RDS	Message	Observações
réplica de leitura	RDS-EVENT-0062	A replicação na réplica de leitura foi interrompida manualmente.	
réplica de leitura	RDS-EVENT-0063	A replicação de uma instância que não é do RDS foi redefinida.	
réplica de leitura	RDS-EVENT-0202	Falha na criação da réplica de leitura.	
réplica de leitura	RDS-EVENT-0357	O *nome* do canal de replicação foi iniciado.	Para ter informações sobre canais de replicação, consulte the section called "Configurar a replicação de várias fontes".
réplica de leitura	RDS-EVENT-0358	O *nome* do canal de replicação foi interrompido.	Para ter informações sobre canais de replicação, consulte the section called "Configurar a replicação de várias fontes".
réplica de leitura	RDS-EVENT-0359	O *nome* do canal de replicação foi interrompido manualmente.	Para ter informações sobre canais de replicação, consulte the section called "Configurar a replicação de várias fontes".
réplica de leitura	RDS-EVENT-0360	O *nome* do canal de replicação foi redefinido.	Para ter informações sobre canais de replicação, consulte the section called "Configurar a replicação de várias fontes".

Categoria	ID do evento do RDS	Message	Observações
recuperação	RDS-EVENT-0020	A recuperação da instância de banco de dados começou. O tempo de recuperação variará dependendo da quantidad e de dados a serem recuperados.	
recuperação	RDS-EVENT-0021	A recuperação da instância de banco de dados está completa.	
recuperação	RDS-EVENT-0023	Solicitação de snapshot emergente: *mensagem*.	Um backup manual foi solicitado, mas o Amazon RDS está no processo de criar um snapshot de banco de dados. Envie o pedido novamente depois que o Amazon RDS tiver concluído o snapshot de banco de dados.
recuperação	RDS-EVENT-0052	A recuperação da instância multi-AZ foi iniciada.	O tempo de recuperação variará dependendo da quantidade de dados a serem recuperados.
recuperação	RDS-EVENT-0053	A recuperação da instância multi-AZ foi concluída . Failover ou ativação pendentes.	

Categoria	ID do evento do RDS	Message	Observações
recuperação	RDS-EVENT-0066	A instância será degradada enquanto o espelhamento for restabelecido: *mensagem*.	A instância de banco de dados do SQL Server está restabelecendo seu espelho. O performance será degradado até o espelho ser restabelecido. Um banco de dados foi encontrado com o modelo de recuperação não FULL. O modelo de recuperação foi alterado de volta para FULL, e a recuperação de espelhamento foi iniciada. (<dbname>: <modelo de recuperação encontrado>[, …])"
recuperação	RDS-EVENT-0166	*mensagem*	A instância de banco de dados do RDS Custom está dentro do perímetro de suporte.
recuperação	RDS-EVENT-0361	A recuperação da instância de banco de dados em espera começou.	A instância de banco de dados em espera é recriada durante o processo de recuperação. O desempenho do banco de dados é afetado durante o processo de recuperação.

Categoria	ID do evento do RDS	Message	Observações
recuperação	RDS-EVENT-0362	A recuperação da instância de banco de dados em espera foi concluída.	A instância de banco de dados em espera é recriada durante o processo de recuperação. O desempenho do banco de dados é afetado durante o processo de recuperação.
restauração	RDS-EVENT-0019	Restauração da instância de banco de dados *nome* para *nome*.	A instância de banco de dados foi restaurada a partir de um backup point-in-time.
segurança	RDS-EVENT-0068	Descriptografando a senha da partição hsm para atualizar a instância.	O RDS está descriptografando a senha da partição do AWS CloudHSM para fazer atualizações na instância de banco de dados. Para obter mais informações, consulte Oracle Database Transparent Data Encryption (TDE) com o AWS CloudHSM, no Guia do usuário do AWS CloudHSM.

Categoria	ID do evento do RDS	Message	Observações
patches de segurança	RDS-EVENT-0230	A atualização do sistema está disponível para a instância de banco de dados. Para obter informações sobre como aplicar atualizações, consulte “Como manter uma instância de banco de dados” no Guia do usuário do RDS.	Uma nova atualização do sistema operacional está disponível. Uma nova atualização de versão secundária do sistema operacional está disponível para sua instância de banco de dados. Para obter informações sobre a aplicação de atualizações, consulte Atualizações do sistema operacional para instâncias de banco de dados do RDS.

Eventos de grupo de parâmetros de banco de dados

A tabela a seguir mostra a categoria de evento e uma lista de eventos quando um grupo de parâmetros de banco de dados é o tipo de origem.

Categoria	ID do evento do RDS	Message	Observações
alteração de configuração	RDS-EVENT-0037	Atualização do parâmetro *name* para *value* pelo método de aplicação *method*.	

Eventos de grupos de segurança de banco de dados

A tabela a seguir mostra a categoria de evento e uma lista de eventos quando um grupo de segurança de banco de dados é o tipo de origem.

> **Note**
>
> Grupos de segurança de banco de dados são recursos do EC2-Classic. O EC2-Classic foi removido em 15 de agosto de 2022. Se ainda não migrou do EC2-Classic para uma VPC, recomendamos que você migre o mais rápido possível. Para ter mais informações, consulte Migrate from EC2-Classic to a VPC (Migrar do EC2-Classic para uma VPC) no Guia do usuário do Amazon EC2 e o blog EC2-Classic Networking is Retiring – Here's How to Prepare (O EC2-Classic Networking será descontinuado. Veja como se preparar).

Categoria	ID do evento do RDS	Message	Observações
alteração de configuração	RDS-EVENT-0038	Alteração aplicada ao grupo de segurança.	
falha	RDS-EVENT-0039	Revogando a autorização como *`usuário`*.	O grupo de segurança de propriedade de *`usuário`* não existe. A autorização para o grupo de segurança foi revogada porque é inválida.

Eventos de snapshot de banco de dados

A tabela a seguir mostra a categoria de evento e uma lista de eventos quando um snapshot de banco de dados é o tipo de origem.

Categoria	ID do evento do RDS	Message	Observações
criação	RDS-EVENT-0040	Crie um snapshot manual.	

Categoria	ID do evento do RDS	Message	Observações
criação	RDS-EVENT-0042	Snapshot manual criado.	
criação	RDS-EVENT-0090	Criando snapshot automático.	
criação	RDS-EVENT-0091	Snapshot automático criado.	
exclusão	RDS-EVENT-0041	Snapshot de usuário excluído.	
notificação	RDS-EVENT-0059	Cópia iniciada do snapshot *nome* da região *nome*.	Essa é uma cópia de snapshot entre regiões
notificação	RDS-EVENT-0060	Cópia finalizada do snapshot *nome* da região *nome* em *minutos*.	Essa é uma cópia de snapshot entre regiões
notificação	RDS-EVENT-0061	Solicitação cancelada da cópia de snapshot de *nome* da região *nome*.	Essa é uma cópia de snapshot entre regiões
notificação	RDS-EVENT-0159	Falha na tarefa de exportação do snapshot.	
notificação	RDS-EVENT-0160	A tarefa de exportação do snapshot foi cancelada.	
notificação	RDS-EVENT-0161	A tarefa de exportação do snapshot foi concluída.	
notificação	RDS-EVENT-0196	Cópia iniciada do snapshot *nome* na região *nome*.	Essa é uma cópia local do snapshot.

Categoria	ID do evento do RDS	Message	Observações
notificação	RDS-EVENT-0197	Cópia do snapshot concluída *nome* na região *nome*.	Essa é uma cópia local do snapshot.
notificação	RDS-EVENT-0190	Solicitação cancelada da cópia de snapshot de *nome* na região *nome*.	Essa é uma cópia local do snapshot.
restauração	RDS-EVENT-0043	Restauração com base no snapshot *nome*.	Uma instância de banco de dados está sendo restaurada a partir de um snapshot de banco de dados.

Eventos de snapshot de cluster de banco de dados

A tabela a seguir mostra a categoria de evento e uma lista de eventos quando um snapshot de cluster de banco de dados é o tipo de origem.

Categoria	ID do evento do RDS	Message	Observações
backup	RDS-EVENT-0074	Criação de um snapshot manual do cluster.	
backup	RDS-EVENT-0075	Snapshot de cluster manual criado.	
backup	RDS-EVENT-0168	Criando snapshot de cluster automatizado.	
backup	RDS-EVENT-0169	Snapshot de cluster automatizado criado.	

Eventos do RDS Proxy

As tabelas a seguir mostram a categoria de evento e uma lista de eventos quando um proxy do RDS é o tipo de fonte.

Categoria	ID do evento do RDS	Message	Observações
alteração de configuração	RDS-EVENT-0204	Proxy de banco de dados *nome* modificado pelo RDS.	
alteração de configuração	RDS-EVENT-0207	O RDS modificou o endpoint do proxy de banco de dados *nome*.	
alteração de configuração	RDS-EVENT-0213	O RDS detectou a adição da instância de banco de dados e a adicionou automaticamente ao grupo de destino do *nome* do proxy de banco de dados.	
alteração de configuração	RDS-EVENT-0213	O RDS detectou a criação da instância de banco de dados *nome* e a adicionou automaticamente no grupo de destino *nome* do proxy de banco de dados *nome*.	
alteração de configuração	RDS-EVENT-0214	O RDS detectou a exclusão da instância de banco de dados *nome* e a removeu automaticamente do grupo de destino *nome* do proxy de banco de dados *nome*.	

Categoria	ID do evento do RDS	Message	Observações
alteração de configuração	RDS-EVENT-0215	O RDS detectou a exclusão do cluster de banco de dados *nome* e o removeu automaticamente do grupo de destino *nome* do proxy de banco de dados *nome*.	
criação	RDS-EVENT-0203	O RDS criou o proxy de banco de dados *nome*.	
criação	RDS-EVENT-0206	O RDS criou o endpoint para *nome* para o proxy do banco de dados *nome*.	
exclusão	RDS-EVENT-0205	O RDS excluiu o proxy do banco de dados *nome*.	
exclusão	RDS-EVENT-0208	O RDS excluiu o endpoint *nome* para o proxy do banco de dados *nome*.	
falha	RDS-EVENT-0243	O RDS não conseguiu provisionar capacidad e para o *nome* do proxy porque não há endereços IP suficientes disponíveis em suas sub-redes: *nome*. Para resolver o problema, as sub-redes devem ter o número mínimo de endereços IP não usados, conforme recomendado na documentação do proxy do RDS.	Para determinar o número recomendado para sua classe de instância , consulte Planejar a capacidade de endereços IP.

Categoria	ID do evento do RDS	Message	Observações
falha	RDS-EVENT-0275	O RDS limitou algumas conexões com o proxy de banco de dados *nome*. O número de solicitações de conexão simultâneas do cliente para o proxy excedeu o limite.	

Eventos de implantação azul/verde

A tabela a seguir mostra a categoria de evento e uma lista de eventos quando uma implantação azul/verde é o tipo de origem.

Para ter mais informações sobre implantações azul/verde, consulte Usar implantações azul/verde do Amazon RDS para atualizações de banco de dados.

Categoria	ID do evento do Amazon RDS	Message	Observações
criação	RDS-EVENT-0244	Tarefas de implantação azul/verde concluídas. Você pode fazer outras modificações nos bancos de dados no ambiente verde ou fazer a transição da implantação.	
falha	RDS-EVENT-0245	A criação da implantação azul/verde falhou porque (a instância/o cluster) de banco de dados de (origem/destino) não foi encontrado.	
exclusão	RDS-EVENT-0246	Implantação azul/verde excluída.	

Categoria	ID do evento do Amazon RDS	Message	Observações
notificação	RDS-EVENT-0247	A transição do *azul* para o *verde* começou.	
notificação	RDS-EVENT-0248	Transição concluída na implantação azul/verde.	
falha	RDS-EVENT-0249	Transição cancelada na implantação azul/verde.	
notificação	RDS-EVENT -0143RDS-EVENT-0143	*A transição do de dados de réplica primária/de leitura azul para verde foi iniciada.*	
notificação	RDS-EVENT -0143RDS-EVENT-0143	*A transição do de dados de réplica primária/de leitura azul para verde foi concluída. Renomeado de azul para azul-antigo e verde para azul.*	
falha	RDS-EVENT -0143RDS-EVENT-0143	*A transição do de dados de réplica primária/de leitura azul para verde foi cancelada por um motivo.*	

Categoria	ID do evento do Amazon RDS	Message	Observações
notificação	RDS-EVENT-0143RDS-EVENT-0143	A sincronização de sequência para alternância do *azul* para *verde foi iniciada*. A alternância ao usar sequências pode levar a um tempo de inatividade prolongado.	
notificação	RDS-EVENT-0143RDS-EVENT-0143	A sincronização de sequência para alternância do *azul* para *verde foi concluída* .	
falha	RDS-EVENT-0143RDS-EVENT-0143	A sincronização de sequência para alternância do *azul* para *verde* foi cancelada porque as sequências falharam na sincronização.	

Eventos de versão de mecanismos personalizados

A tabela a seguir mostra a categoria de evento e uma lista de eventos quando a versão de um mecanismo personalizado é o tipo de fonte.

Categoria	ID do evento do Amazon RDS	Message	Observações
criação	RDS-EVENT-0237	Preparando-se para criar um *nome* de versão do mecanismo personalizado. Todo o processo de criação	

Categoria	ID do evento do Amazon RDS	Message	Observações
		pode levar até quatro horas para ser concluído.	
criação	RDS-EVENT-0237	Criação de *nome* de versão de mecanismo personalizado.	
criação	RDS-EVENT-0237	Validando o *nome* da versão do mecanismo personalizado.	
criação	RDS-EVENT-0237	O *nome da versão do* mecanismo personalizado foi criado com sucesso.	
criação	RDS-EVENT-0237	O RDS não pode criar um *nome* de versão de mecanismo personalizado devido a um problema interno. Estamos resolvendo o problema e entraremos em contato com você, se necessário. Para obter mais assistência, entre em contato com o Suporte AWS Premium/.	
falha	RDS-EVENT-0198	Falha ao criar a versão do mecanismo personalizado *nome*. *mensagem*	A *mensagem* inclui detalhes sobre a falha, como arquivos ausentes.
falha	RDS-EVENT-0277	Falha durante a exclusão da versão do mecanismo personalizado *nome*. *mensagem*	A *mensagem* inclui detalhes sobre a falha.

Categoria	ID do evento do Amazon RDS	Message	Observações
restaurar	RDS-EVENT-0352	A contagem máxima de banco de dados compatível para restauração para um ponto no tempo foi alterada.	A *mensagem* inclui detalhes sobre o evento.

Monitorar arquivos de log do Amazon RDS

Cada mecanismo de banco de dados do RDS gera logs que você pode acessar para auditoria e solução de problemas. O tipo dos logs depende do mecanismo do banco de dados.

É possível acessar os logs de banco de dados para instâncias de banco de dados usando o AWS Management Console, a AWS Command Line Interface (AWS CLI) ou a API do Amazon RDS. Você não pode visualizar, nem monitorar, nem baixar logs de transações.

Tópicos

- Como visualizar e listar arquivos de log do banco de dados
- Como baixar um arquivo de log de banco de dados
- Como observar um arquivo de log de banco de dados
- Publicação de logs de banco de dados no Amazon CloudWatch Logs
- Leitura do conteúdo de arquivos de log usando REST
- Arquivos de log do banco de dados do Amazon RDS para Db2
- Arquivos de log do banco de dados MariaDB
- Arquivos de log do banco de dados do Amazon RDS para Microsoft SQL Server
- Arquivos de log do banco de dados MySQL
- Arquivos de log do banco de dados do Amazon RDS para Oracle
- Arquivos de log do banco de dados do RDS para PostgreSQL

Como visualizar e listar arquivos de log do banco de dados

É possível visualizar arquivos de log de banco de dados do mecanismo de banco de dados do Amazon RDS usando o AWS Management Console. Você pode listar quais arquivos de log estão disponíveis para download ou monitoramento usando a AWS CLI ou a API do Amazon RDS.

> **Note**
>
> Se você não encontrar a lista de arquivos de log para uma instância de banco de dados existente do RDS for Oracle, reinicie a instância para visualizar a lista.

Console

Para visualizar um arquivo de log de banco de dados

1. Abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Databases (Bancos de dados).
3. Escolha o nome da instância de banco de dados que contém o arquivo de log que você deseja visualizar.
4. Escolha a guia Logs & events (Logs e eventos).
5. Role para baixo até a seção Logs.
6. (Opcional) Insira um termo de pesquisa para filtrar seus resultados.
7. Escolha o log que você deseja visualizar e, depois, View (Visualizar).

AWS CLI

Para listar os arquivos de log do banco de dados disponíveis para uma instância de banco de dados, use o comando AWS CLI da `describe-db-log-files`.

O exemplo a seguir retorna uma lista de arquivos de log para uma instância de banco de dados chamada `my-db-instance`.

Example

```
aws rds describe-db-log-files --db-instance-identifier my-db-instance
```

API do RDS

Para listar os arquivos de log disponíveis do banco de dados para uma instância de banco de dados, use a ação `DescribeDBLogFiles` da API do Amazon RDS.

Como baixar um arquivo de log de banco de dados

É possível usar o AWS Management Console, a AWS CLI ou a API para baixar um arquivo de log de banco de dados.

Console

Para baixar um arquivo de log de banco de dados

1. Abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Databases (Bancos de dados).
3. Escolha o nome da instância de banco de dados que contém o arquivo de log que você deseja visualizar.
4. Escolha a guia Logs & events (Logs e eventos).
5. Role para baixo até a seção Logs.
6. Na seção Logs, escolha o botão próximo ao log do qual você deseja baixar e escolha Download.
7. Abra o menu de contexto (clique com o botão direito do mouse) para o link fornecido e escolha Save Link As (Salvar link como). Informe o local onde você deseja salvar o arquivo de log e escolha Save (Salvar).



AWS CLI

Para baixar um arquivo de log de banco de dados, use o comando AWS CLI da `download-db-log-file-portion`. Por padrão, esse comando baixa apenas da parte mais recente de um arquivo de log. No entanto, baixe um arquivo inteiro especificando o parâmetro `--starting-token 0`.

O exemplo a seguir mostra como baixar o conteúdo inteiro de um arquivo de log denominado log/ERROR.4 e armazená-lo em um arquivo local denominado errorlog.txt.

Example

Para Linux, macOS ou Unix:

```
aws rds download-db-log-file-portion \
    --db-instance-identifier myexampledb \
    --starting-token 0 --output text \
    --log-file-name log/ERROR.4 > errorlog.txt
```

Para Windows:

```
aws rds download-db-log-file-portion ^
    --db-instance-identifier myexampledb ^
    --starting-token 0 --output text ^
    --log-file-name log/ERROR.4 > errorlog.txt
```

API do RDS

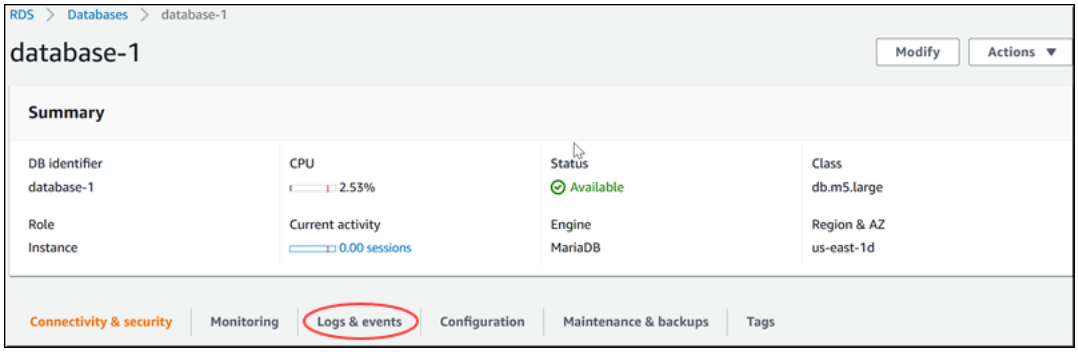
Para baixar um arquivo de log de banco de dados, use a ação `DownloadDBLogFilePortion` da API do Amazon RDS.

Como observar um arquivo de log de banco de dados

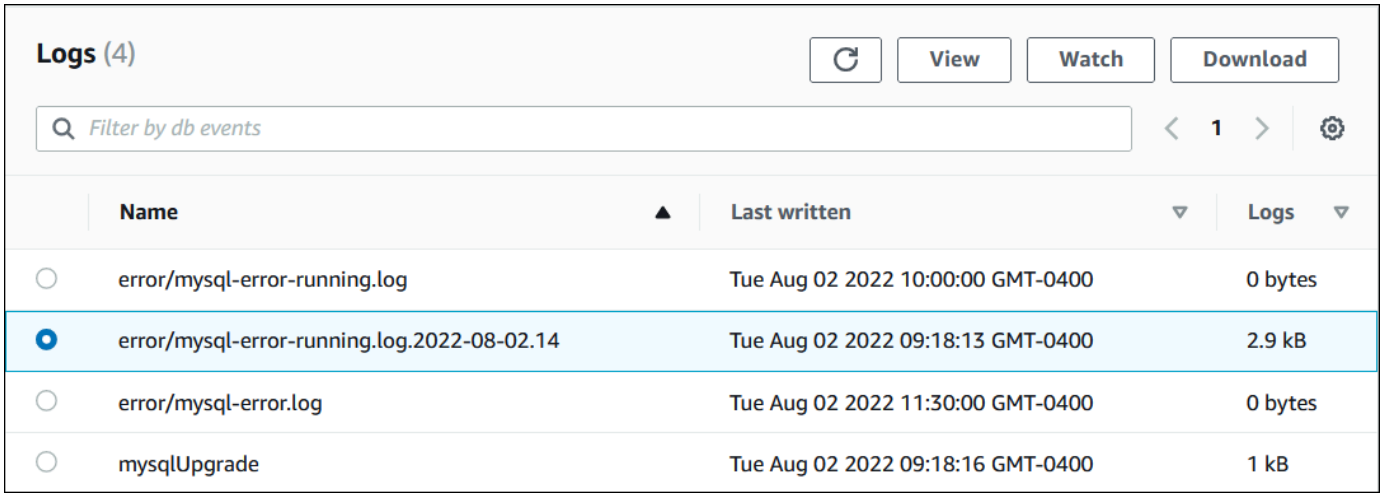
Observar um arquivo de log do banco de dados é equivalente a seguir o arquivo em um sistema UNIX ou Linux. É possível monitorar um arquivo de log usando o AWS Management Console. O RDS atualiza o final do log a cada 5 segundos.

Para observar um arquivo de log de banco de dados

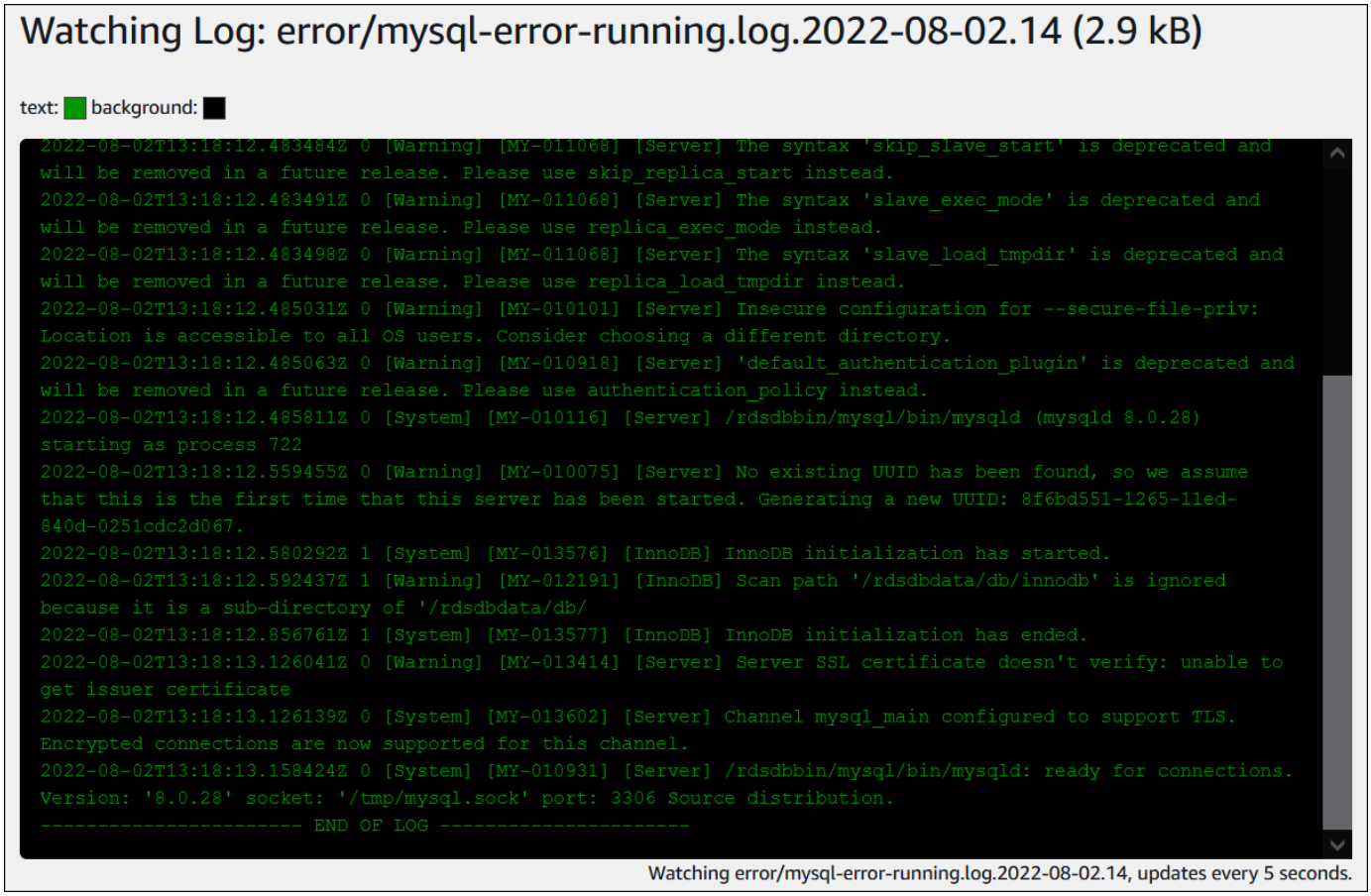
1. Abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Databases (Bancos de dados).
3. Escolha o nome da instância de banco de dados que contém o arquivo de log que você deseja visualizar.
4. Escolha a guia Logs & events (Logs e eventos).



5. Na seção Logs, escolha um arquivo de log e Watch (Observar).



O RDS mostra o final do log, como no exemplo do MySQL a seguir.



Publicação de logs de banco de dados no Amazon CloudWatch Logs

Em um banco de dados local, os registros do banco de dados residem no sistema de arquivos. O Amazon RDS não fornece acesso ao host para os logs de banco de dados no sistema de arquivos de sua instância de banco de dados. Por esse motivo, o Amazon RDS permite exportar logs de banco de dados para o Amazon CloudWatch Logs. Com o CloudWatch Logs, você pode realizar análise em tempo real de dados de log. Você também pode armazenar os dados em um armazenamento resiliente e gerenciar os dados com o agente do CloudWatch Logs.

Tópicos

- Visão geral da integração do RDS com o CloudWatch Logs
- Decidir quais logs publicar no CloudWatch Logs
- Especificar logs para publicar no CloudWatch Logs
- Pesquisar e filtrar logs no CloudWatch Logs

Visão geral da integração do RDS com o CloudWatch Logs

No CloudWatch Logs, um fluxo de logs é uma sequência de eventos de logs que compartilham a mesma origem. Cada origem separada de logs no CloudWatch Logs compõe um fluxo de logs separado. Um grupo de logs é um grupo de fluxos de log que compartilham as mesmas configurações de retenção, monitoramento e controle de acesso.

O Amazon RDS transmite continuamente os registros de logs de sua instância de banco de dados para um grupo de logs. Por exemplo, suponhamos que você tem um grupo de logs `/aws/rds/instance/`*`instance_name`*`/`*`log_type`* para cada tipo de log que publica. Esse grupo de logs está na mesma região da AWS que a instância de banco de dados que gera o log.

A AWS retém os dados de log publicados no CloudWatch Logs por um período indefinido, a menos que você especifique um período de retenção. Para obter mais informações, consulte Alterar a retenção de dados de log no CloudWatch Logs.

Decidir quais logs publicar no CloudWatch Logs

Cada mecanismo de banco de dados do RDS oferece suporte ao seu próprio conjunto de logs. Para saber mais sobre as opções do seu mecanismo de banco de dados, consulte os seguintes tópicos:

- the section called “Publicação de logs do Db2 no Amazon CloudWatch Logs”
- the section called “Publicar logs do MariaDB no Amazon CloudWatch Logs”
- the section called “Publicação de logs do MySQL no Amazon CloudWatch Logs”
- the section called “Publicação de logs da Oracle no Amazon CloudWatch Logs”
- the section called “Publicação de logs do PostgreSQL no Amazon CloudWatch Logs”
- the section called “Publicar logs do SQL Server no Amazon CloudWatch Logs”

Especificar logs para publicar no CloudWatch Logs

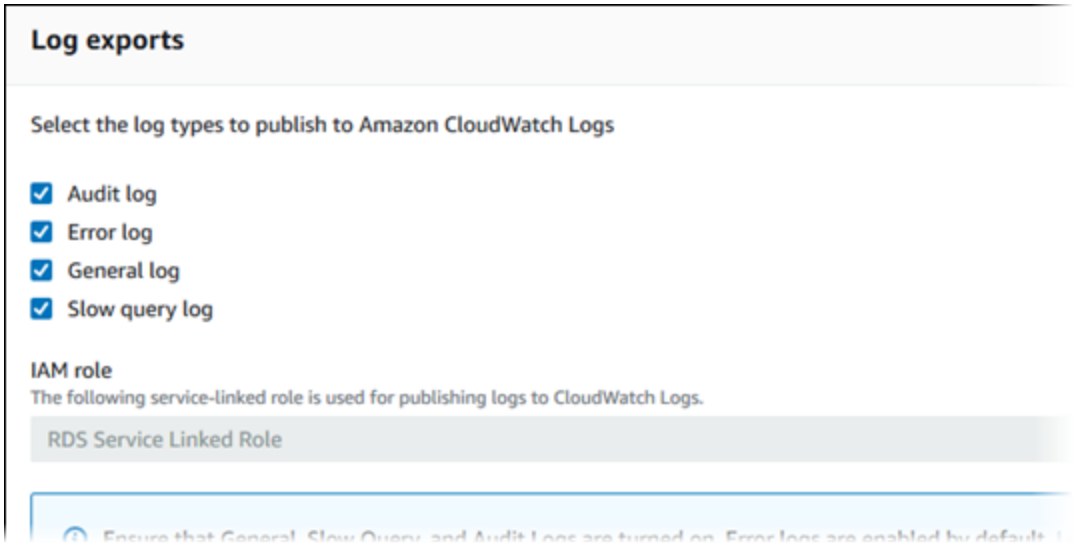
Você especifica quais logs deseja publicar no console. Verifique se você tem um perfil vinculado ao serviço no AWS Identity and Access Management (IAM). Para obter mais informações sobre funções vinculadas ao serviço, consulte Usar funções vinculadas ao serviço do Amazon RDS.

Como especificar os logs que deseja publicar

1. Abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.

2. No painel de navegação, escolha Databases (Bancos de dados).
3. Realize um dos procedimentos a seguir:
   - Selecione Criar banco de dados.
   - Escolha um banco de dados da lista e selecione Modify (Modificar).
4. Em Logs exports (Exportações de logs), escolha os logs para publicar.

   O exemplo a seguir especifica o log de auditoria, os logs de erros, o log geral e o log de consultas lentas de uma instância de banco de dados do RDS para MySQL.



Pesquisar e filtrar logs no CloudWatch Logs

Você pode procurar entradas de log que atendam a critérios especificados usando o console do CloudWatch Logs. Você pode acessar os logs por meio do console do RDS, que leva você ao console do CloudWatch Logs, ou diretamente do console do CloudWatch Logs.

Como pesquisar logs do RDS usando o console do RDS

1. Abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Databases (Bancos de dados).
3. Escolha um instância de banco de dados.
4. Escolher configuração.
5. Em Published logs (Logs publicados), escolha o log de banco de dados que deseja exibir.

Como pesquisar logs do RDS usando o console do CloudWatch Logs

1. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Grupos de logs.
3. Na caixa de filtro, insira **`/aws/rds`**.
4. Em Grupos de logs, escolha o nome do grupo de logs que contém o fluxo de log a ser pesquisado.
5. Em Fluxos de log, escolha o nome do fluxo de log para pesquisa.
6. Em Eventos de log, insira a sintaxe do filtro a ser usada.

Para obter mais informações, consulte Pesquisar e filtrar dados de logs no Guia do usuário do Amazon CloudWatch Logs. Para acessar um tutorial de blog explicando como monitorar logs do RDS, consulte Criar monitoramento de banco de dados proativo para o Amazon RDS com o Amazon CloudWatch Logs, o AWS Lambda e o Amazon SNS.

Leitura do conteúdo de arquivos de log usando REST

O Amazon RDS fornece um endpoint REST que permite acesso aos arquivos de log de instâncias de banco de dados. Isso é útil para gravar uma aplicação para transmitir o conteúdo do arquivo de log do Amazon RDS.

A sintaxe é:

```
GET /v13/downloadCompleteLogFile/DBInstanceIdentifier/LogFileName HTTP/1.1
Content-type: application/json
host: rds.region.amazonaws.com
```

Os seguintes parâmetros são obrigatórios:

- *`DBInstanceIdentifier`*: o nome da instância de banco de dados que contém o arquivo de log do qual você deseja baixar.
- *`LogFileName`*: o nome do arquivo de log que será baixado.

A resposta contém o conteúdo do arquivo de log solicitado, como um fluxo.

O exemplo a seguir baixa o arquivo de log chamado log/ERROR.6 para a instância de banco de dados chamada sample-sql na região us-west-2.

```
GET /v13/downloadCompleteLogFile/sample-sql/log/ERROR.6 HTTP/1.1
host: rds.us-west-2.amazonaws.com
X-Amz-Security-Token: AQoDYXdzEIH//////////
wEa0AIXLhngC5zp9CyB1R6abwKrXHVR5efnAVN3XvR7IwqKYalFSn6UyJuEFTft9nObglx4QJ+GXV9cpACkETq=
X-Amz-Date: 20140903T233749Z
X-Amz-Algorithm: AWS4-HMAC-SHA256
X-Amz-Credential: AKIADQKE4SARGYLE/20140903/us-west-2/rds/aws4_request
X-Amz-SignedHeaders: host
X-Amz-Content-SHA256: e3b0c44298fc1c229afbf4c8996fb92427ae41e4649b934de495991b7852b855
X-Amz-Expires: 86400
X-Amz-Signature: 353a4f14b3f250142d9afc34f9f9948154d46ce7d4ec091d0cdabbcf8b40c558
```

Se você especificar uma instância de banco de dados não existente, a resposta consistirá no erro a seguir:

- `DBInstanceNotFound: DBInstanceIdentifier` não se refere a uma instância de banco de dados existente. (Código de status HTTP: 404)

Arquivos de log do banco de dados do Amazon RDS para Db2

Você pode acessar logs de diagnóstico e logs de notificação do RDS para Db2 usando o console do Amazon RDS, a AWS CLI ou a API do RDS. Para mais informações sobre a visualização, o download e os logs de bancos de dados baseados no monitoramento de arquivos, consulte Monitorar arquivos de log do Amazon RDS.

Tópicos

- Cronograma de retenção
- Publicação de logs do Db2 no Amazon CloudWatch Logs

Cronograma de retenção

Os arquivos de log são revezados a cada dia e sempre que sua instância de banco de dados for reiniciada. A seguir você encontra o cronograma de retenção para logs do RDS para Db2 no Amazon RDS.

Tipo de log	Cronograma de retenção
Logs de diagnóstico	O Db2 exclui logs fora das configurações de retenção na configuração no nível da instância. O Amazon RDS define o parâmetro `diagsize` como 1000.
Logs de notificação	O Db2 exclui logs fora das configurações de retenção na configuração no nível da instância. O Amazon RDS define o parâmetro `diagsize` como 1000.

Publicação de logs do Db2 no Amazon CloudWatch Logs

Com o RDS para Db2, você pode publicar eventos de logs de diagnóstico e de notificação diretamente no Amazon CloudWatch Logs. Analise os dados de logs com o CloudWatch Logs e depois use o CloudWatch para criar alarmes e visualizar métricas.

Com o CloudWatch Logs, você pode fazer o seguinte:

- Armazene os logs em um espaço de armazenamento resiliente e com um período de retenção definido por você.

- Pesquisar e filtrar dados de log.
- Compartilhar dados de log entre contas.
- Exportar logs para o Amazon S3.
- Faça uma transmissão de dados para o Amazon OpenSearch Service.
- Processar dados de log em tempo real com o Amazon Kinesis Data Streams. Para obter mais informações, consulte Trabalhando com o Amazon CloudWatch Logs, no Guia do desenvolvedor do Amazon Kinesis Data Analytics para aplicativos SQL.

O Amazon RDS publica cada log de banco de dados do RDS para Db2 como um fluxo separado de banco de dados no grupo de logs. Por exemplo, se você publicar os logs de diagnóstico e os logs de notificação, os dados de diagnóstico serão armazenados em um fluxo de logs de diagnóstico no grupo de logs `/aws/rds/instance/my_instance/diagnostic`, e os dados de log de notificação serão armazenados no grupo de logs `/aws/rds/instance/my_instance/notify`.

> **Note**
>
> A publicação de logs do RDS para Db2 no CloudWatch Logs não está habilitada por padrão. A publicação de logs de estatísticas do gerenciador de memória autoajustável (STMM) e do otimizador não é permitida. A publicação de logs do RDS para Db2 no CloudWatch Logs é compatível com todas as regiões, exceto Ásia-Pacífico (Hong Kong).

Console

Para publicar logs de banco de dados do RDS para Db2 no CloudWatch Logs pelo AWS Management Console

1. Abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Databases (Bancos de dados) e a instância de banco de dados que você deseja modificar.
3. Selecione Modify.
4. Na seção Log exports (Exportações de log), escolha os logs que deseja começar a publicar no CloudWatch Logs.

   Você pode escolher diag.log, notify.log, ou ambos.

5. Escolha Continue (Continuar) e depois escolha Modify DB Instance (Modificar instância de banco de dados) na página de resumo.

AWS CLI

Para publicar os logs do RDS para Db2, você pode usar o comando `modify-db-instance` com os seguintes parâmetros:

- `--db-instance-identifier`
- `--cloudwatch-logs-export-configuration`

> **Note**
>
> Uma alteração feita na opção `--cloudwatch-logs-export-configuration` sempre é aplicada imediatamente na instância de banco de dados. Por isso, as opções `--apply-immediately` e `--no-apply-immediately` não entram em vigor.

Você também pode publicar os logs do RDS para Db2 usando os seguintes comandos:

- `create-db-instance`
- `restore-db-instance-from-db-snapshot`
- `restore-db-instance-to-point-in-time`

Example

O exemplo a seguir cria uma instância de banco de dados do RDS para Db2 com a publicação no CloudWatch Logs habilitada. O valor `--enable-cloudwatch-logs-exports` é uma matriz JSON de strings que podem incluir `diag.log`, `notify.log` ou ambos.

Para Linux, macOS ou Unix:

```
aws rds create-db-instance \
    --db-instance-identifier mydbinstance \
    --enable-cloudwatch-logs-exports '["diag.log","notify.log"]' \
    --db-instance-class db.m4.large \
    --engine db2-se
```

Para Windows:

```
aws rds create-db-instance ^
    --db-instance-identifier mydbinstance ^
    --enable-cloudwatch-logs-exports "[\"diag.log\",\"notify.log\"]" ^
    --db-instance-class db.m4.large ^
    --engine db2-se
```

> **Note**
>
> Ao usar o prompt de comando do Windows, você deve fazer o escape das aspas duplas (") no código JSON, prefixando-as com uma barra invertida (\).

Example

O exemplo a seguir altera uma instância de banco de dados do RDS para Db2 existente para publicar os arquivos de log no CloudWatch Logs. O valor `--cloudwatch-logs-export-configuration` é um objeto JSON. A chave desse objeto é `EnableLogTypes`, e seu valor é uma matriz de strings que pode incluir `diag.log`, `notify.log` ou ambos.

Para Linux, macOS ou Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --cloudwatch-logs-export-configuration '{"EnableLogTypes":
["diag.log","notify.log"]}'
```

Para Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --cloudwatch-logs-export-configuration "{\"EnableLogTypes\":[\"diag.log\",
\"notify.log\"]}"
```

> **Note**
>
> Ao usar o prompt de comando do Windows, você deve fazer o escape das aspas duplas (") no código JSON, prefixando-as com uma barra invertida (\).

Example

O exemplo a seguir altera uma instância de banco de dados do RDS para Db2 existente para desabilitar a publicação de arquivos de log de diagnóstico no CloudWatch Logs. O valor `--cloudwatch-logs-export-configuration` é um objeto JSON. A chave desse objeto é `DisableLogTypes`, e seu valor é uma matriz de strings que pode incluir `diag.log`, `notify.log` ou ambos.

Para Linux, macOS ou Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --cloudwatch-logs-export-configuration '{"DisableLogTypes":["diag.log"]}'
```

Para Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --cloudwatch-logs-export-configuration "{\"DisableLogTypes\":[\"diag.log\"]}"
```

> **Note**
>
> Ao usar o prompt de comando do Windows, você deve fazer o escape das aspas duplas (") no código JSON, prefixando-as com uma barra invertida (\).

Arquivos de log do banco de dados MariaDB

É possível monitorar o log de erros do MariaDB, o log de erros de consultas lentas, o log de erros de autenticação de banco de dados do IAM e o log geral. O log de erros do MariaDB é gerado por padrão. Você pode gerar a consulta lenta e os logs gerais definindo parâmetros em seu grupo de parâmetros do banco de dados. O Amazon RDS alterna todos os arquivos de log do MariaDB; os intervalos para cada tipo são fornecidos a seguir.

Você pode monitorar os logs do MariaDB diretamente por meio do console do Amazon RDS, da API do Amazon RDS, da CLI do Amazon RDS ou os SDKs da AWS. Você também pode acessar os logs do MariaDB direcionando os logs para uma tabela de banco de dados no banco de dados primário e consultando essa tabela. Você pode usar o utilitário mysqlbinlog para baixar um log de binários.

Para mais informações sobre a visualização, o download e os logs de bancos de dados baseados no monitoramento de arquivos, consulte Monitorar arquivos de log do Amazon RDS.

Tópicos

- Acessar logs de erros do MariaDB
- Acessar os logs gerais e de consultas lentas do MariaDB
- Publicar logs do MariaDB no Amazon CloudWatch Logs
- Alternância e retenção de logs para o MariaDB
- Gerenciar logs do MariaDB com base em tabelas
- Configurar o registro em log binário do MariaDB
- Acessar logs binários do MariaDB
- Habilitar a anotação de log binário do MariaDB

Acessar logs de erros do MariaDB

O log de erros do MariaDB é gravado no arquivo `<host-name>.err`. É possível visualizar esse arquivo usando o console do Amazon RDS. Você também pode recuperar o log usando a API do Amazon RDS, a CLI do Amazon RDS ou AWS SDKs. O arquivo `<host-name>.err` é descarregado a cada 5 minutos, e seu conteúdo é anexado a `mysql-error-running.log`. O arquivo `mysql-error-running.log` é então rotacionado a cada hora, e os arquivos horários gerados durante as últimas 24 horas são mantidos. Cada arquivo de log tem a hora em que foi gerado (em UTC) anexada ao seu nome. Os arquivos de log também possuem um carimbo de data/hora que ajuda você a determinar quando as entradas de log foram gravadas.

O MariaDB grava no log de erros apenas na inicialização, no desligamento e quando encontra erros. Uma instância de banco de dados pode passar horas ou dias sem novas entradas gravadas no log de erros. Se você não vir nenhuma entrada recente, é porque o servidor não encontrou um erro que resultou em uma entrada de log.

Acessar os logs gerais e de consultas lentas do MariaDB

É possível gravar o log de consultas lentas do MariaDB e o log geral em um arquivo ou uma tabela de banco de dados definindo parâmetros em seu grupo de parâmetros de banco de dados. Para obter informações sobre como criar e modificar um grupo de parâmetros de banco de dados, consulte Grupos de parâmetros para Amazon RDS. Você deve definir esses parâmetros antes de visualizar o log de consultas lentas ou o log geral no console do Amazon RDS ou usando a API do Amazon RDS, a AWS CLI ou os SDKs da AWS.

Você pode controlar o registro em log do MariaDB usando os parâmetros nessa lista:

- `slow_query_log` ou `log_slow_query`: para criar o log de consultas lentas, defina como 1. O padrão é 0.
- `general_log`: para criar o log geral, defina como 1. O padrão é 0.
- `long_query_time` ou `log_slow_query_time`: para evitar que as consultas de execução rápida sejam registradas no log de consultas lentas, especifique um valor para o tempo de execução de consultas mais curto a ser registrado, em segundos. O padrão é 10 segundos; o mínimo é 0. Se log_output = FILE, você poderá especificar um valor de ponto flutuante com resolução por microssegundo. Se log_output = TABLE, você deverá especificar um valor inteiro com a segunda resolução. Apenas as consultas cujo tempo de execução excede o valor `long_query_time` ou `log_slow_query_time` são registradas em log. Por exemplo, definir `long_query_time` ou `log_slow_query_time` como 0,1 impede que qualquer consulta que seja executada por menos de 100 milissegundos seja registrada.
- `log_queries_not_using_indexes`: para registrar todas as consultas que não usam um índice no log de consultas lentas, defina esse parâmetro como 1. O padrão é 0. As consultas que não usam um índice são registradas em log, mesmo que seu tempo de execução seja inferior ao valor do parâmetro `long_query_time`.
- `log_output` *`option`*: você pode especificar uma das seguintes opções para o parâmetro `log_output`:
  - TABLE (padrão): grava consultas gerais na tabela `mysql.general_log` e consultas lentas na tabela `mysql.slow_log`.

- FILE: grave logs de consultas gerais e lentas no sistema de arquivos. Arquivos de log são rotacionados de hora em hora.
- NONE: desabilite o registro em log.

Quando o registro em log está habilitado, o Amazon RDS faz o rodízio de logs de tabelas ou exclui arquivos de log em intervalos regulares. Essa medida é uma precaução para reduzir a possibilidade de um arquivo de log grande bloquear o uso do banco de dados ou afetar a performance. O registro em log `FILE` e `TABLE` abordam o rodízio e a exclusão da seguinte maneira:

- Quando o registro em log `FILE` está habilitado, os arquivos de log são examinados a cada hora, e os arquivos de log com mais de 24 horas são excluídos. Em alguns casos, o tamanho do arquivo de log combinado restante após a exclusão pode exceder o limite de 2% do espaço alocado da instância de um banco de dados. Nesses casos, os arquivos de log maiores são excluídos até que o tamanho de arquivo de log não exceda o limite.
- Quando o registro de `TABLE` estiver ativado, em alguns casos as tabelas de log serão rotacionadas a cada 24 horas. Essa alternância ocorrerá se o espaço utilizado pelos logs de tabelas for superior a 20% do espaço de armazenamento alocado. Isso também ocorrerá se o tamanho de todos os logs combinados for superior a 10 GB. Se a quantidade de espaço usada por uma instância de banco de dados for maior que 90% do espaço de armazenamento alocado da instância de banco de dados, os limites para a alternância do log serão reduzidos. As tabelas de logs serão, então, alternadas se o espaço utilizado pelos logs de tabelas for superior a 10% do espaço de armazenamento alocado. Eles também serão alternados se o tamanho de todos os logs combinados for superior a 5 GB.

  Quando as tabelas de log são revezadas, a tabela de log atual é copiada para uma tabela de log de backup e as entradas na tabela de log atual são removidas. Se a tabela de log de backup já existir, então ela será excluída antes que a tabela de log atual seja copiada ao backup. Você pode consultar a tabela de log de backup, se necessário. A tabela de log de backup para a tabela `mysql.general_log` é denominada `mysql.general_log_backup`. A tabela de log de backup para a tabela `mysql.slow_log` é denominada `mysql.slow_log_backup`.

  Você pode rotacionar a tabela `mysql.general_log` chamando o procedimento `mysql.rds_rotate_general_log`. Você pode rotacionar a tabela `mysql.slow_log` chamando o procedimento `mysql.rds_rotate_slow_log`.

  Os logs de tabelas são rotacionados durante um upgrade de versão do banco de dados.

O Amazon RDS registra o rodízio dos logs `TABLE` e `FILE` em um evento do Amazon RDS e envia uma notificação para você.

Para trabalhar com os logs no console do Amazon RDS, na API do Amazon RDS, na CLI do Amazon RDS ou nos SDKs da AWS, defina o parâmetro `log_output` como FILE. Como o log de erros do MariaDB, esses arquivos de log são rotacionados por hora. Os arquivos de log que foram gerados durante as 24 horas anteriores são retidos.

Para obter mais informações sobre os log de consultas gerais e de consultas lentas, acesse os seguintes tópicos na documentação do MariaDB:

- Log de consultas lentas
- Log de consultas gerais

Publicar logs do MariaDB no Amazon CloudWatch Logs

Você pode configurar sua instância de banco de dados MariaDB para publicar dados de log em um grupo de log no Amazon CloudWatch Logs. Com o CloudWatch Logs, você pode executar análise em tempo real de dados de log e usar o CloudWatch para criar alarmes e visualizar métricas. Você pode usar o CloudWatch Logs para armazenar seus registros de log em armazenamento resiliente.

O Amazon RDS publica cada log de banco de dados MariaDB como um fluxo separado de banco de dados no grupo de logs. Por exemplo, suponha que você configure a função de exportação para incluir o log de consultas lentas. Depois, os dados de consultas lentas são armazenados em uma fluxo de log de consultas lentas no grupo de logs `/aws/rds/instance/`*`my_instance`*`/slowquery`.

O log de erros está habilitado por padrão. A tabela a seguir resume os requisitos de outros logs do MariaDB.

Log	Requisito
Log de auditoria	A instância de banco de dados deve usar um grupo de opções personalizado com a opção `MARIADB_AUDIT_PLUGIN` .
Log geral	A instância de banco de dados deve usar um grupo de parâmetros personalizado com a

Log	Requisito
	configuração de parâmetro `general_log = 1` para habilitar o log geral.
Log de consultas lentas	A instância de banco de dados deve usar um grupo de parâmetros personalizado com a configuração de parâmetro `slow_query_log = 1` ou `log_slow_query = 1` para habilitar o log de consultas lentas.
Log de erros de autenticação de banco de dados do IAM	Você deve habilitar o tipo de log `iam-db-auth-error` para uma instância de banco de dados criando ou modificando uma instância de banco de dados.
Resultado de saída do log	A instância de banco de dados deve usar um grupo de parâmetros personalizado com a configuração de parâmetro `log_output = FILE` para gravar logs no sistema de arquivos e publicá-los no CloudWatch Logs.

Console

Para publicar logs do MariaDB no CloudWatch Logs a partir do console

1. Abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Databases (Bancos de dados) e a instância de banco de dados que você deseja modificar.
3. Selecione Modify.
4. Na seção Log exports (Exportações de log), escolha os logs que deseja começar a publicar no CloudWatch Logs.
5. Escolha Continue (Continuar) e depois escolha Modify DB Instance (Modificar instância de banco de dados) na página de resumo.

AWS CLI

Você pode publicar logs do MariaDB com o AWS CLI. Você pode chamar o comando `modify-db-instance` com os seguintes parâmetros:

- `--db-instance-identifier`
- `--cloudwatch-logs-export-configuration`

> **Note**
>
> Uma alteração feita na opção `--cloudwatch-logs-export-configuration` sempre é aplicada imediatamente na instância de banco de dados. Por isso, as opções `--apply-immediately` e `--no-apply-immediately` não entram em vigor.

Também é possível publicar logs do MariaDB seguindo os seguintes comandos AWS CLI:

- `create-db-instance`
- `restore-db-instance-from-db-snapshot`
- `restore-db-instance-from-s3`
- `restore-db-instance-to-point-in-time`

Execute um destes comandos da AWS CLI com as seguintes opções:

- `--db-instance-identifier`
- `--enable-cloudwatch-logs-exports`
- `--db-instance-class`
- `--engine`

Outras opções podem ser obrigatórias, dependendo do comando da AWS CLI executado.

Example

O exemplo a seguir altera uma instância de Banco de Dados MariaDB existente para publicar os arquivos de log no CloudWatch Logs. O valor `--cloudwatch-logs-export-configuration` é

um objeto JSON. A chave para esse objeto é `EnableLogTypes`, e seu valor é uma matriz de strings com qualquer combinação de `audit`, `error`, `general` e `slowquery`.

Para Linux, macOS ou Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --cloudwatch-logs-export-configuration '{"EnableLogTypes":
["audit","error","general","slowquery"]}'
```

Para Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --cloudwatch-logs-export-configuration '{"EnableLogTypes":
["audit","error","general","slowquery"]}'
```

Example

O comando a seguir cria uma instância de banco de dados do MariaDB e publica arquivos de log no CloudWatch Logs. O valor `--enable-cloudwatch-logs-exports` é uma matriz de strings JSON. As strings podem ser qualquer combinação de `audit`, `error`, `general` e `slowquery`.

Para Linux, macOS ou Unix:

```
aws rds create-db-instance \
    --db-instance-identifier mydbinstance \
    --enable-cloudwatch-logs-exports '["audit","error","general","slowquery"]' \
    --db-instance-class db.m4.large \
    --engine mariadb
```

Para Windows:

```
aws rds create-db-instance ^
    --db-instance-identifier mydbinstance ^
    --enable-cloudwatch-logs-exports '["audit","error","general","slowquery"]' ^
    --db-instance-class db.m4.large ^
    --engine mariadb
```

API do RDS

Você pode publicar logs MariaDB com a API da RDS. É possível chamar a operação `ModifyDBInstance` com os seguintes parâmetros:

- `DBInstanceIdentifier`
- `CloudwatchLogsExportConfiguration`

> **Note**
>
> Uma alteração feita no parâmetro `CloudwatchLogsExportConfiguration` sempre é aplicada imediatamente na instância de banco de dados. Por isso, o parâmetro `ApplyImmediately` não entra em vigor.

Também é possível publicar logs do MariaDB chamando as seguintes operações da API do RDS:

- `CreateDBInstance`
- `RestoreDBInstanceFromDBSnapshot`
- `RestoreDBInstanceFromS3`
- `RestoreDBInstanceToPointInTime`

Execute uma destas operações da API do RDS com os seguintes parâmetros:

- `DBInstanceIdentifier`
- `EnableCloudwatchLogsExports`
- `Engine`
- `DBInstanceClass`

Outros parâmetros podem ser exigidos dependendo do comando AWS CLI que você executa.

Alternância e retenção de logs para o MariaDB

Quando o registro em log está habilitado, o Amazon RDS faz o rodízio de logs de tabelas ou exclui arquivos de log em intervalos regulares. Essa medida é uma precaução para reduzir a possibilidade de um arquivo de log grande bloquear o uso do banco de dados ou afetar o desempenho.

Os tamanhos de arquivo do log de consultas lentas, do log de erros e do log geral do MariaDB são restritos a não mais de 2% do espaço de armazenamento alocado para uma instância de banco de dados. Para manter esse limite, os logs são rotacionados automaticamente a cada hora, e os arquivos de log com mais de 24 horas são removidos. Se o tamanho do arquivo de log combinado exceder o limite após a remoção dos arquivos de log antigos, os arquivos de log maiores serão excluídos até o tamanho do arquivo de log deixar de exceder esse limite.

O Amazon RDS alterna os arquivos de log de erros de autenticação de banco de dados do IAM maiores que 10 MB. O Amazon RDS remove os arquivos de log de erros de autenticação de banco de dados do IAM com mais de cinco dias ou maiores que 100 MB.

Gerenciar logs do MariaDB com base em tabelas

Você pode direcionar os logs de consultas gerais e lentas para tabelas na instância de banco de dados. Para fazer isso, crie um grupo de parâmetros do banco de dados e defina o parâmetro `log_output` do servidor como `TABLE`. As consultas gerais são registradas na tabela `mysql.general_log` e as consultas lentas são registradas na tabela `mysql.slow_log`. Você pode consultar as tabelas para acessar as informações do log. Habilitar esse registro aumenta a quantidade de dados gravados no banco de dados, o que pode degradar a performance.

O log geral e os logs de consultas lentas estão desabilitados por padrão. Para habilitar o registro em log de tabelas, também é necessário definir os seguintes parâmetros de servidor como `1`:

- `general_log`
- `slow_query_log` ou `log_slow_query`

As tabelas de log continuarão crescendo até que as respectivas atividades de log sejam desativadas com a redefinição do parâmetro apropriado como `0`. Uma grande quantidade de dados geralmente se acumula ao longo do tempo, o que pode consumir uma porcentagem considerável do espaço de armazenamento alocado. O Amazon RDS não permite truncar tabelas de log, mas é possível mover o respectivo conteúdo. Rotacionar uma tabela salva seu conteúdo em uma tabela de backup e, em seguida, cria uma nova tabela de log vazia. Você pode rotacionar manualmente as tabelas de log

com os seguintes procedimentos de linha de comando, em que o prompt de comando é indicado por PROMPT>:

```
PROMPT> CALL mysql.rds_rotate_slow_log;
PROMPT> CALL mysql.rds_rotate_general_log;
```

Para remover completamente os dados antigos e recuperar o espaço em disco, chame o procedimento apropriado duas vezes sucessivamente.

Configurar o registro em log binário do MariaDB

O log binário é um conjunto de arquivos de log que contêm informações sobre modificações de dados feitas em uma instância do servidor MariaDB. O log binário contém informações como as seguintes:

- Eventos que descrevem alterações no banco de dados, como criação de tabela ou modificações de linha
- Informações sobre a duração de cada instrução que atualizou dados
- Eventos para declarações que poderiam ter dados atualizados, mas não foram

O log binário registra instruções que são enviadas durante a replicação. Também é necessário para algumas operações de recuperação. Para obter mais informações, consulte Log binário na documentação do MariaDB.

O recurso de backups automatizados determina se o registro em log binário está ativado ou desativado para o MariaDB. Você tem as seguintes opções:

Ativar o registro em log binário

Defina o período de retenção de backup para um valor positivo diferente de zero.

Desativar o registro em log binário

Defina o período de retenção de backup para 0.

Para ter mais informações, consulte Ativar backups automáticos.

O MariaDB no Amazon RDS oferece suporte a formatos do registro de logs binários baseados em linha, baseados em instrução e mistos. O formato de registro em log binário padrão é combinado.

Para obter detalhes sobre os diferentes formatos de logs binários do MariaDB, consulte o tópico sobre Formatos de logs binários na documentação do MariaDB.

Se você planeja usar a replicação, o formato de registro em log binário é importante. O motivo disso é porque ele determina o registro de alterações de dados que é registrado na origem e enviado aos destinos de replicação. Para obter informações sobre as vantagens e as desvantagens de formatos de registro em logs binários para replicação, consulte Vantagens e desvantagens da replicação baseada em instrução e baseada em linha na documentação do MySQL.

> **Important**
>
> Definir o formato de registro em log de binários como baseado em linha pode resultar em arquivos de log de binários muito grandes. Arquivos de log binários grandes reduzem a quantidade de armazenamento disponível para uma instância de banco de dados. Eles também podem aumentar o tempo necessário para realizar uma operação de restauração de uma instância de banco de dados.
>
> A replicação baseada em instrução pode causar inconsistências entre a instância de banco de dados de origem e uma réplica de leitura. Para obter mais informações, consulte Instruções não seguras para replicação baseada na instrução na documentação do MariaDB.
>
> Habilitar o registro em log binário aumenta o número de operações de E/S do disco de gravação na instância de banco de dados. Você pode monitorar o uso de IOPS com a métrica `WriteIOPS` do CloudWatch.

Para definir o formato de registro em log binário do MariaDB

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, selecione Parameter groups.
3. Escolha o grupo de parâmetros usado pela instância de banco de dados que você deseja modificar.

   Não é possível modificar um grupo de parâmetros padrão. Se a instância de banco de dados estiver usando um grupo de parâmetros padrão, crie um novo grupo de parâmetros e o associe à instância de banco de dados.

   Para obter mais informações sobre grupos de parâmetros de banco de dados, consulte Grupos de parâmetros para Amazon RDS.

4. Em Parameter group actions (Ações do grupo de parâmetros), escolha Edit (Editar).
5. Defina o parâmetro `binlog_format` para o formato de registro em log binário escolhido (ROW, STATEMENT ou MIXED).

   Você pode desativar o registro em log binário definindo o período de retenção de backup de uma instância de banco de dados como zero, mas isso desativa os backups automatizados diários. Desabilitar os backups automatizados desativa ou desabilita a variável de sessão `log_bin`. Isso desabilita o registro em log binário na instância de banco de dados do RDS para MariaDB que, por sua vez, redefine a variável de sessão `binlog_format` como o valor padrão de `ROW` no banco de dados. Recomendamos não desabilitar os backups. Para receber mais informações sobre a configuração Período de retenção de backup, consulte Configurações para instâncias de banco de dados.
6. Escolha Save changes (Salvar alterações) para salvar as atualizações no grupo de parâmetros de banco de dados.

Como o parâmetro `binlog_format` é dinâmico no RDS para MariaDB, você não precisa reinicializar a instância de banco de dados para que as alterações sejam aplicadas.

> **Important**
>
> Alterar um grupo de parâmetros de banco de dados afeta todas as instâncias de banco de dados que usam esse grupo de parâmetros. Se você quiser especificar diferentes formatos do registro em log binário para diferentes instâncias de banco de dados MariaDB em uma região da AWS, as instâncias de banco de dados deverão usar diferentes grupos de parâmetros de banco de dados. Esses grupos de parâmetros identificam diferentes formatos de log. Atribua o grupo de parâmetros de banco de dados apropriado a cada instância de banco de dados.

Acessar logs binários do MariaDB

Você pode usar o utilitário mysqlbinlog para baixar logs binários no formato de texto das instâncias de bancos de dados MariaDB. O log binário é baixado para o computador local. Para obter mais informações sobre como usar o utilitário mysqlbinlog, acesse o tópico sobre como usar mysqlbinlog na documentação do MariaDB.

Para executar o utilitário mysqlbinlog em uma instância do Amazon RDS, use as seguintes opções:

- Especifique a opção `--read-from-remote-server`.
- `--host`: especifique o nome DNS do endpoint da instância.
- `--port`: especifique a porta usada pela instância.
- `--user`: especifique um usuário do MariaDB ao qual foi concedida a permissão de escravo de replicação.
- `--password`: especifique a senha para o usuário ou omita um valor de senha para que o utilitário solicite uma senha.
- `--result-file`: especifique o arquivo local que recebe a saída.
- Especifique os nomes de um ou mais arquivos de log binários. Para obter uma lista dos logs disponíveis, use o comando SQL SHOW BINARY LOGS.

Para obter mais informações sobre as opções de mysqlbinlog, acesse Opções de mysqlbinlog na documentação do MariaDB.

Veja um exemplo a seguir:

Para Linux, macOS ou Unix:

```
mysqlbinlog \
    --read-from-remote-server \
    --host=mariadbinstance1.1234abcd.region.rds.amazonaws.com \
    --port=3306  \
    --user ReplUser \
    --password <password> \
    --result-file=/tmp/binlog.txt
```

Para Windows:

```
mysqlbinlog ^
    --read-from-remote-server ^
    --host=mariadbinstance1.1234abcd.region.rds.amazonaws.com ^
    --port=3306  ^
    --user ReplUser ^
    --password <password> ^
    --result-file=/tmp/binlog.txt
```

Normalmente, o Amazon RDS limpa um log binário o mais rápido possível. No entanto, o log binário ainda deve estar disponível na instância para ser acessado por mysqlbinlog. Para especificar o

número de horas durante as quais o RDS vai reter os logs binários, use o procedimento armazenado `mysql.rds_set_configuration`. Especifique um período com tempo suficiente para baixar os logs. Após configurar o período de retenção, monitore o uso de armazenamento da instância de banco de dados para garantir que os logs binários retidos não consumam muito armazenamento.

O exemplo a seguir define o período de retenção como 1 dia.

```
call mysql.rds_set_configuration('binlog retention hours', 24);
```

Para exibir a configuração atual, use o procedimento armazenado `mysql.rds_show_configuration`.

```
call mysql.rds_show_configuration;
```

Habilitar a anotação de log binário do MariaDB

Em uma instância de banco de dados MariaDB, você pode usar o evento `Annotate_rows` para anotar um evento de linha com uma cópia da consulta SQL que causou esse evento de linha. Essa abordagem fornece funcionalidade semelhante para habilitar o parâmetro `binlog_rows_query_log_events` em uma instância de banco de dados do RDS para MySQL.

Você pode habilitar anotações de logs binários globalmente criando um grupo de parâmetros personalizado e definindo o parâmetro `binlog_annotate_row_events` como **1**.
Você também pode habilitar anotações em nível de sessão, chamando `SET SESSION binlog_annotate_row_events = 1`. Use a opção `replicate_annotate_row_events` para replicar anotações de logs binários para a instância de réplica se o registro em log binário estiver habilitado nela. Não são necessários privilégios especiais para usar essas configurações.

A seguir está um exemplo de uma transação com base em linha no MariaDB. O uso do registro em log baseado em fila é acionado definindo o nível de isolamento da transação como confirmado na leitura.

```
CREATE DATABASE IF NOT EXISTS test;
USE test;
CREATE TABLE square(x INT PRIMARY KEY, y INT NOT NULL) ENGINE = InnoDB;
SET SESSION TRANSACTION ISOLATION LEVEL READ COMMITTED;
BEGIN
INSERT INTO square(x, y) VALUES(5, 5 * 5);
COMMIT;
```

Sem anotações, as entradas de logs binários para a transação são as seguintes:

```
BEGIN
/*!*/;
# at 1163
# at 1209
#150922  7:55:57 server id 1855786460  end_log_pos 1209         Table_map:
 `test`.`square` mapped to number 76
#150922  7:55:57 server id 1855786460  end_log_pos 1247         Write_rows: table id 76
 flags: STMT_END_F
### INSERT INTO `test`.`square`
### SET
###   @1=5
###   @2=25
# at 1247
#150922  7:56:01 server id 1855786460  end_log_pos 1274         Xid = 62
COMMIT/*!*/;
```

A seguinte instrução permite anotações em nível de sessão para essa mesma transação e as desabilita em seguida confirmando a transação:

```
CREATE DATABASE IF NOT EXISTS test;
USE test;
CREATE TABLE square(x INT PRIMARY KEY, y INT NOT NULL) ENGINE = InnoDB;
SET SESSION TRANSACTION ISOLATION LEVEL READ COMMITTED;
SET SESSION binlog_annotate_row_events = 1;
BEGIN;
INSERT INTO square(x, y) VALUES(5, 5 * 5);
COMMIT;
SET SESSION binlog_annotate_row_events = 0;
```

Com anotações, as entradas de logs binários para a transação são as seguintes:

```
BEGIN
/*!*/;
# at 423
# at 483
# at 529
#150922  8:04:24 server id 1855786460  end_log_pos 483  Annotate_rows:
#Q> INSERT INTO square(x, y) VALUES(5, 5 * 5)
#150922  8:04:24 server id 1855786460  end_log_pos 529  Table_map: `test`.`square`
 mapped to number 76
```

```
#150922  8:04:24 server id 1855786460  end_log_pos 567  Write_rows: table id 76 flags:
 STMT_END_F
### INSERT INTO `test`.`square`
### SET
###   @1=5
###   @2=25
# at 567
#150922  8:04:26 server id 1855786460  end_log_pos 594  Xid = 88
COMMIT/*!*/;
```

Arquivos de log do banco de dados do Amazon RDS para Microsoft SQL Server

Você pode acessar os logs de erros, logs de agentes, arquivos de rastreamento e arquivos de despejo do Microsoft SQL Server usando o console do Amazon RDS, a AWS CLI ou a API do RDS. Para mais informações sobre a visualização, o download e os logs de bancos de dados baseados no monitoramento de arquivos, consulte Monitorar arquivos de log do Amazon RDS.

Cronograma de retenção

Os arquivos de log são revezados a cada dia e sempre que sua instância de banco de dados for reiniciada. A seguir você encontra o cronograma de retenção para logs do Microsoft SQL Server no Amazon RDS.

Tipo de log	Cronograma de retenção
Logs de erro	No máximo, 30 logs de erros são retidos. O Amazon RDS pode excluir logs de erros com mais de 7 dias.
Logs de agente	No máximo, 10 logs de agente são retidos. O Amazon RDS pode excluir logs de agente com mais de 7 dias.
Arquivos de rastreamento	Os arquivos de rastreamento são mantidos de acordo com o período de retenção de arquivos de rastreamento de sua instância de banco de dados. O período de retenção padrão de arquivos de rastreamento é de 7 dias. Para modificar o período de retenção de arquivos de rastreamento de sua instância de banco de dados, consulte Como definir o período de retenção para arquivos de rastreamento e despejo.
Arquivos de despejo	Os arquivos de despejo são mantidos de acordo com o período de retenção de arquivos de despejo de sua instância de banco de dados. O período de retenção padrão de arquivos de despejo é de 7 dias. Para modificar o período de retenção de arquivos de despejo de sua instância de banco de dados, consulte Como definir o período de retenção para arquivos de rastreamento e despejo.

Visualizar o log de erros do SQL Server usando o procedimento rds_read_error_log

Você pode usar o procedimento armazenado `rds_read_error_log` do Amazon RDS para visualizar logs de erros e logs de agente. Para obter mais informações, consulte Visualizar logs do agente ou erros.

Publicar logs do SQL Server no Amazon CloudWatch Logs

Com o Amazon RDS para SQL Server, você pode publicar eventos de log de erros e de agentes diretamente no Amazon CloudWatch Logs. Analise os dados de logs com o CloudWatch Logs e depois use o CloudWatch para criar alarmes e visualizar métricas.

Com o CloudWatch Logs, você pode fazer o seguinte:

- Armazene os logs em um espaço de armazenamento resiliente e com um período de retenção definido por você.
- Pesquisar e filtrar dados de log.
- Compartilhar dados de log entre contas.
- Exportar logs para o Amazon S3.
- Faça uma transmissão de dados para o Amazon OpenSearch Service.
- Processar dados de log em tempo real com o Amazon Kinesis Data Streams. Para obter mais informações, consulte Trabalhando com o Amazon CloudWatch Logs, no Guia do desenvolvedor do Amazon Kinesis Data Analytics para aplicativos SQL.

O Amazon RDS publica cada log de banco de dados SQL Server como um fluxo separado de banco de dados no grupo de logs. Por exemplo, se você publicar os logs de erro e os logs do agente, os dados de erro serão armazenados em um fluxo de logs de erros no grupo de logs `/aws/rds/instance/`*`my_instance.node1`*`/error`, e os dados de log do agente serão armazenados no grupo de logs `/aws/rds/instance/`*`my_instance.node1`*`/agent`.

Para instâncias de banco de dados multi-AZ, o Amazon RDS publica o log do banco de dados como dois fluxos separados no grupo de logs. Por exemplo, se você publicar logs de erro, os dados de erro serão armazenados nos fluxos de log de erros `/aws/rds/instance/`*`my_instance.node1`*`/error` e `/aws/rds/instance/`*`my_instance.node2`*`/error` respectivamente. Os fluxos de log não mudam durante um failover e o fluxo de logs de erros de cada nó pode conter logs de erros da instância primária ou secundária. Com o multi-AZ, um fluxo de logs é criado automaticamente para

`/aws/rds/instance/my_instance/rds-events` para armazenar dados de eventos, como failovers de instâncias de banco de dados.

> **Note**
>
> A publicação de logs do SQL Server no CloudWatch Logs não está habilitada por padrão. Não há suporte para a publicação de arquivos de rastreamento e despejo. A publicação de logs do SQL Server no CloudWatch Logs é compatível com todas as regiões, exceto Ásia-Pacífico (Hong Kong).

Console

Para publicar logs de banco de dados do SQL Server no CloudWatch Logs a partir do AWS Management Console

1. Abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Databases (Bancos de dados) e a instância de banco de dados que você deseja modificar.
3. Selecione Modify.
4. Na seção Log exports (Exportações de log), escolha os logs que deseja começar a publicar no CloudWatch Logs.

   Você pode escolher Agent log (Log do agente), Error log (Log de erros) ou ambos.
5. Escolha Continue (Continuar) e depois escolha Modify DB Instance (Modificar instância de banco de dados) na página de resumo.

AWS CLI

Para publicar os logs do SQL Server, você pode usar o comando `modify-db-instance` com os seguintes parâmetros:

- `--db-instance-identifier`
- `--cloudwatch-logs-export-configuration`

> **Note**
>
> Uma alteração feita na opção `--cloudwatch-logs-export-configuration` sempre é aplicada imediatamente na instância de banco de dados. Por isso, as opções `--apply-immediately` e `--no-apply-immediately` não entram em vigor.

Você também pode publicar logs da SQL Server usando os seguintes comandos:

- `create-db-instance`
- `restore-db-instance-from-db-snapshot`
- `restore-db-instance-to-point-in-time`

Example

O exemplo a seguir cria uma instância de banco de dados SQL Server com a publicação no CloudWatch Logs habilitada. O valor `--enable-cloudwatch-logs-exports` é uma matriz JSON de strings que podem incluir `error`, `agent` ou ambos.

Para Linux, macOS ou Unix:

```
aws rds create-db-instance \
    --db-instance-identifier mydbinstance \
    --enable-cloudwatch-logs-exports '["error","agent"]' \
    --db-instance-class db.m4.large \
    --engine sqlserver-se
```

Para Windows:

```
aws rds create-db-instance ^
    --db-instance-identifier mydbinstance ^
    --enable-cloudwatch-logs-exports "[\"error\",\"agent\"]" ^
    --db-instance-class db.m4.large ^
    --engine sqlserver-se
```

> **Note**
>
> Ao usar o prompt de comando do Windows, você deve fazer o escape das aspas duplas (") no código JSON, prefixando-as com uma barra invertida (\).

Example

O exemplo a seguir altera uma instância de banco de dados SQL Server existente para publicar os arquivos de log no CloudWatch Logs. O valor `--cloudwatch-logs-export-configuration` é um objeto JSON. A chave desse objeto é `EnableLogTypes`, e seu valor é uma matriz de strings que pode incluir `error`, `agent` ou ambos.

Para Linux, macOS ou Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --cloudwatch-logs-export-configuration '{"EnableLogTypes":["error","agent"]}'
```

Para Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --cloudwatch-logs-export-configuration "{\"EnableLogTypes\":[\"error\",\"agent\"]}"
```

> **Note**
>
> Ao usar o prompt de comando do Windows, você deve fazer o escape das aspas duplas (") no código JSON, prefixando-as com uma barra invertida (\).

Example

O exemplo a seguir altera uma instância de banco de dados SQL Server existente para desabilitar a publicação de arquivos de log do agente no CloudWatch Logs. O valor `--cloudwatch-logs-export-configuration` é um objeto JSON. A chave desse objeto é `DisableLogTypes`, e seu valor é uma matriz de strings que pode incluir `error`, `agent` ou ambos.

Para Linux, macOS ou Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --cloudwatch-logs-export-configuration '{"DisableLogTypes":["agent"]}'
```

Para Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --cloudwatch-logs-export-configuration "{\"DisableLogTypes\":[\"agent\"]}"
```

> **Note**
>
> Ao usar o prompt de comando do Windows, você deve fazer o escape das aspas duplas (") no código JSON, prefixando-as com uma barra invertida (\).

Arquivos de log do banco de dados MySQL

Você pode monitorar os logs do MySQL diretamente por meio do console do Amazon RDS, da API do Amazon RDS, da AWS CLI ou dos SDKs da AWS. Você também pode acessar os logs do MySQL direcionando os logs para uma tabela de banco de dados no banco de dados primário e consultando essa tabela. Você pode usar o utilitário mysqlbinlog para baixar um log de binários.

Para mais informações sobre a visualização, o download e os logs de bancos de dados baseados no monitoramento de arquivos, consulte Monitorar arquivos de log do Amazon RDS.

Tópicos

- Visão geral dos logs de banco de dados do RDS para MySQL
- Publicação de logs do MySQL no Amazon CloudWatch Logs
- Enviar a saída de log do MySQL para tabelas
- Configurar o registro em log binário do RDS para MySQL para bancos de dados single-AZ
- Configuração do registro em log binário do MySQL para clusters de banco de dados multi-AZ
- Acessar logs binários do MySQL

Visão geral dos logs de banco de dados do RDS para MySQL

Você pode monitorar os seguintes tipos de arquivos de log do RDS para MySQL:

- Log de erros
- Log de consultas lentas
- Log geral
- Log de auditoria
- Log de instância
- Log de erros de autenticação de banco de dados do IAM

O log de erros do RDS para MySQL é gerado por padrão. Você pode gerar a consulta lenta e os logs gerais definindo parâmetros no seu grupo de parâmetros do banco de dados.

Tópicos

- Logs de erro do RDS para MySQL

- Logs gerais e de consultas lentas do RDS para MySQL
- Log de auditoria do MySQL
- Alternância e retenção de logs do RDS para MySQL
- Limites de tamanho em redo logs

Logs de erro do RDS para MySQL

O RDS para MySQL grava erros no arquivo `mysql-error.log`. Cada arquivo de log tem a hora em que foi gerado (em UTC) anexada ao seu nome. Os arquivos de log também possuem um carimbo de data/hora que ajuda você a determinar quando as entradas de log foram gravadas.

O RDS para MySQL grava no log de erros apenas na inicialização, no desligamento e quando encontra erros. Uma instância de banco de dados pode passar horas ou dias sem novas entradas gravadas no log de erros. Se você não vir nenhuma entrada recente, é porque o servidor não encontrou nenhum erro que tenha gerado uma entrada de log.

Por padrão, os logs de erros são filtrados para que apenas eventos inesperados, como erros, sejam exibidos. No entanto, os logs de erros também contêm algumas informações adicionais do banco de dados, por exemplo, o andamento da consulta, que não são mostradas. Portanto, mesmo sem erros reais, o tamanho dos logs de erros pode aumentar devido às atividades em andamento do banco de dados. Embora você possa ver determinado tamanho em bytes ou quilobytes para os logs de erros no AWS Management Console, eles poderão ter 0 byte quando você fizer download deles.

O RDS para MySQL grava o `mysql-error.log` no disco a cada cinco minutos. Ele acrescenta o conteúdo do log ao `mysql-error-running.log`.

O RDS para MySQL alterna o arquivo `mysql-error-running.log` de hora em hora. Ele retém os logs gerados durante as últimas duas semanas.

> **Note**
>
> Observe que o período de retenção é diferente entre o Amazon RDS e o Aurora.

Logs gerais e de consultas lentas do RDS para MySQL

É possível gravar o log de consultas lentas e o log geral do RDS para MySQL em um arquivo ou em uma tabela de banco de dados. Para isso, defina parâmetros em seu grupo de parâmetros de banco

de dados. Para obter informações sobre como criar e modificar um grupo de parâmetros de banco de dados, consulte Grupos de parâmetros para Amazon RDS. Você deve definir esses parâmetros antes de visualizar o log de consultas lentas ou o log geral no console do Amazon RDS ou usando a API do Amazon RDS, a CLI do Amazon RDS ou os SDKs da AWS.

Você pode controlar o registro em log do RDS para MySQL usando os parâmetros nesta lista:

- `slow_query_log`: para criar o log de consultas lentas, defina como 1. O padrão é 0.
- `general_log`: para criar o log geral, defina como 1. O padrão é 0.
- `long_query_time`: para evitar que as consultas de execução rápida sejam registradas no log de consultas lentas, especifique um valor para o tempo de execução de consultas mais curto a ser registrado, em segundos. O padrão é 10 segundos; o mínimo é 0. Se log_output = FILE, você poderá especificar um valor de ponto flutuante com resolução por microssegundo. Se log_output = TABLE, você deverá especificar um valor inteiro com a segunda resolução. Apenas as consultas cujo tempo de execução excede o valor `long_query_time` são registradas em log. Por exemplo, definir `long_query_time` como 0.1 impede que qualquer consulta que seja executada por menos de 100 milissegundos seja registrada.
- `log_queries_not_using_indexes`: para registrar todas as consultas que não usam um índice no log de consultas lentas, defina como 1. As consultas que não usam um índice são registradas, mesmo que seu tempo de execução seja inferior ao valor do parâmetro `long_query_time`. O padrão é 0.
- `log_output` *`option`*: você pode especificar uma das seguintes opções para o parâmetro `log_output`.
  - TABLE (padrão): grava consultas gerais na tabela `mysql.general_log` e consultas lentas na tabela `mysql.slow_log`.
  - FILE: grave logs de consultas gerais e lentas no sistema de arquivos.
  - NONE: desabilite o registro em log.

Para que dados de consulta lenta apareçam no Amazon CloudWatch Logs, as seguintes condições devem ser atendidas:

- O CloudWatch Logs deve ser configurado para incluir logs de consultas lentas.
- `slow_query_log` deve estar habilitado.
- `log_output` deve ser definido como `FILE`.
- A consulta deve demorar mais do que o tempo configurado para `long_query_time`.

Para obter mais informações sobre os log de consultas gerais e de consultas lentas, acesse os seguintes tópicos na documentação do MySQL:

- O log de consultas lentas
- O log de consultas gerais

Log de auditoria do MySQL

Para acessar o log de auditoria, a instância de banco de dados deve usar um grupo de opções personalizado com a opção `MARIADB_AUDIT_PLUGIN`. Para ter mais informações, consulte Suporte ao plug-in de auditoria do MariaDB para MySQL.

Alternância e retenção de logs do RDS para MySQL

Quando o registro em log está habilitado, o Amazon RDS faz o rodízio de logs de tabelas ou exclui arquivos de log em intervalos regulares. Essa medida é uma precaução para reduzir a possibilidade de um arquivo de log grande bloquear o uso do banco de dados ou afetar o desempenho. O RDS para MySQL lida com a alternância e a exclusão da seguinte forma:

- Os tamanhos de arquivo do log de consultas lentas, do log de erros e do log geral do MySQL são restritos a não mais de 2% do espaço de armazenamento alocado para uma instância de banco de dados. Para manter esse limite, os logs são alternados automaticamente a cada hora. O MySQL remove arquivos de log criados há mais de duas semanas. Se o tamanho do arquivo de log combinado exceder o limite após a remoção dos arquivos de log antigos, os arquivos de log mais antigos serão excluídos até o tamanho do arquivo de log deixar de exceder esse limite.
- Quando o registro em log `FILE` é ativado, os arquivos de log são examinados a cada hora e os arquivos de log com mais de duas semanas são excluídos. Em alguns casos, o tamanho do arquivo de log combinado restante após a exclusão pode exceder o limite de 2% do espaço alocado da instância de um banco de dados. Nesses casos, os arquivos de log mais antigos serão excluídos até que o tamanho do arquivo de log não exceda o limite.
- Quando o registro de `TABLE` estiver ativado, em alguns casos as tabelas de log serão rotacionadas a cada 24 horas. Essa alternância ocorrerá se o espaço utilizado pelos logs de tabelas for superior a 20% do espaço de armazenamento alocado. Isso também ocorrerá se o tamanho de todos os logs combinados for superior a 10 GB. Se a quantidade de espaço usada por uma instância de banco de dados for maior que 90 por cento do espaço de armazenamento alocado da instância de banco de dados, os limites para o rodízio do log serão reduzidos. As tabelas de logs serão, então, alternadas se o espaço utilizado pelos logs de tabelas for superior a 10% do espaço de armazenamento alocado. Eles também serão alternados se o tamanho de

todos os logs combinados for superior a 5 GB. Você pode assinar a categoria de evento `low storage` para ser notificado quando tabelas de log forem rotacionadas para liberar espaço. Para ter mais informações, consulte Trabalhar com a notificação de eventos do Amazon RDS.

Quando as tabelas de log são alternadas, a tabela de logs atual é copiada primeiro em uma tabela de logs de backup. Depois, as entradas na tabela de logs atual são removidas. Se a tabela de log de backup já existir, então ela será excluída antes que a tabela de log atual seja copiada ao backup. Você pode consultar a tabela de log de backup, se necessário. A tabela de log de backup para a tabela `mysql.general_log` é denominada `mysql.general_log_backup`. A tabela de log de backup para a tabela `mysql.slow_log` é denominada `mysql.slow_log_backup`.

Você pode rotacionar a tabela `mysql.general_log` chamando o procedimento `mysql.rds_rotate_general_log`. Você pode rotacionar a tabela `mysql.slow_log` chamando o procedimento `mysql.rds_rotate_slow_log`.

Os logs de tabelas são rotacionados durante um upgrade de versão do banco de dados.

Para trabalhar com os logs no console do Amazon RDS, na API do Amazon RDS, na CLI do Amazon RDS ou nos SDKs da AWS, defina o parâmetro `log_output` como FILE. Como o log de erros do MySQL, esses arquivos de log são rotacionados por hora. Os arquivos de log que foram gerados durante as duas semanas anteriores são retidos. Observe que o período de retenção é diferente entre o Amazon RDS e o Aurora.

Limites de tamanho em redo logs

Para o RDS para MySQL versão 8.0.32 e anterior, o valor padrão desse parâmetro é 256 MB. Esse valor é obtido multiplicando o valor padrão do parâmetro `innodb_log_file_size` (128 MB) pelo valor padrão do parâmetro `innodb_log_files_in_group` (2). Para ter mais informações, consulte Best practices for configuring parameters for Amazon RDS for MySQL, part 1: Parameters related to performance.

Para o RDS para MySQL versão 8.0.33 e posterior e versões secundárias posteriores, o Amazon RDS usa o parâmetro `innodb_redo_log_capacity` em vez do `innodb_log_file_size`. O valor padrão do parâmetro `innodb_redo_log_capacity` do Amazon RDS é 2 GB. Para obter mais informações, consulte Changes in MySQL 8.0.30 (Alterações no MySQL 8.0.30) na documentação do MySQL.

A partir do MySQL 8.4, o Amazon RDS habilita o parâmetro `innodb_dedicated_server` por padrão. Com o parâmetro `innodb_dedicated_server`, o mecanismo do banco de dados calcula

os parâmetros `innodb_buffer_pool_size` e `innodb_redo_log_capacity`. Para ter mais informações, consulte Configuração do tamanho do pool de buffer e a capacidade de log redo no MySQL 8.4.

Publicação de logs do MySQL no Amazon CloudWatch Logs

Você pode configurar sua instância de banco de dados MySQL para publicar dados de log em um grupo de log no Amazon CloudWatch Logs. Com o CloudWatch Logs, você pode executar análise em tempo real de dados de log e usar o CloudWatch para criar alarmes e visualizar métricas. Você pode usar o CloudWatch Logs para armazenar seus registros de log em armazenamento resiliente.

O Amazon RDS publica cada log de banco de dados MySQL como um fluxo separado de banco de dados no grupo de logs. Por exemplo, se você configurar a função de exportação para incluir o log de consultas lentas, os dados de consultas lentas serão armazenados em um fluxo de log de consultas lentas no grupo de logs `/aws/rds/instance/my_instance/slowquery`.

O log de erros está habilitado por padrão. A tabela a seguir resume os requisitos de outros logs do MySQL.

Log	Requisito
Log de auditoria	A instância de banco de dados deve usar um grupo de opções personalizado com a opção `MARIADB_AUDIT_PLUGIN` .
Log geral	A instância de banco de dados deve usar um grupo de parâmetros personalizado com a configuração de parâmetro `general_log = 1` para habilitar o log geral.
Log de consultas lentas	A instância de banco de dados deve usar um grupo de parâmetros personalizado com a configuração de parâmetro `slow_query_log = 1` para habilitar o log de consulta lenta.
Log de erros de autenticação de banco de dados do IAM	Você deve habilitar o tipo de log `iam-db-auth-error` para uma instância de banco de

Log	Requisito
	dados criando ou modificando uma instância de banco de dados.
Resultado de saída do log	A instância de banco de dados deve usar um grupo de parâmetros personalizado com a configuração de parâmetro `log_output = FILE` para gravar logs no sistema de arquivos e publicá-los no CloudWatch Logs.

Console

Para publicar logs do MySQL no CloudWatch Logs usando o console

1. Abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Databases (Bancos de dados) e a instância de banco de dados que você deseja modificar.
3. Selecione Modify.
4. Na seção Log exports (Exportações de log), escolha os logs que deseja começar a publicar no CloudWatch Logs.
5. Escolha Continue (Continuar) e depois escolha Modify DB Instance (Modificar instância de banco de dados) na página de resumo.

AWS CLI

Você pode publicar logs do MySQL com o AWS CLI. Você pode chamar o comando `modify-db-instance` com os seguintes parâmetros:

- `--db-instance-identifier`
- `--cloudwatch-logs-export-configuration`

> **Note**
>
> Uma alteração feita na opção `--cloudwatch-logs-export-configuration` sempre é aplicada imediatamente na instância de banco de dados. Por isso, as opções `--apply-immediately` e `--no-apply-immediately` não entram em vigor.

Também é possível publicar logs do MySQL chamando os seguintes comandos AWS CLI:

- `create-db-instance`
- `restore-db-instance-from-db-snapshot`
- `restore-db-instance-from-s3`
- `restore-db-instance-to-point-in-time`

Execute um destes comandos da AWS CLI com as seguintes opções:

- `--db-instance-identifier`
- `--enable-cloudwatch-logs-exports`
- `--db-instance-class`
- `--engine`

Outras opções podem ser obrigatórias, dependendo do comando da AWS CLI executado.

Example

O exemplo a seguir altera uma instância de Banco de Dados MySQL existente para publicar os arquivos de log no CloudWatch Logs. O valor `--cloudwatch-logs-export-configuration` é um objeto JSON. A chave para esse objeto é `EnableLogTypes`, e seu valor é uma matriz de strings com qualquer combinação de `audit`, `error`, `general` e `slowquery`.

Para Linux, macOS ou Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --cloudwatch-logs-export-configuration '{"EnableLogTypes":
["audit","error","general","slowquery"]}'
```

Para Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --cloudwatch-logs-export-configuration '{"EnableLogTypes":
["audit","error","general","slowquery"]}'
```

Example

O exemplo a seguir cria uma instância de Banco de Dados MySQL e publica os arquivos de log no CloudWatch Logs. O valor `--enable-cloudwatch-logs-exports` é uma matriz de strings JSON. As strings podem ser qualquer combinação de `audit`, `error`, `general` e `slowquery`.

Para Linux, macOS ou Unix:

```
aws rds create-db-instance \
    --db-instance-identifier mydbinstance \
    --enable-cloudwatch-logs-exports '["audit","error","general","slowquery"]' \
    --db-instance-class db.m4.large \
    --engine MySQL
```

Para Windows:

```
aws rds create-db-instance ^
    --db-instance-identifier mydbinstance ^
    --enable-cloudwatch-logs-exports '["audit","error","general","slowquery"]' ^
    --db-instance-class db.m4.large ^
    --engine MySQL
```

API do RDS

Você pode publicar logs do MySQL com a API da RDS. Você pode chamar a ação `ModifyDBInstance` com os seguintes parâmetros:

- `DBInstanceIdentifier`
- `CloudwatchLogsExportConfiguration`

> **Note**
>
> Uma alteração feita no parâmetro `CloudwatchLogsExportConfiguration` sempre é aplicada imediatamente na instância de banco de dados. Por isso, o parâmetro `ApplyImmediately` não entra em vigor.

Também é possível publicar logs do MySQL chamando as seguintes operações da API do RDS:

- `CreateDBInstance`
- `RestoreDBInstanceFromDBSnapshot`
- `RestoreDBInstanceFromS3`
- `RestoreDBInstanceToPointInTime`

Execute uma destas operações da API do RDS com os seguintes parâmetros:

- `DBInstanceIdentifier`
- `EnableCloudwatchLogsExports`
- `Engine`
- `DBInstanceClass`

Outros parâmetros podem ser exigidos dependendo do comando AWS CLI que você executa.

Enviar a saída de log do MySQL para tabelas

Você pode direcionar os logs de consultas gerais e lentas para tabelas na instância de banco de dados, criando um grupo de parâmetros de banco de dados e definindo o parâmetro do servidor `log_output` como `TABLE`. As consultas gerais são registradas na tabela `mysql.general_log` e as consultas lentas são registradas na tabela `mysql.slow_log`. Você pode consultar as tabelas para acessar as informações do log. Habilitar esse registro aumenta a quantidade de dados gravados no banco de dados, o que pode degradar a performance.

O log geral e os logs de consultas lentas estão desabilitados por padrão. Para habilitar o registro em log de tabelas, você também deve definir os parâmetros de servidor `general_log` e `slow_query_log` como 1.

As tabelas de log continuarão crescendo até que as respectivas atividades de log sejam desativadas com a redefinição do parâmetro apropriado como `0`. Uma grande quantidade de dados geralmente se acumula ao longo do tempo, o que pode consumir uma porcentagem considerável do espaço de armazenamento alocado. O Amazon RDS não permite truncar tabelas de log, mas é possível mover o conteúdo delas. Rotacionar uma tabela salva seu conteúdo em uma tabela de backup e, em seguida, cria uma nova tabela de log vazia. Você pode rotacionar manualmente as tabelas de log com os seguintes procedimentos de linha de comando, em que o prompt de comando é indicado por `PROMPT>`:

```
PROMPT> CALL mysql.rds_rotate_slow_log;
PROMPT> CALL mysql.rds_rotate_general_log;
```

Para remover completamente os dados antigos e recuperar o espaço em disco, chame o procedimento apropriado duas vezes sucessivamente.

Configurar o registro em log binário do RDS para MySQL para bancos de dados single-AZ

O log binário é um conjunto de arquivos de log que contêm informações sobre modificações de dados feitas em uma instância do servidor MySQL. O log binário contém informações como as seguintes:

- Eventos que descrevem alterações no banco de dados, como criação de tabela ou modificações de linha
- Informações sobre a duração de cada instrução que atualizou dados
- Eventos para declarações que poderiam ter dados atualizados, mas não foram

O log binário registra instruções que são enviadas durante a replicação. Também é necessário para algumas operações de recuperação. Para obter mais informações, consulte The Binary Log na documentação do MySQL.

O recurso de backups automatizados determina se o registro em log binário está ativado ou desativado para o MySQL. Você tem as seguintes opções:

Ativar o registro em log binário

Defina o período de retenção de backup para um valor positivo diferente de zero.

Desativar o registro em log binário

Defina o período de retenção de backup para 0.

Para ter mais informações, consulte Ativar backups automáticos.

O MySQL no Amazon RDS é compatível com os formatos de registros em log binários baseados em linha, baseados em instrução e mistos. Recomendamos misto, a menos que você precise de um formato específico de log binário. Para obter detalhes sobre os diferentes formatos de logs binários do MySQL, consulte Formatos de registro em log binário na documentação do MySQL.

Se você pretende usar replicação, o formato do registro em log binário é importante porque determina o registro de alterações feitas nos dados salvas na origem e enviadas para os destinos de replicação. Para obter informações sobre as vantagens e as desvantagens de formatos de registro em logs binários para replicação, consulte Vantagens e desvantagens da replicação baseada em instrução e baseada em linha na documentação do MySQL.

Important

Com o MySQL 8.0.34, o MySQL descontinuou o parâmetro `binlog_format`. Em versões posteriores do MySQL, o MySQL pretende remover o parâmetro e oferecer suporte somente à replicação baseada em linhas. Como resultado, recomendamos o uso de registro baseado em linhas para novas configurações de replicação do MySQL. Para obter mais informações, consulte binlog_format na documentação do MySQL.
As versões 8.0 e 8.4 do MySQL aceitam o parâmetro `binlog_format`. Ao usar esse parâmetro, o MySQL emite um aviso de descontinuação. Em uma futura versão principal, o MySQL removerá o parâmetro `binlog_format`.
A replicação baseada em instrução pode causar inconsistências entre a instância de banco de dados de origem e uma réplica de leitura. Para obter mais informações, consulte Determinação de instruções seguras e inseguras no registro de logs binários. na documentação do MySQL.
Habilitar o registro em log binário aumenta o número de operações de E/S do disco de gravação na instância de banco de dados. Você pode monitorar o uso de IOPS com a métrica `WriteIOPS` do CloudWatch.

Para definir o formato de registro em log binário do MySQL

1. Abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, selecione Parameter groups.
3. Selecione o grupo de parâmetros do de banco de dados associado à instância de banco de dados que você deseja modificar.

   Não é possível modificar um grupo de parâmetros padrão. Se a instância de banco de dados estiver usando um grupo de parâmetros padrão, crie outro grupo de parâmetros e o associe à instância de banco de dados.

   Para ter mais informações sobre grupos de parâmetros, consulte Grupos de parâmetros para Amazon RDS.
4. Em Ações, selecione Editar.
5. Defina o parâmetro `binlog_format` como o formato de registro em log binário de sua escolha (`ROW`, `STATEMENT` ou `MIXED`).

   Você pode desativar o registro em log binário definindo o período de retenção de backup de uma instância de banco de dados como zero, mas isso desativa os backups automatizados diários. Desabilitar os backups automatizados desativa ou desabilita a variável de sessão `log_bin`. Isso desabilita o registro em log binário na instância de banco de dados do RDS para MySQL que, por sua vez, redefine a variável de sessão `binlog_format` como o valor padrão de `ROW` no banco de dados. Recomendamos não desabilitar os backups. Para receber mais informações sobre a configuração Período de retenção de backup, consulte Configurações para instâncias de banco de dados.
6. Escolha Salvar alterações para salvar as atualizações no grupo de parâmetros de banco de dados.

Como o parâmetro `binlog_format` é dinâmico no RDS para MySQL, você não precisa reinicializar a instância de banco de dados para que as alterações sejam aplicadas. (Observe que, no Aurora MySQL, esse parâmetro é estático. Consulte mais informações em Configurar o registro em log binário do Aurora MySQL.)

> **Important**
>
> Alterar um grupo de parâmetros de banco de dados afeta todas as instâncias de banco de dados que usam esse grupo de parâmetros. Se você quiser especificar diferentes

formatos do registro em log binário para diferentes instâncias de banco de dados MySQL em uma região da AWS, as instâncias de banco de dados deverão usar diferentes grupos de parâmetros de banco de dados. Esses grupos de parâmetros identificam diferentes formatos de log. Atribua o grupo de parâmetros de banco de dados apropriado a cada instância de banco de dados.

Configuração do registro em log binário do MySQL para clusters de banco de dados multi-AZ

O registro em log binário no Amazon RDS para MySQL para clusters de banco de dados multi-AZ registra todas as alterações do banco de dados para oferecer suporte a replicação, recuperação point-in-time e auditoria. Em clusters de banco de dados multi-AZ, os logs binários sincronizam os nós secundários com o nó primário, garantindo a consistência de dados em todas as zonas de disponibilidade e permitindo failovers contínuos.

Para otimizar o registro em log binário, o Amazon RDS oferece suporte à compactação de transações de log binário, o que reduz os requisitos de armazenamento para logs binários e melhora a eficiência da replicação.

Tópicos

- Compactação de transação de log binário para clusters de banco de dados multi-AZ
- Configuração da compactação de transação de log binário para clusters de banco de dados multi-AZ

Compactação de transação de log binário para clusters de banco de dados multi-AZ

A compactação de transações de log binário usa o algoritmo zstd para reduzir o tamanho dos dados de transação armazenados em logs binários. Quando habilitado, o mecanismo de banco de dados MySQL comprime as cargas úteis de transação em um único evento, minimizando a sobrecarga de E/S e armazenamento. Esse recurso melhora o desempenho do banco de dados, reduz o tamanho do log binário e otimiza o uso de recursos para gerenciar e replicar logs em clusters de banco de dados multi-AZ.

O Amazon RDS fornece compactação de transação de log binário para clusters de banco de dados multi-AZ do RDS para MySQL por meio dos seguintes parâmetros:

- `binlog_transaction_compression`: quando habilitado (1), o mecanismo de banco de dados compacto as cargas úteis da transação e as grava no log binário como um único evento. Isso reduz o uso do armazenamento e a sobrecarga de E/S. Por padrão, o parâmetro é desabilitado.
- `binlog_transaction_compression_level_zstd`: configura o nível de compressão zstd para transações de log binário. Valores mais altos aumentam a taxa de compactação, reduzindo ainda mais os requisitos de armazenamento, mas aumentando o uso da CPU e da memória para compactação. O valor padrão é 3, com um intervalo de 1 a 22.

Esses parâmetros permitem ajustar a compactação de logs binários com base nas características da workload e na disponibilidade de recursos. Para obter mais informações, consulte Compactação de transações de log binário na documentação do MySQL.

A compactação de transações de log binário oferece os seguintes benefícios principais:

- A compactação diminui o tamanho dos logs binários, especialmente para workloads com grandes transações ou altos volumes de gravação.
- Logs binários menores reduzem a sobrecarga de rede e de E/S, aprimorando o desempenho da replicação.
- O parâmetro `binlog_transaction_compression_level_zstd` fornece controle sobre o equilíbrio entre a taxa de compressão e o consumo de recursos.

Configuração da compactação de transação de log binário para clusters de banco de dados multi-AZ

Para configurar a compactação de transações de log binário para um cluster de banco de dados multi-AZ do RDS para MySQL, modifique as configurações relevantes dos parâmetros do cluster de acordo com seus requisitos de workload.

Console

Como habilitar a compactação de transações de log binário

1. Modifique o grupo de parâmetros de cluster de banco de dados para definir o parâmetro `binlog_transaction_compression` como 1.
2. (Opcional) Ajuste o valor do parâmetro `binlog_transaction_compression_level_zstd` com base nos requisitos de workload e na disponibilidade de recursos.

Para ter mais informações, consulte the section called “Modificar parâmetros em um grupo de parâmetros de cluster de banco de dados”.

AWS CLI

Para configurar a compactação de transações de log binário usando a AWS CLI, use o comando modify-db-cluster-parameter-group.

Example

Para Linux, macOS ou Unix:

```
aws rds modify-db-cluster-parameter-group \
  --db-cluster-parameter-group-name your-cluster-parameter-group \
  --parameters
 "ParameterName=binlog_transaction_compression,ParameterValue=1,ApplyMethod=pending-
reboot"
```

Para Windows:

```
aws rds modify-db-cluster-parameter-group ^
  --db-cluster-parameter-group-name your-cluster-parameter-group ^
  --parameters
 "ParameterName=binlog_transaction_compression,ParameterValue=1,ApplyMethod=pending-
reboot"
```

API do RDS

Para configurar a compactação de transações de log binário usando a API do Amazon RDS, use a operação `ModifyDBClusterParameterGroup`.

Acessar logs binários do MySQL

É possível usar o utilitário mysqlbinlog para baixar ou transmitir os logs binários de instâncias do RDS para instâncias de banco de dados do MySQL. O log binário é baixado para o computador local, onde você pode realizar ações como reproduzir o log usando o utilitário mysql. Para ter mais informações sobre como usar o utilitário mysqlbinlog, acesse Usar mysqlbinlog para fazer backup de arquivos de log binários na documentação do MySQL.

Para executar o utilitário mysqlbinlog em uma instância do Amazon RDS, use as seguintes opções:

- `--read-from-remote-server` – obrigatório.
- `--host`: o nome DNS do endpoint da instância.
- `--port`: a porta usada pela instância.
- `--user`: um usuário do MySQL ao qual foi concedida a permissão `REPLICATION SLAVE`.
- `--password`: a senha do usuário do MySQL ou omita um valor de senha de forma que o utilitário solicite uma senha.
- `--raw`: baixe o arquivo em formato binário.
- `--result-file`: o arquivo local para receber a saída bruta.
- `--stop-never`: transmita os arquivos de log binários.
- `--verbose`: ao usar o formato de log binário `ROW`, inclua essa opção para ver os eventos de linha como instruções pseudo-SQL. Para ter mais informações sobre a opção `--verbose`, consulte Exibição de evento da linha mysqlbinlog na documentação do MySQL.
- Especifique os nomes de um ou mais arquivos de log binários. Para obter uma lista dos logs disponíveis, use o comando SQL `SHOW BINARY LOGS`.

Para ter mais informações sobre as opções de mysqlbinlog, consulte mysqlbinlog: utilitário para processar arquivos de log binários na documentação do MySQL.

Os exemplos a seguir mostram como usar o utilitário mysqlbinlog.

Para Linux, macOS ou Unix:

```
mysqlbinlog \
    --read-from-remote-server \
    --host=MySQLInstance1.cg034hpkmmjt.region.rds.amazonaws.com \
    --port=3306  \
    --user ReplUser \
    --password \
    --raw \
    --verbose \
    --result-file=/tmp/ \
    binlog.00098
```

Para Windows:

```
mysqlbinlog ^
    --read-from-remote-server ^
```

```
    --host=MySQLInstance1.cg034hpkmmjt.region.rds.amazonaws.com ^
    --port=3306  ^
    --user ReplUser ^
    --password ^
    --raw ^
    --verbose ^
    --result-file=/tmp/ ^
    binlog.00098
```

Os logs binários devem permanecer disponíveis na instância de banco de dados para que o utilitário mysqlbinlog possa acessá-los. Para garantir a disponibilidade deles, use o procedimento armazenado mysql.rds_set_configuration e especifique um período com tempo suficiente para você baixar os logs. Se essa configuração não for definida, o Amazon RDS purgará os logs binários o mais rápido possível, causando lacunas nos logs binários que o utilitário mysqlbinlog recupera.

O exemplo a seguir define o período de retenção como 1 dia.

```
call mysql.rds_set_configuration('binlog retention hours', 24);
```

Para exibir a configuração atual, use o procedimento armazenado mysql.rds_show_configuration.

```
call mysql.rds_show_configuration;
```

Arquivos de log do banco de dados do Amazon RDS para Oracle

Você pode acessar os logs de alerta, os arquivos de auditoria e os arquivos de rastreamento da Oracle usando o console ou a API do Amazon RDS. Para mais informações sobre a visualização, o download e os logs de bancos de dados baseados no monitoramento de arquivos, consulte Monitorar arquivos de log do Amazon RDS.

Os arquivos de auditoria da Oracle fornecidos são os arquivos de auditoria padrão da Oracle. O Amazon RDS é compatível com o recurso de auditoria refinada (FGA) da Oracle. Contudo, o acesso ao log não fornece acesso a eventos do FGA que são armazenados na tabela `SYS.FGA_LOG$` e são acessados por meio da visualização `DBA_FGA_AUDIT_TRAIL`.

A operação da API `DescribeDBLogFiles` que lista os arquivos de log da Oracle disponíveis para uma instância de banco de dados ignora o parâmetro `MaxRecordse` retorna até 1.000 registros. A chamada retorna `LastWritten` como uma data POSIX em milissegundos.

Tópicos

- Cronograma de retenção
- Trabalhar com arquivos de rastreamento da Oracle
- Publicação de logs da Oracle no Amazon CloudWatch Logs
- Acesso aos logs de alerta e logs de receptor

Cronograma de retenção

O mecanismo de banco de dados Oracle pode fazer a rotação dos arquivos de log se eles ficarem muito grandes. Baixe esses arquivos de auditoria ou de rastreamento para retê-los. Se você armazenar os arquivos localmente, reduzirá os custos de armazenamento do Amazon RDS e disponibilizará mais espaço para seus dados.

A tabela a seguir mostra o cronograma de retenção para os logs de alerta, arquivos de auditoria e arquivos de rastreamento da Oracle no Amazon RDS.

Tipo de log	Cronograma de retenção
Logs de alerta	O log de alertas de texto é alternado diariamente com uma retenção de 30 dias gerenciada pelo Amazon RDS. O log de alerta de XML é

Tipo de log	Cronograma de retenção
	retido por pelo menos sete dias. Você pode acessar esse log usando a visualização `ALERTLOG`.
Arquivos de auditoria	O período de retenção padrão para arquivos de rastreamento é de sete dias. O Amazon RDS poderá excluir arquivos de auditoria com mais de sete dias.
Arquivos de rastreamento	O período de retenção padrão para arquivos de rastreamento é de sete dias. O Amazon RDS poderá excluir arquivos de rastreamento com mais de sete dias.
Logs de listener	O período de retenção padrão para os logs de listener é sete dias. O Amazon RDS poderá excluir logs de listener com mais de sete dias.

> **Note**
>
> Arquivos de auditoria e de rastreamento compartilham a mesma configuração de retenção.

Trabalhar com arquivos de rastreamento da Oracle

A seguir, você pode localizar descrições de procedimentos do Amazon RDS para criar, atualizar, acessar e excluir arquivos de rastreamento.

Tópicos

- Listar arquivos
- Gerar arquivos de rastreamento e rastrear uma sessão
- Recuperar arquivos de rastreamento
- Limpar arquivos de rastreamento

Listar arquivos

Você pode usar qualquer um dos dois procedimentos para permitir acesso a qualquer arquivo no caminho `background_dump_dest`. O primeiro procedimento atualiza uma visualização que contém uma lista de todos os arquivos que estão em no moment `background_dump_dest`.

```
EXEC rdsadmin.manage_tracefiles.refresh_tracefile_listing;
```

Depois que a visualização for atualizada, consulte a visualização a seguir para acessar os resultados.

```
SELECT * FROM rdsadmin.tracefile_listing;
```

Uma alternativa ao processo anterior é usar `FROM table` para transmitir dados não relacionais que não estejam em tabelas em um formato semelhante ao de uma tabela para listar o conteúdo do diretório de banco de dados.

```
SELECT * FROM TABLE(rdsadmin.rds_file_util.listdir('BDUMP'));
```

A consulta a seguir mostra o texto de um arquivo de log.

```
SELECT text FROM
 TABLE(rdsadmin.rds_file_util.read_text_file('BDUMP','alert_dbname.log.date'));
```

Em uma réplica de leitura, obtenha o nome do diretório BDUMP consultando `V$DATABASE.DB_UNIQUE_NAME`. Se o nome exclusivo for `DATABASE_B`, o diretório BDUMP será `BDUMP_B`. O exemplo a seguir consulta o nome BDUMP em uma réplica e usa esse nome para consultar o conteúdo de `alert_DATABASE.log.2020-06-23`.

```
SELECT 'BDUMP' || (SELECT regexp_replace(DB_UNIQUE_NAME,'.*(_[A-Z])', '\1') FROM V
$DATABASE) AS BDUMP_VARIABLE FROM DUAL;

BDUMP_VARIABLE
--------------
BDUMP_B

SELECT TEXT FROM
 table(rdsadmin.rds_file_util.read_text_file('BDUMP_B','alert_DATABASE.log.2020-06-23'));
```

Gerar arquivos de rastreamento e rastrear uma sessão

Como não há restrições em `ALTER SESSION`, muitos métodos padrão para gerar arquivos de rastreamento da Oracle permanecem disponíveis para uma instância de banco de dados do Amazon RDS. Os procedimentos a seguir são fornecidos para arquivos de rastreamento que exigem maior acesso.

Método Oracle	Método do Amazon RDS
`oradebug hanganalyze 3`	`EXEC rdsadmin.manage_tracefiles.hanganalyze;`
`oradebug dump systemstate 266`	`EXEC rdsadmin.manage_tracefiles.dump_systemstate;`

Você pode usar muitos métodos padrão para rastrear sessões individuais conectadas a uma instância de banco de dados Oracle no Amazon RDS. Para habilitar o rastreamento de uma sessão, é possível executar subprogramas em pacotes PL/SQL fornecidos pela Oracle, como `DBMS_SESSION` e `DBMS_MONITOR`. Para obter mais informações, consulte Habilitar o rastreamento de uma sessão na documentação da Oracle.

Recuperar arquivos de rastreamento

Você pode recuperar qualquer arquivo de rastreamento em `background_dump_dest` usando uma consulta SQL padrão em uma tabela externa gerenciada pelo Amazon RDS. Para usar este método, execute o procedimento e defina o local desta tabela para o arquivo de rastreamento específico.

Por exemplo, você pode usar a visualização `rdsadmin.tracefile_listing` mencionada anteriormente para listar todos os arquivos de rastreamento no sistema. Em seguida, você pode definir a visualização `tracefile_table` para apontar para o arquivo de rastreamento pretendido usando o seguinte procedimento.

```
EXEC
 rdsadmin.manage_tracefiles.set_tracefile_table_location('CUST01_ora_3260_SYSTEMSTATE.trc');
```

O exemplo a seguir criar uma tabela externa no esquema atual com o local definido para o arquivo fornecido. É possível recuperar o conteúdo em um arquivo local usando uma consulta SQL.

```
SPOOL /tmp/tracefile.txt
SELECT * FROM tracefile_table;
SPOOL OFF;
```

Limpar arquivos de rastreamento

Os arquivos de rastreamento podem acumular e consumir espaço em disco. Por padrão, o Amazon RDS limpa os arquivos de rastreamento e os arquivos de log com mais de sete dias. Você pode visualizar e definir o período de retenção do arquivo de rastreamento usando o procedimento `show_configuration`. Você deve executar o comando `SET SERVEROUTPUT ON` para que possa visualizar os resultados da configuração.

O exemplo a seguir mostra o período de retenção do arquivo de rastreamento atual e, em seguida, define um novo período de retenção para o arquivo de rastreamento.

```
# Show the current tracefile retention
SQL> EXEC rdsadmin.rdsadmin_util.show_configuration;
NAME:tracefile retention
VALUE:10080
DESCRIPTION:tracefile expiration specifies the duration in minutes before tracefiles in
 bdump are automatically deleted.

# Set the tracefile retention to 24 hours:
SQL> EXEC rdsadmin.rdsadmin_util.set_configuration('tracefile retention',1440);
SQL> commit;

#show the new tracefile retention
SQL> EXEC rdsadmin.rdsadmin_util.show_configuration;
NAME:tracefile retention
VALUE:1440
DESCRIPTION:tracefile expiration specifies the duration in minutes before tracefiles in
 bdump are automatically deleted.
```

Além do processo periódico de limpeza, você pode remover arquivos de manualment `background_dump_dest`. O exemplo a seguir mostra como limpar todos os arquivos que tenham mais de cinco minutos.

```
EXEC rdsadmin.manage_tracefiles.purge_tracefiles(5);
```

Também limpe todos os arquivos correspondentes a um padrão específico (do contrário, não inclua a extensão do arquivo, como .trc). O exemplo a seguir mostra como limpar todos os arquivos que começam com `SCHPOC1_ora_5935`.

```
EXEC rdsadmin.manage_tracefiles.purge_tracefiles('SCHPOC1_ora_5935');
```

Publicação de logs da Oracle no Amazon CloudWatch Logs

É possível configurar a instância de banco de dados do RDS para Oracle para publicar dados de log em um grupo de logs no Amazon CloudWatch Logs. Com o CloudWatch Logs, você pode analisar os dados de log e usar o CloudWatch para criar alarmes e visualizar métricas. Você pode usar o CloudWatch Logs para armazenar seus registros de log em armazenamento resiliente.

O Amazon RDS publica cada log de banco de dados Oracle como um fluxo separado de banco de dados no grupo de logs. Por exemplo, se você configurar a função de exportação para incluir o log de auditoria, os dados de auditoria serão armazenados em um fluxo de log de auditoria no grupo de logs `/aws/rds/instance/my_instance/audit`. A tabela a seguir resume os requisitos do RDS para Oracle para publicar logs no Amazon CloudWatch Logs.

Nome do log	Requisito	Padrão
Log de alertas	Nenhum. Não é possível desabilitar esse log.	Habilitado
Log de rastreamento	Defina o parâmetro `trace_enabled` como TRUE ou deixe-o definido como padrão.	TRUE
Log de auditoria	Defina o parâmetro `audit_trail` como qualquer um dos seguintes valores permitidos: `{ none | os | db [, extended] | xml [, extended] }`	none
Log de listener	Nenhum. Não é possível desabilitar esse log.	Habilitado
Log do Oracle Management Agent	Nenhum. Não é possível desabilitar esse log.	Habilitado

Esse log do Oracle Management Agent consiste nos grupos de logs mostrados na tabela a seguir.

Nome do log	Grupo de logs do CloudWatch
emctl.log	oemagent-emctl
emdctlj.log	oemagent-emdctlj
gcagent.log	oemagent-gcagent
gcagent_errors.log	oemagent-gcagent-errors
emagent.nohup	oemagent-emagent-nohup
secure.log	oemagent-secure

Para obter mais informações, consulte Locating Management Agent Log and Trace Files (Localizar arquivos de log e rastreamento do agente de gerenciamento) na documentação do Oracle.

Console

Para publicar logs de banco de dados Oracle no CloudWatch Logs pelo AWS Management Console

1. Abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Databases (Bancos de dados) e a instância de banco de dados que você deseja modificar.
3. Selecione Modify.
4. Na seção Log exports (Exportações de log), escolha os logs que deseja começar a publicar no CloudWatch Logs.
5. Escolha Continue (Continuar) e depois escolha Modify DB Instance (Modificar instância de banco de dados) na página de resumo.

AWS CLI

Para publicar os logs da Oracle, você pode usar o comando `modify-db-instance` com os seguintes parâmetros:

- `--db-instance-identifier`
- `--cloudwatch-logs-export-configuration`

> **Note**
>
> Uma alteração feita na opção `--cloudwatch-logs-export-configuration` sempre é aplicada imediatamente na instância de banco de dados. Por isso, as opções `--apply-immediately` e `--no-apply-immediately` não entram em vigor.

Você também pode publicar os logs da Oracle usando os seguintes comandos:

- `create-db-instance`
- `restore-db-instance-from-db-snapshot`
- `restore-db-instance-from-s3`
- `restore-db-instance-to-point-in-time`

Example

O exemplo a seguir cria uma instância de Banco de Dados Oracle com a publicação no CloudWatch Logs habilitada. O valor `--cloudwatch-logs-export-configuration` é uma matriz de strings JSON. As strings podem ser qualquer combinação de `alert`, `audit`, `listener` e `trace`.

Para Linux, macOS ou Unix:

```
aws rds create-db-instance \
    --db-instance-identifier mydbinstance \
    --cloudwatch-logs-export-configuration
 '["trace","audit","alert","listener","oemagent"]' \
    --db-instance-class db.m5.large \
    --allocated-storage 20 \
    --engine oracle-ee \
    --engine-version 19.0.0.0.ru-2024-04.rur-2024-04.r1 \
    --license-model bring-your-own-license \
    --master-username myadmin \
    --manage-master-user-password
```

Para Windows:

```
aws rds create-db-instance ^
    --db-instance-identifier mydbinstance ^
    --cloudwatch-logs-export-configuration trace alert audit listener oemagent ^
```

```
    --db-instance-class db.m5.large ^
    --allocated-storage 20 ^
    --engine oracle-ee ^
    --engine-version 19.0.0.0.ru-2024-04.rur-2024-04.r1 ^
    --license-model bring-your-own-license ^
    --master-username myadmin ^
    --manage-master-user-password
```

Example

O exemplo a seguir altera uma instância de Banco de Dados Oracle existente para publicar os arquivos de log no CloudWatch Logs. O valor `--cloudwatch-logs-export-configuration` é um objeto JSON. A chave para esse objeto é `EnableLogTypes`, e seu valor é uma matriz de strings com qualquer combinação de `alert`, `audit`, `listener` e `trace`.

Para Linux, macOS ou Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --cloudwatch-logs-export-configuration '{"EnableLogTypes":
["trace","alert","audit","listener","oemagent"]}'
```

Para Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --cloudwatch-logs-export-configuration EnableLogTypes=\"trace\",\"alert\",\"audit
\",\"listener\",\"oemagent\"
```

Example

O exemplo a seguir modifica uma instância de Banco de Dados Oracle existente para desabilitar a publicação de arquivos de log de auditoria e de listener no CloudWatch Logs. O valor `--cloudwatch-logs-export-configuration` é um objeto JSON. A chave para esse objeto é `DisableLogTypes`, e seu valor é uma matriz de strings com qualquer combinação de `alert`, `audit`, `listener` e `trace`.

Para Linux, macOS ou Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
```

```
    --cloudwatch-logs-export-configuration '{"DisableLogTypes":["audit","listener"]}'
```

Para Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --cloudwatch-logs-export-configuration DisableLogTypes=\"audit\",\"listener\"
```

API do RDS

Você pode publicar logs de banco de dados Oracle com a API do RDS. Você pode chamar a ação `ModifyDBInstance` com os seguintes parâmetros:

- `DBInstanceIdentifier`
- `CloudwatchLogsExportConfiguration`

> **Note**
>
> Uma alteração feita no parâmetro `CloudwatchLogsExportConfiguration` sempre é aplicada imediatamente na instância de banco de dados. Por isso, o parâmetro `ApplyImmediately` não entra em vigor.

Também publique logs Oracle chamando as seguintes operações da API do RDS:

- `CreateDBInstance`
- `RestoreDBInstanceFromDBSnapshot`
- `RestoreDBInstanceFromS3`
- `RestoreDBInstanceToPointInTime`

Execute uma destas operações da API do RDS com os seguintes parâmetros:

- `DBInstanceIdentifier`
- `EnableCloudwatchLogsExports`
- `Engine`
- `DBInstanceClass`

Outros parâmetros podem ser obrigatórios, dependendo da operação do RDS executada.

Acesso aos logs de alerta e logs de receptor

Você pode visualizar o log de alertas usando o console do Amazon RDS. Também é possível usar a instrução SQL a seguir.

```
SELECT message_text FROM alertlog;
```

Acesse o log do receptor usando o Amazon CloudWatch Logs.

> **Note**
>
> A Oracle faz a rotação dos logs de alerta e de ouvinte quando ultrapassam 10 MB e, a partir deste ponto, eles ficam indisponíveis nas visualizações do Amazon RDS.

Arquivos de log do banco de dados do RDS para PostgreSQL

Você pode monitorar os seguintes tipos de arquivo de log do :

- Log do PostgreSQL
- Log de atualização
- Log de erros de autenticação de banco de dados do IAM

> **Note**
>
> Para habilitar os logs de erro de autenticação de banco de dados do IAM, você deve primeiro habilitar a autenticação de banco de dados do IAM para a instância de banco de dados do RDS para PostgreSQL. Para ter mais informações sobre a autenticação de banco de dados do IAM, consulte Habilitar e desabilitar a autenticação de banco de dados do IAM.

O RDS para PostgreSQL registra as atividades do banco de dados no arquivo de log padrão do PostgreSQL. Para uma instância de banco de dados PostgreSQL on-premises, essas mensagens são armazenadas localmente em `log/postgresql.log`. Para uma instância de banco de dados do RDS para PostgreSQL, o arquivo de log está disponível na instância do Amazon RDS. Esses logs também podem ser acessados por meio do AWS Management Console, onde você pode visualizá-los ou baixá-los. O nível de registro em log padrão captura falhas de login, erros fatais do servidor, deadlocks e falhas de consulta.

Para ter mais informações sobre a visualização, o download e os logs de banco de dados baseados no monitoramento de arquivos, consulte Monitorar arquivos de log do Amazon RDS. Para saber mais sobre logs do PostgreSQL, consulte Working with Amazon RDS and Aurora PostgreSQL logs: Part 1 (Trabalhar com o Amazon RDS e logs do Aurora PostgreSQL: parte 1) e Working with RDS and Aurora PostgreSQL logs: Part 2 (Trabalhar com o Amazon RDS e logs do Aurora PostgreSQL: parte 2).

Além dos logs padrão do PostgreSQL abordados neste tópico, o RDS para PostgreSQL também é compatível com a extensão Audit do PostgreSQL (`pgAudit`). A maioria dos setores regulamentados e agências governamentais precisa manter um log de auditoria ou uma trilha de auditoria das alterações feitas nos dados para cumprir os requisitos legais. Para obter informações sobre a instalação e o uso de pgAudit, consulte Usar pgAudit para registrar a atividade do banco de dados.

Tópicos

- Parâmetros para registro em log no RDS para PostgreSQL
- Ativar o registro em log de consultas para sua instância de banco de dados do RDS para PostgreSQL
- Publicação de logs do PostgreSQL no Amazon CloudWatch Logs

Parâmetros para registro em log no RDS para PostgreSQL

Você pode personalizar o comportamento do registro em log da instância de banco de dados do RDS para PostgreSQL modificando vários parâmetros. Na tabela a seguir, você encontra os parâmetros que afetam por quanto tempo os logs são armazenados, quando alternar o log e se ele deve ser gerado no formato CSV (valor separado por vírgula). Você também pode encontrar a saída de texto enviada para STDERR, entre outras configurações. Para alterar as configurações dos parâmetros que podem ser modificados, use um grupo de parâmetros de banco de dados para o Instância do RDS para PostgreSQL. Para ter mais informações, consulte Grupos de parâmetros de banco de dados para instâncias de banco de dados do Amazon RDS.

Parameter	Padrão	Descrição
log_destination	stderr	Define o formato de saída para o log. O padrão é `stderr`, mas você também pode especificar valores separados por vírgula (CSV) adicionan do `csvlog` à configuração. Para obter mais informações, consulte Definir o destino dos logs (`stderr`, `csvlog`).
log_filename	postgresql.log.%Y-%m-%d-%H%M	Especifica o padrão para o nome do arquivo de log. Além do padrão, esse parâmetro é compatível com `postgresql.log.%Y-%m-%d` e `postgresql.log.%Y-%m-%d-%H%M` para o padrão de nome de arquivo.
log_line_prefix	%t:%r:%u@%d:[%p]:	Define o prefixo para cada linha de log que é gravada em `stderr`, para anotar a hora (%t), o host remoto (%r), o usuário (%u), o banco de dados (%d) e o ID do processo (%p).

Parameter	Padrão	Descrição
log_rotation_age	60	Minutos após os quais o arquivo de log é alternado automaticamente. É possível alterar esse valor no intervalo de 1 a e 1.440 minutos. Para obter mais informações, consulte Configurar a alternância do arquivo de log.
log_rotation_size	–	O tamanho (kB) no qual o log é alternado automaticamente. Por padrão, esse parâmetro não é usado porque os logs são trocados com base no parâmetro `log_rotation_age` . Para saber mais, consulte Configurar a alternância do arquivo de log.
rds.log_retention_period	4320	Os logs do PostgreSQL mais antigos que o número especificado de minutos são excluídos . O valor padrão de 4.320 minutos exclui os arquivos de log após três dias. Para ter mais informações, consulte Definir o período de retenção de log.

Para identificar problemas de aplicações, você pode procurar falhas de consulta, falhas de login, deadlocks e erros fatais de servidor no log. Por exemplo, suponha que você converta uma aplicação herdada do Oracle no Amazon RDS PostgreSQL, mas nem todas as consultas foram convertidas corretamente. Essas consultas formatadas incorretamente geram mensagens de erro nos logs, que você pode usar para identificar problemas. Para ter mais informações sobre o registro em log de consultas, consulte Ativar o registro em log de consultas para sua instância de banco de dados do RDS para PostgreSQL.

Nos tópicos a seguir, você encontrará informações sobre como definir vários parâmetros que controlam os detalhes básicos de seus logs do PostgreSQL.

Tópicos

- Definir o período de retenção de log
- Configurar a alternância do arquivo de log
- Definir o destino dos logs (stderr, csvlog)

- Noções básicas sobre o parâmetro log_line_prefix

Definir o período de retenção de log

O parâmetro `rds.log_retention_period` especifica por quanto tempo sua instância de banco de dados do Aurora PostgreSQL mantém seus arquivos de log. A configuração padrão é de três dias (4.320 minutos), mas você pode definir esse valor de um dia (1.440 minutos) a sete dias (10.080 minutos). Seu cluster de banco de dados do Aurora PostgreSQLSua instância de banco de dados do RDS para PostgreSQL deve ter armazenamento suficiente para armazenar os arquivos de log durante o período.

Recomendamos que você publique habitualmente seus logs no Amazon CloudWatch Logs para que possa visualizar e analisar os dados do sistema muito tempo depois que os logs tiverem sido removidos do Instância de banco de dados do RDS para PostgreSQL. Para ter mais informações, consulte Publicação de logs do PostgreSQL no Amazon CloudWatch Logs.

Configurar a alternância do arquivo de log

Por padrão, o Amazon RDS cria arquivos de log a cada hora. O tempo é controlado pelo parâmetro `log_rotation_age`. Esse parâmetro tem um valor padrão de 60 (minutos), mas você pode definir qualquer valor entre 1 minuto e 24 horas (1.440 minutos). Quando chegar o momento da alternância, será criado um novo arquivo de log distinto. O arquivo é nomeado de acordo com o padrão especificado pelo parâmetro `log_filename`.

Também é possível alternar os arquivos de log de acordo com o tamanho, conforme especificado no parâmetro `log_rotation_size`. Esse parâmetro especifica que o log deva ser alternado quando atingir o tamanho determinado (em kilobytes). Para uma instância de banco de dados do RDS para PostgreSQL, `log_rotation_size` não é definido, ou seja, não há valor especificado. No entanto, é possível definir o parâmetro de 0 a 2.097.151 kB (kilobytes).

Os nomes dos arquivos de log são baseados no padrão de nome do arquivo do parâmetro `log_filename`. As configurações disponíveis para esse parâmetro são as seguintes:

- `postgresql.log.%Y-%m-%d`: formato padrão do nome do arquivo de log. Inclui o ano, o mês e a data no nome do arquivo de log.
- `postgresql.log.%Y-%m-%d-%H`: inclui a hora no formato do nome do arquivo de log.

Para ter mais informações, consulte `log_rotation_age` e `log_rotation_size` na documentação do PostgreSQL.

Definir o destino dos logs (**`stderr`**, **`csvlog`**)

Por padrão, o Amazon RDS PostgreSQL gera logs no formato de erro padrão (stderr). Esse formato é a configuração padrão do parâmetro `log_destination`. Cada mensagem é prefixada usando o padrão especificado no parâmetro `log_line_prefix`. Para ter mais informações, consulte Noções básicas sobre o parâmetro log_line_prefix.

O RDS para PostgreSQL também pode gerar os registros no formato `csvlog`. O `csvlog` é útil para analisar os dados de log como dados de valores separados por vírgula (CSV). Por exemplo, digamos que você use a extensão `log_fdw` para trabalhar com seus logs como tabelas externas. A tabela externa criada nos arquivos de log do `stderr` contém uma única coluna com dados de eventos de log. Ao adicionar `csvlog` ao parâmetro `log_destination`, você obtém o arquivo de log no formato CSV com demarcações para as várias colunas da tabela externa. Agora você pode classificar e analisar os logs com maior facilidade. Para saber como usar o `log_fdw` com o `csvlog`, consulte Usar a extensão log_fdw para acessar o log de banco de dados usando SQL.

Se você especificar `csvlog` para esse parâmetro, lembre-se de que os arquivos `stderr` e `csvlog` são gerados. Monitore o armazenamento consumido pelos logs, levando em consideração o `rds.log_retention_period` e outras configurações que afetam o armazenamento e a rotatividade dos logs. O uso de `stderr` e `csvlog` mais do que dobra o armazenamento consumido pelos logs.

Se você adicionar `csvlog` a `log_destination` e quiser reverter para o `stderr`, precisará redefinir o parâmetro. Para fazer isso, use o console do Amazon RDS e, depois, abra o grupo de parâmetros do banco de dados para sua instância. Selecione o parâmetro `log_destination`, Edit parameter (Editar parâmetro) e depois Reset (Redefinir).

Para ter mais informações sobre como configurar o registro em log, consulte Trabalhar com logs do Amazon RDS e do Aurora PostgreSQL: parte 1.

Noções básicas sobre o parâmetro log_line_prefix

O formato de log `stderr` prefixa cada mensagem de log com os detalhes especificados pelo parâmetro `log_line_prefix`. O valor padrão é:

```
%t:%r:%u@%d:[%p]:t
```

A partir do Aurora PostgreSQL versão 16, você também pode escolher:

```
%m:%r:%u@%d:[%p]:%l:%e:%s:%v:%x:%c:%q%a
```

Cada entrada de log enviada para stderr inclui as seguintes informações com base no valor selecionado:

- `%t`: hora da entrada do log sem milissegundos.
- `%m`: hora da entrada do log com milissegundos.
- `%r`: endereço do host remoto.
- `%u@%d`: nome de usuário no nome do banco de dados.
- `[%p]`: ID do processo, se disponível.
- `%l`: número da linha de log por sessão.
- `%e`: código de erro SQL.
- `%s`: data e hora de início do processo.
- `%v`: ID da transação virtual.
- `%x`: ID da transação.
- `%c`: ID da sessão.
- `%q`: terminador não relacionado à sessão.
- `%a`: nome da aplicação.

Ativar o registro em log de consultas para sua instância de banco de dados do RDS para PostgreSQL

Você pode coletar informações mais detalhadas sobre suas atividades de banco de dados, inclusive consultas, consultas à espera de bloqueios, pontos de verificação e muitos outros detalhes definindo alguns dos parâmetros listados na tabela a seguir. Este tópico se concentra no registro em log de consultas.

Parameter	Padrão	Descrição
log_connections	–	Registra cada conexão bem-sucedida.
log_disconnections	–	Registra o final de cada sessão e sua duração.
log_checkpoints	1	Registra em log cada ponto de verificação.
log_lock_waits	–	Registra esperas de bloqueio longas. Por padrão, esse parâmetro não está definido.

Parameter	Padrão	Descrição
log_min_duration_sample	–	(ms) Define o tempo de execução mínimo acima do qual uma amostra de declaraçõ es será registrada. O tamanho da amostra é definido usando o parâmetro `log_statement_sample_rate` .
log_min_duration_statement	–	Todas as instruções SQL executadas pelo menos por um período especificado ou mais é registrada. Por padrão, esse parâmetro não está definido. Ativar esse parâmetro pode ajudar a encontrar consultas não otimizadas.
log_statement	–	Define o tipo de instruções registradas. Por padrão, esse parâmetro não está definido, mas você pode alterá-lo para `all`, `ddl` ou `mod` para especificar os tipos de declaração SQL que você deseja registrar. Se você especificar algo diferente de `none` para esse parâmetro, você também deve tomar medidas adicionais para evitar a exposição de senhas nos arquivos de log. Para ter mais informações, consulte Reduzir o risco de exposição de senhas ao usar o registro em log de consultas.
log_statement_sample_rate	–	A porcentagem de declarações que excedem o tempo especificado em `log_min_duration_sample` para serem registradas, expressa como um valor de ponto flutuante entre 0,0 e 1,0.
log_statement_stats	–	Grava estatísticas de desempenho cumulativas no log do servidor.

Usar o registro em log para encontrar consultas de baixa performance

Você pode registrar consultas e declarações SQL para ajudar a encontrar consultas com a performance lenta. Você ativa esse recurso modificando as configurações dos parâmetros `log_statement` e `log_min_duration` conforme descrito nesta seção. Antes de ativar o registro em log de consultas para sua instância de banco de dados do RDS para PostgreSQL, você deve estar ciente da possível exposição de senhas nos logs e de como reduzir os riscos. Para ter mais informações, consulte Reduzir o risco de exposição de senhas ao usar o registro em log de consultas.

A seguir, você encontrará informações de referência sobre os parâmetros `log_statement` e `log_min_duration`.

log_statement

Esse parâmetro especifica o tipo de declarações SQL que devem ser enviadas ao log. O valor padrão é `none`. Se você alterar esse parâmetro para `all`, `ddl` ou `mod`, realize algumas das ações recomendadas para reduzir o risco de expor senhas nos logs. Para ter mais informações, consulte Reduzir o risco de exposição de senhas ao usar o registro em log de consultas.

tudo

Registra todas as declarações. Essa configuração é recomendada para fins de depuração.

ddl

Registra todas as declarações de linguagem de definição de dados (DDL), como CREATE, ALTER, DROP etc.

mod

Registra todas as declarações DDL e declarações de linguagem de manipulação de dados (INSERT, UPDATE e DELETE) que modificam os dados.

nenhuma

Nenhuma declaração SQL é registrada. Recomendamos essa configuração para evitar o risco de expor senhas nos logs.

log_min_duration_statement

Todas as instruções SQL executadas pelo menos por um período especificado ou mais é registrada. Por padrão, esse parâmetro não está definido. Ativar esse parâmetro pode ajudar a encontrar consultas não otimizadas.

–1–2147483647

O número de milissegundos (ms) de tempo de execução durante o qual uma declaração é registrada.

Como configurar o registro em log de consultas

Essas etapas pressupõem que o A instância de banco de dados do RDS para PostgreSQL usa um grupo de parâmetros de banco de dados personalizado.

1. Defina o parâmetro `log_statement` como `all`. O exemplo a seguir mostra a informação gravada no arquivo `postgresql.log`com essa configuração de parâmetro.

```
2022-10-05 22:05:52 UTC:52.95.4.1(11335):postgres@labdb:[3639]:LOG: statement:
 SELECT feedback, s.sentiment,s.confidence
FROM support,aws_comprehend.detect_sentiment(feedback, 'en') s
ORDER BY s.confidence DESC;
2022-10-05 22:05:52 UTC:52.95.4.1(11335):postgres@labdb:[3639]:LOG: QUERY
 STATISTICS
2022-10-05 22:05:52 UTC:52.95.4.1(11335):postgres@labdb:[3639]:DETAIL: ! system
 usage stats:
! 0.017355 s user, 0.000000 s system, 0.168593 s elapsed
! [0.025146 s user, 0.000000 s system total]
! 36644 kB max resident size
! 0/8 [0/8] filesystem blocks in/out
! 0/733 [0/1364] page faults/reclaims, 0 [0] swaps
! 0 [0] signals rcvd, 0/0 [0/0] messages rcvd/sent
! 19/0 [27/0] voluntary/involuntary context switches
2022-10-05 22:05:52 UTC:52.95.4.1(11335):postgres@labdb:[3639]:STATEMENT: SELECT
 feedback, s.sentiment,s.confidence
FROM support,aws_comprehend.detect_sentiment(feedback, 'en') s
ORDER BY s.confidence DESC;
2022-10-05 22:05:56 UTC:52.95.4.1(11335):postgres@labdb:[3639]:ERROR: syntax error
 at or near "ORDER" at character 1
2022-10-05 22:05:56 UTC:52.95.4.1(11335):postgres@labdb:[3639]:STATEMENT: ORDER BY
 s.confidence DESC;
```

```
---------------------- END OF LOG ----------------------
```

2. Defina o parâmetro `log_min_duration_statement`. O exemplo a seguir mostra a informação gravada no arquivo `postgresql.log` quando o parâmetro estiver definido como `1`.

   As consultas que excedem a duração especificada no parâmetro `log_min_duration_statement` são registradas. Por exemplo: Você pode visualizar o arquivo de log de sua instância de banco de dados do RDS para PostgreSQL no console do Amazon RDS.

```
2022-10-05 19:05:19 UTC:52.95.4.1(6461):postgres@labdb:[6144]:LOG: statement: DROP
 table comments;
2022-10-05 19:05:19 UTC:52.95.4.1(6461):postgres@labdb:[6144]:LOG: duration:
 167.754 ms
2022-10-05 19:08:07 UTC::@:[355]:LOG: checkpoint starting: time
2022-10-05 19:08:08 UTC::@:[355]:LOG: checkpoint complete: wrote 11 buffers
 (0.0%); 0 WAL file(s) added, 0 removed, 0 recycled; write=1.013 s, sync=0.006 s,
 total=1.033 s; sync files=8, longest=0.004 s, average=0.001 s; distance=131028 kB,
 estimate=131028 kB
---------------------- END OF LOG ----------------------
```

Reduzir o risco de exposição de senhas ao usar o registro em log de consultas

Recomendamos manter `log_statement` definido como `none` para evitar a exposição de senhas. Se você definir `log_statement` como `all`, `ddl` oumod, recomendamos que você siga uma ou mais destas etapas.

- Para o cliente, criptografe informações confidenciais. Para ter mais informações consulte Encryption Options (Opções de criptografia) na documentação do PostgreSQL. Use as opções `ENCRYPTED` (e `UNENCRYPTED`) das declarações `CREATE` e `ALTER`. Para ter mais informações, consulte CREATE USER na documentação do PostgreSQL.
- Para instância de banco de dados do RDS para PostgreSQL, configure e use a extensão de auditoria do PostgreSQL (pgAudit). Essa extensão remove informações confidenciais das declarações CREATE e ALTER enviadas ao log. Para ter mais informações, consulte Usar pgAudit para registrar a atividade do banco de dados.
- Restringir o acesso aos logs CloudWatch.
- Use mecanismos de autenticação mais fortes, como IAM.

Publicação de logs do PostgreSQL no Amazon CloudWatch Logs

Para armazenar os registros de log do PostgreSQL em armazenamento resiliente, você pode usar o Amazon CloudWatch Logs. Com o CloudWatch Logs, é possível realizar análise em tempo real de dados e usar o CloudWatch para visualizar métricas e criar alarmes. Por exemplo, se você definir `log_statement` como `ddl`, poderá configurar um alarme para alertar sempre que uma declaração DDL for executada. Você pode optar por fazer upload dos logs do PostgreSQL no CloudWatch Logs durante o processo de criação de sua instância de banco de dados do RDS para PostgreSQL. Se você optar por não fazer upload de logs no momento, poderá modificar sua instância posteriormente para começar a fazer upload dos logs a partir desse momento. Em outras palavras, não é feito upload dos logs existentes. Só é feito upload de novos logs, pois eles são criados em sua instância de banco de dados do RDS para PostgreSQL modificada.

Todas as versões do RDS para PostgreSQL atualmente disponíveis são compatíveis com a publicação de arquivos de log no CloudWatch Logs. Para obter informações, consulte Amazon RDS for PostgreSQL updates (Atualizações do Amazon RDS para PostgreSQL) nas Amazon RDS for PostgreSQL Release Notes (Notas de versão do Amazon RDS para PostgreSQL).

Para trabalhar com o CloudWatch Logs, configure o RDS para a instância de banco de dados do PostgreSQL para publicar dados de log a um grupo de log.

Você pode publicar os seguintes tipos de log no CloudWatch Logs do RDS do PostgreSQL:

- Log do PostgreSQL
- Log de atualização
- Log de erros de autenticação de banco de dados do IAM

Depois que você concluir a configuração, o Amazon RDS publicará os eventos de log em fluxos de logs dentro de um grupo de logs do CloudWatch. Por exemplo, os dados de log do PostgreSQL são armazenados no grupo de logs `/aws/rds/instance/`*`my_instance`*`/postgresql`. Para visualizar os logs, abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/.

Console

Para publicar logs do PostgreSQL no CloudWatch Logs usando o console

1. Abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Databases (Bancos de dados).

3. Escolha a instância de banco de dados que você deseja modificar e escolha Modify (Modificar).
4. Na seção Log exports (Exportações de log), escolha os logs que deseja começar a publicar no CloudWatch Logs.

   A seção Log exports (Exportações de log) só está disponível para versões do PostgreSQL compatível com a publicação no CloudWatch Logs.
5. Escolha Continue (Continuar) e depois escolha Modify DB Instance (Modificar instância de banco de dados) na página de resumo.

AWS CLI

Publique logs do PostgreSQL com a AWS CLI. Você pode chamar o comando `modify-db-instance` com os parâmetros a seguir.

- `--db-instance-identifier`
- `--cloudwatch-logs-export-configuration`

> **Note**
>
> Uma alteração feita na opção `--cloudwatch-logs-export-configuration` sempre é aplicada imediatamente na instância de banco de dados. Por isso, as opções `--apply-immediately` e `--no-apply-immediately` não entram em vigor.

Também publique logs do PostgreSQL chamando os seguintes comandos de CLI:

- `create-db-instance`
- `restore-db-instance-from-db-snapshot`
- `restore-db-instance-to-point-in-time`

Execute um destes comandos da CLI com as seguintes opções:

- `--db-instance-identifier`
- `--enable-cloudwatch-logs-exports`
- `--db-instance-class`
- `--engine`

Outras opções podem ser obrigatórias, dependendo do comando da CLI executado.

Example Modificar uma instância para publicar logs no CloudWatch Logs

O exemplo a seguir modifica uma instância de banco de dados PostgreSQL existente para publicar arquivos de log em CloudWatch Logs. O valor `--cloudwatch-logs-export-configuration` é um objeto JSON. A chave desse objeto é `EnableLogTypes`, e o valor é uma matriz de strings com uma combinação qualquer de `postgresql` e `upgrade`.

Para Linux, macOS ou Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --cloudwatch-logs-export-configuration '{"EnableLogTypes":["postgresql",
 "upgrade"]}'
```

Para Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --cloudwatch-logs-export-configuration '{"EnableLogTypes":
["postgresql","upgrade"]}'
```

Example Criar uma instância para publicar logs no CloudWatch Logs

O exemplo a seguir cria uma instância de banco de dados PostgreSQL e publica arquivos de log em CloudWatch Logs. O valor `--enable-cloudwatch-logs-exports` é uma matriz de strings JSON. As strings podem ser uma combinação qualquer de `postgresql` e `upgrade`.

Para Linux, macOS ou Unix:

```
aws rds create-db-instance \
    --db-instance-identifier mydbinstance \
    --enable-cloudwatch-logs-exports '["postgresql","upgrade"]' \
    --db-instance-class db.m4.large \
    --engine postgres
```

Para Windows:

```
aws rds create-db-instance ^
    --db-instance-identifier mydbinstance ^
    --enable-cloudwatch-logs-exports '["postgresql","upgrade"]' ^
    --db-instance-class db.m4.large ^
    --engine postgres
```

API do RDS

Publique logs do PostgreSQL com a API do RDS. Você pode chamar a ação `ModifyDBInstance` com os seguintes parâmetros:

- `DBInstanceIdentifier`
- `CloudwatchLogsExportConfiguration`

> **Note**
>
> Uma alteração feita no parâmetro `CloudwatchLogsExportConfiguration` sempre é aplicada imediatamente na instância de banco de dados. Por isso, o parâmetro `ApplyImmediately` não entra em vigor.

Também é possível publicar logs do PostgreSQL chamando as seguintes operações da API do RDS:

- `CreateDBInstance`
- `RestoreDBInstanceFromDBSnapshot`
- `RestoreDBInstanceToPointInTime`

Execute uma destas operações da API do RDS com os seguintes parâmetros:

- `DBInstanceIdentifier`
- `EnableCloudwatchLogsExports`
- `Engine`
- `DBInstanceClass`

Outros parâmetros podem ser obrigatórios, dependendo da operação executada.

Monitorar chamadas de API do Amazon RDSno AWS CloudTrail

O AWS CloudTrail é um serviço da AWS que ajuda a auditar a conta da AWS. O AWS CloudTrail é ativado na sua conta da AWS quando ela é criada. Para obter mais informações sobre o CloudTrail, consulte o Guia do usuário do AWS CloudTrail.

Tópicos

- Integração do CloudTrail com o Amazon RDS
- Entradas do arquivo de log do Amazon RDS

Integração do CloudTrail com o Amazon RDS

Todas as ações do Amazon RDS são registradas em log pelo CloudTrail. O CloudTrail fornece um registro de ações executadas por um usuário, uma função ou um serviço da AWS no Amazon RDS.

Eventos do CloudTrail

O CloudTrail captura as chamadas de API do Amazon RDS como eventos. Um evento representa uma única solicitação de qualquer origem e inclui informações sobre a ação solicitada, a data e a hora da ação, os parâmetros de solicitação e assim por diante. Os eventos incluem as chamadas do console do Amazon RDS e as chamadas de código para as operações de API do Amazon RDS.

A atividade do Amazon RDS é registrada em um evento do CloudTrail em Event history (Histórico de eventos). Você pode usar o console do CloudTrail para visualizar os últimos 90 dias de eventos e de atividades de API registrados em uma região da AWS. Para obter mais informações, consulte Visualizar eventos com o histórico de eventos do CloudTrail.

Trilhas do CloudTrail

Para obter um registro contínuo de eventos na conta da AWS, incluindo eventos do Amazon RDS, crie uma trilha. Uma trilha é uma configuração que permite a entrega de eventos a um bucket do Amazon S3 especificado. O CloudTrail normalmente fornece os arquivos de log em até 15 minutos após uma atividade da conta.

> **Note**
>
> Se você não configurar uma trilha, ainda poderá visualizar os eventos mais recentes no console do CloudTrail no Histórico de eventos.

É possível criar dois tipos de trilhas para uma conta da AWS: uma trilha que se aplica a todas as regiões ou uma trilha que se aplica a uma região. Por padrão, quando uma trilha é criada no console, a mesma é aplicada a todas as regiões da .

Além disso, é possível configurar outros serviços da AWS para analisar e agir melhor com base nos dados de eventos coletados nos logs do CloudTrail. Para obter mais informações, consulte:

- Visão geral da criação de uma trilha
- Serviços e integrações compatíveis com o CloudTrail
- Configurar notificações do Amazon SNS para o CloudTrail
- Receber arquivos de log do CloudTrail de várias regiões e Receber arquivos de log do CloudTrail de várias contas

Entradas do arquivo de log do Amazon RDS

Os arquivos de log do CloudTrail contêm uma ou mais entradas de log. Os arquivos de log do CloudTrail não são um rastreamento de pilha ordenada das chamadas de API pública. Dessa forma, eles não são exibidos em uma ordem específica.

O exemplo a seguir mostra uma entrada de log do CloudTrail que demonstra a ação `CreateDBInstance`.


```
{
    "eventVersion": "1.04",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "AKIAIOSFODNN7EXAMPLE",
        "arn": "arn:aws:iam::123456789012:user/johndoe",
        "accountId": "123456789012",
        "accessKeyId": "AKIAI44QH8DHBEXAMPLE",
        "userName": "johndoe"
    },
    "eventTime": "2018-07-30T22:14:06Z",
    "eventSource": "rds.amazonaws.com",
    "eventName": "CreateDBInstance",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "aws-cli/1.15.42 Python/3.6.1 Darwin/17.7.0 botocore/1.10.42",
    "requestParameters": {
```

```
        "enableCloudwatchLogsExports": [
            "audit",
            "error",
            "general",
            "slowquery"
        ],
        "dBInstanceIdentifier": "test-instance",
        "engine": "mysql",
        "masterUsername": "myawsuser",
        "allocatedStorage": 20,
        "dBInstanceClass": "db.m1.small",
        "masterUserPassword": "****"
    },
    "responseElements": {
        "dBInstanceArn": "arn:aws:rds:us-east-1:123456789012:db:test-instance",
        "storageEncrypted": false,
        "preferredBackupWindow": "10:27-10:57",
        "preferredMaintenanceWindow": "sat:05:47-sat:06:17",
        "backupRetentionPeriod": 1,
        "allocatedStorage": 20,
        "storageType": "standard",
        "engineVersion": "8.0.28",
        "dbInstancePort": 0,
        "optionGroupMemberships": [
            {
                "status": "in-sync",
                "optionGroupName": "default:mysql-8-0"
            }
        ],
        "dBParameterGroups": [
            {
                "dBParameterGroupName": "default.mysql8.0",
                "parameterApplyStatus": "in-sync"
            }
        ],
        "monitoringInterval": 0,
        "dBInstanceClass": "db.m1.small",
        "readReplicaDBInstanceIdentifiers": [],
        "dBSubnetGroup": {
            "dBSubnetGroupName": "default",
            "dBSubnetGroupDescription": "default",
            "subnets": [
                {
                    "subnetAvailabilityZone": {"name": "us-east-1b"},
```

```
                "subnetIdentifier": "subnet-cbfff283",
                "subnetStatus": "Active"
            },
            {
                "subnetAvailabilityZone": {"name": "us-east-1e"},
                "subnetIdentifier": "subnet-d7c825e8",
                "subnetStatus": "Active"
            },
            {
                "subnetAvailabilityZone": {"name": "us-east-1f"},
                "subnetIdentifier": "subnet-6746046b",
                "subnetStatus": "Active"
            },
            {
                "subnetAvailabilityZone": {"name": "us-east-1c"},
                "subnetIdentifier": "subnet-bac383e0",
                "subnetStatus": "Active"
            },
            {
                "subnetAvailabilityZone": {"name": "us-east-1d"},
                "subnetIdentifier": "subnet-42599426",
                "subnetStatus": "Active"
            },
            {
                "subnetAvailabilityZone": {"name": "us-east-1a"},
                "subnetIdentifier": "subnet-da327bf6",
                "subnetStatus": "Active"
            }
        ],
        "vpcId": "vpc-136a4c6a",
        "subnetGroupStatus": "Complete"
    },
    "masterUsername": "myawsuser",
    "multiAZ": false,
    "autoMinorVersionUpgrade": true,
    "engine": "mysql",
    "cACertificateIdentifier": "rds-ca-2015",
    "dbiResourceId": "db-ETDZIIXHEWY5N7GXVC4SH7H5IA",
    "dBSecurityGroups": [],
    "pendingModifiedValues": {
        "masterUserPassword": "****",
        "pendingCloudwatchLogsExports": {
            "logTypesToEnable": [
                "audit",
```

```
                    "error",
                    "general",
                    "slowquery"
                ]
            }
        },
        "dBInstanceStatus": "creating",
        "publiclyAccessible": true,
        "domainMemberships": [],
        "copyTagsToSnapshot": false,
        "dBInstanceIdentifier": "test-instance",
        "licenseModel": "general-public-license",
        "iAMDatabaseAuthenticationEnabled": false,
        "performanceInsightsEnabled": false,
        "vpcSecurityGroups": [
            {
                "status": "active",
                "vpcSecurityGroupId": "sg-f839b688"
            }
        ]
    },
    "requestID": "daf2e3f5-96a3-4df7-a026-863f96db793e",
    "eventID": "797163d3-5726-441d-80a7-6eeb7464acd4",
    "eventType": "AwsApiCall",
    "recipientAccountId": "123456789012"
}
```

Conforme mostrado no elemento `userIdentity` no exemplo anterior, cada evento ou entrada de log contém informações sobre quem gerou a solicitação. As informações de identidade ajudam a determinar:

- Se a solicitação foi feita com credenciais de usuário-raiz ou usuário do IAM.
- Se a solicitação foi feita com credenciais de segurança temporárias de uma função ou de um usuário federado.
- Se a solicitação foi feita por outro serviço da AWS.

Para obter mais informações sobre o `userIdentity`, consulte o Elemento userIdentity do CloudTrail. Para obter mais informações sobre `CreateDBInstance` e outras ações do Amazon RDS, consulte a Referência da API do Amazon RDS.

Monitorar o Amazon RDS com o recurso Database Activity Streams

Usando o recurso Database Activity Streams, você pode monitorar fluxos quase em tempo real da atividade de um banco de dados.

Tópicos

- Visão geral dos Database Activity Streams
- Configurar a auditoria unificada do Oracle Database
- Configurar uma política de auditoria para o Amazon RDS para Microsoft SQL Server
- Iniciar um stream de atividade de banco de dados
- Modificar um fluxo de atividades de banco de dados para o Amazon RDS
- Obter o status de um fluxo de atividade de banco de dados
- Interromper um fluxo de atividade de banco de dados
- Monitorar fluxos de atividades de banco de dados
- Exemplos de políticas do IAM para fluxos de atividades de banco de dados

Visão geral dos Database Activity Streams

Como administrador de um banco de dados do Amazon RDS, você precisa proteger o banco de dados e atender aos requisitos regulatórios e de compatibilidade. Uma estratégia é integrar fluxos de atividades de banco de dados às suas ferramentas de monitoramento. Dessa forma, você monitora e define alarmes para realizar a auditoria da atividade no banco de dados.

As ameaças à segurança são externas e internas. Para se proteger contra ameaças internas, você pode controlar o acesso do administrador aos transmissões de dados configurando o recurso Database Activity Streams. Os administradores de bancos de dados do Amazon RDS não têm acesso à coleta, transmissão, armazenamento e processamento das transmissões.

Sumário

- Como os fluxos de atividade do banco de dados funcionam
- Auditoria no Oracle Database e no banco de dados Microsoft SQL
  - Auditoria unificada no Oracle Database
  - Auditoria no Microsoft SQL Server
  - Campos de auditoria não nativos para Oracle Database e SQL Server

  - Substituição do grupo de parâmetros de banco de dados
- Modo assíncrono para transmissões de atividades de banco de dados
- Requisitos e limitações para fluxos de atividade de banco de dados
- Disponibilidade de regiões e versões
- Classes de instância de banco de dados compatíveis para transmissões de atividades de banco de dados

Como os fluxos de atividade do banco de dados funcionam

O Amazon RDS envia atividades para um fluxo de dados do Amazon Kinesis quase em tempo real. O fluxo do Kinesis é criado automaticamente. No Kinesis, é possível configurar serviços da AWS, como o Amazon Data Firehose e o AWS Lambda, para consumir o fluxo e armazenar os dados.

> **Important**
>
> O uso do recurso Database Activity Streams é gratuito no Amazon RDS, mas o Amazon Kinesis cobra pelo fluxo de dados. Para ter mais informações, consulte Definição de preço do Amazon Kinesis Data Streams.

É possível configurar aplicações para gerenciamento de conformidade com o objetivo de consumir fluxos de atividades do banco de dados. Essas aplicações podem usar o fluxo para gerar alertas e auditar atividades em seu banco de dados.

O Amazon RDS é compatível com fluxos de atividade de banco de dados em implantações multi-AZ. Nesse caso, os fluxos de atividades do banco de dados auditam as instâncias principal e em espera.

Auditoria no Oracle Database e no banco de dados Microsoft SQL

Auditoria é o monitoramento e registro de ações de banco de dados configuradas. O Amazon RDS não captura a atividade de um banco de dados por padrão. Você mesmo cria e gerencia políticas de auditoria no banco de dados.

Tópicos

- Auditoria unificada no Oracle Database
- Auditoria no Microsoft SQL Server

- Campos de auditoria não nativos para Oracle Database e SQL Server
- Substituição do grupo de parâmetros de banco de dados

Auditoria unificada no Oracle Database

Em um banco de dados Oracle, uma política de auditoria unificada é um grupo chamado de configurações de auditoria que você pode usar para auditar um aspecto do comportamento do usuário. Uma política pode ser tão simples quanto auditar as atividades de um único usuário. Também é possível criar políticas de auditoria complexas que usam condições.

Um banco de dados Oracle grava registros de auditoria, inclusive registros de auditoria do `SYS`, para a trilha de auditoria unificada. Por exemplo, se ocorrer um erro durante uma instrução `INSERT`, a auditoria padrão indicará o número do erro e o SQL que foi executado. A trilha de auditoria reside em uma tabela somente leitura no esquema do `AUDSYS`. Para acessar esses registros,consulte a visualização do dicionário de dados `UNIFIED_AUDIT_TRAIL`.

Normalmente, configure fluxos de atividade do banco de dados da seguinte maneira:

1. Crie uma política de auditoria do Oracle Database usando o comando `CREATE AUDIT POLICY`.

   O Oracle Database gera registros de auditoria.
2. Ative a política de auditoria usando o comando `AUDIT POLICY`.
3. Configurar fluxos de atividade do banco de dados.

   Somente as atividades que correspondem às políticas de auditoria do Oracle Database são capturadas e enviadas ao fluxo de dados do Amazon Kinesis. Quando fluxos de atividades de banco de dados são habilitados, um administrador de banco de dados Oracle não pode alterar a política de auditoria nem remover registros de auditoria.

Para saber mais sobre as políticas de auditoria unificadas, consulte Sobre atividades de auditoria com políticas de auditoria unificadas e AUDIT no Guia de segurança do Oracle Database.

Auditoria no Microsoft SQL Server

O fluxo de atividade de banco de dados usa o recurso SQLAudit para auditar o banco de dados do SQL Server.

A instância do RDS para SQL Server contém o seguinte:

- Auditoria de servidor: a auditoria de servidor SQL coleta uma única instância de ações no nível do servidor ou do banco de dados, e um grupo de ações a serem monitoradas. As auditorias `RDS_DAS_AUDIT` e `RDS_DAS_AUDIT_CHANGES` em nível de servidor são gerenciadas pelo RDS.
- Especificação de auditoria de servidor: a especificação de auditoria de servidor registra os eventos no nível do servidor. Você pode modificar a especificação `RDS_DAS_SERVER_AUDIT_SPEC`. Essa especificação está vinculada à auditoria de servidor `RDS_DAS_AUDIT`. A especificação `RDS_DAS_CHANGES_AUDIT_SPEC` é gerenciada pelo RDS.
- Especificação de auditoria de banco de dados: a especificação de auditoria de banco de dados registra os eventos no nível do banco de dados. Você pode criar uma especificação de auditoria de banco de dados `RDS_DAS_DB_<name>` e vinculá-la à auditoria de servidor `RDS_DAS_AUDIT`.

Você pode configurar fluxos de atividade de banco de dados usando o console ou a CLI. Normalmente, configure fluxos de atividade do banco de dados da seguinte maneira:

1. (Opcional) Crie uma especificação de auditoria de banco de dados com o comando `CREATE DATABASE AUDIT SPECIFICATION` e vincule-a à auditoria de servidor `RDS_DAS_AUDIT`.
2. (Opcional) Modifique a especificação de auditoria de servidor com o comando `ALTER SERVER AUDIT SPECIFICATION` e defina as políticas.
3. Ative as políticas de auditoria de banco de dados e servidor. Por exemplo:

   ```
   ALTER DATABASE AUDIT SPECIFICATION [<Your database specification>] WITH (STATE=ON)

   ALTER SERVER AUDIT SPECIFICATION [RDS_DAS_SERVER_AUDIT_SPEC] WITH (STATE=ON)
   ```
4. Configurar fluxos de atividade do banco de dados.

   Somente as atividades que correspondem às políticas de auditoria de servidor e banco de dados são capturadas e enviadas ao fluxo de dados do Amazon Kinesis. Quando fluxos de atividade de banco de dados estão habilitados e as políticas estão bloqueadas, um administrador de banco de dados não pode alterar a política de auditoria nem remover registros de auditoria.

> **Important**
>
> Se a especificação de auditoria de banco de dados para um banco de dados específico estiver habilitada e a política estiver em um estado bloqueado, o banco de dados não poderá ser descartado.

Para ter mais informações sobre auditoria do SQL Server, consulte Componentes de auditoria do SQL Server na documentação do Microsoft SQL Server.

Campos de auditoria não nativos para Oracle Database e SQL Server

Quando você inicia um fluxo de atividade de banco de dados, cada evento de banco de dados gera um evento de fluxo de atividade correspondente. Por exemplo, um usuário de banco de dados pode executar as instruções `SELECT` e `INSERT`. O banco de dados audita esses eventos e os envia para um fluxo de dados do Amazon Kinesis.

Os eventos são representados no fluxo como objetos JSON. Um objeto JSON contém um `DatabaseActivityMonitoringRecord`, que contém uma matriz `databaseActivityEventList`. Os campos predefinidos na matriz incluem `class`, `clientApplication` e `command`.

Por padrão, um fluxo de atividade não contém campos de auditoria nativos do mecanismo. É possível configurar o Amazon RDS para Oracle e o SQL Server de modo que inclua esses campos extras no objeto JSON `engineNativeAuditFields`.

No Oracle Database, a maioria dos eventos na trilha de auditoria unificada são mapeados para campos do fluxo de atividade de dados do RDS. Por exemplo, o campo `UNIFIED_AUDIT_TRAIL.SQL_TEXT` em mapas de auditoria unificada para o campo `commandText` em um fluxo de atividade de banco de dados. No entanto, campos de auditoria do Oracle Database, como `OS_USERNAME`, não são mapeados para campos predefinidos em um fluxo de atividade de banco de dados.

No SQL Server, a maioria dos campos do evento que são registrados pelo SQLAudit são mapeados para os campos no fluxo de atividade do banco de dados RDS. Por exemplo, o campo `code` de `sys.fn_get_audit_file` na auditoria é mapeado para o campo `commandText` em um fluxo de atividade de banco de dados. No entanto, campos de auditoria de banco de dados do SQL

Server, como `permission_bitmask`, não são mapeados para campos predefinidos em um fluxo de atividade de banco de dados.

Para ter mais informações sobre databaseActivityEventList, consulte Matriz JSON databaseActivityEventList para fluxos de atividades de banco de dados.

Substituição do grupo de parâmetros de banco de dados

Normalmente, você ativa a auditoria unificada no RDS for Oracle anexando um grupo de parâmetros. No entanto, o recurso Database Activity Streams requer configuração adicional. Para melhorar a experiência do cliente, o Amazon RDS faz o seguinte:

- Se você ativar um fluxo de atividade, o RDS para Oracle vai ignorar os parâmetros de auditoria no grupo de parâmetros.
- Se você desativar um fluxo de atividade, o RDS para Oracle deixará de ignorar os parâmetros de auditoria.

O fluxo de atividade de banco de dados para o SQL Server não depende de nenhum parâmetro definido na opção de auditoria SQL.

Modo assíncrono para transmissões de atividades de banco de dados

Os fluxos de atividade no Amazon RDS são sempre assíncronos. Quando uma sessão de banco de dados gera um evento de fluxo de atividade, a sessão retorna às atividades normais imediatamente. Em segundo plano, o Amazon RDS transforma o evento do fluxo de atividade em um registro durável.

Se ocorrer um erro na tarefa de segundo plano, o Amazon RDS gera um evento. Este evento indica o início e o final de qualquer janela de tempo em que os registros do evento do fluxo de atividade podem ter sido perdidos. O modo assíncrono favorece a performance do banco de dados sobre a precisão do fluxo de atividade.

Requisitos e limitações para fluxos de atividade de banco de dados

No RDS, os fluxos de atividade de banco de dados têm os seguintes requisitos e limites:

- O Amazon Kinesis é imprescindível para os fluxos de atividades do banco de dados.
- O AWS Key Management Service (AWS KMS) é imprescindível para fluxos de atividades de banco de dados, pois eles são sempre criptografados.

- A aplicação de criptografia adicional ao seu fluxo de dados do Amazon Kinesis é incompatível com fluxos de atividade do banco de dados, que já estão criptografados com sua chave do AWS KMS.
- Você mesmo cria e gerencia políticas de auditoria. Por padrão, diferente do Amazon Aurora, o RDS for Oracle não captura atividades.
- Você mesmo cria e gerencia políticas e especificações de auditoria. Por padrão, ao contrário do Amazon Aurora, o Amazon RDS não captura a atividade de um banco de dados.
- Em uma implantação multi-AZ, inicie o fluxo de atividade de banco de dados apenas na instância de banco de dados primária. O fluxo de atividades audita as instâncias de banco de dados primário e em espera automaticamente. Não é necessária nenhuma etapa adicional durante um failover.
- Renomear uma instância de banco de dados não criará um novo fluxo do Kinesis.
- O RDS para Oracle não oferece suporte a CDBs.
- As réplicas de leitura não são compatíveis.

Disponibilidade de regiões e versões

A disponibilidade e a compatibilidade de recursos variam entre versões específicas de cada mecanismo de banco de dados e entre Regiões da AWS. Para ter mais informações sobre a disponibilidade de versões e regiões para fluxos de atividades de banco de dados, consulte Regiões e mecanismos de banco de dados compatíveis com fluxos de atividades de banco de dados no Amazon RDS.

Classes de instância de banco de dados compatíveis para transmissões de atividades de banco de dados

Para o RDS para Oracle, é possível utilizar fluxos de atividade de banco de dados com as seguintes classes de instâncias de banco de dados:

- db.m4.*large
- db.m5.*large
- db.m5d.*large
- db.m6i.*large
- db.r4.*large
- db.r5.*large
- db.r5.*large.tpc*.mem*x
- db.r5b.*large

- db.r5b.*large.tpc*.mem*x
- db.r5d.*large
- db.r6i.*large
- db.r6i.*large.tpc*.mem*x
- db.x2idn.*large
- db.x2iedn.*large
- db.x2iezn.*large
- db.z1d.*large

Para o RDS para SQL Server, é possível utilizar fluxos de atividade de banco de dados com as seguintes classes de instâncias de banco de dados:

- db.m4.*large
- db.m5.*large
- db.m5d.*large
- db.m6i.*large
- db.r4.*large
- db.r5.*large
- db.r5b.*large
- db.r5d.*large
- db.r6i.*large
- db.x1e.*large
- db.x2iedn.*large
- db.z1d.*large

Para mais informações sobre tipos de classes da instância, consulte Classes de instâncias de banco de dados .

Configurar a auditoria unificada do Oracle Database

Ao configurar a auditoria unificada para uso com fluxos de atividades de banco de dados, as seguintes situações são possíveis:

- A auditoria unificada não está configurada para o banco de dados Oracle.

  Nesse caso, crie políticas com o comando `CREATE AUDIT POLICY`, depois ative-as com o comando `AUDIT POLICY`. O exemplo a seguir cria e ativa uma política para monitorar usuários com privilégios e perfis específicos.

  ```
  CREATE AUDIT POLICY table_pol
  PRIVILEGES CREATE ANY TABLE, DROP ANY TABLE
  ROLES emp_admin, sales_admin;

  AUDIT POLICY table_pol;
  ```

  Para obter instruções completas, consulte Configuring Audit Policies (Configurar políticas de auditoria) na documentação do Oracle Database.

- A auditoria unificada está configurada para o banco de dados Oracle.

  Quando você ativa um fluxo de atividade de banco de dados, o RDS para Oracle limpa automaticamente os dados de auditoria existentes. Também revoga privilégios de trilha de auditoria. O RDS for Oracle não pode mais fazer o seguinte:

  - Elimine os registros de trilha de auditoria unificada.
  - Adicione, exclua ou modifique a política de auditoria unificada.
  - Atualize o último carimbo de data/hora arquivado.

  > **Important**
  >
  > Recomendamos fortemente que você faça backup dos dados de auditoria antes de ativar um fluxo de atividade de banco de dados.

  Para obter uma descrição da visualização `UNIFIED_AUDIT_TRAIL`, consulte UNIFIED_AUDIT_TRAIL. Se você tiver uma conta do Oracle Support, consulte Como depurar a TRILHA DE AUDITORIA UNIFICADA.

Configurar uma política de auditoria para o Amazon RDS para Microsoft SQL Server

Uma instância de banco de dados do SQL Server tem a auditoria de servidor `RDS_DAS_AUDIT`, que é gerenciada pelo Amazon RDS. Você pode definir as políticas para registrar eventos do servidor na especificação de auditoria de servidor `RDS_DAS_SERVER_AUDIT_SPEC`. Você pode criar uma especificação de auditoria de banco de dados, como `RDS_DAS_DB_<name>`, e definir as políticas para registrar eventos do banco de dados. Para obter a lista de grupos de ações de auditoria em nível de servidor e banco de dados, consulte Ações e grupos de ações de auditoria do SQL Server na documentação do Microsoft SQL Server.

A política de servidor padrão monitora somente falhas de logins e alterações em bancos de dados ou especificações de auditoria de servidor para fluxos de atividade de banco de dados.

As limitações para a auditoria e para as especificações de auditoria incluem o seguinte:

- Você não pode modificar as especificações de auditoria de servidor ou banco de dados quando o fluxo de atividade do banco de dados está em um estado bloqueado.
- Você não pode modificar a especificação de auditoria de servidor `RDS_DAS_AUDIT`.
- Você não pode modificar a auditoria `RDS_DAS_CHANGES` do SQL Server ou sua especificação de auditoria de servidor relacionada `RDS_DAS_CHANGES_AUDIT_SPEC`.
- Ao criar uma especificação de auditoria de banco de dados, você deve usar o formato `RDS_DAS_DB_<name>`; por exemplo, `RDS_DAS_DB_databaseActions`.

> **Important**
>
> Para classes de instâncias menores, recomendamos que você não audite todos os dados, apenas os necessários. Isso ajuda a reduzir o impacto na performance do recurso Database Activity Streams nessas classes de instâncias.

O exemplo de código abaixo modifica a especificação de auditoria de servidor `RDS_DAS_SERVER_AUDIT_SPEC` e realiza a auditoria de todas as ações de desconexão e de login bem-sucedido:

```
ALTER SERVER AUDIT SPECIFICATION [RDS_DAS_SERVER_AUDIT_SPEC]
```

```
      WITH (STATE=OFF);
ALTER SERVER AUDIT SPECIFICATION [RDS_DAS_SERVER_AUDIT_SPEC]
      ADD (LOGOUT_GROUP),
      ADD (SUCCESSFUL_LOGIN_GROUP)
      WITH (STATE = ON );
```

O exemplo de código abaixo cria uma especificação de auditoria de banco de dados `RDS_DAS_DB_database_spec` e a anexa à auditoria de servidor `RDS_DAS_AUDIT`:

```
USE testDB;
CREATE DATABASE AUDIT SPECIFICATION [RDS_DAS_DB_database_spec]
     FOR SERVER AUDIT [RDS_DAS_AUDIT]
     ADD ( INSERT, UPDATE, DELETE
          ON testTable BY testUser )
     WITH (STATE = ON);
```

Depois que as especificações de auditoria forem configuradas, certifique-se de que as especificações `RDS_DAS_SERVER_AUDIT_SPEC` e `RDS_DAS_DB_<name>` estejam definidas no estado `ON`. Agora elas podem enviar os dados de auditoria para o fluxo de atividade do banco de dados.

Iniciar um stream de atividade de banco de dados

Ao iniciar um fluxo de atividade para instância de banco de dados, cada evento da atividade do banco de dados que você configurou na política de auditoria gera um evento do fluxo de atividade. Comandos SQL, como `CONNECT` e `SELECT`, geram eventos de acesso. Comandos SQL, como `CREATE` e `INSERT`, geram eventos de alteração.

> **Important**
>
> Ativar um fluxo de atividade para uma instância de banco de dados Oracle limpa os dados de auditoria existentes. Também revoga privilégios de trilha de auditoria. Quando o fluxo está habilitado, o RDS for Oracle não pode mais fazer o seguinte:
>
> - Elimine os registros de trilha de auditoria unificada.
> - Adicione, exclua ou modifique a política de auditoria unificada.
> - Atualize o último carimbo de data/hora arquivado.

Console

Para iniciar um fluxo de atividade de banco de dados

1. Abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Databases (Bancos de dados).
3. Escolha a instância de banco de dados do Amazon RDS em que deseja iniciar um fluxo de atividade. Em uma implantação multi-AZ, inicie o fluxo apenas na instância principal. O fluxo de atividades audita as instâncias principal e em espera.
4. Em Actions (Ações), escolha Start activity (Iniciar atividade).

   A janela Start database activity stream: *name* (Iniciar fluxo de atividade do banco de dados: nome) aparece. Nela, *name* (nome) é a sua instância do RDS.
5. Insira as seguintes configurações:

   - Em AWS KMS key, escolha uma chave na lista de AWS KMS keys.

     O Amazon RDS usa a chave do KMS para criptografar a chave que, por sua vez, criptografa a atividade do banco de dados. Escolha uma chave do KMS diferente da chave padrão. Para obter mais informações sobre as chaves de criptografia e o AWS KMS, consulte O que é o AWS Key Management Service? no Guia do desenvolvedor do AWS Key Management Service.
   - Em Eventos de atividades do banco de dados, escolha Habilitar os campos de auditoria nativos do mecanismo para incluir campos de auditoria específicos do mecanismo.
   - Escolha Immediately (Imediatamente).

     Quando você escolhe Immediately (Imediatamente), a instância do RDS reinicia imediatamente. Se você escolher During the next maintenance window, (Durante a próxima janela de manutenção), a instância do RDS não reinicia imediatamente. Nesse caso, o stream de atividade do banco de dados não é iniciado até a próxima janela de manutenção.
6. Selecione Start database activity stream (Iniciar fluxo de atividades do banco de dados).

   O status para o banco de dados mostra que o fluxo de atividade está começando.

> **Note**
>
> Se você receber o erro `You can't start a database activity stream in this configuration`, confira Classes de instância de banco de dados compatíveis para transmissões de atividades de banco de dados para ver se seu cluster de banco de dados está usando uma classe de instância compatível.

AWS CLI

Para iniciar fluxos de atividade de banco de dados para uma instância de banco de dados, configure o banco de dados utilizando o comando start-activity-stream da AWS CLI.

- `--resource-arn` *`arn`*: especifica o nome do recurso da Amazon (ARN) da instância de banco de dados.
- `--kms-key-id` *`key`*: especifica o identificador de chave KMS para criptografar mensagens no fluxo de atividade do banco de dados. O identificador da chave do KMS da AWS é o ARN da chave, o ID da chave ou o ARN do alias ou o nome do alias para AWS KMS key.
- `--engine-native-audit-fields-included`: contém campos de auditoria específicos do mecanismo no fluxo de dados. Para excluir esses campos, especifique `--no-engine-native-audit-fields-included` (padrão).

O exemplo a seguir inicia um fluxo de atividade de banco de dados para uma instância de banco de dados no modo assíncrono.

Para Linux, macOS ou Unix:

```
aws rds start-activity-stream \
    --mode async \
    --kms-key-id my-kms-key-arn \
    --resource-arn my-instance-arn \
    --engine-native-audit-fields-included \
    --apply-immediately
```

Para Windows:

```
aws rds start-activity-stream ^
```

```
    --mode async ^
    --kms-key-id my-kms-key-arn ^
    --resource-arn my-instance-arn ^
    --engine-native-audit-fields-included ^
    --apply-immediately
```

Amazon RDS API

Para iniciar fluxos de atividade de banco de dados para uma instância de banco de dados, configure a instância utilizando a operação StartActivityStream.

Chame a ação com os parâmetros abaixo:

- `Region`
- `KmsKeyId`
- `ResourceArn`
- `Mode`
- `EngineNativeAuditFieldsIncluded`

Modificar um fluxo de atividades de banco de dados para o Amazon RDS

Talvez você queira personalizar sua política de auditoria do Amazon RDS quando seu fluxo de atividade for iniciado. Se não quiser perder tempo e dados interrompendo o fluxo de atividades, você pode alterar o estado da política de auditoria para uma das seguintes configurações:

Bloqueada (padrão)

As políticas de auditoria em seu banco de dados são somente leitura.

Unlocked (Desbloqueada)

As políticas de auditoria em seu banco de dados são de leitura/gravação.

As etapas básicas são:

1. Modifique o estado da política de auditoria para desbloqueada.
2. Personalize sua política de auditoria.
3. Modifique o estado da política de auditoria para bloqueada.

Console

Como modificar o estado da política de auditoria do fluxo de atividades

1. Abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Databases (Bancos de dados).
3. Em Actions (Ações), escolha Modify database activity stream (Modificar fluxo de atividades do banco de dados).

   A janela Modify database activity stream: ***name*** (Iniciar fluxo de atividade do banco de dados: name) aparece, em que *name* é sua instância do RDS.
4. Escolha uma das seguintes opções:

   Locked (Bloqueada)

   Quando você bloqueia sua política de auditoria, ela se torna somente leitura. Você não pode editar sua política de auditoria, a menos que desbloqueie a política ou interrompa o fluxo de atividades.

   Unlocked (Desbloqueada)

   Quando você desbloqueia sua política de auditoria, ela se torna de leitura/gravação. Você pode editar sua política de auditoria enquanto o fluxo de atividades é iniciado.
5. Escolha Modify DB activity stream (Modificar fluxo de atividades de banco de dados).

   O status do banco de dados do Amazon RDS mostra Configuração do fluxo de atividades.
6. (Opcional) Escolha o link da instância de banco de dados. Em seguida, escolha a guia Configuration (Configuração).

   O campo Audit policy status (Status da política de auditoria) mostra um dos seguintes valores:

   - Locked (Bloqueada)
   - Unlocked (Desbloqueada)
   - Locking policy (Política de bloqueio)
   - Unlocking policy (Política de desbloqueio)

AWS CLI

Para modificar o estado do fluxo de atividade de uma instância de banco de dados, use o comando modify-activity-stream da AWS CLI.

Opção	Obrigatório?	Descrição
`--resource-arn` *my-instance-ARN*	Sim	O nome do recurso da Amazon (ARN) da instância de banco de dados do RDS.
`--audit-policy-state`	Não	O novo estado da política de auditoria para o fluxo de atividades do banco de dados em sua instância : `locked` ou `unlocked`.

O exemplo a seguir desbloqueia a política de auditoria para o fluxo de atividades iniciado em *my-instance-ARN*.

Para Linux, macOS ou Unix:

```
aws rds modify-activity-stream \
    --resource-arn my-instance-ARN \
    --audit-policy-state unlocked
```

Para Windows:

```
aws rds modify-activity-stream ^
    --resource-arn my-instance-ARN ^
    --audit-policy-state unlocked
```

O exemplo a seguir descreve a instância *my-instance*. A saída parcial da amostra mostra que a política de auditoria está desbloqueada.

```
aws rds describe-db-instances --db-instance-identifier my-instance

{
    "DBInstances": [
        {
```

```
            ...
            "Engine": "oracle-ee",
            ...
            "ActivityStreamStatus": "started",
            "ActivityStreamKmsKeyId": "ab12345e-1111-2bc3-12a3-ab1cd12345e",
            "ActivityStreamKinesisStreamName": "aws-rds-das-db-
AB1CDEFG23GHIJK4LMNOPQRST",
            "ActivityStreamMode": "async",
            "ActivityStreamEngineNativeAuditFieldsIncluded": true,
            "ActivityStreamPolicyStatus": "unlocked",
            ...
        }
    ]
}
```

API do RDS

Para modificar o estado da política do seu fluxo de atividades de banco de dados, use a operação ModifyActivityStream.

Chame a ação com os parâmetros abaixo:

- `AuditPolicyState`
- `ResourceArn`

Obter o status de um fluxo de atividade de banco de dados

É possível obter o status de um fluxo de atividade para sua instância de banco de dados do Amazon RDS usando o console ou a AWS CLI.

Console

Para obter o status de um fluxo de atividade de banco de dados

1. Abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Databases (Bancos de dados) e escolha o link da instância de banco de dados.
3. Escolha a guia Configuração e verifique o status do Stream de atividades do banco de dados.

AWS CLI

É possível obter a configuração do fluxo de atividade para uma instância de banco de dados como a resposta a uma solicitação de CLI describe-db-instances.

O exemplo a seguir descreve *my-instance*.

```
aws rds --region my-region describe-db-instances --db-instance-identifier my-db
```

O exemplo a seguir mostra uma resposta JSON. Os campos a seguir são mostrados:

- `ActivityStreamKinesisStreamName`
- `ActivityStreamKmsKeyId`
- `ActivityStreamStatus`
- `ActivityStreamMode`
- `ActivityStreamPolicyStatus`

```
{
    "DBInstances": [
        {
            ...
            "Engine": "oracle-ee",
            ...
            "ActivityStreamStatus": "starting",
            "ActivityStreamKmsKeyId": "ab12345e-1111-2bc3-12a3-ab1cd12345e",
            "ActivityStreamKinesisStreamName": "aws-rds-das-db-
AB1CDEFG23GHIJK4LMNOPQRST",
            "ActivityStreamMode": "async",
            "ActivityStreamEngineNativeAuditFieldsIncluded": true,
            "ActivityStreamPolicyStatus": locked",
            ...
        }
    ]
}
```

API do RDS

É possível obter a configuração do fluxo de atividades para um banco de dados como a resposta a uma operação DescribeDBInstances.

Interromper um fluxo de atividade de banco de dados

É possível interromper um fluxo de atividade usando o console ou a AWS CLI.

Se você excluir sua instância de banco de dados do Amazon RDS, o fluxo de atividade será interrompido e o fluxo subjacente do Amazon Kinesis será excluído automaticamente.

Console

Como desativar um fluxo de atividade

1. Abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Databases (Bancos de dados).
3. Escolha um banco de dados de onde deseja interromper o fluxo de atividade de banco de dados.
4. Em Actions (Ações), escolha Stop activity (Interromper atividade). A janela Database Activity Stream (Fluxo de atividade de banco de dados) é exibida.
   a. Escolha Immediately (Imediatamente).

      Quando você escolhe Immediately (Imediatamente), a instância do RDS reinicia imediatamente. Se você escolher During the next maintenance window, (Durante a próxima janela de manutenção), a instância do RDS não reinicia imediatamente. Nesse caso, o fluxo de atividade do banco de dados não será interrompido até a próxima janela de manutenção.
   b. Escolha Continue.

AWS CLI

Para interromper fluxos de atividades de banco de dados para seu banco de dados, configure a instância de banco de dados usando o comando da AWS CLI stop-activity-stream. Identifique a região da AWS para a instância de banco de dados usando o parâmetro `--region`. O parâmetro `--apply-immediately` é opcional.

Para Linux, macOS ou Unix:

```
aws rds --region MY_REGION \
    stop-activity-stream \
    --resource-arn MY_DB_ARN \
    --apply-immediately
```

Para Windows:

```
aws rds --region MY_REGION ^
    stop-activity-stream ^
    --resource-arn MY_DB_ARN ^
    --apply-immediately
```

API do RDS

Para interromper os fluxos de atividades de banco de dados para o banco de dados, configure a instância de banco de dados usando a operação StopActivityStream. Identifique a região da AWS para a instância de banco de dados usando o parâmetro `Region`. O parâmetro `ApplyImmediately` é opcional.

Monitorar fluxos de atividades de banco de dados

Os fluxos de atividades de banco de dados monitoram e relatam atividades. O stream de atividade é coletado e transmitido para o Amazon Kinesis. No Kinesis, é possível monitorar o stream de atividades ou outros serviços e aplicações podem consumir o stream de atividades para análise posterior. É possível encontrar o nome do fluxo subjacente do Kinesis utilizando o comando `describe-db-instances` da AWS CLI ou a operação `DescribeDBInstances` da API do RDS.

O Amazon RDS gerencia o fluxo do Kinesis para você da seguinte forma:

- O Amazon RDS cria o fluxo do Kinesis automaticamente com um período de retenção de 24 horas.
- O Amazon RDS escala o fluxo do Kinesis, se necessário.
- Se você interromper o fluxo de atividades do banco de dados ou excluir a instância de banco de dados, o Amazon RDS excluirá o fluxo do Kinesis.

As categorias de atividade a seguir são monitoradas e colocadas no log de auditoria do fluxo de atividade:

- Comandos SQL: todos os comandos SQL são auditados e também instruções preparadas, funções integradas e funções em PL/SQL. Chamadas para procedimentos armazenados são auditadas. Quaisquer instruções SQL emitidas dentro de procedimentos armazenados ou funções também são auditadas.
- Outras informações do banco de dados: atividades monitoradas incluem a instrução SQL completa, a contagem de linhas afetadas dos comandos DML, os objetos acessados e o nome

do banco de dados exclusivo. Os fluxos de atividades de banco de dados também monitoram as variáveis de ligação e os parâmetros de procedimento armazenados.

> **Important**
>
> O texto SQL completo de cada instrução é visível no log de auditoria do stream de atividades, incluindo quaisquer dados confidenciais. No entanto, as senhas do usuário do banco de dados serão editadas se o Oracle puder determiná-las pelo contexto, como na instrução SQL a seguir.
>
> ```
> ALTER ROLE role-name WITH password
> ```

- Informações de conexão: a atividade monitorada inclui informações de rede e sessão, o ID do processo de servidor e códigos de saída.

Se um fluxo de atividade tiver uma falha ao monitorar uma instância de banco de dados, você será notificado por meio de eventos do RDS.

Nas seções a seguir, é possível acessar, auditar e processar fluxos de atividades do banco de dados.

Tópicos

- Acessar um fluxo de atividades no Amazon Kinesis
- Conteúdo do log de auditoria e exemplos de log de fluxos de atividade do banco de dados
- Matriz JSON databaseActivityEventList para fluxos de atividades de banco de dados
- Processar um fluxo de atividade usando o SDK da AWS

Acessar um fluxo de atividades no Amazon Kinesis

Ao habilitar um fluxo de atividade para um banco de dados, um fluxo do Kinesis será criado para você. No Kinesis, você pode monitorar a atividade de banco de dados em tempo real. Para analisar detalhadamente a atividade de banco de dados, é possível conectar o stream do Kinesis para aplicações consumidoras. Também é possível conectar o fluxo a aplicações de gerenciamento de compatibilidade como o IBM Security Guardium ou Imperva's SecureSphere Database Audit and Protection.

Você pode acessar seu fluxo do Kinesis a partir do console do RDS ou do console do Kinesis.

Como acessar um fluxo de atividade pelo Kinesis usando o console do RDS

1. Abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Databases (Bancos de dados).
3. Escolha a instância de banco de dados do Amazon RDS em que iniciou um fluxo de atividade.
4. Escolher configuração.
5. Em Database activity stream (Fluxo de atividades do banco de dados), clique no link em Kinesis stream (Fluxo do Kinesis).
6. No console do Kinesis, selecione Monitoring (Monitoramento) para começar a observar a atividade de banco de dados.

Como acessar um fluxo de atividade pelo Kinesis usando o console do Kinesis

1. Abra o console do Kinesis em https://console.aws.amazon.com/kinesis.
2. Escolha o fluxo de atividade na lista de fluxos do Kinesis.

   O nome de um fluxo de atividade contém o prefixo `aws-rds-das-db-` seguido pelo ID de recurso do banco de dados. Veja um exemplo a seguir.

   ```
   aws-rds-das-db-NHVOV4PCLWHGF52NP
   ```

   Para utilizar o console do Amazon RDS para encontrar o ID do recurso do banco de dados, escolha a instância de banco de dados na lista de bancos de dados e escolha a guia Configuration (Configuração).

   Para utilizar a AWS CLI a fim de encontrar o nome completo do fluxo do Kinesis para um fluxo de atividades, use uma solicitação de CLI describe-db-instances e anote os valores de `ActivityStreamKinesisStreamName` na resposta.
3. Escolha Monitoring (Monitoramento) para começar a observar a atividade de banco de dados.

Para obter mais informações sobre como usar o Amazon Kinesis, consulte O que é o Amazon Kinesis Data Streams?.

Conteúdo do log de auditoria e exemplos de log de fluxos de atividade do banco de dados

Os eventos monitorados são representados no fluxo de atividade do banco de dados como strings JSON. A estrutura consiste em um objeto JSON que contém um `DatabaseActivityMonitoringRecord`, que, por sua vez, contém uma matriz `databaseActivityEventList` de eventos de atividade.

> **Note**
>
> Para fluxos de atividades do banco de dados, a matriz JSON `paramList` não inclui valores nulos de aplicações Hibernate.

Tópicos

- Exemplos de log de auditoria para fluxos de atividade
- Objeto JSON DatabaseActivityMonitoringRecords
- Objeto JSON databaseActivityEvents

Exemplos de log de auditoria para fluxos de atividade

Veja a seguir exemplos de log de auditoria JSON descriptografados de registros de evento de atividade.

Example Registro de evento de atividade de uma instrução SQL CONNECT

O registro de eventos de atividade a seguir mostra um login com o uso de uma instrução SQL `CONNECT` (`command`) por um cliente JDBC Thin (`clientApplication`) para seu banco de dados Oracle.

```
{
    "class": "Standard",
    "clientApplication": "JDBC Thin Client",
    "command": "LOGON",
    "commandText": null,
    "dbid": "0123456789",
    "databaseName": "ORCL",
    "dbProtocol": "oracle",
    "dbUserName": "TEST",
```

```
    "endTime": null,
    "errorMessage": null,
    "exitCode": 0,
    "logTime": "2021-01-15 00:15:36.233787",
    "netProtocol": "tcp",
    "objectName": null,
    "objectType": null,
    "paramList": [],
    "pid": 17904,
    "remoteHost": "123.456.789.012",
    "remotePort": "25440",
    "rowCount": null,
    "serverHost": "987.654.321.098",
    "serverType": "oracle",
    "serverVersion": "19.0.0.0.ru-2020-01.rur-2020-01.r1.EE.3",
    "serviceName": "oracle-ee",
    "sessionId": 987654321,
    "startTime": null,
    "statementId": 1,
    "substatementId": null,
    "transactionId": "0000000000000000",
    "engineNativeAuditFields": {
        "UNIFIED_AUDIT_POLICIES": "TEST_POL_EVERYTHING",
        "FGA_POLICY_NAME": null,
        "DV_OBJECT_STATUS": null,
        "SYSTEM_PRIVILEGE_USED": "CREATE SESSION",
        "OLS_LABEL_COMPONENT_TYPE": null,
        "XS_SESSIONID": null,
        "ADDITIONAL_INFO": null,
        "INSTANCE_ID": 1,
        "DBID": 123456789
        "DV_COMMENT": null,
        "RMAN_SESSION_STAMP": null,
        "NEW_NAME": null,
        "DV_ACTION_NAME": null,
        "OLS_PROGRAM_UNIT_NAME": null,
        "OLS_STRING_LABEL": null,
        "RMAN_SESSION_RECID": null,
        "OBJECT_PRIVILEGES": null,
        "OLS_OLD_VALUE": null,
        "XS_TARGET_PRINCIPAL_NAME": null,
        "XS_NS_ATTRIBUTE": null,
        "XS_NS_NAME": null,
        "DBLINK_INFO": null,
```

```
        "AUTHENTICATION_TYPE": "(TYPE\u003d(DATABASE));(CLIENT ADDRESS\u003d((ADDRESS
\u003d(PROTOCOL\u003dtcp)(HOST\u003d205.251.233.183)(PORT\u003d25440))));",
        "OBJECT_EDITION": null,
        "OLS_PRIVILEGES_GRANTED": null,
        "EXCLUDED_USER": null,
        "DV_ACTION_OBJECT_NAME": null,
        "OLS_LABEL_COMPONENT_NAME": null,
        "EXCLUDED_SCHEMA": null,
        "DP_TEXT_PARAMETERS1": null,
        "XS_USER_NAME": null,
        "XS_ENABLED_ROLE": null,
        "XS_NS_ATTRIBUTE_NEW_VAL": null,
        "DIRECT_PATH_NUM_COLUMNS_LOADED": null,
        "AUDIT_OPTION": null,
        "DV_EXTENDED_ACTION_CODE": null,
        "XS_PACKAGE_NAME": null,
        "OLS_NEW_VALUE": null,
        "DV_RETURN_CODE": null,
        "XS_CALLBACK_EVENT_TYPE": null,
        "USERHOST": "a1b2c3d4e5f6.amazon.com",
        "GLOBAL_USERID": null,
        "CLIENT_IDENTIFIER": null,
        "RMAN_OPERATION": null,
        "TERMINAL": "unknown",
        "OS_USERNAME": "sumepate",
        "OLS_MAX_READ_LABEL": null,
        "XS_PROXY_USER_NAME": null,
        "XS_DATASEC_POLICY_NAME": null,
        "DV_FACTOR_CONTEXT": null,
        "OLS_MAX_WRITE_LABEL": null,
        "OLS_PARENT_GROUP_NAME": null,
        "EXCLUDED_OBJECT": null,
        "DV_RULE_SET_NAME": null,
        "EXTERNAL_USERID": null,
        "EXECUTION_ID": null,
        "ROLE": null,
        "PROXY_SESSIONID": 0,
        "DP_BOOLEAN_PARAMETERS1": null,
        "OLS_POLICY_NAME": null,
        "OLS_GRANTEE": null,
        "OLS_MIN_WRITE_LABEL": null,
        "APPLICATION_CONTEXTS": null,
        "XS_SCHEMA_NAME": null,
        "DV_GRANTEE": null,
```

```
        "XS_COOKIE": null,
        "DBPROXY_USERNAME": null,
        "DV_ACTION_CODE": null,
        "OLS_PRIVILEGES_USED": null,
        "RMAN_DEVICE_TYPE": null,
        "XS_NS_ATTRIBUTE_OLD_VAL": null,
        "TARGET_USER": null,
        "XS_ENTITY_TYPE": null,
        "ENTRY_ID": 1,
        "XS_PROCEDURE_NAME": null,
        "XS_INACTIVITY_TIMEOUT": null,
        "RMAN_OBJECT_TYPE": null,
        "SYSTEM_PRIVILEGE": null,
        "NEW_SCHEMA": null,
        "SCN": 5124715
    }
}
```

O registro de eventos de atividade a seguir mostra uma falha de login no banco de dados do SQL Server.

```
{
    "type": "DatabaseActivityMonitoringRecord",
    "clusterId": "",
    "instanceId": "db-4JCWQLUZVFYP7DIWP6JVQ77O3Q",
    "databaseActivityEventList": [
        {
            "class": "LOGIN",
            "clientApplication": "Microsoft SQL Server Management Studio",
            "command": "LOGIN FAILED",
            "commandText": "Login failed for user 'test'. Reason: Password did not
 match that for the login provided. [CLIENT: local-machine]",
            "databaseName": "",
            "dbProtocol": "SQLSERVER",
            "dbUserName": "test",
            "endTime": null,
            "errorMessage": null,
            "exitCode": 0,
            "logTime": "2022-10-06 21:34:42.7113072+00",
            "netProtocol": null,
            "objectName": "",
            "objectType": "LOGIN",
            "paramList": null,
```

```
            "pid": null,
            "remoteHost": "local machine",
            "remotePort": null,
            "rowCount": 0,
            "serverHost": "172.31.30.159",
            "serverType": "SQLSERVER",
            "serverVersion": "15.00.4073.23.v1.R1",
            "serviceName": "sqlserver-ee",
            "sessionId": 0,
            "startTime": null,
            "statementId": "0x1eb0d1808d34a94b9d3dcf5432750f02",
            "substatementId": 1,
            "transactionId": "0",
            "type": "record",
            "engineNativeAuditFields": {
                "target_database_principal_id": 0,
                "target_server_principal_id": 0,
                "target_database_principal_name": "",
                "server_principal_id": 0,
                "user_defined_information": "",
                "response_rows": 0,
                "database_principal_name": "",
                "target_server_principal_name": "",
                "schema_name": "",
                "is_column_permission": false,
                "object_id": 0,
                "server_instance_name": "EC2AMAZ-NFUJJNO",
                "target_server_principal_sid": null,
                "additional_information": "<action_info "xmlns=\"http://
schemas.microsoft.com/sqlserver/2008/sqlaudit_data\"><pooled_connection>0</
pooled_connection><error>0x00004818</error><state>8</state><address>local machine</
address><PasswordFirstNibbleHash>B</PasswordFirstNibbleHash></action_info>"-->,
                "duration_milliseconds": 0,
                "permission_bitmask": "0x00000000000000000000000000000000",
                "data_sensitivity_information": "",
                "session_server_principal_name": "",
                "connection_id": "98B4F537-0F82-49E3-AB08-B9D33B5893EF",
                "audit_schema_version": 1,
                "database_principal_id": 0,
                "server_principal_sid": null,
                "user_defined_event_id": 0,
                "host_name": "EC2AMAZ-NFUJJNO"
            }
        }
```

```
    ]
}
```

> **Note**
>
> Se um fluxo de atividade do banco de dados não estiver habilitado, o último campo no documento JSON será `"engineNativeAuditFields": { }`.

Example Registro de evento de atividades de uma instrução CREATE TABLE do

O exemplo a seguir mostra um evento `CREATE TABLE` para seu banco de dados Oracle.

```
{
    "class": "Standard",
    "clientApplication": "sqlplus@ip-12-34-5-678 (TNS V1-V3)",
    "command": "CREATE TABLE",
    "commandText": "CREATE TABLE persons(\n    person_id NUMBER GENERATED BY DEFAULT AS
 IDENTITY,\n    first_name VARCHAR2(50) NOT NULL,\n    last_name VARCHAR2(50) NOT NULL,
\n    PRIMARY KEY(person_id)\n)",
    "dbid": "0123456789",
    "databaseName": "ORCL",
    "dbProtocol": "oracle",
    "dbUserName": "TEST",
    "endTime": null,
    "errorMessage": null,
    "exitCode": 0,
    "logTime": "2021-01-15 00:22:49.535239",
    "netProtocol": "beq",
    "objectName": "PERSONS",
    "objectType": "TEST",
    "paramList": [],
    "pid": 17687,
    "remoteHost": "123.456.789.0",
    "remotePort": null,
    "rowCount": null,
    "serverHost": "987.654.321.01",
    "serverType": "oracle",
    "serverVersion": "19.0.0.0.ru-2020-01.rur-2020-01.r1.EE.3",
    "serviceName": "oracle-ee",
    "sessionId": 1234567890,
    "startTime": null,
```

```
    "statementId": 43,
    "substatementId": null,
    "transactionId": "090011007F0D0000",
    "engineNativeAuditFields": {
        "UNIFIED_AUDIT_POLICIES": "TEST_POL_EVERYTHING",
        "FGA_POLICY_NAME": null,
        "DV_OBJECT_STATUS": null,
        "SYSTEM_PRIVILEGE_USED": "CREATE SEQUENCE, CREATE TABLE",
        "OLS_LABEL_COMPONENT_TYPE": null,
        "XS_SESSIONID": null,
        "ADDITIONAL_INFO": null,
        "INSTANCE_ID": 1,
        "DV_COMMENT": null,
        "RMAN_SESSION_STAMP": null,
        "NEW_NAME": null,
        "DV_ACTION_NAME": null,
        "OLS_PROGRAM_UNIT_NAME": null,
        "OLS_STRING_LABEL": null,
        "RMAN_SESSION_RECID": null,
        "OBJECT_PRIVILEGES": null,
        "OLS_OLD_VALUE": null,
        "XS_TARGET_PRINCIPAL_NAME": null,
        "XS_NS_ATTRIBUTE": null,
        "XS_NS_NAME": null,
        "DBLINK_INFO": null,
        "AUTHENTICATION_TYPE": "(TYPE\u003d(DATABASE));(CLIENT ADDRESS\u003d((PROTOCOL
\u003dbeq)(HOST\u003d123.456.789.0)));",
        "OBJECT_EDITION": null,
        "OLS_PRIVILEGES_GRANTED": null,
        "EXCLUDED_USER": null,
        "DV_ACTION_OBJECT_NAME": null,
        "OLS_LABEL_COMPONENT_NAME": null,
        "EXCLUDED_SCHEMA": null,
        "DP_TEXT_PARAMETERS1": null,
        "XS_USER_NAME": null,
        "XS_ENABLED_ROLE": null,
        "XS_NS_ATTRIBUTE_NEW_VAL": null,
        "DIRECT_PATH_NUM_COLUMNS_LOADED": null,
        "AUDIT_OPTION": null,
        "DV_EXTENDED_ACTION_CODE": null,
        "XS_PACKAGE_NAME": null,
        "OLS_NEW_VALUE": null,
        "DV_RETURN_CODE": null,
        "XS_CALLBACK_EVENT_TYPE": null,
```

```
        "USERHOST": "ip-10-13-0-122",
        "GLOBAL_USERID": null,
        "CLIENT_IDENTIFIER": null,
        "RMAN_OPERATION": null,
        "TERMINAL": "pts/1",
        "OS_USERNAME": "rdsdb",
        "OLS_MAX_READ_LABEL": null,
        "XS_PROXY_USER_NAME": null,
        "XS_DATASEC_POLICY_NAME": null,
        "DV_FACTOR_CONTEXT": null,
        "OLS_MAX_WRITE_LABEL": null,
        "OLS_PARENT_GROUP_NAME": null,
        "EXCLUDED_OBJECT": null,
        "DV_RULE_SET_NAME": null,
        "EXTERNAL_USERID": null,
        "EXECUTION_ID": null,
        "ROLE": null,
        "PROXY_SESSIONID": 0,
        "DP_BOOLEAN_PARAMETERS1": null,
        "OLS_POLICY_NAME": null,
        "OLS_GRANTEE": null,
        "OLS_MIN_WRITE_LABEL": null,
        "APPLICATION_CONTEXTS": null,
        "XS_SCHEMA_NAME": null,
        "DV_GRANTEE": null,
        "XS_COOKIE": null,
        "DBPROXY_USERNAME": null,
        "DV_ACTION_CODE": null,
        "OLS_PRIVILEGES_USED": null,
        "RMAN_DEVICE_TYPE": null,
        "XS_NS_ATTRIBUTE_OLD_VAL": null,
        "TARGET_USER": null,
        "XS_ENTITY_TYPE": null,
        "ENTRY_ID": 12,
        "XS_PROCEDURE_NAME": null,
        "XS_INACTIVITY_TIMEOUT": null,
        "RMAN_OBJECT_TYPE": null,
        "SYSTEM_PRIVILEGE": null,
        "NEW_SCHEMA": null,
        "SCN": 5133083
    }
}
```

O exemplo a seguir mostra um evento `CREATE TABLE` para o seu banco de dados do SQL Server.


```
{
    "type": "DatabaseActivityMonitoringRecord",
    "clusterId": "",
    "instanceId": "db-4JCWQLUZVFYP7DIWP6JVQ77O3Q",
    "databaseActivityEventList": [
        {
            "class": "SCHEMA",
            "clientApplication": "Microsoft SQL Server Management Studio - Query",
            "command": "ALTER",
            "commandText": "Create table [testDB].[dbo].[TestTable2](\r\ntextA
 varchar(6000),\r\n    textB varchar(6000)\r\n)",
            "databaseName": "testDB",
            "dbProtocol": "SQLSERVER",
            "dbUserName": "test",
            "endTime": null,
            "errorMessage": null,
            "exitCode": 1,
            "logTime": "2022-10-06 21:44:38.4120677+00",
            "netProtocol": null,
            "objectName": "dbo",
            "objectType": "SCHEMA",
            "paramList": null,
            "pid": null,
            "remoteHost": "local machine",
            "remotePort": null,
            "rowCount": 0,
            "serverHost": "172.31.30.159",
            "serverType": "SQLSERVER",
            "serverVersion": "15.00.4073.23.v1.R1",
            "serviceName": "sqlserver-ee",
            "sessionId": 84,
            "startTime": null,
            "statementId": "0x5178d33d56e95e419558b9607158a5bd",
            "substatementId": 1,
            "transactionId": "4561864",
            "type": "record",
            "engineNativeAuditFields": {
                "target_database_principal_id": 0,
                "target_server_principal_id": 0,
                "target_database_principal_name": "",
                "server_principal_id": 2,
                "user_defined_information": "",
```

```
                "response_rows": 0,
                "database_principal_name": "dbo",
                "target_server_principal_name": "",
                "schema_name": "",
                "is_column_permission": false,
                "object_id": 1,
                "server_instance_name": "EC2AMAZ-NFUJJNO",
                "target_server_principal_sid": null,
                "additional_information": "",
                "duration_milliseconds": 0,
                "permission_bitmask": "0x00000000000000000000000000000000",
                "data_sensitivity_information": "",
                "session_server_principal_name": "test",
                "connection_id": "EE1FE3FD-EF2C-41FD-AF45-9051E0CD983A",
                "audit_schema_version": 1,
                "database_principal_id": 1,
                "server_principal_sid":
 "0x0105000000000005150000000bdc2795e2d0717901ba6998cf4010000",
                "user_defined_event_id": 0,
                "host_name": "EC2AMAZ-NFUJJNO"
            }
        }
    ]
}
```

Example Registro de evento de atividades de uma instrução SELECT do

O exemplo a seguir mostra um evento SELECT para seu banco de dados Oracle.

```
{
    "class": "Standard",
    "clientApplication": "sqlplus@ip-12-34-5-678 (TNS V1-V3)",
    "command": "SELECT",
    "commandText": "select count(*) from persons",
    "databaseName": "1234567890",
    "dbProtocol": "oracle",
    "dbUserName": "TEST",
    "endTime": null,
    "errorMessage": null,
    "exitCode": 0,
    "logTime": "2021-01-15 00:25:18.850375",
    "netProtocol": "beq",
    "objectName": "PERSONS",
    "objectType": "TEST",
```

```
    "paramList": [],
    "pid": 17687,
    "remoteHost": "123.456.789.0",
    "remotePort": null,
    "rowCount": null,
    "serverHost": "987.654.321.09",
    "serverType": "oracle",
    "serverVersion": "19.0.0.0.ru-2020-01.rur-2020-01.r1.EE.3",
    "serviceName": "oracle-ee",
    "sessionId": 1080639707,
    "startTime": null,
    "statementId": 44,
    "substatementId": null,
    "transactionId": null,
    "engineNativeAuditFields": {
        "UNIFIED_AUDIT_POLICIES": "TEST_POL_EVERYTHING",
        "FGA_POLICY_NAME": null,
        "DV_OBJECT_STATUS": null,
        "SYSTEM_PRIVILEGE_USED": null,
        "OLS_LABEL_COMPONENT_TYPE": null,
        "XS_SESSIONID": null,
        "ADDITIONAL_INFO": null,
        "INSTANCE_ID": 1,
        "DV_COMMENT": null,
        "RMAN_SESSION_STAMP": null,
        "NEW_NAME": null,
        "DV_ACTION_NAME": null,
        "OLS_PROGRAM_UNIT_NAME": null,
        "OLS_STRING_LABEL": null,
        "RMAN_SESSION_RECID": null,
        "OBJECT_PRIVILEGES": null,
        "OLS_OLD_VALUE": null,
        "XS_TARGET_PRINCIPAL_NAME": null,
        "XS_NS_ATTRIBUTE": null,
        "XS_NS_NAME": null,
        "DBLINK_INFO": null,
        "AUTHENTICATION_TYPE": "(TYPE\u003d(DATABASE));(CLIENT ADDRESS\u003d((PROTOCOL
\u003dbeq)(HOST\u003d123.456.789.0)));",
        "OBJECT_EDITION": null,
        "OLS_PRIVILEGES_GRANTED": null,
        "EXCLUDED_USER": null,
        "DV_ACTION_OBJECT_NAME": null,
        "OLS_LABEL_COMPONENT_NAME": null,
        "EXCLUDED_SCHEMA": null,
```

```
        "DP_TEXT_PARAMETERS1": null,
        "XS_USER_NAME": null,
        "XS_ENABLED_ROLE": null,
        "XS_NS_ATTRIBUTE_NEW_VAL": null,
        "DIRECT_PATH_NUM_COLUMNS_LOADED": null,
        "AUDIT_OPTION": null,
        "DV_EXTENDED_ACTION_CODE": null,
        "XS_PACKAGE_NAME": null,
        "OLS_NEW_VALUE": null,
        "DV_RETURN_CODE": null,
        "XS_CALLBACK_EVENT_TYPE": null,
        "USERHOST": "ip-12-34-5-678",
        "GLOBAL_USERID": null,
        "CLIENT_IDENTIFIER": null,
        "RMAN_OPERATION": null,
        "TERMINAL": "pts/1",
        "OS_USERNAME": "rdsdb",
        "OLS_MAX_READ_LABEL": null,
        "XS_PROXY_USER_NAME": null,
        "XS_DATASEC_POLICY_NAME": null,
        "DV_FACTOR_CONTEXT": null,
        "OLS_MAX_WRITE_LABEL": null,
        "OLS_PARENT_GROUP_NAME": null,
        "EXCLUDED_OBJECT": null,
        "DV_RULE_SET_NAME": null,
        "EXTERNAL_USERID": null,
        "EXECUTION_ID": null,
        "ROLE": null,
        "PROXY_SESSIONID": 0,
        "DP_BOOLEAN_PARAMETERS1": null,
        "OLS_POLICY_NAME": null,
        "OLS_GRANTEE": null,
        "OLS_MIN_WRITE_LABEL": null,
        "APPLICATION_CONTEXTS": null,
        "XS_SCHEMA_NAME": null,
        "DV_GRANTEE": null,
        "XS_COOKIE": null,
        "DBPROXY_USERNAME": null,
        "DV_ACTION_CODE": null,
        "OLS_PRIVILEGES_USED": null,
        "RMAN_DEVICE_TYPE": null,
        "XS_NS_ATTRIBUTE_OLD_VAL": null,
        "TARGET_USER": null,
        "XS_ENTITY_TYPE": null,
```

```
        "ENTRY_ID": 13,
        "XS_PROCEDURE_NAME": null,
        "XS_INACTIVITY_TIMEOUT": null,
        "RMAN_OBJECT_TYPE": null,
        "SYSTEM_PRIVILEGE": null,
        "NEW_SCHEMA": null,
        "SCN": 5136972
    }
}
```

O exemplo a seguir mostra um evento SELECT para o seu banco de dados do SQL Server.

```
{
    "type": "DatabaseActivityMonitoringRecord",
    "clusterId": "",
    "instanceId": "db-4JCWQLUZVFYP7DIWP6JVQ77O3Q",
    "databaseActivityEventList": [
        {
            "class": "TABLE",
            "clientApplication": "Microsoft SQL Server Management Studio - Query",
            "command": "SELECT",
            "commandText": "select * from [testDB].[dbo].[TestTable]",
            "databaseName": "testDB",
            "dbProtocol": "SQLSERVER",
            "dbUserName": "test",
            "endTime": null,
            "errorMessage": null,
            "exitCode": 1,
            "logTime": "2022-10-06 21:24:59.9422268+00",
            "netProtocol": null,
            "objectName": "TestTable",
            "objectType": "TABLE",
            "paramList": null,
            "pid": null,
            "remoteHost": "local machine",
            "remotePort": null,
            "rowCount": 0,
            "serverHost": "172.31.30.159",
            "serverType": "SQLSERVER",
            "serverVersion": "15.00.4073.23.v1.R1",
            "serviceName": "sqlserver-ee",
            "sessionId": 62,
            "startTime": null,
```

```
            "statementId": "0x03baed90412f564fad640ebe51f89b99",
            "substatementId": 1,
            "transactionId": "4532935",
            "type": "record",
            "engineNativeAuditFields": {
                "target_database_principal_id": 0,
                "target_server_principal_id": 0,
                "target_database_principal_name": "",
                "server_principal_id": 2,
                "user_defined_information": "",
                "response_rows": 0,
                "database_principal_name": "dbo",
                "target_server_principal_name": "",
                "schema_name": "dbo",
                "is_column_permission": true,
                "object_id": 581577110,
                "server_instance_name": "EC2AMAZ-NFUJJNO",
                "target_server_principal_sid": null,
                "additional_information": "",
                "duration_milliseconds": 0,
                "permission_bitmask": "0x00000000000000000000000000000001",
                "data_sensitivity_information": "",
                "session_server_principal_name": "test",
                "connection_id": "AD3A5084-FB83-45C1-8334-E923459A8109",
                "audit_schema_version": 1,
                "database_principal_id": 1,
                "server_principal_sid":
 "0x0105000000000005150000000bdc2795e2d0717901ba6998cf4010000",
                "user_defined_event_id": 0,
                "host_name": "EC2AMAZ-NFUJJNO"
            }
        }
    ]
}
```


Objeto JSON DatabaseActivityMonitoringRecords

Os registros de eventos de atividade de banco de dados estão em um objeto JSON que contém as informações a seguir.

Campo JSON	Tipo de dados	Descrição
`type`	string	O tipo de registro JSON. O valor é `DatabaseActivityMonitoringRecords` .
`version`	string	A versão dos registros de monitoramento de atividade de banco de dados. O banco de dados Oracle usa a versão 1.3 e o SQL Server usa a versão 1.4. Essas versões do mecanismo apresentam o objeto JSON `engineNativeAuditFields` .
databaseActivityEvents	string	Um objeto JSON que contém os eventos de atividade.
key	string	Uma chave de criptografia que você usa para descriptografar o Matriz JSON databaseActivityEventList

Objeto JSON databaseActivityEvents

O objeto JSON `databaseActivityEvents` contém as informações a seguir.

Campos de nível superior no registro JSON

Cada evento no log de auditoria é incluído em um registro no formato JSON. Esse registro contém os campos a seguir.

type

Esse campo sempre tem o valor `DatabaseActivityMonitoringRecords`.

versão

Esse campo representa a versão do contrato ou do protocolo de dados de streaming da atividade do banco de dados. Define quais campos estão disponíveis.

databaseActivityEvents

Uma string criptografada que representa um ou mais eventos de atividade. Ela é representada como uma matriz de bytes base64. Quando você descriptografa a string, o resultado é um registro no formato JSON com campos, conforme mostrado nos exemplos nesta seção.

chave

A chave de dados criptografada usada para criptografar a string `databaseActivityEvents`. Esta é a mesma AWS KMS key que você forneceu ao iniciar o fluxo de atividade do banco de dados.

O exemplo a seguir mostra o formato desse registro.

```
{
  "type":"DatabaseActivityMonitoringRecords",
  "version":"1.3",
  "databaseActivityEvents":"encrypted audit records",
  "key":"encrypted key"
}
```

```
          "type":"DatabaseActivityMonitoringRecords",
          "version":"1.4",
          "databaseActivityEvents":"encrypted audit records",
          "key":"encrypted key"
```

Siga as seguintes etapas para descriptografar o conteúdo do campo `databaseActivityEvents`:

1. Descriptografe o valor no campo JSON da `key` usando a chave do KMS que forneceu ao iniciar o fluxo de atividade do banco de dados. Fazer isso retorna a chave de criptografia de dados em texto não criptografado.
2. Decodifique em base64 o valor no campo JSON `databaseActivityEvents` para obter o texto cifrado, em formato binário, da carga útil de auditoria.
3. Descriptografe o texto cifrado binário com a chave de criptografia de dados que você decodificou na primeira etapa.
4. Descompacte a carga útil descriptografada.
   - A carga criptografada está no campo `databaseActivityEvents`.

- O campo `databaseActivityEventList` contém uma matriz de registros de auditoria. Os campos `type` na matriz podem ser `record` ou `heartbeat`.

O registro do evento de atividade do log de auditoria é um objeto JSON que contém as informações a seguir.

Campo JSON	Tipo de dados	Descrição
`type`	string	O tipo de registro JSON. O valor é `DatabaseActivityMonitoringRecord` .
`instanceId`	string	O identificador de recurso da instância de banco de dados. Ele corresponde ao atributo de instância de banco de dados `DbiResourceId` .
Matriz JSON databaseActivityEventList	string	Uma matriz de registros de auditoria de atividade ou mensagens de pulsação.

Matriz JSON databaseActivityEventList para fluxos de atividades de banco de dados

A carga do log de auditoria é uma matriz JSON `databaseActivityEventList` criptografada. As tabelas a seguir listam em ordem alfabética os campos para cada evento de atividade na matriz `DatabaseActivityEventList` descriptografada de um log de auditoria.

Quando a auditoria unificada é habilitada no Oracle Database, os registros de auditoria são preenchidos nessa nova trilha de auditoria. A visualização do `UNIFIED_AUDIT_TRAIL` exibe registros de auditoria em forma tabular, recuperando os registros de auditoria da trilha de auditoria. Quando você inicia um fluxo de atividade de banco de dados, uma coluna no `UNIFIED_AUDIT_TRAIL` mapeia para um campo na matriz `databaseActivityEventList`.

> **Important**
>
> A estrutura do evento está sujeita a alterações. O Amazon RDS pode adicionar novos campos a eventos de atividade no futuro. Em aplicações que analisam os dados JSON,

confirme se o código pode ignorar ou executar ações apropriadas para nomes de campo desconhecidos.

Campos databaseActivityEventList para o Amazon RDS para Oracle

Veja a seguir campos `databaseActivityEventList` do Amazon RDS para Oracle.

Campo	Tipo de dados	Origem	Descrição
`class`	string	Coluna `AUDIT_TYPE` no `UNIFIED_AUDIT_TRAIL`	A classe do evento de atividade. Corresponde à coluna `AUDIT_TYPE` na visualização do `UNIFIED_AUDIT_TRAIL`. Os valores válidos para o Amazon RDS para Oracle são os seguintes: • `Standard` • `FineGrainedAudit` • `XS` • `Database Vault` • `Label Security` • `RMAN_AUDIT` • `Datapump` • `Direct path API` Para obter mais informações, consulte UNIFIED_AUDIT_TRAIL na documentação do Oracle.

Campo	Tipo de dados	Origem	Descrição
clientApplication	string	CLIENT_PROGRAM_NAME em UNIFIED_AUDIT_TRAIL	A aplicação que o cliente usou para se conectar conforme relatado pelo cliente. O cliente não precisa fornecer essas informações, então o valor pode ser nulo. Um valor de amostra é JDBC Thin Client.
command	string	Coluna ACTION_NAME no UNIFIED_AUDIT_TRAIL	Nome da ação executada pelo usuário. Para entender a ação completa, leia o nome do comando e o valor AUDIT_TYPE . Um valor de amostra é ALTER DATABASE.
commandText	string	Coluna SQL_TEXT no UNIFIED_AUDIT_TRAIL	A instrução SQL associada ao evento. Um valor de amostra é ALTER DATABASE BEGIN BACKUP.
databaseName	string	Coluna NAME no V$DATABASE	O nome do banco de dados.
dbid	numb	Coluna DBID no UNIFIED_AUDIT_TRAIL	Identificador numérico do banco de dados. Um valor de amostra é 1559204751 .

Campo	Tipo de dados	Origem	Descrição
`dbProtocol`	string	N/D	o protocolo do banco de dados. Neste beta, o valor é `oracle`.
`dbUserName`	string	Coluna `DBUSERNAME` no `UNIFIED_AUDIT_TRAIL`	Nome do usuário do banco de dados cujas ações foram auditadas . Um valor de amostra é `RDSADMIN`.
`endTime`	string	N/D	Este campo não é usado para o RDS for Oracle e é sempre nulo.

Campo	Tipo de dados	Origem	Descrição

`engineNativeAuditFields`	objeto	`UNIFIED_AUDIT_TRAIL`	Por padrão, este objeto permanece em branco. Quando você inicia o fluxo de atividade com a opção `--engine-native-audit-fields-included` , esse objeto inclui as seguintes colunas e seus valores:
			`ADDITIONAL_INFO`
			`APPLICATION_CONTEXTS`
			`AUDIT_OPTION`
			`AUTHENTICATION_TYPE`
			`CLIENT_IDENTIFIER`
			`CURRENT_USER`
			`DBLINK_INFO`
			`DBPROXY_USERNAME`
			`DIRECT_PATH_NUM_COLUMNS_LOADED`
			`DP_BOOLEAN_PARAMETERS1`
			`DP_TEXT_PARAMETERS1`
			`DV_ACTION_CODE`
			`DV_ACTION_NAME`
			`DV_ACTION_OBJECT_NAME`
			`DV_COMMENT`
			`DV_EXTENDED_ACTION_CODE`
			`DV_FACTOR_CONTEXT`
			`DV_GRANTEE`
			`DV_OBJECT_STATUS`

Campo	Tipo de dados	Origem	Descrição
			DV_RETURN_CODE DV_RULE_SET_NAME ENTRY_ID EXCLUDED_OBJECT EXCLUDED_SCHEMA EXCLUDED_USER EXECUTION_ID EXTERNAL_USERID FGA_POLICY_NAME GLOBAL_USERID INSTANCE_ID KSACL_SER VICE_NAME KSACL_SOURCE_LOCA TION KSACL_USER_NAME NEW_NAME NEW_SCHEMA OBJECT_EDITION OBJECT_PRIVILEGES OLS_GRANTEE OLS_LABEL_COM PONENT_NAME OLS_LABEL_COMPO NENT_TYPE OLS_MAX_READ_LABEL OLS_MAX_WRITE_LA BEL OLS_MIN_WRITE_ LABEL OLS_NEW_VALUE OLS_OLD_VALUE OLS_PARENT_ GROUP_NAME OLS_POLICY_NAME OLS_PRIVILEGES_GRA NTED OLS_PRIVILEGE S_USED

Campo	Tipo de dados	Origem	Descrição
			OLS_PROGRAM_UNIT_NAME OLS_STRING_LABEL OS_USERNAME PROTOCOL_ACTION_NAME PROTOCOL_MESSAGE PROTOCOL_RETURN_CODE PROTOCOL_SESSION_ID PROTOCOL_USERHOST PROXY_SESSIONID RLS_INFO RMAN_DEVICE_TYPE RMAN_OBJECT_TYPE RMAN_OPERATION RMAN_SESSION_RECID RMAN_SESSION_STAMP ROLE SCN SYSTEM_PRIVILEGE SYSTEM_PRIVILEGE_USED TARGET_USER TERMINAL UNIFIED_AUDIT_POLICIES USERHOST XS_CALLBACK_EVENT_TYPE XS_COOKIE XS_DATASEC_POLICY_NAME XS_ENABLED_ROLE XS_ENTITY_TYPE XS_INACTIVITY_TIMEOUT XS_NS_ATTRIBUTE

Campo	Tipo de dados	Origem	Descrição
			`XS_NS_ATTRI BUTE_NEW_VAL` `XS_NS_ATTRIBUT E_OLD_VAL` `XS_NS_NAME` `XS_PACKAGE_NAME` `XS_PROCEDURE_NAME` `XS_PROXY_USER_NAME` `XS_SCHEMA_NAME` `XS_SESSIONID` `XS_TARGET_PRINC IPAL_NAME` `XS_USER_NAME` Para obter mais informações, consulte UNIFIED_AUDIT_TRAI L na documentação do Oracle Database.
`errorMessage`	string	N/D	Este campo não é usado para o RDS for Oracle e é sempre nulo.
`exitCode`	numb	Coluna `RETURN_CODE` no `UNIFIED_AUDIT_TRAIL`	Código de erro do Oracle Database gerado pela ação. Se a ação for realizada, o valor será `0`.
`logTime`	string	Coluna `EVENT_TIM ESTAMP_UTC` no `UNIFIED_AUDIT_TRAIL`	Carimbo de data/hora da criação da entrada da trilha de auditoria. Um valor de amostra é `2020-11-27 06:56:14.981404` .

Campo	Tipo de dados	Origem	Descrição
netProtocol	string	Coluna AUTHENTICATION_TYPE no UNIFIED_AUDIT_TRAIL	O protocolo de comunicação da rede Um valor de amostra é TCP.
objectName	string	Coluna OBJECT_NAME no UNIFIED_AUDIT_TRAIL	O nome do objeto afetado pela ação. Um valor de amostra é employees .
objectType	string	Coluna OBJECT_SCHEMA no UNIFIED_AUDIT_TRAIL	O nome do esquema do objeto afetado pela ação. Um valor de amostra é hr.
paramList	lista	Coluna SQL_BINDS no UNIFIED_AUDIT_TRAIL	A lista de variáveis de ligação, se houver, associadas a SQL_TEXT. Um valor de amostra é parameter_1,parameter_2 .
pid	numb	Coluna OS_PROCESS no UNIFIED_AUDIT_TRAIL	Identificador do processo do sistema operacional do processo do banco de dados Oracle. Um valor de amostra é 22396.
remoteHost	string	Coluna AUTHENTICATION_TYPE no UNIFIED_AUDIT_TRAIL	O nome de cliente ou endereço IP do host do qual a sessão foi gerada. Um valor de amostra é 123.456.789.123 .

Campo	Tipo de dados	Origem	Descrição
`remotePort`	string	Coluna `AUTHENTICATION_TYPE` no `UNIFIED_AUDIT_TRAIL`	O número da porta do cliente. Um valor típico nos ambientes do Oracle Database é `1521`.
`rowCount`	numb	N/D	Este campo não é usado para o RDS for Oracle e é sempre nulo.
`serverHost`	string	Host do banco de dados	O endereço IP do host do servidor de banco de dados. Um valor de amostra é `123.456.789.123` .
`serverType`	string	N/D	O tipo de servidor do banco de dados. O valor é sempre `ORACLE`.
`serverVersion`	string	Host do banco de dados	A versão do Amazon RDS para Oracle, a atualização de release (RU) e a revisão de atualização de release (RUR). Um valor de amostra é `19.0.0.0.ru-2020-01.rur-2020-01.r1.EE.3` .
`serviceName`	string	Host do banco de dados	O nome do serviço da Um valor de amostra é `oracle-ee` .

Campo	Tipo de dados	Origem	Descrição
`sessionId`	numb	Coluna `SESSIONID` no `UNIFIED_AUDIT_TRAIL`	O identificador da sessão da auditoria. Um exemplo é `1894327130` .
`startTime`	string	N/D	Este campo não é usado para o RDS for Oracle e é sempre nulo.
`statementId`	numb	Coluna `STATEMENT_ID` no `UNIFIED_AUDIT_TRAIL`	O ID numérico para cada execução de instrução . Uma instrução pode causar muitas ações. Um valor de amostra é `142197`.
`substatementId`	N/D	N/D	Este campo não é usado para o RDS for Oracle e é sempre nulo.
`transactionId`	string	Coluna `TRANSACTION_ID` no `UNIFIED_AUDIT_TRAIL`	O identificador da transação na qual o objeto é modificado. Um valor de amostra é `02000800D5030000` .

Campos databaseActivityEventList para o Amazon RDS para SQL Server

Veja a seguir campos `databaseActivityEventList` do Amazon RDS para SQL Server.

Campo	Tipo de dados	Origem	Descrição
`class`	string	`sys.fn_get_audit_file.class_type` mapeado para `sys.dm_audit_class_type_map.class_type_desc`	A classe do evento de atividade . Para obter mais informações, consulte Auditoria do SQL Server (mecanismo de banco de dados) na documentação da Microsoft.
`clientApplication`	string	`sys.fn_get_audit_file.application_name`	A aplicação à qual o cliente se conecta, conforme relatado pelo cliente (SQL Server versão 14 e posteriores). Esse campo fica nulo no SQL Server versão 13.
`command`	string	`sys.fn_get_audit_file.action_id` mapeado para `sys.dm_audit_actions.name`	A categoria geral da instrução SQL. O valor deste campo depende do valor da classe.
`commandText`	string	`sys.fn_get_audit_file.statement`	Este campo indica a instrução SQL.
`databaseName`	string	`sys.fn_get_audit_file.database_name`	O nome do banco de dados.
`dbProtocol`	string	N/D	o protocolo do banco de dados. Este valor é `SQLSERVER` .
`dbUserName`	string	`sys.fn_get_audit_file.server_principal_name`	O usuário do banco de dados para autenticação do cliente.
`endTime`	string	N/D	Este campo não é usado pelo Amazon RDS para SQL Server e o valor é nulo.

Campo	Tipo de dados	Origem	Descrição
`engineNativeAuditFields`	objeto	Cada campo em `sys.fn_get_audit_file` que não está listado nessa coluna.	Por padrão, este objeto permanece em branco. Quando você inicia o fluxo de atividade com a opção `--engine-native-audit-fields-included`, esse objeto inclui outros campos nativos de auditoria nativos do mecanismo, que não são retornados por esse mapa JSON.
`errorMessage`	string	N/D	Este campo não é usado pelo Amazon RDS para SQL Server e o valor é nulo.
`exitCode`	integer	`sys.fn_get_audit_file.succeeded`	Indica se a ação que iniciou o evento foi bem-sucedida. Este campo não pode ficar nulo. Para todos os eventos, exceto eventos de login, esse campo informa se a verificação de permissão foi bem-sucedida ou falhou, mas não se a operação foi bem-sucedida ou falhou. Os valores incluem: • `0`: falha • `1`: êxito
`logTime`	string	`sys.fn_get_audit_file.event_time`	O carimbo de data/hora do evento que é registrado pelo SQL Server.

Campo	Tipo de dados	Origem	Descrição
`netProtocol`	string	N/D	Este campo não é usado pelo Amazon RDS para SQL Server e o valor é nulo.
`objectName`	string	`sys.fn_get_audit_file.object_name`	O nome do objeto do banco de dados se a instrução SQL estiver operando em um objeto.
`objectType`	string	`sys.fn_get_audit_file.class_type` mapeado para `sys.dm_audit_class_type_map.class_type_desc`	O tipo do objeto do banco de dados se a instrução SQL estiver operando em um tipo de objeto.
`paramList`	string	N/D	Este campo não é usado pelo Amazon RDS para SQL Server e o valor é nulo.
`pid`	integer	N/D	Este campo não é usado pelo Amazon RDS para SQL Server e o valor é nulo.
`remoteHost`	string	`sys.fn_get_audit_file.client_ip`	O endereço IP ou o nome de host do cliente que emitiu a instrução SQL (SQL Server versão 14 e posteriores). Esse campo fica nulo no SQL Server versão 13.
`remotePort`	integer	N/D	Este campo não é usado pelo Amazon RDS para SQL Server e o valor é nulo.

Campo	Tipo de dados	Origem	Descrição
`rowCount`	integer	`sys.fn_get_audit_file.affected_rows`	O número de linhas de tabela afetadas pela instrução SQL (SQL Server versão 14 e posteriores). Esse campo existe no SQL Server versão 13.
`serverHost`	string	Host do banco de dados	O endereço IP do servidor de banco de dados host.
`serverType`	string	N/D	O tipo de servidor do banco de dados. O valor é `SQLSERVER` .
`serverVersion`	string	Host do banco de dados	A versão do servidor do banco de dados; por exemplo, 15.00.4073.23.v1.R1 para SQL Server 2017.
`serviceName`	string	Host do banco de dados	O nome do serviço da Um exemplo de valor é `sqlserver-ee` .
`sessionId`	integer	`sys.fn_get_audit_file.session_id`	Um identificador exclusivo da sessão.
`startTime`	string	N/D	Este campo não é usado pelo Amazon RDS para SQL Server e o valor é nulo.

Campo	Tipo de dados	Origem	Descrição
`statementId`	string	`sys.fn_get_audit_file.sequence_group_id`	Um identificador exclusivo para a instrução SQL do cliente. O identificador é diferente para cada evento gerado. Um valor de amostra é `0x38eaf4156267184094bb82071aaab644` .
`substatementId`	integer	`sys.fn_get_audit_file.sequence_number`	Um identificador para determinar o número de uma instrução na sequência. Esse identificador ajuda quando registros grandes são divididos em vários registros.
`transactionId`	integer	`sys.fn_get_audit_file.transaction_id`	Um identificador de uma transação. Se não houver nenhuma transação ativa, o valor será zero.
`type`	string	Fluxo de atividade de banco de dados gerado	O tipo de evento. Os valores são `record` ou `heartbeat` .

Processar um fluxo de atividade usando o SDK da AWS

É possível processar de maneira programática um fluxo de atividade usando o SDK da AWS. Veja a seguir exemplos de Java e Python em total funcionamento de como usar registros de fluxo de atividades de banco de dados do. Veja a seguir exemplos de Java e Python em total funcionamento de como você pode habilitar com base em exemplos de.

Java

```
import java.io.ByteArrayInputStream;
import java.io.ByteArrayOutputStream;
import java.io.IOException;
```

```
import java.net.InetAddress;
import java.nio.ByteBuffer;
import java.nio.charset.StandardCharsets;
import java.security.NoSuchAlgorithmException;
import java.security.NoSuchProviderException;
import java.security.Security;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import java.util.UUID;
import java.util.zip.GZIPInputStream;

import javax.crypto.Cipher;
import javax.crypto.NoSuchPaddingException;
import javax.crypto.spec.SecretKeySpec;

import com.amazonaws.auth.AWSStaticCredentialsProvider;
import com.amazonaws.auth.BasicAWSCredentials;
import com.amazonaws.encryptionsdk.AwsCrypto;
import com.amazonaws.encryptionsdk.CryptoInputStream;
import com.amazonaws.encryptionsdk.jce.JceMasterKey;
import
 com.amazonaws.services.kinesis.clientlibrary.exceptions.InvalidStateException;
import com.amazonaws.services.kinesis.clientlibrary.exceptions.ShutdownException;
import com.amazonaws.services.kinesis.clientlibrary.exceptions.ThrottlingException;
import com.amazonaws.services.kinesis.clientlibrary.interfaces.IRecordProcessor;
import
 com.amazonaws.services.kinesis.clientlibrary.interfaces.IRecordProcessorCheckpointer;
import
 com.amazonaws.services.kinesis.clientlibrary.interfaces.IRecordProcessorFactory;
import
 com.amazonaws.services.kinesis.clientlibrary.lib.worker.InitialPositionInStream;
import
 com.amazonaws.services.kinesis.clientlibrary.lib.worker.KinesisClientLibConfiguration;
import com.amazonaws.services.kinesis.clientlibrary.lib.worker.ShutdownReason;
import com.amazonaws.services.kinesis.clientlibrary.lib.worker.Worker;
import com.amazonaws.services.kinesis.clientlibrary.lib.worker.Worker.Builder;
import com.amazonaws.services.kinesis.model.Record;
import com.amazonaws.services.kms.AWSKMS;
import com.amazonaws.services.kms.AWSKMSClientBuilder;
import com.amazonaws.services.kms.model.DecryptRequest;
import com.amazonaws.services.kms.model.DecryptResult;
import com.amazonaws.util.Base64;
import com.amazonaws.util.IOUtils;
```

```
import com.google.gson.Gson;
import com.google.gson.GsonBuilder;
import com.google.gson.annotations.SerializedName;
import org.bouncycastle.jce.provider.BouncyCastleProvider;

public class DemoConsumer {

    private static final String STREAM_NAME = "aws-rds-das-[instance-external-
resource-id]"; // aws-rds-das-db-ABCD123456
    private static final String APPLICATION_NAME = "AnyApplication"; //unique
 application name for dynamo table generation that holds kinesis shard tracking
    private static final String AWS_ACCESS_KEY =
 "[AWS_ACCESS_KEY_TO_ACCESS_KINESIS]";
    private static final String AWS_SECRET_KEY =
 "[AWS_SECRET_KEY_TO_ACCESS_KINESIS]";
    private static final String RESOURCE_ID = "[external-resource-id]"; // db-
ABCD123456
    private static final String REGION_NAME = "[region-name]"; //us-east-1, us-
east-2...
    private static final BasicAWSCredentials CREDENTIALS = new
 BasicAWSCredentials(AWS_ACCESS_KEY, AWS_SECRET_KEY);
    private static final AWSStaticCredentialsProvider CREDENTIALS_PROVIDER = new
 AWSStaticCredentialsProvider(CREDENTIALS);

    private static final AwsCrypto CRYPTO = new AwsCrypto();
    private static final AWSKMS KMS = AWSKMSClientBuilder.standard()
            .withRegion(REGION_NAME)
            .withCredentials(CREDENTIALS_PROVIDER).build();

    class Activity {
        String type;
        String version;
        String databaseActivityEvents;
        String key;
    }

    class ActivityEvent {
        @SerializedName("class") String _class;
        String clientApplication;
        String command;
        String commandText;
        String databaseName;
        String dbProtocol;
        String dbUserName;
```

```
        String endTime;
        String errorMessage;
        String exitCode;
        String logTime;
        String netProtocol;
        String objectName;
        String objectType;
        List<String> paramList;
        String pid;
        String remoteHost;
        String remotePort;
        String rowCount;
        String serverHost;
        String serverType;
        String serverVersion;
        String serviceName;
        String sessionId;
        String startTime;
        String statementId;
        String substatementId;
        String transactionId;
        String type;
    }

    class ActivityRecords {
        String type;
        String clusterId; // note that clusterId will contain an empty string on RDS
Oracle and RDS SQL Server
        String instanceId;
        List<ActivityEvent> databaseActivityEventList;
    }

    static class RecordProcessorFactory implements IRecordProcessorFactory {
        @Override
        public IRecordProcessor createProcessor() {
            return new RecordProcessor();
        }
    }

    static class RecordProcessor implements IRecordProcessor {

        private static final long BACKOFF_TIME_IN_MILLIS = 3000L;
        private static final int PROCESSING_RETRIES_MAX = 10;
        private static final long CHECKPOINT_INTERVAL_MILLIS = 60000L;
```

```
        private static final Gson GSON = new
GsonBuilder().serializeNulls().create();

        private static final Cipher CIPHER;
        static {
            Security.insertProviderAt(new BouncyCastleProvider(), 1);
            try {
                CIPHER = Cipher.getInstance("AES/GCM/NoPadding", "BC");
            } catch (NoSuchAlgorithmException | NoSuchPaddingException |
NoSuchProviderException e) {
                throw new ExceptionInInitializerError(e);
            }
        }

        private long nextCheckpointTimeInMillis;

        @Override
        public void initialize(String shardId) {
        }

        @Override
        public void processRecords(final List<Record> records, final
IRecordProcessorCheckpointer checkpointer) {
            for (final Record record : records) {
                processSingleBlob(record.getData());
            }

            if (System.currentTimeMillis() > nextCheckpointTimeInMillis) {
                checkpoint(checkpointer);
                nextCheckpointTimeInMillis = System.currentTimeMillis() +
CHECKPOINT_INTERVAL_MILLIS;
            }
        }

        @Override
        public void shutdown(IRecordProcessorCheckpointer checkpointer,
ShutdownReason reason) {
            if (reason == ShutdownReason.TERMINATE) {
                checkpoint(checkpointer);
            }
        }

        private void processSingleBlob(final ByteBuffer bytes) {
            try {
```

```
                // JSON $Activity
                final Activity activity = GSON.fromJson(new String(bytes.array(),
StandardCharsets.UTF_8), Activity.class);

                // Base64.Decode
                final byte[] decoded =
Base64.decode(activity.databaseActivityEvents);
                final byte[] decodedDataKey = Base64.decode(activity.key);

                Map<String, String> context = new HashMap<>();
                context.put("aws:rds:db-id", RESOURCE_ID);

                // Decrypt
                final DecryptRequest decryptRequest = new DecryptRequest()

.withCiphertextBlob(ByteBuffer.wrap(decodedDataKey)).withEncryptionContext(context);
                final DecryptResult decryptResult = KMS.decrypt(decryptRequest);
                final byte[] decrypted = decrypt(decoded,
getByteArray(decryptResult.getPlaintext()));

                // GZip Decompress
                final byte[] decompressed = decompress(decrypted);
                // JSON $ActivityRecords
                final ActivityRecords activityRecords = GSON.fromJson(new
String(decompressed, StandardCharsets.UTF_8), ActivityRecords.class);

                // Iterate throught $ActivityEvents
                for (final ActivityEvent event :
activityRecords.databaseActivityEventList) {
                    System.out.println(GSON.toJson(event));
                }
            } catch (Exception e) {
                // Handle error.
                e.printStackTrace();
            }
        }

        private static byte[] decompress(final byte[] src) throws IOException {
            ByteArrayInputStream byteArrayInputStream = new
ByteArrayInputStream(src);
            GZIPInputStream gzipInputStream = new
GZIPInputStream(byteArrayInputStream);
            return IOUtils.toByteArray(gzipInputStream);
        }
```

```
       private void checkpoint(IRecordProcessorCheckpointer checkpointer) {
           for (int i = 0; i < PROCESSING_RETRIES_MAX; i++) {
               try {
                   checkpointer.checkpoint();
                   break;
               } catch (ShutdownException se) {
                   // Ignore checkpoint if the processor instance has been shutdown
(fail over).
                   System.out.println("Caught shutdown exception, skipping
checkpoint." + se);
                   break;
               } catch (ThrottlingException e) {
                   // Backoff and re-attempt checkpoint upon transient failures
                   if (i >= (PROCESSING_RETRIES_MAX - 1)) {
                       System.out.println("Checkpoint failed after " + (i + 1) +
"attempts." + e);
                       break;
                   } else {
                       System.out.println("Transient issue when checkpointing -
attempt " + (i + 1) + " of " + PROCESSING_RETRIES_MAX + e);
                   }
               } catch (InvalidStateException e) {
                   // This indicates an issue with the DynamoDB table (check for
table, provisioned IOPS).
                   System.out.println("Cannot save checkpoint to the DynamoDB table
used by the Amazon Kinesis Client Library." + e);
                   break;
               }
               try {
                   Thread.sleep(BACKOFF_TIME_IN_MILLIS);
               } catch (InterruptedException e) {
                   System.out.println("Interrupted sleep" + e);
               }
           }
       }
   }

   private static byte[] decrypt(final byte[] decoded, final byte[] decodedDataKey)
throws IOException {
       // Create a JCE master key provider using the random key and an AES-GCM
encryption algorithm
       final JceMasterKey masterKey = JceMasterKey.getInstance(new
SecretKeySpec(decodedDataKey, "AES"),
```

```
                "BC", "DataKey", "AES/GCM/NoPadding");
        try (final CryptoInputStream<JceMasterKey> decryptingStream =
 CRYPTO.createDecryptingStream(masterKey, new ByteArrayInputStream(decoded));
             final ByteArrayOutputStream out = new ByteArrayOutputStream()) {
            IOUtils.copy(decryptingStream, out);
            return out.toByteArray();
        }
    }

    public static void main(String[] args) throws Exception {
        final String workerId = InetAddress.getLocalHost().getCanonicalHostName() +
 ":" + UUID.randomUUID();
        final KinesisClientLibConfiguration kinesisClientLibConfiguration =
                new KinesisClientLibConfiguration(APPLICATION_NAME, STREAM_NAME,
 CREDENTIALS_PROVIDER, workerId);

 kinesisClientLibConfiguration.withInitialPositionInStream(InitialPositionInStream.LATEST);
        kinesisClientLibConfiguration.withRegionName(REGION_NAME);
        final Worker worker = new Builder()
                .recordProcessorFactory(new RecordProcessorFactory())
                .config(kinesisClientLibConfiguration)
                .build();

        System.out.printf("Running %s to process stream %s as worker %s...\n",
 APPLICATION_NAME, STREAM_NAME, workerId);

        try {
            worker.run();
        } catch (Throwable t) {
            System.err.println("Caught throwable while processing data.");
            t.printStackTrace();
            System.exit(1);
        }
        System.exit(0);
    }

    private static byte[] getByteArray(final ByteBuffer b) {
        byte[] byteArray = new byte[b.remaining()];
        b.get(byteArray);
        return byteArray;
    }
}
```

Python

```
import base64
import json
import zlib
import aws_encryption_sdk
from aws_encryption_sdk import CommitmentPolicy
from aws_encryption_sdk.internal.crypto import WrappingKey
from aws_encryption_sdk.key_providers.raw import RawMasterKeyProvider
from aws_encryption_sdk.identifiers import WrappingAlgorithm, EncryptionKeyType
import boto3

REGION_NAME = '<region>'                    # us-east-1
RESOURCE_ID = '<external-resource-id>'      # db-ABCD123456
STREAM_NAME = 'aws-rds-das-' + RESOURCE_ID  # aws-rds-das-db-ABCD123456

enc_client =
 aws_encryption_sdk.EncryptionSDKClient(commitment_policy=CommitmentPolicy.FORBID_ENCRYPT_AL

class MyRawMasterKeyProvider(RawMasterKeyProvider):
    provider_id = "BC"

    def __new__(cls, *args, **kwargs):
        obj = super(RawMasterKeyProvider, cls).__new__(cls)
        return obj

    def __init__(self, plain_key):
        RawMasterKeyProvider.__init__(self)
        self.wrapping_key =
 WrappingKey(wrapping_algorithm=WrappingAlgorithm.AES_256_GCM_IV12_TAG16_NO_PADDING,
                                        wrapping_key=plain_key,
 wrapping_key_type=EncryptionKeyType.SYMMETRIC)

    def _get_raw_key(self, key_id):
        return self.wrapping_key


def decrypt_payload(payload, data_key):
    my_key_provider = MyRawMasterKeyProvider(data_key)
    my_key_provider.add_master_key("DataKey")
    decrypted_plaintext, header = enc_client.decrypt(
        source=payload,

 materials_manager=aws_encryption_sdk.materials_managers.default.DefaultCryptoMaterialsManag
```

```
    return decrypted_plaintext


def decrypt_decompress(payload, key):
    decrypted = decrypt_payload(payload, key)
    return zlib.decompress(decrypted, zlib.MAX_WBITS + 16)


def main():
    session = boto3.session.Session()
    kms = session.client('kms', region_name=REGION_NAME)
    kinesis = session.client('kinesis', region_name=REGION_NAME)

    response = kinesis.describe_stream(StreamName=STREAM_NAME)
    shard_iters = []
    for shard in response['StreamDescription']['Shards']:
        shard_iter_response = kinesis.get_shard_iterator(StreamName=STREAM_NAME,
 ShardId=shard['ShardId'],

 ShardIteratorType='LATEST')
        shard_iters.append(shard_iter_response['ShardIterator'])

    while len(shard_iters) > 0:
        next_shard_iters = []
        for shard_iter in shard_iters:
            response = kinesis.get_records(ShardIterator=shard_iter, Limit=10000)
            for record in response['Records']:
                record_data = record['Data']
                record_data = json.loads(record_data)
                payload_decoded =
 base64.b64decode(record_data['databaseActivityEvents'])
                data_key_decoded = base64.b64decode(record_data['key'])
                data_key_decrypt_result =
 kms.decrypt(CiphertextBlob=data_key_decoded,

 EncryptionContext={'aws:rds:db-id': RESOURCE_ID})
                print (decrypt_decompress(payload_decoded,
 data_key_decrypt_result['Plaintext']))
            if 'NextShardIterator' in response:
                next_shard_iters.append(response['NextShardIterator'])
        shard_iters = next_shard_iters


if __name__ == '__main__':
```

```
    main()
```

Exemplos de políticas do IAM para fluxos de atividades de banco de dados

Qualquer usuário com privilégios de função do AWS Identity and Access Management (IAM) para os fluxos de atividades de banco de dados pode criar, iniciar, interromper e modificar as configurações do fluxo de atividade para uma instância de banco de dados. Essas ações estão incluídas no log de auditoria do fluxo. Para realizar as práticas recomendadas de conformidade, recomendamos não fornecer esses privilégios a DBAs.

Você define o acesso aos fluxos de atividades de banco de dados usando políticas do IAM. Para obter mais informações sobre autenticação do Amazon RDS, consulte Gerenciamento de identidade e acesso no Amazon RDS. Para obter mais informações sobre como criar políticas do IAM, consulte Criar e usar uma política do IAM para acesso do banco de dados do IAM.

Example Política para permitir a configuração de fluxos de atividades de banco de dados

Para conceder aos usuários acesso refinado para modificar fluxos de atividade, use as chaves de contexto de operação específicas de serviço `rds:StartActivityStream` e `rds:StopActivityStream` em uma política do IAM. O exemplo de política do IAM a seguir permite que um usuário ou uma função configure fluxos de atividade.

```
{
   "Version":"2012-10-17",
   "Statement":[
        {
            "Sid":"ConfigureActivityStreams",
            "Effect":"Allow",
            "Action": [
                "rds:StartActivityStream",
                "rds:StopActivityStream"
            ],
            "Resource":"*",
          }
     ]
}
```

Example Política para permitir o início de fluxos de atividades de banco de dados

O exemplo de política do IAM a seguir permite que um usuário ou uma função inicie fluxos de atividade.

```
{
    "Version":"2012-10-17",
    "Statement":[
        {
            "Sid":"AllowStartActivityStreams",
            "Effect":"Allow",
            "Action":"rds:StartActivityStream",
            "Resource":"*"
        }
    ]
}
```

Example Política para permitir a interrupção de fluxos de atividades de banco de dados

O exemplo de política do IAM a seguir permite que um usuário ou uma função interrompa fluxos de atividade.

```
{
    "Version":"2012-10-17",
    "Statement":[
        {
            "Sid":"AllowStopActivityStreams",
            "Effect":"Allow",
            "Action":"rds:StopActivityStream",
            "Resource":"*"
        }
     ]
}
```

Example Política para negar o início de fluxos de atividades de banco de dados

O exemplo de política do IAM a seguir impede que um usuário ou uma função inicie fluxos de atividades.

```
{
    "Version":"2012-10-17",
    "Statement":[
```

```
        {
            "Sid":"DenyStartActivityStreams",
            "Effect":"Deny",
            "Action":"rds:StartActivityStream",
            "Resource":"*"
        }
     ]
}
```

Example Política para interromper o início de fluxos de atividades de banco de dados

O exemplo de política do IAM a seguir impede que um usuário ou uma função interrompa fluxos de atividades.

```
{
    "Version":"2012-10-17",
    "Statement":[
        {
            "Sid":"DenyStopActivityStreams",
            "Effect":"Deny",
            "Action":"rds:StopActivityStream",
            "Resource":"*"
        }
    ]
}
```

Monitorar ameaças com o Amazon GuardDuty RDS Protection

O Amazon GuardDuty é um serviço de detecção de ameaças que ajuda a proteger contas, contêineres, workloads e dados no ambiente da AWS. Usando modelos de machine learning (ML) e recursos de detecção de anomalias e ameaças, o GuardDuty monitora continuamente diferentes fontes de log e atividades de runtime para identificar e priorizar possíveis riscos de segurança e atividades maliciosas no seu ambiente.

O GuardDuty RDS Protection analisa e traça o perfil de eventos de login em busca de possíveis ameaças de acesso aos seus bancos de dados do Amazon RDS. Quando você ativa o RDS Protection, o GuardDuty consome eventos de login do RDS de seus bancos de dados do RDS. O RDS Protection monitora esses eventos e traça o perfil de possíveis ameaças internas ou agentes externos.

Para ter mais informações sobre como habilitar a Proteção do GuardDuty RDS, consulte Proteção do GuardDuty RDS no Guia do usuário do Amazon GuardDuty.

Quando o RDS Protection detecta uma ameaça em potencial, como um padrão incomum em tentativas de login bem-sucedidas ou malsucedidas, o GuardDuty gera uma nova descoberta com detalhes sobre o banco de dados possivelmente comprometido. Você pode visualizar os detalhes da descoberta na seção de resumo da descoberta no console do Amazon GuardDuty. Os detalhes da descoberta variam de acordo com o tipo de descoberta. Os detalhes principais, o tipo de recurso e o perfil do recurso determinam o tipo de informação disponível para qualquer descoberta. Para ter mais informações sobre os detalhes comumente disponíveis para descobertas e os tipos de descoberta, consulte Encontrar detalhes e Tipos de descoberta da Proteção do GuardDuty RDS, respectivamente, no Guia do usuário do Amazon GuardDuty.

Você pode ativar ou desativar o recurso RDS Protection para qualquer Conta da AWS em qualquer Região da AWS onde esse recurso esteja disponível. Quando o RDS Protection não está ativado, o GuardDuty não detecta bancos de dados do RDS possivelmente comprometidos nem fornece detalhes sobre o comprometimento.

Uma conta do GuardDuty existente pode habilitar a Proteção do RDS com um período de teste de 30 dias. Para uma nova conta do GuardDuty, a Proteção do RDS já está habilitada e incluída no período de teste gratuito de 30 dias. Para ter mais informações, consulte Custo da Proteção do GuardDuty RDS no Guia do usuário do Amazon GuardDuty.

Para saber em quais Regiões da AWS não é possível usar o GuardDuty com a Proteção do RDS, consulte Region-specific feature availability no Guia do usuário do Amazon GuardDuty.

Para ter informações sobre as versões do banco de dados do RDS que o GuardDuty RDS Protection aceita, consulte Supported Amazon Aurora, Amazon RDS, and Aurora Limitless databases no Guia do usuário do Amazon GuardDuty.

Amazon RDS Custom

O Amazon RDS Custom automatiza tarefas e operações de administração do banco de dados. O RDS Custom permite que você, como administrador de banco de dados, acesse e personalize seu ambiente de banco de dados e sistema operacional. Com o RDS Custom, você pode personalizar para atender aos requisitos de aplicações herdadas, personalizadas e empacotadas.

Para ver os webinars e blogs mais recentes sobre o RDS Custom, consulte Amazon RDS Custom resources (Recursos do Amazon RDS Custom).

Tópicos

- Enfrentar o desafio da personalização do banco de dados
- Modelo de gerenciamento e benefícios para o Amazon RDS Custom
- Arquitetura do Amazon RDS Custom
- Segurança no Amazon RDS Custom
- Trabalhar com o RDS Custom for Oracle
- Trabalhar com o RDS Custom for SQL Server

Enfrentar o desafio da personalização do banco de dados

O Amazon RDS Custom traz os benefícios do Amazon RDS para um mercado que não pode migrar facilmente para um serviço totalmente gerenciado devido às personalizações que são necessárias com aplicações de terceiros. O Amazon RDS Custom economiza tempo administrativo, é durável e se expande no ritmo da sua empresa.

Se você precisa que todo o banco de dados e o sistema operacional sejam totalmente gerenciados pela AWS, recomendamos o Amazon RDS. Se você precisa de direitos administrativos sobre o banco de dados e o sistema operacional subjacente para disponibilizar aplicações dependentes, o Amazon RDS Custom é a melhor escolha. Se você deseja responsabilidade total pelo gerenciamento e apenas precisa de um serviço computacional gerenciado, a melhor opção é autogerenciar seus bancos de dados comerciais no Amazon EC2.

Para oferecer uma experiência de serviço gerenciada, o Amazon RDS não permite o acesso ao host subjacente. O Amazon RDS também restringe o acesso a alguns procedimentos e objetos que exigem privilégios de alto nível. No entanto, para algumas aplicações, talvez seja necessário realizar operações como um usuário privilegiado do sistema operacional.

Por exemplo, você pode precisar fazer o seguinte:

- Instale patches e pacotes personalizados de banco de dados e SO.
- Defina configurações específicas do banco de dados.
- Configure sistemas de arquivos para compartilhar arquivos diretamente com as suas aplicações.

Antes, se você precisasse personalizar sua aplicação, era necessário implantar seu banco de dados on-premises ou no Amazon EC2. Nesse caso, você assume a maior parte da responsabilidade, ou toda a responsabilidade, pelo gerenciamento de banco de dados, conforme resumido na tabela a seguir.

Recurso	Responsabilidade on-premises	Responsabilidade do Amazon EC2	Responsabilidade do Amazon RDS
Otimização de aplicações	Cliente	Cliente	Cliente
Escalabilidade	Cliente	Cliente	AWS
Alta disponibilidade	Cliente	Cliente	AWS
Backups de banco de dados	Cliente	Cliente	AWS
Aplicação de patches de softwares para banco de dados	Cliente	Cliente	AWS
Instalação de softwares para banco de dados	Cliente	Cliente	AWS
Aplicação de patches de sistema operacional	Cliente	Cliente	AWS
Instalação do sistema operacional	Cliente	Cliente	AWS

Recurso	Responsabilidade on-premises	Responsabilidade do Amazon EC2	Responsabilidade do Amazon RDS
Manutenção do servidor	Cliente	AWS	AWS
Ciclo de vida do hardware	Cliente	AWS	AWS
Energia, rede e desaquecimento	Cliente	AWS	AWS

Gerenciando o software de banco de dados por conta própria, você ganha mais controle, mas as chances de erros dos usuários também aumenta. Por exemplo, ao fazer alterações manualmente, você pode acidentalmente causar tempo de inatividade na aplicação. Você pode perder horas verificando todas as alterações para identificar e corrigir um problema. O ideal é ter um serviço de banco de dados gerenciado que automatize tarefas comuns do DBA, mas que também ofereça suporte ao acesso privilegiado ao banco de dados e ao sistema operacional subjacente.

Modelo de gerenciamento e benefícios para o Amazon RDS Custom

O Amazon RDS Custom é um serviço de banco de dados gerenciado para aplicações herdadas, personalizadas e em pacote que exigem acesso ao sistema operacional subjacente e ao ambiente de banco de dados. O Amazon RDS Custom automatiza as tarefas de configuração, operação e escalabilidade de bancos de dados na Nuvem AWS, ao mesmo tempo em que concede acesso ao banco de dados e ao sistema operacional subjacente. Com esse acesso, é possível definir configurações, instalar patches e habilitar recursos nativos para atender aos requisitos da aplicação dependente. Com o RDS Custom, a workload do banco de dados pode ser executada utilizando o AWS Management Console ou a AWS CLI.

No momento, o RDS Custom comporta somente os mecanismos de banco de dados Oracle Database e Microsoft SQL Server.

Tópicos

- Modelo de responsabilidade compartilhada no RDS Custom

- Perímetro de suporte e configurações sem suporte no RDS Custom
- Principais benefícios do RDS Custom

Modelo de responsabilidade compartilhada no RDS Custom

Com o RDS Custom, você usa os recursos gerenciados do Amazon RDS, mas gerencia o host e personaliza o sistema operacional da mesma forma que faz no Amazon EC2. Você assume responsabilidades adicionais pelo gerenciamento do banco de dados além do que é feito no Amazon RDS. O resultado é que você tem maior controle sobre o gerenciamento de bancos de dados e instâncias de banco de dados do que no Amazon RDS, além de se beneficiar da automação do RDS.

Responsabilidade compartilhada significa o seguinte:

1. Você detém parte do processo ao usar um recurso do RDS Custom.

   Por exemplo, no RDS Custom para Oracle, você controla quais patches de banco de dados Oracle deve usar e quando aplicá-los às instâncias de banco de dados.
2. Você é responsável por garantir que todas as personalizações dos recursos personalizados do RDS funcionem corretamente.

   Para ajudar na proteção contra a personalização inválida, o RDS Custom conta com um software de automação que é executado fora da instância de banco de dados. Se a instância do Amazon EC2 subjacente ficar comprometida, o RDS Custom tentará resolver esses problemas reiniciando ou substituindo a instância do EC2. A única alteração visível para o usuário é um novo endereço IP. Para obter mais informações, consulte Substituição do host do Amazon RDS Custom.

A tabela a seguir detalha o modelo de responsabilidade compartilhada para diferentes recursos do RDS Custom.

Recurso	Responsabilidade do Amazon EC2	Responsabilidade do Amazon RDS	Responsabilidade pelo RDS Custom for Oracle	Responsabilidade pelo RDS Custom for SQL Server
Otimização de aplicações	Cliente	Cliente	Cliente	Cliente

Recurso	Responsabilidade do Amazon EC2	Responsabilidade do Amazon RDS	Responsabilidade pelo RDS Custom for Oracle	Responsabilidade pelo RDS Custom for SQL Server
Escalabilidade	Cliente	AWS	Compartilhado	Compartilhado
Alta disponibilidade	Cliente	AWS	AWS	AWS
Backups de banco de dados	Cliente	AWS	Compartilhado	AWS
Aplicação de patches de softwares para banco de dados	Cliente	AWS	Compartilhado	AWS para RPEV, cliente para CEV[1]
Instalação de softwares para banco de dados	Cliente	AWS	Compartilhado	AWS para RPEV, cliente para CEV[1]
Aplicação de patches de sistema operacional	Cliente	AWS	Cliente	AWS para RPEV, cliente para CEV[1]
Instalação do sistema operacional	Cliente	AWS	Compartilhado	AWS
Manutenção do servidor	AWS	AWS	AWS	AWS
Ciclo de vida do hardware	AWS	AWS	AWS	AWS
Energia, rede e desaquecimento	AWS	AWS	AWS	AWS

[1] Uma versão de mecanismo personalizada (CEV) é um snapshot de volume binário de uma versão do banco de dados e de uma imagem de máquina da Amazon (AMI). Uma versão de mecanismo fornecida pelo RDS (RPEV) é a instalação da imagem de máquina da Amazon (AMI) padrão e do Microsoft SQL Server.

Você pode criar uma instância de banco de dados do RDS Custom utilizando o Microsoft SQL Server. Neste caso:

- É possível escolher entre dois modelos de licenciamento: Licença inclusa (LI) e Traga a sua própria mídia (BYOM).
- Com a LI, não é necessário adquirir licenças do SQL Server separadamente. A AWS mantém a licença para o software de banco de dados SQL Server.
- Com a BYOM, você fornece e instala seus próprios binários e licenças do Microsoft SQL Server.

Você pode criar uma instância de banco de dados do RDS Custom utilizando o Oracle Database. Nesse caso, você faz o seguinte:

- Gerencia sua própria mídia.

  Ao utilizar o RDS Custom, você carrega seus próprios arquivos e patches de instalação de banco de dados. Você cria uma versão de mecanismo personalizada (CEV) a partir desses arquivos. Em seguida, pode criar uma instância de banco de dados do RDS Custom utilizando essa CEV.
- Gerencia suas próprias licenças.

  Você traz suas próprias licenças do Oracle Database e as gerencia por conta própria.

Perímetro de suporte e configurações sem suporte no RDS Custom

O RDS Custom fornece um recurso de monitoramento denominado perímetro de suporte. Esse recurso garante que o ambiente de host e banco de dados sejam configurados corretamente. Se uma alteração sua fizer com que a instância de banco de dados saia do perímetro de suporte, o RDS Custom alterará o status da instância para `unsupported-configuration` até que você corrija manualmente os problemas de configuração. Para obter mais informações, consulte Perímetro de suporte do RDS Custom.

Principais benefícios do RDS Custom

Com o RDS Custom, é possível fazer o seguinte:

- Automatizar muitas das mesmas tarefas administrativas que o Amazon RDS, entre elas:
  - Gerenciamento do ciclo de vida de bancos de dados
  - Backups automatizados e recuperação em um ponto anterior no tempo (PITR)
  - Monitorar a integridade das instâncias de banco de dados do RDS Custom e observar as alterações na infraestrutura, no sistema operacional e nos processos do bancos de dados
  - Notificação ou medidas para corrigir problemas dependendo da interrupção na instância de banco de dados
- Instalar aplicações de terceiros.

  Você pode instalar softwares para executar aplicações e agentes personalizados. Como você tem acesso privilegiado ao host, pode modificar sistemas de arquivos para oferecer suporte a aplicações herdadas.
- Instalar patches personalizados.

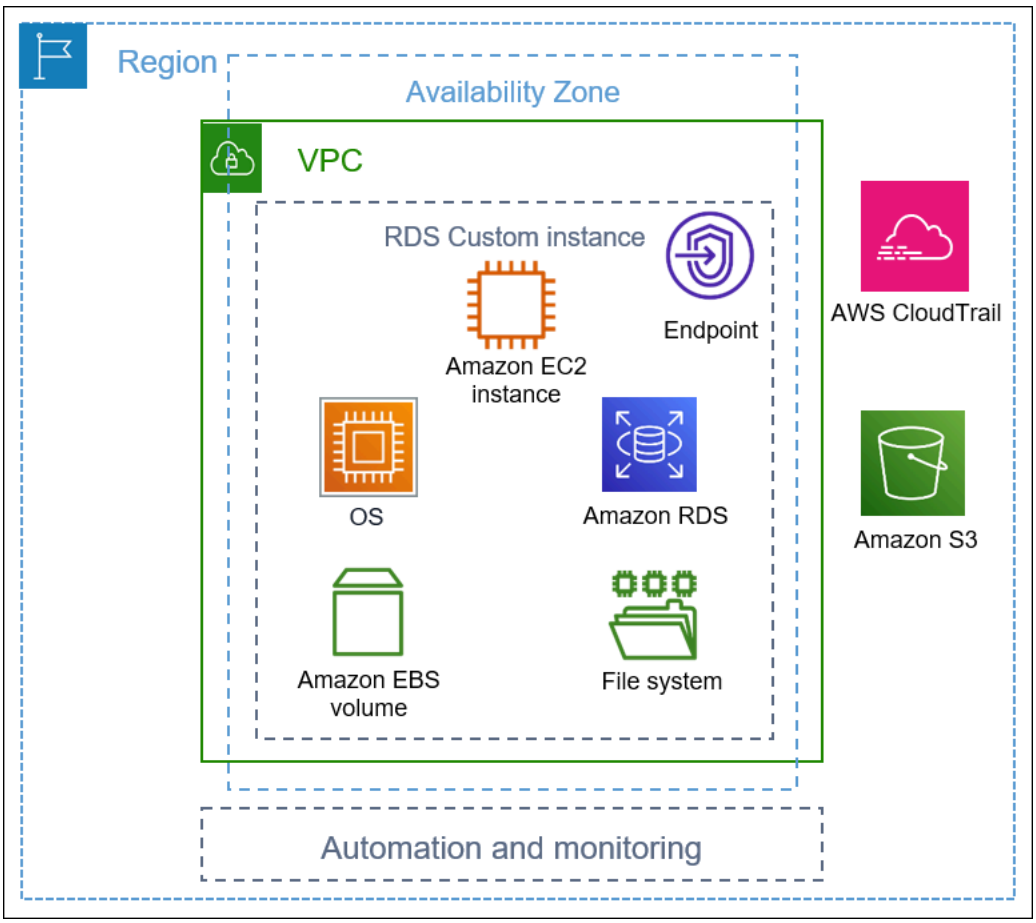
  Você pode aplicar patches de banco de dados personalizados ou modificar pacotes do SO nas suas instâncias de banco de dados do RDS Custom.
- Preparar um banco de dados local antes de movê-lo para um serviço totalmente gerenciado.

  Se você gerencia seu próprio banco de dados on-premises, pode prepará-lo para o RDS Custom no estado em que ele se encontra. Depois de se familiarizar com o ambiente de nuvem, você pode migrar seu banco de dados para uma instância de banco de dados do Amazon RDS totalmente gerenciada.
- Criar sua própria automação.

  Você pode criar, programar e executar scripts de automação personalizados para ferramentas de relatórios, gerenciamento ou diagnóstico.

Arquitetura do Amazon RDS Custom

A arquitetura do Amazon RDS Custom se baseia no Amazon RDS, com diferenças importantes. O diagrama a seguir mostra os principais componentes da arquitetura do RDS Custom.



Tópicos

- VPC
- Automação e monitoramento do RDS Custom
- Amazon S3
- AWS CloudTrail

VPC

Como no Amazon RDS, a instância de banco de dados do RDS Custom reside em uma nuvem privada virtual (VPC).



A instância de banco de dados do RDS Custom é formada pelos seguintes componentes principais:

- Instância do Amazon EC2
- Endpoint da instância
- Sistema operacional instalado na instância do Amazon EC2
- Armazenamento do Amazon EBS, que contém quaisquer sistemas de arquivos adicionais

Automação e monitoramento do RDS Custom

O RDS Custom tem um software de automação que é executado fora da instância de banco de dados. Esse software se comunica com agentes na instância de banco de dados e com outros componentes dentro do ambiente geral do RDS Custom.

Os recursos de monitoramento e recuperação do RDS Custom oferecem funcionalidade semelhante à do Amazon RDS. Por padrão, o RDS Custom está no modo de automação total. O software de automação tem as seguintes responsabilidades principais:

- Coletar métricas e enviar notificações
- Realizar a recuperação automática de instâncias

Uma responsabilidade importante da automação do RDS Custom é responder a problemas com a sua instância do Amazon EC2. Por vários motivos, o host pode ficar prejudicado ou inacessível. O RDS Custom resolve esses problemas reiniciando ou substituindo a instância do Amazon EC2.

Tópicos

- Substituição do host do Amazon RDS Custom
- Perímetro de suporte do RDS Custom

Substituição do host do Amazon RDS Custom

Se o host do Amazon EC2 for afetado, o RDS Custom tentará reinicializá-lo. Se esse esforço falhar, o RDS Custom utilizará o mesmo recurso de parada e início incluído no Amazon EC2. A única alteração visível para o cliente quando um host é substituído é um novo endereço IP público.

Tópicos

- Interromper e iniciar o host
- Efeitos da substituição do host
- Práticas recomendadas para o Amazon EC2

Interromper e iniciar o host

O RDS Custom executa automaticamente as seguintes etapas, sem a necessidade de intervenção do usuário:

1. Interrompe o host do Amazon EC2.

   A instância do EC2 realiza um desligamento normal e encerra a execução. Todos os volumes do Amazon EBS permanecem associados à instância, e seus dados persistem. Todos os dados armazenados nos volumes do armazenamento de instâncias (sem suporte no RDS Custom) ou na RAM do computador host desaparecem.

   Para obter mais informações, consulte Encerrar e iniciar sua instância no Guia do Usuário do Amazon EC2.
2. Inicia o host do Amazon EC2.

   A instância do EC2 é migrada para um novo hardware de host subjacente. Em alguns casos, a instância de banco de dados do RDS Custom permanece no host original.

Efeitos da substituição do host

No RDS Custom, você tem controle total sobre o volume do dispositivo raiz e os volumes de armazenamento do Amazon EBS. O volume raiz pode conter dados e configurações importantes que você não quer perder.

O RDS Custom for Oracle retém todos os dados do banco de dados e do cliente após a operação, incluindo os dados de volume raiz. Nenhuma intervenção do usuário é necessária. No RDS Custom para SQL Server, os dados do banco de dados são retidos, mas todos os dados na unidade `C:`, incluindo dados do sistema operacional e do cliente, são perdidos.

Após o processo de substituição de host, o host do Amazon EC2 tem um novo endereço IP público. O novo host retém os dados e os metadados mostrados na tabela a seguir.

Metadados ou dados	Retenção do RDS Custom para Oracle	Retenção do RDS Custom para SQL Server
IDs de instância EC2	Sim	Não
Metadados da instância do EC2	Sim	Sim
Dados de volumes de armazenamento de dados	Sim	Sim
Dados do volume raiz	Sim	Não
Endereços IP privados	Sim	Sim
Endereços IP elásticos	Sim	Sim

Práticas recomendadas para o Amazon EC2

O recurso de substituição de host do Amazon EC2 abrange a maioria dos cenários de comprometimento do Amazon EC2. Convém seguir estas práticas recomendadas:

- Antes de alterar a configuração ou o sistema operacional, faça backup dos seus dados. Se o volume raiz ou o sistema operacional ficar corrompido, a substituição do host não conseguirá repará-lo. Suas únicas opções são restaurar de um snapshot do banco de dados ou recuperação em um ponto anterior no tempo.

- Não pare ou encerre manualmente o host físico do Amazon EC2. Ambas as ações fazem com que a instância seja colocada fora do perímetro de suporte do RDS Custom.
- (RDS Custom para SQL Server) Se você anexar volumes adicionais ao host do Amazon EC2, configure-os para remontagem após a reinicialização. Se o host estiver afetado, o RDS Custom poderá pará-lo e iniciá-lo automaticamente.

Perímetro de suporte do RDS Custom

O RDS Custom fornece recurso de monitoramento adicional chamado de perímetro de suporte. Esse monitoramento adicional garante que a instância do RDS Custom utilize uma infraestrutura da AWS, um sistema operacional e um banco de dados compatíveis.

O perímetro de suporte verificará se a instância de banco de dados está em conformidade com os requisitos listados em Corrigir configurações não compatíveis no RDS Custom para Oracle e Corrigir configurações não compatíveis no RDS Custom para SQL Server. Se algum desses requisitos não for atendido, o RDS Custom considerará que a instância de banco de dados está fora do perímetro de suporte.

Tópicos

- Configurações não compatíveis no RDS Custom
- Solução de problemas de configurações não compatíveis

Configurações não compatíveis no RDS Custom

Quando a instância de banco de dados está fora do perímetro de suporte, o RDS Custom altera o status da respectiva instância para `unsupported-configuration` e envia notificações de eventos. Depois de corrigir os problemas de configuração, o RDS Custom alterará o status da instância de banco de dados para `available`.

Enquanto a instância de banco de dados está no estado `unsupported-configuration`, o seguinte se aplica:

- Seu banco de dados permanece acessível. Exceto quando a instância de banco de dados está na `unsupported-configuration` porque o banco de dados está encerrando inesperadamente.
- Não é possível modificá-la.
- Não é possível obter snapshots de banco de dados.

- Não são criados backups automáticos.
- Somente no caso de instâncias de banco de dados do RDS Custom para SQL Server, o RDS Custom não substituirá a instância subjacente do Amazon EC2 se ela ficar prejudicada. Para obter mais informações sobre substituição de host, consulte Substituição do host do Amazon RDS Custom.
- Você pode excluir a instância de banco de dados, mas a maioria das outras operações de API do RDS Custom não está disponível.
- O RDS Custom continua comportando a recuperação para um ponto no tempo (PITR) ao arquivar arquivos de redo log e enviá-los ao Amazon S3. A PITR em um estado `unsupported-configuration` apresenta as seguintes diferenças:
  - A PITR pode demorar muito tempo para restaurar completamente para uma nova instância de banco de dados do RDS Custom. Essa situação ocorre porque não é possível obter snapshots automatizados ou manuais enquanto a instância está no estado `unsupported-configuration`.
  - A PITR precisa reproduzir mais logs de redo a partir do snapshot mais recente obtido antes da entrada da instância no estado `unsupported-configuration`.
  - Em alguns casos, a instância de banco de dados está no estado`unsupported-configuration` porque você fez uma alteração que impediu o carregamento de arquivos de redo log arquivados. Exemplos incluem interromper a instância do EC2, interromper o agente do RDS Custom e desanexar volumes do EBS. Nesses casos, a PITR não pode restaurar a instância de banco de dados para o último momento restaurável.

Solução de problemas de configurações não compatíveis

O RDS Custom fornece orientações para solução de problemas referentes ao estado `unsupported-configuration`. Embora algumas orientações se apliquem ao RDS Custom para Oracle e ao RDS Custom para SQL Server, outras orientações dependem do seu mecanismo de banco de dados. Para obter informações de solução de problemas específicas ao mecanismo, consulte os seguintes tópicos:

- Corrigir configurações não compatíveis no RDS Custom para Oracle
- Corrigir configurações não compatíveis no RDS Custom para SQL Server

Amazon S3

Se você usar o RDS Custom for Oracle, carregará a mídia de instalação para um bucket do Amazon S3 criado pelo usuário. O RDS Custom for Oracle usa a mídia neste bucket para criar uma versão de mecanismo personalizada (CEV). Uma CEV é um snapshot de volume binário de uma versão do banco de dados e de uma imagem de máquina da Amazon (AMI). A partir da CEV, você cria uma instância de banco de dados do RDS Custom. Para obter mais informações, consulte Trabalhar com versões de mecanismo personalizadas para o Amazon RDS Custom for Oracle.

Tanto para o RDS Custom for Oracle quanto para o RDS Custom for SQL Server, o RDS Custom cria automaticamente um bucket do Amazon S3 prefixado com a string `do-not-delete-rds-custom-`. O RDS Custom usa o bucket do S3 do `do-not-delete-rds-custom-` para armazenar os seguintes tipos de arquivos:

- Logs do AWS CloudTrail para a trilha criada pelo RDS Custom
- Artefatos de perímetro de suporte (consulte Perímetro de suporte do RDS Custom)
- Arquivos de redo log de banco de dados (somente no RDS Custom for Oracle)
- Logs de transações (somente no RDS Custom for SQL Server)
- Artefatos de versão do mecanismo personalizado (somente no RDS Custom for Oracle)

O RDS Custom cria o bucket do S3 do `do-not-delete-rds-custom-` quando você cria um dos seguintes recursos:

- Seu primeiro CEV para RDS Custom for Oracle
- Sua primeira instância de banco de dados para o RDS Custom for SQL Server

O RDS Custom cria um bucket para cada combinação dos seguintes tipos:

- ID da Conta da AWS
- Tipo de mecanismo (RDS Custom for Oracle ou RDS Custom for SQL Server)
- Região da AWS

Por exemplo, se você criar CEVs do RDS Custom for Oracle em um único Região da AWS, existirá um bucket do `do-not-delete-rds-custom-`. Se você criar várias instâncias do RDS Custom for SQL Server e elas residirem em um Regiões da AWS diferente, existirá um bucket do `do-not-delete-rds-custom-` em cada Região da AWS. Se você criar uma instância do RDS Custom for

Oracle e duas instâncias do RDS Custom for SQL Server em uma única Região da AWS, existirão dois buckets do `do-not-delete-rds-custom-`.

AWS CloudTrail

O RDS Custom cria automaticamente uma trilha AWS CloudTrail cujo nome começa com `do-not-delete-rds-custom-`. O perímetro de suporte do RDS Custom depende dos eventos do CloudTrail para determinar se suas ações afetam a automação do RDS Custom. Para obter mais informações, consulte Solução de problemas de configurações não compatíveis.

O RDS Custom cria a trilha quando você cria sua primeira instância de banco de dados. O RDS Custom cria uma trilha para cada combinação do seguinte:

- ID da Conta da AWS
- Tipo de mecanismo (RDS Custom for Oracle ou RDS Custom for SQL Server)
- Região da AWS

Quando você exclui uma instância de banco de dados do RDS Custom, o CloudTrail dessa instância não é removido automaticamente. Nesse caso, a Conta da AWS continua sendo cobrada pelo CloudTrail não excluído. O RDS Custom não é responsável pela exclusão desse recurso. Para saber como remover o CloudTrail manualmente, consulte Excluir uma trilha no Guia do usuário do AWS CloudTrail.

Segurança no Amazon RDS Custom

Familiarize-se com as considerações de segurança do RDS Custom.

Consulte mais informações sobre segurança do RDS Custom nos tópicos a seguir.

- Proteger o bucket do Amazon S3 contra o problema do substituto confuso
- Alternar as credenciais do RDS Custom para Oracle para programas de conformidade

Como o RDS Custom gerencia com segurança as tarefas em seu nome

O RDS Custom utiliza as seguintes ferramentas e técnicas para executar operações com segurança em seu nome:

Perfil vinculado a serviço AWSServiceRoleForRDSCustom

Um perfil vinculado a serviço é predefinido pelo serviço e inclui todas as permissões que o serviço precisa para chamar outros Serviços da AWS em seu nome. Para o RDS Custom, `AWSServiceRoleForRDSCustom` é um perfil vinculado a serviço que é definido de acordo com o princípio do privilégio mínimo. O RDS Custom usa as permissões em `AmazonRDSCustomServiceRolePolicy`, que é a política associada a esse perfil, para realizar a maioria das tarefas de provisionamento e todas as tarefas de gerenciamento fora do host. Para obter mais informações, consulte AmazonRDSCustomServiceRolePolicy.

Ao executar tarefas no host, a automação do RDS Custom usa credenciais do perfil vinculado a serviço para executar comandos usando o AWS Systems Manager. Você pode auditar o histórico de comandos por meio do histórico de comandos do Systems Manager e pelo AWS CloudTrail. O Systems Manager se conecta à sua instância de banco de dados do RDS Custom usando sua configuração de rede. Para ter mais informações, consulte Etapa 4: Configurar o IAM para RDS Custom para Oracle.

Credenciais temporárias do IAM

Ao provisionar ou excluir recursos, o RDS Custom às vezes utiliza credenciais temporárias derivadas das credenciais da entidade principal do IAM que realiza a chamada. Essas credenciais do IAM são restringidas pelas políticas do IAM anexadas a essa entidade principal e expiram após a conclusão da operação. Para saber mais sobre as permissões necessárias para entidades principais do IAM que usam o RDS Custom, consulte Etapa 5: Conceder as permissões necessárias ao usuário ou ao perfil do IAM.

Perfil de instância do Amazon EC2

Um perfil de instância do EC2 é um contêiner para um perfil do IAM que pode ser usado para transmitir as informações do perfil para uma instância do EC2. Uma instância do EC2 é a base de uma instância de banco de dados do RDS Custom. Você fornece um perfil de instância ao criar uma instância de banco de dados do RDS Custom. O RDS Custom usa as credenciais do perfil de instância do EC2 ao executar tarefas de gerenciamento baseadas em host, como backups. Para ter mais informações, consulte Criar seu perfil do IAM e perfil de instância manualmente.

Par de chaves SSH

Quando o RDS Custom cria a instância do EC2 como base de uma instância de banco de dados, ele cria um par de chaves SSH em seu nome. A chave utiliza o prefixo de nome `do-not-delete-rds-custom-ssh-privatekey-db-` ou `rds-custom!oracle-do-not-delete-`*`db_resource_id-uuid`*`-ssh-privatekey`. O AWS Secrets Manager armazena essa chave privada SSH como um segredo em sua Conta da AWS. O Amazon RDS não armazena, nem acessa, nem usa essas credenciais. Para obter mais informações, consulte Pares de chaves do Amazon EC2 e instâncias do Linux.

Certificados SSL

As instâncias de banco de dados personalizadas do RDS não comportam certificados SSL gerenciados. Se quiser implantar o SSL, você pode autogerenciar certificados SSL em sua própria carteira e criar um receptor SSL para proteger as conexões entre o banco de dados do cliente ou para a replicação do banco de dados. Para obter mais informações, consulte Configuring Transport Layer Security Authentication na documentação do Oracle Database.

Proteger o bucket do Amazon S3 contra o problema do substituto confuso

Quando você cria uma versão de mecanismo personalizada (CEV) do Amazon RDS Custom para Oracle ou uma instância de banco de dados do RDS Custom para SQL Server, o RDS Custom cria um bucket do Amazon S3. O bucket do S3 armazena arquivos como artefatos de CEV, logs de redo (transação), itens de configuração para o perímetro de suporte e logs do AWS CloudTrail.

Você pode tornar esses buckets do S3 mais seguros usando as chaves de contexto de condição globais para evitar o problema de representante confuso. Para ter mais informações, consulte Prevenção do problema do substituto confuso entre serviços.

O exemplo do RDS Custom para Oracle a seguir mostra o uso das chaves de contexto de condição globais `aws:SourceArn` e `aws:SourceAccount` em uma política de bucket do S3. Para o RDS

Custom para Oracle, sempre inclua os nomes de recurso da Amazon (ARNs) das CEVs e das instâncias de banco de dados. Para o RDS Custom para SQL Server, inclua o ARN das instâncias de banco de dados.

```
...
{
  "Sid": "AWSRDSCustomForOracleInstancesObjectLevelAccess",
  "Effect": "Allow",
  "Principal": {
     "Service": "custom.rds.amazonaws.com"
  },
  "Action": [
     "s3:GetObject",
     "s3:GetObjectVersion",
     "s3:DeleteObject",
     "s3:DeleteObjectVersion",
     "s3:GetObjectRetention",
     "s3:BypassGovernanceRetention"
  ],
  "Resource": "arn:aws:s3:::do-not-delete-rds-custom-123456789012-us-east-2-c8a6f7/
RDSCustomForOracle/Instances/*",
  "Condition": {
     "ArnLike": {
        "aws:SourceArn": [
            "arn:aws:rds:us-east-2:123456789012:db:*",
            "arn:aws:rds:us-east-2:123456789012:cev:*/*"
        ]
     },
     "StringEquals": {
        "aws:SourceAccount": "123456789012"
    }
  }
},
...
```

Alternar as credenciais do RDS Custom para Oracle para programas de conformidade

Alguns programas de conformidade exigem que as credenciais do usuário do banco de dados sejam alteradas periodicamente, por exemplo, a cada 90 dias. O RDS Custom para Oracle alterna automaticamente as credenciais de alguns usuários predefinidos do banco de dados.

Tópicos

- Alternância automática de credenciais para usuários predefinidos
- Diretrizes para alternância de credenciais de usuário
- Alternar manualmente as credenciais do usuário

Alternância automática de credenciais para usuários predefinidos

Se sua instância de banco de dados do RDS Custom para Oracle estiver hospedada no Amazon RDS, as credenciais dos seguintes usuários predefinidos da Oracle serão alternadas automaticamente a cada 30 dias. As credenciais dos usuários anteriores residem em AWS Secrets Manager.

Usuário do banco de dados	Criado por	Versões compatíveis do mecanismo	Observações
`SYS`	Oracle	custom-oracle-ee custom-oracle-ee-cdb custom-oracle-se2 custom-oracle-se2-cdb	
`SYSTEM`	Oracle	custom-oracle-ee custom-oracle-ee-cdb custom-oracle-se2 custom-oracle-se2-cdb	
`RDSADMIN`	RDS	custom-oracle-ee custom-oracle-se2	
`C##RDSADMIN`	RDS	custom-oracle-ee-cdb custom-oracle-se2-cdb	Nomes de usuário com um prefixo `C##` só existem em CDBs. Para ter mais informaçõ

Usuário do banco de dados	Criado por	Versões compatíveis do mecanismo	Observações
			es sobre CDBs, consulte Visão geral da arquitetura do Amazon RDS Custom para Oracle.
`RDS_DATAGUARD`	RDS	custom-oracle-ee	Esse usuário existe somente em réplicas de leitura, bancos de dados de origem para réplicas de leitura e bancos de dados que você migrou fisicamente para o RDS Custom usando o Oracle Data Guard.
`C##RDS_DATAGUARD`	RDS	custom-oracle-ee-cdb	Esse usuário existe somente em réplicas de leitura, bancos de dados de origem para réplicas de leitura e bancos de dados que você migrou fisicamente para o RDS Custom usando o Oracle Data Guard. Nomes de usuário com um prefixo `C##` só existem em CDBs. Para ter mais informações sobre CDBs, consulte Visão geral da arquitetura do Amazon RDS Custom para Oracle.

Uma exceção à alternância automática de credenciais é uma instância de banco de dados do RDS Custom para Oracle que você configurou manualmente como um banco de dados de espera. O RDS alterna somente as credenciais para réplicas de leitura que você criou usando o comando `create-db-instance-read-replica` DA CLI ou a API `CreateDBInstanceReadReplica`.

Diretrizes para alternância de credenciais de usuário

Para garantir que suas credenciais sejam alternadas de acordo com seu programa de conformidade, observe as seguintes diretrizes:

- Se sua instância de banco de dados alternar as credenciais automaticamente, não altere nem exclua manualmente um segredo, um arquivo de senha ou uma senha para usuários listados em Usuários predefinidos da Oracle. Caso contrário, o RDS Custom pode colocar sua instância de banco de dados fora do perímetro de suporte, o que suspende a alternância automática.
- O usuário principal do RDS não é predefinido, então você é responsável por alterar a senha manualmente ou configurar a alternância automática no Secrets Manager. Para ter mais informações, consulte Alternar segredos do AWS Secrets Manager.

Alternar manualmente as credenciais do usuário

Para as seguintes categorias de bancos de dados, o RDS não alterna automaticamente as credenciais dos usuários listados em Usuários predefinidos da Oracle:

- Um banco de dados que você configurou manualmente para funcionar como um banco de dados de espera.
- Um banco de dados on-premises.
- Uma instância de banco de dados que está fora do perímetro de suporte ou em um estado em que a automação do RDS Custom não pode ser executada. Nesse caso, o RDS Custom também não alterna as chaves.

Se seu banco de dados estiver em qualquer uma das categorias anteriores, você deverá alternar suas credenciais de usuário manualmente.

Como alternar as credenciais do usuário manualmente para uma instância de banco de dados

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. Em Bancos de dados, garanta que o RDS não esteja fazendo backup de sua instância de banco de dados nem realizando operações como configuração de alta disponibilidade.
3. Na página de detalhes do banco de dados, selecione Configuração e anote o ID do recurso para a instância de banco de dados. Você pode usar o comando `describe-db-instances` da AWS CLI.

4. Abra o console do Secrets Manager em https://console.aws.amazon.com/secretsmanager/.
5. Na caixa de pesquisa, insira o ID do recurso do seu banco de dados e pesquise um segredo usando uma das seguintes convenções de nomenclatura:

```
do-not-delete-rds-custom-resource_id-uuid
rds-custom!oracle-do-not-delete-resource_id-uuid
```

Esse segredo armazena a senha para RDSADMIN, SYS e SYSTEM. As seguintes chaves de exemplo são para a instância de banco de dados com o ID de recurso db-ABCDEFG12HIJKLNMNOPQRS3TUVWX e o UUID 123456:

```
do-not-delete-rds-custom-db-ABCDEFG12HIJKLNMNOPQRS3TUVWX-123456
rds-custom!oracle-do-not-delete-db-ABCDEFG12HIJKLNMNOPQRS3TUVWX-123456
```

> **Important**
>
> Se sua instância de banco de dados for uma réplica de leitura e usar o mecanismo custom-oracle-ee-cdb, haverá dois segredos com o sufixo *db-resource_id-uuid*, um para o usuário principal e outro para RDSADMIN, SYS e SYSTEM. Para encontrar o segredo correto, execute o seguinte comando no host:
>
> ```
> cat /opt/aws/rdscustomagent/config/database_metadata.json | python3 -c
>  "import sys,json; print(json.load(sys.stdin)['dbMonitoringUserPassword'])"
> ```
>
> O atributo dbMonitoringUserPassword indica o segredo para RDSADMIN, SYS e SYSTEM.

6. Se sua instância de banco de dados estiver em uma configuração do Oracle Data Guard, pesquise um segredo usando uma das seguintes convenções de nomenclatura:

```
do-not-delete-rds-custom-resource_id-uuid-dg
rds-custom!oracle-do-not-delete-resource_id-uuid-dg
```

Esse segredo armazena a senha para RDS_DATAGUARD. As seguintes chaves de exemplo são para a instância de banco de dados com o ID de recurso db-ABCDEFG12HIJKLNMNOPQRS3TUVWX e o UUID 789012:

```
do-not-delete-rds-custom-db-ABCDEFG12HIJKLNMNOPQRS3TUVWX-789012-dg
rds-custom!oracle-do-not-delete-db-ABCDEFG12HIJKLNMNOPQRS3TUVWX-789012-dg
```

7. Para todos os usuários do banco de dados listados em Usuários predefinidos da Oracle, atualize as senhas seguindo as instruções em Modificar um segredo do AWS Secrets Manager.
8. Se seu banco de dados for um banco de dados independente ou de origem em uma configuração do Oracle Data Guard:

   a. Inicie seu cliente Oracle SQL e faça login como SYS.

   b. Execute uma instrução SQL no seguinte formato para cada usuário do banco de dados listado em Usuários Oracle predefinidos:

   ```
   ALTER USER user-name IDENTIFIED BY pwd-from-secrets-manager ACCOUNT UNLOCK;
   ```

   Por exemplo, se a nova senha do RDSADMIN armazenada no Secrets Manager for pwd-123, execute a seguinte instrução:

   ```
   ALTER USER RDSADMIN IDENTIFIED BY pwd-123 ACCOUNT UNLOCK;
   ```

9. Se sua instância de banco de dados executa o Oracle Database 12c Release 1 (12.1) e é gerenciada pelo Oracle Data Guard, copie manualmente o arquivo de senha (orapw) da instância de banco de dados primária para cada instância de banco de dados de espera.

   Se sua instância de banco de dados estiver hospedada no Amazon RDS, o local do arquivo de senha será /rdsdbdata/config/orapw. Para bancos de dados que não estão hospedados no Amazon RDS, o local padrão é $ORACLE_HOME/dbs/orapw$ORACLE_SID no Linux e no UNIX e %ORACLE_HOME%\database\PWD%ORACLE_SID%.ora no Windows.

Trabalhar com o RDS Custom for Oracle

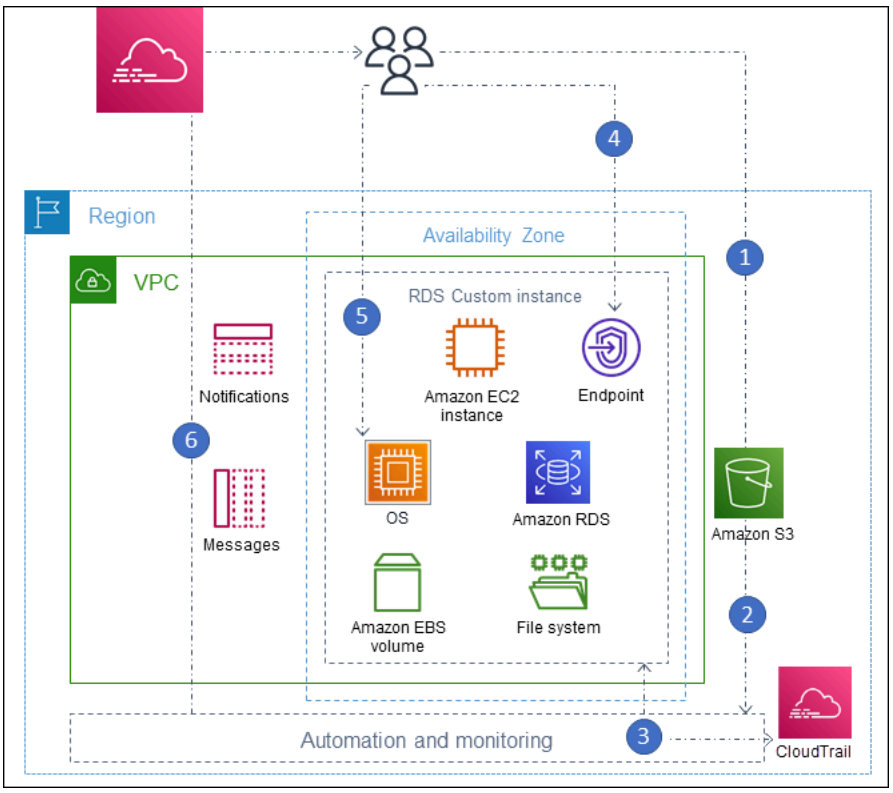
Veja a seguir instruções para criar, gerenciar e manter as instâncias de banco de dados do RDS Custom for Oracle.

Tópicos

- Fluxo de trabalho do RDS Custom for Oracle
- Arquitetura de banco de dados para o Amazon RDS Custom para Oracle
- Disponibilidade de recursos e suporte do RDS Custom para Oracle
- Requisitos e limitações do RDS Custom for Oracle
- Configurar seu ambiente para o Amazon RDS Custom for Oracle
- Trabalhar com versões de mecanismo personalizadas para o Amazon RDS Custom for Oracle
- Configurar uma instância de banco de dados para o Amazon RDS Custom para Oracle
- Gerenciar uma instância de banco de dados do Amazon RDS Custom for Oracle
- Gerenciar uma implantação multi-AZ para o RDS Custom para Oracle
- Trabalhar com réplicas do Oracle para o RDS Custom para Oracle
- Fazer o backup e a restauração de uma instância de banco de dados do Amazon RDS Custom for Oracle
- Trabalhar com grupos de opções no RDS Custom para Oracle
- Migrar um banco de dados on-premises para o RDS Custom para Oracle
- Fazer upgrade de uma instância de banco de dados para o Amazon RDS Custom for Oracle
- Solucionar problemas de banco de dados do Amazon RDS Custom para Oracle
- Problemas conhecidos do Amazon RDS Custom para Oracle

Fluxo de trabalho do RDS Custom for Oracle

O diagrama a seguir mostra o fluxo de trabalho típico do RDS Custom for Oracle.



As etapas são:

1. Carregar o software do banco de dados no bucket do Amazon S3.

   Para ter mais informações, consulte Etapa 3: Fazer upload de seus arquivos de instalação no Amazon S3.

2. Crie uma versão de mecanismo personalizada (CEV) do RDS Custom para Oracle a partir de sua mídia.

   Escolha a arquitetura CBD ou a arquitetura tradicional que não seja de CDB. Para obter mais informações, consulte Criar um CEV.

3. Crie uma instância de banco de dados do RDS Custom para Oracle a partir de uma CEV.

   Para ter mais informações, consulte Criar uma instância de banco de dados do RDS Custom for Oracle.

4. Conectar sua aplicação ao endpoint da instância de banco de dados.

   Para ter mais informações, consulte Conectar-se à sua instância de banco de dados do RDS Custom via SSH e Conectar-se à sua instância de banco de dados do RDS Custom usando o Session Manager.

5. (Opcional) Acessar o host para personalizar o software.
6. Monitore notificações e mensagens geradas pela automação do RDS Custom.

Arquivos de instalação do banco de dados

Sua responsabilidade pela mídia é uma diferença fundamental entre o Amazon RDS e o RDS Custom. O Amazon RDS, que é um serviço totalmente gerenciado, fornece a imagem de máquina da Amazon (AMI) e o software de banco de dados. Como o software de banco de dados do Amazon RDS é pré-instalado, você precisa apenas escolher um mecanismo de banco de dados e uma versão e criar seu banco de dados.

Para o RDS Custom, você fornece sua própria mídia. Quando você cria uma versão de mecanismo personalizada, o RDS Custom instala essa mídia fornecida. A mídia do RDS Custom contém arquivos e patches de instalação do banco de dados. Esse modelo de serviço é chamado de Bring Your Own Media (BYOM - Traga sua própria mídia).

Trabalhar com versões de mecanismo personalizadas para o RDS Custom for Oracle

Uma Versão de mecanismo personalizada (CEV) do RDS Custom for Oracle é um snapshot de volume binário de uma versão de banco de dados e AMI. Por padrão, o RDS Custom para Oracle usa a AMI mais recente disponibilizada pelo Amazon EC2. Você também pode optar por reutilizar uma IAM existente.

Manifesto de CEV

Depois de baixar os arquivos de instalação do banco de dados Oracle no site da Oracle, faça upload deles em um bucket do Amazon S3. Ao criar a CEV, especifique os nomes de arquivo em um documento JSON chamado Manifesto de CEV. O RDS Custom para Oracle usa os arquivos especificados e a AMI para criar a CEV.

O RDS Custom para Oracle fornece modelos de manifesto JSON com nossos arquivos .zip recomendados para cada versão compatível do Oracle Database. Por exemplo, o modelo a seguir é para a RU 19.17.0.0.0.

```
{
    "mediaImportTemplateVersion": "2020-08-14",
    "databaseInstallationFileNames": [
        "V982063-01.zip"
    ],
```

```
    "opatchFileNames": [
        "p6880880_190000_Linux-x86-64.zip"
    ],
    "psuRuPatchFileNames": [
        "p34419443_190000_Linux-x86-64.zip",
        "p34411846_190000_Linux-x86-64.zip"
    ],
    "otherPatchFileNames": [
        "p28852325_190000_Linux-x86-64.zip",
        "p29997937_190000_Linux-x86-64.zip",
        "p31335037_190000_Linux-x86-64.zip",
        "p32327201_190000_Linux-x86-64.zip",
        "p33613829_190000_Linux-x86-64.zip",
        "p34006614_190000_Linux-x86-64.zip",
        "p34533061_190000_Linux-x86-64.zip",
        "p34533150_190000_Generic.zip",
        "p28730253_190000_Linux-x86-64.zip",
        "p29213893_1917000DBRU_Generic.zip",
        "p33125873_1917000DBRU_Linux-x86-64.zip",
        "p34446152_1917000DBRU_Linux-x86-64.zip"
    ]
}
```

Você também pode especificar parâmetros de instalação no manifesto JSON. Por exemplo, é possível definir valores não padrão para a base do Oracle, o Oracle Home e o ID e o nome do usuário e do grupo do UNIX/Linux. Para obter mais informações, consulte Campos JSON no manifesto de CEV.

Formato de nomenclatura de CEV

Dê um nome para sua CEV utilizando uma string especificada pelo cliente. O formato do nome é o seguinte, dependendo da versão do Oracle Database:

- `19.customized_string`
- `18.customized_string`
- `12.2.customized_string`
- `12.1.customized_string`

Você pode usar de 1 a 50 caracteres alfanuméricos, sublinhados, travessões e pontos. Por exemplo, sua CEV pode se chamar `19.my_cev1`.

Arquitetura multilocatário da Oracle no RDS Custom for Oracle

A arquitetura multilocatário da Oracle permite que um banco de dados Oracle funcione como um banco de dados de contêiner (CDB). Um CDB inclui zero, um ou vários bancos de dados conectáveis (PDBs) criados pelo cliente. Um PDB é uma coleção portátil de esquemas e objetos que aparece para uma aplicação como não CDB tradicional.

Ao criar um RDS Custom para Oracle CEV, especifique a arquitetura CDB ou não CDB. Você pode criar um CDB do RDS Custom para Oracle somente quando a CEV usada para criá-lo usa a arquitetura multilocatário da Oracle. Para obter mais informações, consulte Trabalhar com versões de mecanismo personalizadas para o Amazon RDS Custom for Oracle.

Criar uma instância de banco de dados do RDS Custom for Oracle

Após a criação da CEV, ela estará disponível para uso. É possível criar várias CEVs e várias instâncias de banco de dados do RDS Custom para Oracle a partir de qualquer CEV. Também é possível alterar o status de uma CEV para torná-la disponível ou inativa.

É possível criar sua instância de banco de dados do RDS Custom para Oracle com a arquitetura de multilocação Oracle (tipo de mecanismo `custom-oracle-ee-cdb` ou `custom-oracle-se2-cdb`) ou com a arquitetura tradicional que não é de CDB (tipo de mecanismo `custom-oracle-ee` ou `custom-oracle-se2`). Quando você cria um banco de dados de contêiner (CDB), ele contém um banco de dados conectável (PDB) e uma semente de PDB. Você pode criar PDBs adicionais manualmente utilizando o Oracle SQL.

Para criar sua instância de banco de dados do RDS Custom for Oracle, utilize o comando `create-db-instance`. Nesse comando, especifique qual CEV usar. O procedimento é semelhante à criação de uma instância de banco de dados do Amazon RDS. No entanto, alguns parâmetros são diferentes. Para ter mais informações, consulte Configurar uma instância de banco de dados para o Amazon RDS Custom para Oracle.

Conexão de banco de dados

Como em uma instância de banco de dados do Amazon RDS, uma instância de banco de dados do RDS Custom reside em uma nuvem privada virtual (VPC). Sua aplicação se conecta ao banco de dados Oracle utilizando um receptor do Oracle.

Se seu banco de dados for um CDB, você poderá usar o receptor `L_RDSCDB_001` para se conectar à raiz do CDB e a um PDB. Se você conectar um não CDB a um CDB, configure

`USE_SID_AS_SERVICE_LISTENER = ON` para que as aplicações migradas mantenham as mesmas configurações.

Quando você se conecta a um não CDB, o usuário principal é o usuário do não CDB. Quando você se conecta a um CDB, o usuário principal é o usuário do CDB. Para se conectar à raiz do CDB, faça login no host, inicie um cliente SQL e crie um usuário administrativo com comandos SQL.

Personalização do RDS Custom

É possível acessar o host do RDS Custom para instalar ou personalizar softwares. Para evitar conflitos entre suas alterações e a automação do RDS Custom, é possível pausar a automação por um período especificado. Durante esse período, o RDS Custom não realiza o monitoramento ou a recuperação de instâncias. No final do período, o RDS Custom retomará a automação completa. Para ter mais informações, consulte Pausar e retomar sua instância de banco de dados do RDS Custom.

Arquitetura de banco de dados para o Amazon RDS Custom para Oracle

O RDS Custom para Oracle oferece suporte às arquiteturas multilocatário e não multilocatário.

Tópicos

- Arquiteturas de banco de dados Oracle compatíveis
- Tipos de mecanismo compatíveis
- Recursos compatíveis com a arquitetura de multilocação da Oracle

Arquiteturas de banco de dados Oracle compatíveis

A arquitetura multilocatário, também chamada de arquitetura CDB, permite que um banco de dados Oracle funcione como um banco de dados de contêiner (CDB). Um CDB inclui bancos de dados conectáveis (PDBs). Um PDB é uma coleção portátil de esquemas e objetos que aparece para uma aplicação como um banco de dados Oracle tradicional. Para obter mais informações, consulte Introduction to the Multitenant Architecture no Oracle Multitenant Administrator's Guide.

As arquiteturas CDB e não CDB são mutuamente exclusivas. Se um banco de dados não for CDB, será não CDB, portanto não poderá conter outros bancos de dados. No RDS Custom para Oracle, somente o Oracle Database 19c é compatível com a arquitetura multilocatário. Assim, se você criar instâncias usando versões anteriores do banco de dados, só poderá criar não CDBs. Para ter mais informações, consulte Considerações sobre arquitetura multilocatário.

Tipos de mecanismo compatíveis

Ao criar uma CEV ou uma instância de banco de dados do Amazon RDS Custom para Oracle, selecione um tipo de mecanismo de CDB ou não CDB:

- `custom-oracle-ee-cdb` e `custom-oracle-se2-cdb`

  Esses tipos de mecanismo especificam a arquitetura de multilocação da Oracle. Essa opção só está disponível para Oracle Database 19c. Quando você cria uma instância de banco de dados do RDS para Oracle usando a arquitetura multilocatário, o CDB inclui os seguintes contêineres:

  - Raiz do CDB (`CDB$ROOT`)
  - Semente do PDB (`PDB$SEED`)
  - PDB inicial

Você pode criar mais PDBs usando o comando `CREATE PLUGGABLE DATABASE` do Oracle SQL. Não é possível usar as APIs do RDS para criar ou excluir PDBs.

- `custom-oracle-ee` e `custom-oracle-se2`

  Esses tipos de mecanismo especificam a arquitetura tradicional que não é CDB. Um não CDB não pode conter bancos de dados conectáveis (PDBs).

Para ter mais informações, consulte Considerações sobre arquitetura multilocatário.

Recursos compatíveis com a arquitetura de multilocação da Oracle

Uma instância de CDB do RDS Custom para Oracle é compatível com os seguintes atributos:

- Backups
- Restauração e restauração pontual (PITR) a partir de backups
- Réplicas de leitura
- Atualizações de versões secundárias

Disponibilidade de recursos e suporte do RDS Custom para Oracle

Neste tópico, é possível encontrar um resumo do suporte e da disponibilidade dos recursos do RDS Custom para Oracle para referência rápida.

Tópicos

- Suporte a versões de banco de dados e Região da AWS do RDS Custom para Oracle
- Suporte a versões de banco de dados do RDS Custom para Oracle
- Suporte a edições e licenciamento do RDS Custom para Oracle
- Suporte a classes de instâncias de banco de dados do RDS Custom for Oracle
- Compatibilidade com grupos de opções do RDS Custom para Oracle

Suporte a versões de banco de dados e Região da AWS do RDS Custom para Oracle

A disponibilidade e a compatibilidade de recursos variam entre versões específicas de cada mecanismo de banco de dados e entre Regiões da AWS. Para obter mais informações sobre a disponibilidade de versões e regiões do RDS Custom para Oracle, consulte Regiões e mecanismos de banco de dados compatíveis com o RDS Custom.

Suporte a versões de banco de dados do RDS Custom para Oracle

O RDS para Oracle é compatível com as seguintes versões do banco de dados Oracle:

- Oracle Database 19c
- Oracle Database 18c
- Oracle Database 12c Versão 2 (12.2)
- Oracle Database 12c Versão 1 (12.1)

Suporte a edições e licenciamento do RDS Custom para Oracle

O RDS Custom para Oracle é compatível com a Enterprise Edition (EE) e a Standard Edition 2 (SE2) no modelo BYOL.

Observe as seguintes limitações da Standard Edition 2:

- O Oracle Data Guard não é compatível. Assim, não é possível criar réplicas de leitura da Oracle.

- Você só pode usar classes de instância de banco de dados que tenham até 16 vCPUs (até 4xlarge).
- Uma instância de CDB na Standard Edition 2 comporta até três bancos de dados de locatários.
- Não é possível migrar dados entre a Enterprise Edition e a Standard Edition 2.

Suporte a classes de instâncias de banco de dados do RDS Custom for Oracle

O RDS Custom para Oracle é compatível com as classes de instância de banco de dados a seguir. Se você criar uma instância de banco de dados na Standard Edition 2, só poderá usar classes de instância com até 16 vCPUs (até 4x large).

Tipo	Tamanho
db.r7i	db.r7i.large |db.r7i.xlarge |db.r7i.2xlarge |db.r7i.4xlarge |db.r7i.8xlarge |db.r7i.12xlarge |db.r7i.16xlarge |db.r7i.24xlarge |db.r7i.48xlarge
db.r6i	db.r6i.large |db.r6i.xlarge |db.r6i.2xlarge |db.r6i.4xlarge |db.r6i.8xlarge |db.r6i.12xlarge |db.r6i.16xlarge |db.r6i.24xlarge |db.r6i.32xlarge
db.r5b	db.r5b.large |db.r5b.xlarge |db.r5b.2xlarge |db.r5b.4xlarge |db.r5b.8xlarge |db.r5b.12xlarge |db.r5b.16xlarge |db.r5b.24xlarge
db.r5	db.r5.large |db.r5.xlarge |db.r5.2xlarge |db.r5.4xlarge | db.r5.8xlarge |db.r5.12xlarge |db.r5.16xlarge |db.r5.24xlarge
db.x2iec	db.x2iedn.xlarge |db.x2iedn.2xlarge |db.x2iedn.4xlarge | db.x2iedn.8xlarge |db.x2iedn.16xlarge |db.x2iedn.24xlarge | db.x2iedn.32xlarge
db.x2iez	db.x2iezn.2xlarge |db.x2iezn.4xlarge |db.x2iezn.6xlarge | db.x2iezn.8xlarge |db.x2iezn.12xlarge
db.m7i	db.m7i.large |db.m7i.xlarge |db.m7i.2xlarge |db.m7i.4xlarge |db.m7i.8xlarge |db.m7i.12xlarge |db.m7i.16xlarge |db.m7i.24xlarge |db.m7i.48xlarge

Tipo	Tamanho
db.m6i	db.m6i.large | db.m6i.xlarge | db.m6i.2xlarge | db.m6i.4xlarge | db.m6i.8xlarge | db.m6i.12xlarge | db.m6i.16xlarge | db.m6i.24xlarge | db.m6i.32xlarge
db.m5	db.m5.large | db.m5.xlarge | db.m5.2xlarge | db.m5.4xlarge | db.m5.8xlarge | db.m5.12xlarge | db.m5.16xlarge | db.m5.24xlarge
db.t3	db.t3.medium | db.t3.large | db.t3.xlarge | db.t3.2xlarge

Compatibilidade com grupos de opções do RDS Custom para Oracle

É possível especificar um grupo de opções ao criar ou modificar uma instância de banco de dados do RDS Custom para Oracle. Para obter mais informações, consulte Trabalhar com grupos de opções no RDS Custom para Oracle.

Requisitos e limitações do RDS Custom for Oracle

Neste tópico, você encontra um resumo dos requisitos e da disponibilidade dos recursos do Amazon RDS Custom para Oracle para referência rápida.

Tópicos

- Requisitos gerais do RDS Custom for Oracle
- Limitações gerais do RDS Custom para Oracle
- Limitações de CEV e AMI do RDS Custom para Oracle
- Configurações não aceitas para criar e modificar fluxos de trabalho
- Cotas de instância de banco de dados para a Conta da AWS

Requisitos gerais do RDS Custom for Oracle

Siga estes requisitos para o Amazon RDS Custom para Oracle:

- Você tem acesso ao My Oracle Support e ao Oracle Software Delivery Cloud para baixar a lista compatível de arquivos de instalação e patches para o RDS Custom para Oracle. Se você utilizar um patch desconhecido, a criação da versão de mecanismo personalizada (CEV) falhará. Nesse caso, entre em contato com a equipe de suporte do RDS Custom e peça que ela adicione o patch ausente. Para ter mais informações, consulte Etapa 2: Baixar arquivos de instalação do banco de dados e patches do Oracle Software Delivery Cloud.
- Você tem acesso ao Amazon S3. Esse serviço é necessário pelas seguintes razões:
  - Você faz upload de seus arquivos de instalação do Oracle para buckets do S3. Você vai usar os arquivos de instalação carregados para criar sua CEV do RDS Custom.
  - O RDS Custom para Oracle usa scripts baixados de buckets do S3 definidos internamente para executar ações em suas instâncias de banco de dados. Esses scripts são necessários para a integração e a automação do RDS Custom.
  - O RDS Custom para Oracle faz upload de determinados arquivos em buckets do S3 localizados em sua conta de cliente. Esses buckets usam o seguinte formato de nomenclatura: `do-not-delete-rds-custom-`*`account_id-region-uuid`*. Por exemplo, você pode ter um bucket denominado `do-not-delete-rds-custom-123456789012-us-east-1-12a3b4`.

  Para ter mais informações, consulte Etapa 3: Fazer upload de seus arquivos de instalação no Amazon S3 e Criar um CEV.

- Você deve usar as classes de instância de banco de dados listadas em Suporte a classes de instâncias de banco de dados do RDS Custom for Oracle para criar instâncias de banco de dados do RDS Custom para Oracle.
- Suas instâncias de banco de dados do RDS Custom para Oracle executam o Oracle Linux 8 (recomendado) ou o Oracle Linux 7. Se você precisar do Oracle Linux 7, entre em contato com o Suporte. Para obter mais informações, consulte Considerações sobre atualizações do banco de dados do RDS Custom para Oracle.
- Especifique as unidades de estado sólido gp2, gp3 ou io1 para armazenamento do Amazon EBS. O tamanho máximo de armazenamento é 64 TiB.
- Você tem uma chave do AWS KMS para criar uma instância de banco de dados do RDS Custom para Oracle. Para obter mais informações, consulte Etapa 1: Criar ou reutilizar uma chave AWS KMS de criptografia simétrica.
- Você tem o perfil do AWS Identity and Access Management (IAM) e o perfil de instância necessários para criar instâncias de banco de dados do RDS Custom para Oracle. Para obter mais informações, consulte Etapa 4: Configurar o IAM para RDS Custom para Oracle.
- O usuário do AWS Identity and Access Management (IAM) que cria uma CVE ou uma instância de banco de dados do RDS Custom tem as permissões necessárias para o IAM, o CloudTrail e o Amazon S3.

  Para obter mais informações, consulte Etapa 5: Conceder as permissões necessárias ao usuário ou ao perfil do IAM.
- Você fornece a sua própria configuração de nuvem privada virtual (VPC) e grupo de segurança. Para obter mais informações, consulte Etapa 6: Configurar a VPC para RDS Custom para Oracle.
- Forneça uma configuração de rede que o RDS Custom para Oracle pode utilizar para acessar outros Serviços da AWS. Para conhecer os requisitos específicos, consulte Etapa 4: Configurar o IAM para RDS Custom para Oracle.

Limitações gerais do RDS Custom para Oracle

As limitações a seguir se aplicam ao RDS Custom for Oracle:

- Você não pode modificar o identificador da instância de banco de dados existente do RDS Custom para Oracle.
- Não é possível especificar a arquitetura multilocatária para nenhuma versão, exceto o Oracle Database 19c.

- Não é possível criar vários bancos de dados Oracle em uma instância de banco de dados do RDS Custom para Oracle.
- Não altere sua instância de banco de dados do RDS Custom para Oracle ou sua instância do Amazon EC2 subjacente. A cobrança por uma instância de banco de dados do RDS Custom para Oracle não pode ser interrompida.
- Não é possível usar o gerenciamento automático de memória compartilhada porque o RDS Custom para Oracle comporta somente o gerenciamento automático de memória. Para ter mais informações, consulte Gerenciamento automático de memória no Guia do administrador do Oracle Database.
- Não altere o `DB_UNIQUE_NAME` da instância de banco de dados primária. Alterar o nome faz com que qualquer operação de restauração fique travada.
- Não é possível fazer mais de vinte cópias de snapshot simultaneamente na mesma região.
- Você não pode usar a API `describe-reserved-db-instances` para instâncias de banco de dados do RDS Custom para Oracle.

Para saber as limitações específicas da modificação de uma instância de banco de dados do RDS Custom para Oracle, consulteModificar sua instância de banco de dados do RDS Custom para Oracle. Para limitações de replicação, consulte Limitações gerais da replicação do RDS Custom para Oracle.

Limitações de CEV e AMI do RDS Custom para Oracle

As seguintes limitações se aplicam a CEVs e AMIs do RDS Custom para Oracle:

- Não é possível fornecer a própria AMI para uso em uma CEV do RDS Custom para Oracle. É possível especificar a AMI padrão, que usa o Oracle Linux 8, ou uma AMI que tenha sido usada anteriormente por uma CEV do RDS Custom para Oracle.

> **Note**
>
> O RDS Custom para Oracle lança uma nova AMI padrão quando vulnerabilidades e exposições comuns são descobertas. Nenhum cronograma fixo está disponível ou garantido. O RDS Custom para Oracle tende a publicar uma nova AMI padrão a cada trinta dias.

- Não é possível modificar uma CEV para usar uma AMI diferente.

- Não é possível criar uma instância de CDB usando uma CEV que utiliza os tipos de mecanismo `custom-oracle-ee` ou `custom-oracle-se2`. A CEV deve usar `custom-oracle-ee-cdb` ou `custom-oracle-se2-cdb`.
- Atualmente, o RDS Custom para Oracle não permite atualizar o sistema operacional da instância de banco de dados do RDS Custom para Oracle com chamadas de API do RDS. Como alternativa, atualize manualmente o sistema operacional com o seguinte comando: `sudo yum update --security`.

Configurações não aceitas para criar e modificar fluxos de trabalho

Por exemplo, ao criar ou modificar uma instância de banco de dados do RDS Custom para Oracle, não é possível fazer o seguinte:

- Modifique o número de núcleos de CPU e de threads por núcleo na classe da instância de banco de dados.
- Habilite a escalabilidade automática de armazenamento.
- Definir a retenção de backup como `0`.
- Configurar a autenticação Kerberos
- Especificar seu próprio grupo de parâmetros de banco de dados ou grupo de opções.
- Habilite o Performance Insights.
- Habilite o upgrade automático de versões secundárias.

Cotas de instância de banco de dados para a Conta da AWS

Certifique-se de que o número combinado de instâncias de banco de dados do RDS Custom e do Amazon RDS não exceda o limite de cota. Por exemplo, se a cota do Amazon RDS para 40 instâncias de banco de dados, você poderá ter 20 instâncias de banco de dados do RDS Custom for Oracle e 20 instâncias de banco de dados do Amazon RDS.

Configurar seu ambiente para o Amazon RDS Custom for Oracle

Antes de criar uma instância de banco de dados do Amazon RDS Custom para Oracle, realize as tarefas a seguir.

Tópicos

- Etapa 1: Criar ou reutilizar uma chave AWS KMS de criptografia simétrica
- Etapa 2: Baixar e instalar a AWS CLI
- Etapa 3: Extrair os modelos do CloudFormation para RDS Custom para Oracle
- Etapa 4: Configurar o IAM para RDS Custom para Oracle
- Etapa 5: Conceder as permissões necessárias ao usuário ou ao perfil do IAM
- Etapa 6: Configurar a VPC para RDS Custom para Oracle

Etapa 1: Criar ou reutilizar uma chave AWS KMS de criptografia simétrica

Chaves gerenciadas pelo cliente são AWS KMS keys em sua conta da AWS criadas, pertencentes e gerenciadas por você. Uma chave do KMS de criptografia simétrica gerenciada pelo cliente é necessária para o RDS Custom. Ao criar uma instância de banco de dados do RDS Custom for Oracle, forneça o identificador de chave do KMS. Para ter mais informações, consulte Configurar uma instância de banco de dados para o Amazon RDS Custom para Oracle.

Você tem as seguintes opções:

- Se você tiver uma chave do KMS gerenciada pelo cliente em sua Conta da AWS, poderá usá-la com o RDS Custom. Nenhuma outra ação é necessária.
- Se você já tiver criado uma chave do KMS de criptografia simétrica gerenciada pelo cliente para outro mecanismo do RDS Custom, poderá reutilizar a mesma chave do KMS. Nenhuma outra ação é necessária.
- Se você não tiver uma chave do KMS de criptografia simétrica gerenciada pelo cliente em sua conta, crie uma seguindo as instruções em Criar chaves no Guia do desenvolvedor do AWS Key Management Service.
- Se você estiver criando uma instância de banco de dados do CEV ou do RDS Custom e sua chave do KMS estiver em outra Conta da AWS, use a AWS CLI. Você não pode usar o console da AWS com chaves do KMS de várias contas.

> **Important**
>
> O RDS Custom não é compatível com chaves do KMS gerenciadas pela AWS.

A chave de criptografia simétrica deve conceder acesso às operações `kms:Decrypt` e `kms:GenerateDataKey` ao perfil do AWS Identity and Access Management (IAM) em seu perfil de instância do IAM. Se você tiver uma nova chave de criptografia simétrica em sua conta, nenhuma alteração será necessária. Do contrário, a política de sua chave de criptografia simétrica precisará conceder acesso a essas operações.

Para ter mais informações, consulte Etapa 4: Configurar o IAM para RDS Custom para Oracle.

Para ter mais informações sobre como configurar o IAM para o RDS Custom para Oracle, consulte Etapa 4: Configurar o IAM para RDS Custom para Oracle.

Etapa 2: Baixar e instalar a AWS CLI

A AWS fornece uma interface de linha de comando para utilizar os recursos do RDS Custom. Você pode utilizar a versão 1 ou a versão 2 do AWS CLI.

Para obter informações sobre como baixar e instalar a AWS CLI, consulte Instalar ou atualizar para a versão mais recente da AWS CLI.

Ignore esta etapa se uma das seguintes situações for verdadeira:

- Você planeja acessar o RDS Custom somente por meio do AWS Management Console.
- Você já baixou a AWS CLI para o Amazon RDS ou outro mecanismo de banco de dados do RDS Custom.

Etapa 3: Extrair os modelos do CloudFormation para RDS Custom para Oracle

Para simplificar a configuração, é altamente recomendável utilizar modelos do AWS CloudFormation para criar pilhas do CloudFormation. Se você planeja configurar o IAM e a VPC manualmente, ignore esta etapa.

Tópicos

- Etapa 3a: Baixar os arquivos de modelo do CloudFormation
- Etapa 3b: Extrair custom-oracle-iam.json

- Etapa 3c: Extrair custom-vpc.json

Etapa 3a: Baixar os arquivos de modelo do CloudFormation

Um modelo do CloudFormation é uma declaração de recursos da AWS que compõem uma pilha. O modelo é armazenado como um arquivo JSON.

Como baixar os arquivos de modelo do CloudFormation

1. Abra o menu de contexto (clique com o botão direito do mouse) do link custom-oracle-iam.zip e selecione Save Link As (Salvar link como).
2. Salve o arquivo no computador.
3. Repita as etapas anteriores para o link custom-vpc.zip.

   Se já tiver configurado sua VPC para o RDS Custom, ignore essa etapa.

Etapa 3b: Extrair custom-oracle-iam.json

Abra o arquivo `custom-oracle-iam.zip` que você baixou e, depois, extraia o arquivo `custom-oracle-iam.json`. O início do arquivo tem a aparência a seguir.

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Parameters": {
    "EncryptionKey": {
      "Type": "String",
      "Default": "*",
      "Description": "KMS Key ARN for encryption of data managed by RDS Custom and by
 DB Instances."
    }
  },
  "Resources": {
    "RDSCustomInstanceServiceRole": {
      "Type": "AWS::IAM::Role",
      "Properties": {
        "RoleName": { "Fn::Sub": "AWSRDSCustomInstanceRole-${AWS::Region}" },
        "AssumeRolePolicyDocument": {
          "Version": "2012-10-17",
          "Statement": [
            {
              "Action": "sts:AssumeRole",
```

```
            "Effect": "Allow",
            "Principal": {
              "Service": "ec2.amazonaws.com"
            }
          }
        ]
      },...
```

Etapa 3c: Extrair custom-vpc.json

> **Note**
>
> Se você já tiver configurado uma VPC para o RDS Custom para Oracle, poderá ignorar esta etapa. Para obter mais informações, consulte Configurar a VPC manualmente para RDS Custom para Oracle.

Abra o arquivo `custom-vpc.zip` que você baixou e, depois, extraia o arquivo `custom-vpc.json`. O início do arquivo tem a aparência a seguir.

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Parameters": {
    "PrivateVpc": {
      "Type": "AWS::EC2::VPC::Id",
      "Description": "Private VPC Id to use for RDS Custom DB Instances"
    },
    "PrivateSubnets": {
      "Type": "List<AWS::EC2::Subnet::Id>",
      "Description": "Private Subnets to use for RDS Custom DB Instances"
    },
    "RouteTable": {
      "Type": "String",
      "Description": "Route Table that must be associated with the PrivateSubnets and
 used by S3 VPC Endpoint",
      "AllowedPattern": "rtb-[0-9a-z]+"
    }
  },
  "Resources": {
    "DBSubnetGroup": {
      "Type": "AWS::RDS::DBSubnetGroup",
      "Properties": {
```

```
        "DBSubnetGroupName": "rds-custom-private",
        "DBSubnetGroupDescription": "RDS Custom Private Network",
        "SubnetIds": {
          "Ref": "PrivateSubnets"
        }
      }
    },...
```

Etapa 4: Configurar o IAM para RDS Custom para Oracle

Use um perfil ou um usuário do IAM (conhecido como entidade do IAM) para criar uma instância de banco de dados do RDS Custom usando o console ou a AWS CLI. Essa entidade do IAM deve ter as permissões necessárias para a criação de instância.

É possível configurar o IAM usando o CloudFormation ou etapas manuais.

> **Important**
>
> É altamente recomendável configurar seu ambiente do RDS Custom para Oracle usando AWS CloudFormation. Essa técnica é a mais fácil e menos propensa a erros.

Tópicos

- Configurar o IAM utilizando o CloudFormation
- Criar seu perfil do IAM e perfil de instância manualmente

Configurar o IAM utilizando o CloudFormation

Quando você usa o modelo do CloudFormation para IAM, ele cria os seguintes recursos necessários:

- Um perfil de instância denominado `AWSRDSCustomInstanceProfile-`*`region`*.
- Um perfil de serviço denominado `AWSRDSCustomInstanceRole-`*`region`*
- Uma política de acesso denominada `AWSRDSCustomIamRolePolicy` anexada ao perfil de serviço

Para configurar o IAM utilizando o CloudFormation

1. Abra o console do CloudFormation em https://console.aws.amazon.com/cloudformation.

2. Inicie o assistente de criação de pilha e selecione Create Stack (Criar pilha).
3. Na página Create a stack (Criar uma pilha), faça o seguinte:
   a. Em Prepare template (Preparar modelo), selecione Template is ready (O modelo está pronto).
   b. Para Template source (Origem do template), escolha Upload a template file (Fazer upload de um arquivo de template).
   c. Em Escolher arquivo, navegue até custom-oracle-iam.json e selecione-o.
   d. Escolha Próximo.
4. Na página Specify stack details (Especificar detalhes da pilha), faça o seguinte:
   a. Para Stack name (Nome da pilha), insira **`custom-oracle-iam`**.
   b. Escolha Avançar.
5. Na página Configurar opções de pilha, selecione Avançar.
6. Na página Review custom-oracle-iam (Revisar custom-oracle-iam), faça o seguinte:
   a. Marque a caixa de seleção Confirmo que o AWS CloudFormation pode criar recursos do IAM com nomes personalizados.
   b. Selecione Enviar.

   O CloudFormation cria as funções do IAM necessárias para o RDS Custom for Oracle. No painel esquerdo, quando custom-oracle-iam mostrar CREATE_COMPLETE, vá para a próxima etapa.
7. No painel à esquerda, selecione custom-oracle-iam. No painel à direita, faça o seguinte:
   a. Selecione Informações da pilha. Sua pilha tem um ID no formato arn:aws:cloudformation:***`region`***:***`account-no`***:stack/custom-oracle-iam/***`identifier`***.
   b. Escolha atributos. Você deve ver o seguinte:
      - Um perfil de instância denominado AWSRDSCustomInstanceProfile-***`region`***
      - Um perfil de serviço denominado AWSRDSCustomInstanceRole-***`region`***

      Ao criar sua instância de banco de dados do RDS Custom, você precisa fornecer o ID do perfil de instância.

Criar seu perfil do IAM e perfil de instância manualmente

A configuração é mais fácil quando você usa o CloudFormation. No entanto, você também pode configurar o IAM manualmente. Para configuração manual, faça o seguinte:

- Etapa 1: Criar o perfil do IAM AWSRDSCustomInstanceRoleForRdsCustomInstance.
- Etapa 2: Adicionar uma política de acesso a AWSRDSCustomInstanceRoleForRdsCustomInstance.
- Etapa 2: Adicionar uma política de acesso a AWSRDSCustomInstanceRoleForRdsCustomInstance.
- Etapa 4: Adicionar AWSRDSCustomInstanceRoleForRdsCustomInstance a AWSRDSCustomInstanceProfile.

Etapa 1: Criar o perfil do IAM AWSRDSCustomInstanceRoleForRdsCustomInstance

Nesta etapa, você cria o perfil usando o formato de nomenclatura `AWSRDSCustomInstanceRole-`*`region`*. Utilizando a política de confiança, o Amazon EC2 pode assumir a função. O exemplo a seguir pressupõe que você definiu a variável de ambiente `$REGION` como a Região da AWS onde você deseja criar sua instância de banco de dados.

```
aws iam create-role \
  --role-name AWSRDSCustomInstanceRole-$REGION \
  --assume-role-policy-document '{
    "Version": "2012-10-17",
      "Statement": [
        {
          "Action": "sts:AssumeRole",
          "Effect": "Allow",
          "Principal": {
              "Service": "ec2.amazonaws.com"
          }
        }
      ]
    }'
```

Etapa 2: Adicionar uma política de acesso a AWSRDSCustomInstanceRoleForRdsCustomInstance

Quando você incorpora uma política em linha em uma função do IAM, essa política é utilizada como parte da política de acesso (permissões) da função. Você cria a política

`AWSRDSCustomIamRolePolicy`, que permite ao Amazon EC2 enviar e receber mensagens e realizar várias ações.

O exemplo a seguir cria a política de acesso chamada `AWSRDSCustomIamRolePolicy` e a adiciona à função do IAM `AWSRDSCustomInstanceRole-`*`region`*. Esse exemplo pressupõe que você definiu as seguintes variáveis de ambiente:

`$REGION`

Defina essa variável como a Região da AWS onde você planeja criar sua instância de banco de dados.

`$ACCOUNT_ID`

Defina essa variável como o número de sua Conta da AWS.

`$KMS_KEY`

Defina essa variável como o nome do recurso da Amazon (ARN) da AWS KMS key que você deseja usar para as suas instâncias de banco de dado do RDS Custom. Para especificar mais de uma chave do KMS, adicione-a à seção `Resources` do ID de instrução (Sid) 11.

```
aws iam put-role-policy \
  --role-name AWSRDSCustomInstanceRole-$REGION \
  --policy-name AWSRDSCustomIamRolePolicy \
  --policy-document '{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "1",
            "Effect": "Allow",
            "Action": [
                "ssm:DescribeAssociation",
                "ssm:GetDeployablePatchSnapshotForInstance",
                "ssm:GetDocument",
                "ssm:DescribeDocument",
                "ssm:GetManifest",
                "ssm:GetParameter",
                "ssm:GetParameters",
                "ssm:ListAssociations",
                "ssm:ListInstanceAssociations",
                "ssm:PutInventory",
                "ssm:PutComplianceItems",
```

```
                "ssm:PutConfigurePackageResult",
                "ssm:UpdateAssociationStatus",
                "ssm:UpdateInstanceAssociationStatus",
                "ssm:UpdateInstanceInformation",
                "ssm:GetConnectionStatus",
                "ssm:DescribeInstanceInformation",
                "ssmmessages:CreateControlChannel",
                "ssmmessages:CreateDataChannel",
                "ssmmessages:OpenControlChannel",
                "ssmmessages:OpenDataChannel"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Sid": "2",
            "Effect": "Allow",
            "Action": [
                "ec2messages:AcknowledgeMessage",
                "ec2messages:DeleteMessage",
                "ec2messages:FailMessage",
                "ec2messages:GetEndpoint",
                "ec2messages:GetMessages",
                "ec2messages:SendReply"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Sid": "3",
            "Effect": "Allow",
            "Action": [
                "logs:PutRetentionPolicy",
                "logs:PutLogEvents",
                "logs:DescribeLogStreams",
                "logs:DescribeLogGroups",
                "logs:CreateLogStream",
                "logs:CreateLogGroup"
            ],
            "Resource": [
                "arn:aws:logs:'$REGION':'$ACCOUNT_ID':log-group:rds-custom-instance*"
            ]
```

```
        },
        {
            "Sid": "4",
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:GetObject",
                "s3:GetObjectVersion"
            ],
            "Resource": [
                "arn:aws:s3:::do-not-delete-rds-custom-*/*"
            ]
        },
        {
            "Sid": "5",
            "Effect": "Allow",
            "Action": [
                "cloudwatch:PutMetricData"
            ],
            "Resource": [
                "*"
            ],
            "Condition": {
                "StringEquals": {
                    "cloudwatch:namespace": [
                        "RDSCustomForOracle/Agent"
                    ]
                }
            }
        },
        {
            "Sid": "6",
            "Effect": "Allow",
            "Action": [
                "events:PutEvents"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Sid": "7",
            "Effect": "Allow",
            "Action": [
```

```
                "secretsmanager:GetSecretValue",
                "secretsmanager:DescribeSecret"
            ],
            "Resource": [
                "arn:aws:secretsmanager:'$REGION':'$ACCOUNT_ID':secret:do-not-delete-
rds-custom-*",
                "arn:aws:secretsmanager:'$REGION':'$ACCOUNT_ID':secret:rds-custom!
oracle-do-not-delete-*"
            ]
        },
        {
           "Sid": "8",
           "Effect": "Allow",
           "Action": [
             "s3:ListBucketVersions"
           ],
           "Resource": [
             "arn:aws:s3:::do-not-delete-rds-custom-*"
           ]
         },
         {
            "Sid": "9",
            "Effect": "Allow",
            "Action": "ec2:CreateSnapshots",
            "Resource": [
                "arn:aws:ec2:*:*:instance/*",
                "arn:aws:ec2:*:*:volume/*"
            ],
            "Condition": {
                "StringEquals": {
                    "ec2:ResourceTag/AWSRDSCustom": "custom-oracle"
                }
            }
          },
          {
            "Sid": "10",
            "Effect": "Allow",
            "Action": "ec2:CreateSnapshots",
            "Resource": [
                "arn:aws:ec2:*::snapshot/*"
            ]
          },
          {
            "Sid": "11",
```

```
            "Effect": "Allow",
            "Action": [
              "kms:Decrypt",
              "kms:GenerateDataKey"
            ],
            "Resource": [
              "arn:aws:kms:'$REGION':'$ACCOUNT_ID':key/'$KMS_KEY'"
            ]
          },
          {
            "Sid": "12",
            "Effect": "Allow",
            "Action": "ec2:CreateTags",
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "ec2:CreateAction": [
                        "CreateSnapshots"
                    ]
                }
            }
        },
        {
            "Sid": "13",
            "Effect": "Allow",
            "Action": [
                "sqs:SendMessage",
                "sqs:ReceiveMessage",
                "sqs:DeleteMessage",
                "sqs:GetQueueUrl"
            ],
            "Resource": "arn:aws:sqs:'$REGION':'$ACCOUNT_Id':do-not-delete-rds-custom-
*",
            "Condition": {
                "StringLike": {
                    "aws:ResourceTag/AWSRDSCustom": "custom-oracle"
                }
            }
        }
    ]
}'
```

Etapa 3: Criar o perfil de instância do RDS Custom AWSRDSCustomInstanceProfile

Um perfil de instância é um contêiner que inclui um único perfil do IAM. O RDS Custom usa o perfil de instância para transmitir o perfil à instância.

Se você usar a CLI para criar um perfil, você cria o perfil e o perfil de instância como ações separadas, com nomes possivelmente diferentes. Crie seu perfil de instância do IAM da seguinte maneira, chamando-o de `AWSRDSCustomInstanceProfile-`*`region`*. O exemplo a seguir pressupõe que você definiu a variável de ambiente `$REGION` como a Região da AWS onde você deseja criar sua instância de banco de dados.

```
aws iam create-instance-profile \
    --instance-profile-name AWSRDSCustomInstanceProfile-$REGION
```

Etapa 4: Adicionar AWSRDSCustomInstanceRoleForRdsCustomInstance a AWSRDSCustomInstanceProfile

Adicione seu perfil do IAM ao perfil de instância que você criou anteriormente. O exemplo a seguir pressupõe que você definiu a variável de ambiente `$REGION` como a Região da AWS onde você deseja criar sua instância de banco de dados.

```
aws iam add-role-to-instance-profile \
    --instance-profile-name AWSRDSCustomInstanceProfile-$REGION \
    --role-name AWSRDSCustomInstanceRole-$REGION
```

Etapa 5: Conceder as permissões necessárias ao usuário ou ao perfil do IAM

Garanta que a entidade principal do IAM (usuário ou perfil) que cria a CEV ou a instância de banco de dados do RDS Custom tenha uma das seguintes políticas:

- A política `AdministratorAccess`
- A política `AmazonRDSFullAccess` com permissões necessárias para o Amazon S3 e o AWS KMS, criação da CEV e criação de instâncias de banco de dados.

Tópicos

- Permissões necessárias do IAM para o Amazon S3 e o AWS KMS
- Permissões necessárias do IAM para criar uma CEV

- Permissões necessárias do IAM para criar uma instância de banco de dados com base em uma CEV

Permissões necessárias do IAM para o Amazon S3 e o AWS KMS

Para criar CEVs ou instâncias de banco de dados do RDS Custom para Oracle, a entidade principal do IAM precisa acessar o Amazon S3 e o AWS KMS. O exemplo de política JSON a seguir concede as permissões necessárias.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "CreateS3Bucket",
            "Effect": "Allow",
            "Action": [
                "s3:CreateBucket",
                "s3:PutBucketPolicy",
                "s3:PutBucketObjectLockConfiguration",
                "s3:PutBucketVersioning"
            ],
            "Resource": "arn:aws:s3:::do-not-delete-rds-custom-*"
        },
        {
            "Sid": "CreateKmsGrant",
            "Effect": "Allow",
            "Action": [
                "kms:CreateGrant",
                "kms:DescribeKey"
            ],
            "Resource": "*"
        }
    ]
}
```

Para ter mais informações sobre a permissão `kms:CreateGrant`, consulte Gerenciamento de AWS KMS key.

Permissões necessárias do IAM para criar uma CEV

Para criar uma CEV, a entidade principal do IAM precisa das seguintes permissões adicionais:

```
s3:GetObjectAcl
s3:GetObject
s3:GetObjectTagging
s3:ListBucket
mediaimport:CreateDatabaseBinarySnapshot
```

O exemplo de política JSON a seguir concede permissões adicionais necessárias para acessar o bucket *my-custom-installation-files* e seu conteúdo.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AccessToS3MediaBucket",
            "Effect": "Allow",
            "Action": [
                "s3:GetObjectAcl",
                "s3:GetObject",
                "s3:GetObjectTagging",
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::my-custom-installation-files",
                "arn:aws:s3:::my-custom-installation-files/*"
            ]
        },
        {
            "Sid": "PermissionForByom",
            "Effect": "Allow",
            "Action": [
                "mediaimport:CreateDatabaseBinarySnapshot"
            ],
            "Resource": "*"
        }
    ]
}
```

Você pode conceder permissões semelhantes para o Amazon S3 às contas dos autores das chamadas usando uma política de bucket do S3.

Permissões necessárias do IAM para criar uma instância de banco de dados com base em uma CEV

Para criar uma instância de banco de dados do RDS Custom para Oracle de uma CEV existente, a entidade principal do IAM precisa das seguintes permissões adicionais:

```
iam:SimulatePrincipalPolicy
cloudtrail:CreateTrail
cloudtrail:StartLogging
```

O exemplo de política JSON a seguir concede as permissões necessárias para validar um perfil do IAM e registrar informações em um AWS CloudTrail.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ValidateIamRole",
            "Effect": "Allow",
            "Action": "iam:SimulatePrincipalPolicy",
            "Resource": "*"
        },
        {
            "Sid": "CreateCloudTrail",
            "Effect": "Allow",
            "Action": [
                "cloudtrail:CreateTrail",
                "cloudtrail:StartLogging"
            ],
            "Resource": "arn:aws:cloudtrail:*:*:trail/do-not-delete-rds-custom-*"
        }
    ]
}
```

Etapa 6: Configurar a VPC para RDS Custom para Oracle

Sua instância de banco de dados do RDS Custom está em uma nuvem privada virtual (VPC) com base no serviço da Amazon VPC, assim como uma instância do Amazon EC2 ou do Amazon RDS. Você fornece e configura sua própria VPC. Ao contrário do RDS Custom para SQL Server, o RDS Custom para Oracle não cria uma lista de controle de acesso nem grupos de segurança. Você deve anexar seu próprio grupo de segurança, sub-redes e tabelas de rotas.

É possível configurar a nuvem privada virtual (VPC) utilizando o CloudFormation ou um processo manual.

> **Important**
>
> É altamente recomendável configurar seu ambiente do RDS Custom para Oracle usando AWS CloudFormation. Essa técnica é a mais fácil e menos propensa a erros.

Tópicos

- Configurar a VPC utilizando o CloudFormation (recomendado)
- Configurar a VPC manualmente para RDS Custom para Oracle

Configurar a VPC utilizando o CloudFormation (recomendado)

Se já tiver configurado sua VPC para outro mecanismo do RDS Custom e quiser reutilizar a VPC existente, ignore esta etapa. Esta seção supõe o seguinte:

- Você já usou o CloudFormation para criar seu perfil e sua função de instância do IAM.
- Você sabe o ID da sua tabela de rotas.

  Para que uma instância de banco de dados seja privada, ela deve estar em uma sub-rede privada. Para que uma sub-rede seja privada, ela não deve estar associada a uma tabela de rotas que tenha um gateway de Internet padrão. Para ter mais informações, consulte Configurar tabelas de rotas, no Guia do usuário da Amazon VPC.

Quando você usa o modelo do CloudFormation para a VPC, ele cria os seguintes recursos necessários:

- Uma VPC privada

- Um grupo de sub-redes denominado `rds-custom-private`.
- Os seguintes endpoints da VPC, que a instância de banco de dados usa para se comunicar com Serviços da AWS dependentes:
  - `com.amazonaws.region.ec2messages`
  - `com.amazonaws.region.events`
  - `com.amazonaws.region.logs`
  - `com.amazonaws.region.monitoring`
  - `com.amazonaws.region.s3`
  - `com.amazonaws.region.secretsmanager`
  - `com.amazonaws.region.ssm`
  - `com.amazonaws.region.ssmmessages`

  Se você estiver criando uma implantação multi-AZ:

  - `com.amazonaws.region.sqs`

> **Note**
>
> Para uma configuração de rede complexa com contas existentes, recomendamos configurar o acesso aos serviços dependentes manualmente, caso o acesso ainda não exista. Para obter mais informações, consulte Garanta que a VPC possa acessar os Serviços da AWS dependentes..

Para configurar sua VPC utilizando o CloudFormation

1. Abra o console do CloudFormation em https://console.aws.amazon.com/cloudformation.
2. Inicie o assistente de criação de pilhas e selecione Criar pilha e, depois, Com novos recursos (padrão).
3. Na página Create a stack (Criar uma pilha), faça o seguinte:
   a. Em Prepare template (Preparar modelo), selecione Template is ready (O modelo está pronto).
   b. Para Template source (Origem do template), escolha Upload a template file (Fazer upload de um arquivo de template).

c. Para Choose file (Escolher arquivo), navegue até `custom-vpc.json` e escolha essa opção.

d. Escolha Próximo.

4. Na página Specify stack details (Especificar detalhes da pilha), faça o seguinte:

a. Para Stack name (Nome da pilha), insira **`custom-vpc`**.

b. Para Parameters (Parâmetros), escolha as sub-redes privadas a serem utilizadas para instâncias de banco de dados do RDS Custom.

c. Escolha o ID da VPC privada a ser utilizada para instâncias de banco de dados do RDS Custom.

d. Insira a tabela de rotas principal associada às sub-redes privadas.

e. Escolha Avançar.

5. Na página Configurar opções de pilha, selecione Avançar.

6. Na página Revisar, selecione Enviar.

O CloudFormation configura sua VPC privada. No painel esquerdo, quando custom-vpc mostrar CREATE_COMPLETE, vá para a próxima etapa.

7. (Opcional) Revise os detalhes de sua VPC. No painel Pilhas, selecione custom-vpc. No painel à direita, faça o seguinte:

a. Selecione Informações da pilha. Sua pilha tem um ID no formato arn:aws:cloudformation:***`region`***:***`account-no`***:stack/custom-vpc/***`identifier`***.

b. Escolha atributos. Você deve ver um grupo de sub-redes chamado rds-custom-private e vários endpoints da VPC que usam o formato de nomenclatura vpce-***`string`***. Cada endpoint corresponde a um AWS service (Serviço da AWS) com o qual o RDS Custom precisa se comunicar. Para obter mais informações, consulte Garanta que a VPC possa acessar os Serviços da AWS dependentes..

c. Selecione Parâmetros. Você deve ver as sub-redes privadas, a VPC privada e a tabela de rotas que você especificou ao criar a pilha. Ao criar uma instância de banco de dados, você precisa fornecer o ID da VPC o grupo de sub-redes.

Configurar a VPC manualmente para RDS Custom para Oracle

Como alternativa à automatização da criação da VPC com o AWS CloudFormation, é possível configurar a VPC manualmente. Essa opção pode ser melhor quando você tem uma configuração de rede complexa que usa recursos existentes.

Tópicos

- Garanta que a VPC possa acessar os Serviços da AWS dependentes.
- Configurar o serviço de metadados da instância

Garanta que a VPC possa acessar os Serviços da AWS dependentes.

O RDS Custom envia comunicações de sua instância de banco de dados para outros Serviços da AWS. Garanta que os seguintes serviços estejam acessíveis por meio da sub-rede na qual você cria instâncias de banco de dados do RDS Custom:

- Amazon CloudWatch (`com.amazonaws.`*`region`*`.monitoring`)
- Amazon CloudWatch Logs (`com.amazonaws.`*`region`*`.logs`)
- Amazon CloudWatch Events (`com.amazonaws.`*`region`*`.events`)
- Amazon EC2 (`com.amazonaws.`*`region`*`.ec2` e `com.amazonaws.`*`region`*`.ec2messages`)
- Amazon EventBridge (`com.amazonaws.`*`region`*`.events`)
- Amazon S3 (`com.amazonaws.`*`region`*`.s3`)
- AWS Secrets Manager (`com.amazonaws.`*`region`*`.secretsmanager`)
- AWS Systems Manager (`com.amazonaws.`*`region`*`.ssm` e `com.amazonaws.`*`region`*`.ssmmessages`)

Se você criar implantações multi-AZ

- Amazon Simple Queue Service (`com.amazonaws.`*`region`*`.sqs`)

Se o RDS Custom não conseguir se comunicar com os serviços necessários, ele publicará os seguintes eventos:

```
Database instance in incompatible-network. SSM Agent connection not available. Amazon
 RDS can't connect to the dependent AWS services.
```

```
Database instance in incompatible-network. Amazon RDS can't connect to dependent AWS
 services. Make sure port 443 (HTTPS) allows outbound connections, and try again.
 "Failed to connect to the following services: s3 events"
```

Para evitar erros `incompatible-network`, os componentes da VPC envolvidos na comunicação entre sua instância de banco de dados do RDS Custom e os Serviços da AWS devem cumprir os seguintes requisitos:

- A instância de banco de dados pode fazer conexões de saída na porta 443 para outros Serviços da AWS.
- A VPC permite respostas recebidas a solicitações originadas de sua instância de banco de dados do RDS Custom.
- O RDS Custom pode resolver corretamente os nomes de domínio dos endpoints para cada AWS service (Serviço da AWS).

Se você já configurou uma VPC para outro mecanismo de banco de dados do RDS Custom, poderá reutilizar essa VPC e ignorar esse processo.

Configurar o serviço de metadados da instância

Verifique se a sua instância pode fazer o seguinte:

- Acesse o serviço de metadados de instância utilizando o Instance Metadata Service versão 2 (IMDSv2).
- Permita comunicações de saída por meio da porta 80 (HTTP) com o endereço IP do link IMDS.
- Solicite metadados da instância de `http://169.254.169.254`, o link IMDSv2.

Consulte mais informações em Usar IMDSv2 no Guia do usuário do Amazon EC2.

A automação do RDS Custom for Oracle utiliza o IMDSv2 por padrão, definindo `HttpTokens=enabled` na instância do Amazon EC2 subjacente. No entanto, se quiser, você pode utilizar o IMDSv1. Consulte mais informações em Configurar as opções de metadados da instância no Guia do usuário do Amazon EC2.

Trabalhar com versões de mecanismo personalizadas para o Amazon RDS Custom for Oracle

Uma versão de mecanismo personalizada (CEV) para o Amazon RDS Custom for Oracle é um snapshot de volume binário de um mecanismo de banco de dados e de uma imagem de máquina da Amazon (AMI) específica. Por padrão, o RDS Custom para Oracle usa a AMI mais recente disponível, gerenciada pelo RDS Custom, mas você pode especificar uma AMI que foi usada em uma CEV anterior. Você armazena seus arquivos de instalação do banco de dados no Amazon S3. O RDS Custom usa os arquivos de instalação e a AMI para criar a CEV para você.

Tópicos

- Preparar-se para criar uma CEV
- Criar um CEV
- Modificar o status da CEV
- Visualizar detalhes da CEV com o Amazon RDS Custom para Oracle
- Excluir uma CEV

Preparar-se para criar uma CEV

Para criar uma CEV, acesse os arquivos de instalação e os patches armazenados no bucket do Amazon S3 para qualquer uma das seguintes versões:

- Oracle Database 19c
- Oracle Database 18c
- Oracle Database 12c Versão 2 (12.2)
- Oracle Database 12c Versão 1 (12.1)

Por exemplo, é possível utilizar a RU/RUR de abril de 2021 para o Oracle Database 19c, ou qualquer combinação válida de arquivos de instalação e patches. Para ter mais informações sobre as regiões e as versões compatíveis com o RDS Custom para Oracle, consulte RDS Custom with RDS for Oracle (RDS Custom com RDS para Oracle).

Tópicos

- Etapa 1 (opcional): Baixar os modelos de manifesto

- Etapa 2: Baixar arquivos de instalação do banco de dados e patches do Oracle Software Delivery Cloud
- Etapa 3: Fazer upload de seus arquivos de instalação no Amazon S3
- Etapa 4 (opcional): Compartilhar a mídia de instalação no S3 entreContas da AWS
- Etapa 5: Preparar o manifesto CEV
- Etapa 6 (opcional): Validar o manifesto CEV
- Etapa 7: Adicionar permissões do IAM necessárias

Etapa 1 (opcional): Baixar os modelos de manifesto

Um manifesto CEV é um documento JSON que inclui a lista de arquivos .zip de instalação do banco de dados para sua CEV. Para criar uma CEV, faça o seguinte:

1. Identifique os arquivos de instalação do banco de dados Oracle que você deseja incluir em sua CEV.
2. Baixe os arquivos de instalação.
3. Crie um manifesto JSON que liste os arquivos de instalação.

O RDS Custom para Oracle fornece modelos de manifesto JSON com nossos arquivos .zip recomendados para cada versão compatível do Oracle Database. Por exemplo, o modelo a seguir é para a RU 19.17.0.0.0.

```
{
    "mediaImportTemplateVersion": "2020-08-14",
    "databaseInstallationFileNames": [
        "V982063-01.zip"
    ],
    "opatchFileNames": [
        "p6880880_190000_Linux-x86-64.zip"
    ],
    "psuRuPatchFileNames": [
        "p34419443_190000_Linux-x86-64.zip",
        "p34411846_190000_Linux-x86-64.zip"
    ],
    "otherPatchFileNames": [
        "p28852325_190000_Linux-x86-64.zip",
        "p29997937_190000_Linux-x86-64.zip",
```

```
        "p31335037_190000_Linux-x86-64.zip",
        "p32327201_190000_Linux-x86-64.zip",
        "p33613829_190000_Linux-x86-64.zip",
        "p34006614_190000_Linux-x86-64.zip",
        "p34533061_190000_Linux-x86-64.zip",
        "p34533150_190000_Generic.zip",
        "p28730253_190000_Linux-x86-64.zip",
        "p29213893_1917000DBRU_Generic.zip",
        "p33125873_1917000DBRU_Linux-x86-64.zip",
        "p34446152_1917000DBRU_Linux-x86-64.zip"
    ]
}
```

Cada modelo tem um arquivo leiame associado que inclui instruções para baixar os patches, os URLs para os arquivos .zip e somas de verificação de arquivos. Você pode usar esses modelos como estão ou modificá-los com seus próprios patches. Para revisar os modelos, baixe o custom-oracle-manifest.zip em seu disco local e abra-o com uma aplicação de arquivamento de arquivos. Para ter mais informações, consulte Etapa 5: Preparar o manifesto CEV.

Etapa 2: Baixar arquivos de instalação do banco de dados e patches do Oracle Software Delivery Cloud

Depois de identificar os arquivos de instalação que deseja para sua CEV, baixe-os para o sistema local. Os patches e arquivos de instalação do Oracle Database estão hospedados na Oracle Software Delivery Cloud. Cada CEV exige uma versão de base, como o Oracle Database 19c ou o Oracle Database 12c Versão 2 (12.2) e uma lista opcional de patches.

Como baixar os arquivos de instalação do banco de dados para o Oracle Database

1. Acesse https://edelivery.oracle.com/ e faça login.
2. Na caixa, insira **`Oracle Database Enterprise Edition`** ou **`Oracle Database Standard Edition 2`** e selecione Pesquisar.
3. Selecione uma das seguintes versões de base:

Versão do banco de dados	Edição Enterprise	Standard Edition 2
Oracle Database 19c	DLP: Oracle Database 19c Enterprise Edition 19.3.0.0.0	DLP: Oracle Database 19c Standard Edition 2 19.3.0.0.

Versão do banco de dados	Edição Enterprise	Standard Edition 2
	(Oracle Database Enterprise Edition)	0 (Oracle Database Standard Edition 2)
Oracle Database 18c	DLP: Oracle Database 18c Enterprise Edition 18.0.0.0.0 (Oracle Database Enterprise Edition)	DLP: Oracle Database Standard Edition 2 18.0.0.0.0 (Oracle Database Standard Edition 2)
Oracle Database 12c Versão 2 (12.2.0.1)	DLP: Oracle Database 12c Enterprise Edition 12.2.0.1.0 (Oracle Database Enterprise Edition)	DLP: Oracle Database Standard Edition 2 12.2.0.1.0 (Oracle Database Standard Edition 2)
Oracle Database 12c Versão 1 (12.1.0.2)	DLP: Oracle Database 12c Enterprise Edition 12.1.0.2.0 (Oracle Database Enterprise Edition)	DLP: Oracle Database Standard Edition 2 12.1.0.2.0 (Oracle Database Standard Edition 2)

4. Escolha Continuar.
5. Limpe a caixa de diálogo Download Queue (Baixar fila).
6. Selecione a opção que corresponde à sua versão de base:
   - Oracle Database 19.3.0.0.0: Long Term Release.
   - Oracle Database 18.0.0.0.0
   - Oracle Database 12.2.0.1.0.
   - Oracle Database 12.1.0.2.0.
7. Escolha Linux x86-64 em Platform/Languages (Plataforma/Idiomas).
8. Selecione Continuar e, depois, assine o Contrato de Licença da Oracle.
9. Selecione o arquivo .zip que corresponde à sua versão de base:

Versão e edição do banco de dados	Arquivos zip	Hash SHA-256
19c EE e SE2	V982063-01.zip	BA8329C757133DA313ED3B6D7F86C5AC42CD9970A28BF2E6233F3235233AA8D8
18c EE e SE2	V978967-01.zip	C96A4FD768787AF98272008833FE10B172691CF84E42816B138C12D4DE63AB96
12.2.0.1 EE e SE2	V839960-01.zip	96ED97D21F15C1AC0CCE3749DA6C3DAC7059BB60672D76B008103FC754D22DDE
12.1.0.2 EE	V46095-01_1of2.zip V46095-01_2of2.zip	31FDC2AF41687B4E547A3A18F796424D8C1AF36406D2160F65B0AF6A9CD47355 para V46095-01_1of2.zip 03DA14F5E875304B28F0F3BB02AF0EC33227885B99C9865DF70749D1E220ACCD para V46095-01_1of2.zip
12.1.0.2 SE2	V77388-01_1of2.zip V77388-01_2of2.zip	73873369753230F5A0921F95ACEADB591388CB06ED72A7F3AEA7BCBCEA2403BC para V77388-01_1of2.zip 2492E1BE1E3E3531DA83D0843C09C08E435AC8CEFD9A00C0DF56BE4F15CEEBF3 para V77388-01_2of2.zip

10. Baixe os patches Oracle desejados de `updates.oracle.com` ou `support.oracle.com` para seu sistema local. Você pode encontrar os URLs dos patches nos seguintes locais:

    - Os arquivos leiame no arquivo .zip que você baixou em Etapa 1 (opcional): Baixar os modelos de manifesto
    - Os patches listados em cada atualização de versão (RU) em Notas de versão do Amazon Relational Database Service (Amazon RDS) para Oracle

Etapa 3: Fazer upload de seus arquivos de instalação no Amazon S3

Carregue seus arquivos de patch e instalação do Oracle no Amazon S3 utilizando a AWS CLI. O bucket do S3 que contém seus arquivos de instalação deve estar na mesma região da AWS que a CEV.

Os exemplos nesta seção usam os seguintes espaços reservados:

- *`install-or-patch-file.zip`* – Arquivo de mídia de instalação do Oracle. Por exemplo, p32126828_190000_Linux-x86-64.zip é um patch.
- *`amzn-s3-demo-destination-bucket`* – Seu bucket do Amazon S3 designado aos seus arquivos de instalação carregados.
- *`123456789012/cev1`*: um prefixo opcional no bucket do Amazon S3.
- *`amzn-s3-demo-source-bucket`*: um bucket do Amazon S3 no qual há a opção de preparar arquivos.

Tópicos

- Etapa 3a: Verifique se o bucket do S3 está no local correto Região da AWS
- Etapa 3b: Certifique-se de que sua política de bucket do S3 tenha as permissões corretas
- Etapa 3c: Carregue seus arquivos usando os comandos cp ou sync
- Etapa 3: Listar os arquivos no bucket do S3

Etapa 3a: Verifique se o bucket do S3 está no local correto Região da AWS

Verifique se o bucket do S3 está na região do AWS na qual você planeja executar o comando `create-custom-db-engine-version`.

```
aws s3api get-bucket-location --bucket amzn-s3-demo-destination-bucket
```

Etapa 3b: Certifique-se de que sua política de bucket do S3 tenha as permissões corretas

É possível criar uma CEV do zero ou a partir de uma CEV de origem. Se você planeja criar um novo CEV a partir dos CEVs de origem, certifique-se de que sua política de bucket do S3 tenha as permissões corretas:

1. Identificar o bucket do S3, reservado pelo RDS Custom. O nome do bucket tem o formatodo-not-delete-rds-custom-*account*-*region*-*string*. Por exemplo, ele pode ser do-not-delete-rds-custom-*123456789012*-*us-east-1*-*abc123EXAMPLE*.
2. Certifique-se de que a permissão a seguir esteja anexada à sua política de bucket do S3. Substitua do-not-delete-rds-custom-*123456789012*-*us-east-1*-*abc123EXAMPLE* pelo nome do seu bucket.

```
{
    "Sid": "AWSRDSCustomForOracleCustomEngineVersionGetObject",
    "Effect": "Allow",
    "Principal": {
        "Service": "custom.rds.amazonaws.com"
    },
    "Action": [
        "s3:GetObject",
        "s3:GetObjectTagging"
    ],
    "Resource": "arn:aws:s3:::do-not-delete-rds-custom-123456789012-us-
east-1-abc123EXAMPLE/CustomEngineVersions/*"
}, ...
```

Etapa 3c: Carregue seus arquivos usando os comandos cp ou sync

Escolha uma das seguintes opções:

- Use `aws s3 cp` para carregar um único arquivo .zip.

  Carregue cada arquivo .zip de instalação separadamente. Não combine os arquivos .zip em um único arquivo .zip.
- Use `aws s3 sync` para carregar um diretório.

Example

O exemplo a seguir carrega *install-or-patch-file.zip* na pasta *123456789012/cev1* do bucket do Amazon S3 do RDS Custom. Execute um comando `aws s3` separado para cada .zip que você deseja carregar.

Para Linux, macOS ou Unix:

```
aws s3 cp install-or-patch-file.zip \
    s3://amzn-s3-demo-destination-bucket/123456789012/cev1/
```

Para Windows:

```
aws s3 cp install-or-patch-file.zip ^
    s3://amzn-s3-demo-destination-bucket/123456789012/cev1/
```

Example

O exemplo a seguir carrega os arquivos da pasta *cev1* local na pasta *123456789012/cev1* do seu bucket do Amazon S3.

Para Linux, macOS ou Unix:

```
aws s3 sync cev1 \
    s3://amzn-s3-demo-destination-bucket/123456789012/cev1/
```

Para Windows:

```
aws s3 sync cev1 ^
    s3://amzn-s3-demo-destination-bucket/123456789012/cev1/
```

Example

O exemplo a seguir faz upload de todos os arquivos em *amzn-s3-demo-source-bucket* na pasta *123456789012/cev1* em seu bucket do Amazon S3.

Para Linux, macOS ou Unix:

```
aws s3 sync s3://amzn-s3-demo-source-bucket/ \
    s3://amzn-s3-demo-destination-bucket/123456789012/cev1/
```

Para Windows:

```
aws s3 sync s3://amzn-s3-demo-source-bucket/ ^
    s3://amzn-s3-demo-destination-bucket/123456789012/cev1/
```

Etapa 3: Listar os arquivos no bucket do S3

O exemplo a seguir usa o `s3 ls` comando para listar os arquivos no bucket do Amazon S3 do RDS Custom.

```
aws s3 ls \
    s3://amzn-s3-demo-destination-bucket/123456789012/cev1/
```

Etapa 4 (opcional): Compartilhar a mídia de instalação no S3 entreContas da AWS

Para os fins desta seção, o bucket do Amazon S3 que contém os arquivos de instalação do Oracle enviados é seu bucket de mídia. Sua organização pode usar várias Contas da AWS em uma Região da AWS. Se sim, você pode optar por usar uma Conta da AWS para preencher o bucket de mídia e outra Conta da AWS para criar CEVs. Se você não pretende compartilhar o bucket de mídia, vá para a próxima seção.

Esta seção supõe o seguinte:

- Você tem acesso à conta que criou o bucket de mídia e a uma conta diferente na qual pretende criar CEVs.
- Você pretende criar CEVs em uma única Região da AWS. Se você pretende usar várias regiões, crie um bucket de mídia em cada região.
- Você está usando a CLI. Se estiver usando o console do Amazon S3, adapte as etapas a seguir.

Como configurar o bucket de mídia para compartilhamento entre Contas da AWS

1. Faça login na Conta da AWS que contém o bucket do S3 no qual você fez upload da mídia de instalação.
2. Comece com um modelo de política JSON em branco ou com uma política existente que você possa adaptar.

   O comando a seguir recupera uma política existente e a salva como *my-policy.json*. Neste exemplo, o bucket do S3 que contém os arquivos de instalação é chamado de *amzn-s3-demo-bucket*.

   ```
   aws s3api get-bucket-policy \
       --bucket amzn-s3-demo-bucket \
       --query Policy \
   ```

```
    --output text > my-policy.json
```

3. Edite as permissões do bucket de mídia da seguinte forma:

   - No elemento `Resource` do modelo, especifique o bucket do S3 no qual você fez upload dos arquivos de instalação do Oracle Database.
   - No elemento `Principal`, especifique os ARNs de todas as Contas da AWS que você pretende usar para criar CEVs. Você pode adicionar a raiz, um usuário ou um perfil à lista de permissões do bucket do S3. Para ter mais informações, consulte Identificadores do IAM no Manual do usuário do AWS Identity and Access Management.

```
{
    "Version": "2008-10-17",
    "Statement": [
        {
            "Sid": "GrantAccountsAccess",
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::account-1:root",
                    "arn:aws:iam::account-2:user/user-name-with-path",
                    "arn:aws:iam::account-3:role/role-name-with-path",
                    ...
                ]
            },
            "Action": [
                "s3:GetObject",
                "s3:GetObjectAcl",
                "s3:GetObjectTagging",
                "s3:ListBucket",
                "s3:GetBucketLocation"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket",
                "arn:aws:s3:::amzn-s3-demo-bucket/*"
            ]
        }
    ]
}
```

4. Associe a política ao bucket de mídia.

No exemplo a seguir, *amzn-s3-demo-bucket* é o nome do bucket do S3 que contém os arquivos de instalação e *my-policy.json* é o nome do arquivo JSON.

```
aws s3api put-bucket-policy \
    --bucket amzn-s3-demo-bucket \
    --policy file://my-policy.json
```

5. Faça login em uma Conta da AWS na qual pretende criar CEVs.
6. Verifique se essa conta pode acessar o bucket de mídia na Conta da AWS que o criou.

```
aws s3 ls --query "Buckets[].Name"
```

Para ter mais informações, consulte aws s3 ls na Referência de comandos da AWS CLI.

7. Crie uma CEV seguindo as etapas em Criar um CEV.

Etapa 5: Preparar o manifesto CEV

Um manifesto de CEV é um documento JSON que inclui o seguinte conteúdo:

- (Obrigatório) A lista de arquivos .zip carregados no Amazon S3. O RDS Custom aplica os patches na ordem em que eles estão listados no manifesto.
- (Opcional) Parâmetros de instalação que definem valores não padrão para a base do Oracle, o Oracle home e o ID e o nome do usuário e grupo do UNIX/Linux. Lembre-se de que você não pode modificar os parâmetros de instalação de uma CEV existente ou de uma instância de banco de dados existente. Você também não pode realizar a atualização uma para outra CEV quando os parâmetros de instalação têm configurações diferentes.

Para obter exemplos de manifestos CEV, consulte os modelos JSON que você baixou em Etapa 1 (opcional): Baixar os modelos de manifesto. Também é possível revisar os exemplos em Exemplos de manifesto de CEV.

Tópicos

- Campos JSON no manifesto de CEV
- Criar o manifesto de CEV
- Exemplos de manifesto de CEV

Campos JSON no manifesto de CEV

A tabela a seguir descreve os campos JSON no manifesto.

Campo JSON	Descrição
`MediaImportTemplateVersion`	Versão do manifesto de CEV. A data deve estar no formato `YYYY-MM-DD` .
`databaseInstallationFileNames`	Lista ordenada de arquivos de instalação do banco de dados.
`opatchFileNames`	Lista ordenada de instaladores OPatch utilizados para o mecanismo de banco de dados Oracle. Apenas um valor é válido. Valores para `opatchFileNames` devem começar com `p6880880_` .
`psuRuPatchFileNames`	Os patches PSU e RU desse banco de dados. **Important** Se você incluir `psuRuPatchFileNames` , `opatchFileNames` será obrigatório. Valores para `opatchFileNames` devem começar com `p6880880_` .
`OtherPatchFileNames`	Os patches que não estão na lista de patches PSU e RU. O RDS Custom aplicará esses patches após a aplicação dos patches PSU e RU. **Important** Se você incluir `OtherPatchFileNames` , `opatchFileNames` será obrigatório. Valores para `opatchFileNames` devem começar com `p6880880_` .
`installationParameters`	Configurações não padrão para a base do Oracle, o Oracle Home e o ID e o nome do usuário e do grupo do UNIX/Linux. É possível definir os seguintes parâmetros:

Campo JSON	Descrição
	oracleBase O diretório no qual seus binários do Oracle estão instalados. É o ponto de montagem do volume binário que armazena seus arquivos. O diretório base do Oracle pode incluir vários Oracle homes. Por exemplo, se `/home/oracle/oracle.19.0.0.0.ru-2020-04.rur-2020-04.r1.EE.1` for um dos seus diretórios do Oracle home, `/home/oracle` será o diretório base do Oracle. Um diretório base do Oracle especificado pelo usuário não é um link simbólico. Se você não especificar a base do Oracle, o diretório padrão será `/rdsdbbin` . oracleHome O diretório no qual seus binários do banco de dados Oracle estão instalados. Por exemplo, se você especificar `/home/oracle/` como sua base do Oracle, poderá especificar `/home/oracle/oracle.19.0.0.0.ru-2020-04.rur-2020-04.r1.EE.1/` como seu Oracle home. Um diretório do Oracle home especificado pelo usuário não é um link simbólico. O valor do Oracle home é referenciado pela variável de ambiente `$ORACLE_HOME` . Se você não especificar o Oracle home, o formato de nomenclatura padrão será `/rdsdbbin/oracle.` *major-engine-version* `.custom.r1.` *engine-edition* `.1`. unixUname O nome do usuário do UNIX que possui o software Oracle. O RDS Custom assume esse usuário ao executar comandos do banco de dados local. Se você especificar `unixUid` e `unixUname` , o RDS Custom criará o usuário se ele não existir e, depois, atribuirá o UID ao usuário se ele não for igual ao UID inicial.

Campo JSON	Descrição
	O nome do usuário padrão é `rdsdb`. unixUid O ID do usuário do UNIX que possui o software Oracle. Se você especificar `unixUid` e `unixUname` , o RDS Custom criará o usuário se ele não existir e, depois, atribuirá o UID ao usuário se ele não for igual ao UID inicial. O UID padrão é `61001`. Esse é o UID do usuário `rdsdb`. unixGroupName O nome do grupo do UNIX. O usuário do UNIX que possui o software Oracle pertence a esse grupo. O nome do grupo padrão é `rdsdb`. unixGroupId O ID do grupo do UNIX ao qual pertence o usuário do UNIX. O ID do grupo padrão é `1000`. Este é o ID do grupo `rdsdb`.

Cada versão do Oracle Database tem uma lista diferente de arquivos de instalação compatíveis. Ao criar seu manifesto de CEV, especifique somente os arquivos que são compatíveis com o RDS Custom para Oracle. Caso contrário, a criação da CEV falhará com um erro. Todos os patches listados em Notas de versão do Amazon Relational Database Service (Amazon RDS) para Oracle são compatíveis.

Criar o manifesto de CEV

Como criar um manifesto de CEV

1. Liste todos os arquivos de instalação que você planeja aplicar, na ordem em que deseja aplicá-los.
2. Correlacione os arquivos de instalação com os campos JSON descritos em Campos JSON no manifesto de CEV.
3. Realize um dos procedimentos a seguir:

- Crie o manifesto de CEV como um arquivo de texto JSON.
- Edite o modelo de manifesto de CEV ao criar a CEV no console. Para ter mais informações, consulte Criar um CEV.

Exemplos de manifesto de CEV

Os exemplos a seguir mostram arquivos de manifesto de CEV para diferentes versões do Oracle Database. Se você incluir um campo JSON no manifesto, nunca deixe-o vazio. Por exemplo, o manifesto de CEV a seguir é inválido porque `otherPatchFileNames` está vazio.

```
{
    "mediaImportTemplateVersion": "2020-08-14",
    "databaseInstallationFileNames": [
        "V982063-01.zip"
    ],
    "opatchFileNames": [
        "p6880880_190000_Linux-x86-64.zip"
    ],
    "psuRuPatchFileNames": [
        "p32126828_190000_Linux-x86-64.zip"
    ],
    "otherPatchFileNames": [
    ]
}
```

Tópicos

- Sample CEV manifest for Oracle Database 12c Release 1 (12.1)
- Sample CEV manifest for Oracle Database 12c Release 2 (12.2)
- Sample CEV manifest for Oracle Database 18c
- Sample CEV manifest for Oracle Database 19c

Example Exemplo de manifesto de CEV para Oracle Database 12c versão 1 (12.1)

No exemplo a seguir para a PSU de julho de 2021 do Oracle Database 12c versão 1 (12.1), o RDS Custom aplica os patches na sequência especificada. Assim, o RDS Custom aplica p32768233, depois p32876425, depois p18759211 e assim por diante. O exemplo define novos valores para o usuário e o grupo do UNIX, o Oracle home e a base do Oracle.

```
{
    "mediaImportTemplateVersion":"2020-08-14",
    "databaseInstallationFileNames":[
        "V46095-01_1of2.zip",
        "V46095-01_2of2.zip"
    ],
    "opatchFileNames":[
        "p6880880_121010_Linux-x86-64.zip"
    ],
    "psuRuPatchFileNames":[
        "p32768233_121020_Linux-x86-64.zip"
    ],
    "otherPatchFileNames":[
        "p32876425_121020_Linux-x86-64.zip",
        "p18759211_121020_Linux-x86-64.zip",
        "p19396455_121020_Linux-x86-64.zip",
        "p20875898_121020_Linux-x86-64.zip",
        "p22037014_121020_Linux-x86-64.zip",
        "p22873635_121020_Linux-x86-64.zip",
        "p23614158_121020_Linux-x86-64.zip",
        "p24701840_121020_Linux-x86-64.zip",
        "p25881255_121020_Linux-x86-64.zip",
        "p27015449_121020_Linux-x86-64.zip",
        "p28125601_121020_Linux-x86-64.zip",
        "p28852325_121020_Linux-x86-64.zip",
        "p29997937_121020_Linux-x86-64.zip",
        "p31335037_121020_Linux-x86-64.zip",
        "p32327201_121020_Linux-x86-64.zip",
        "p32327208_121020_Generic.zip",
        "p17969866_12102210119_Linux-x86-64.zip",
        "p20394750_12102210119_Linux-x86-64.zip",
        "p24835919_121020_Linux-x86-64.zip",
        "p23262847_12102201020_Linux-x86-64.zip",
        "p21171382_12102201020_Generic.zip",
        "p21091901_12102210720_Linux-x86-64.zip",
        "p33013352_12102210720_Linux-x86-64.zip",
        "p25031502_12102210720_Linux-x86-64.zip",
        "p23711335_12102191015_Generic.zip",
        "p19504946_121020_Linux-x86-64.zip"
    ],
    "installationParameters": {
        "unixGroupName": "dba",
        "unixGroupId": 12345,
```

```
        "unixUname": "oracle",
        "unixUid": 12345,
        "oracleHome": "/home/oracle/oracle.12.1.0.2",
        "oracleBase": "/home/oracle"
    }
}
```

Example Exemplo de manifesto de CEV para Oracle Database 12c versão 2 (12.2)

No exemplo a seguir para a PSU de outubro de 2021 do Oracle Database 12c versão 2 (12.2), o RDS Custom aplica p33261817, depois p33192662, depois p29213893 e assim por diante. O exemplo define novos valores para o usuário e o grupo do UNIX, o Oracle home e a base do Oracle.

```
{
    "mediaImportTemplateVersion":"2020-08-14",
    "databaseInstallationFileNames":[
        "V839960-01.zip"
    ],
    "opatchFileNames":[
        "p6880880_122010_Linux-x86-64.zip"
    ],
    "psuRuPatchFileNames":[
        "p33261817_122010_Linux-x86-64.zip"
    ],
    "otherPatchFileNames":[
        "p33192662_122010_Linux-x86-64.zip",
        "p29213893_122010_Generic.zip",
        "p28730253_122010_Linux-x86-64.zip",
        "p26352615_12201211019DBOCT2021RU_Linux-x86-64.zip",
        "p23614158_122010_Linux-x86-64.zip",
        "p24701840_122010_Linux-x86-64.zip",
        "p25173124_122010_Linux-x86-64.zip",
        "p25881255_122010_Linux-x86-64.zip",
        "p27015449_122010_Linux-x86-64.zip",
        "p28125601_122010_Linux-x86-64.zip",
        "p28852325_122010_Linux-x86-64.zip",
        "p29997937_122010_Linux-x86-64.zip",
        "p31335037_122010_Linux-x86-64.zip",
        "p32327201_122010_Linux-x86-64.zip",
        "p32327208_122010_Generic.zip"
    ],
    "installationParameters": {
        "unixGroupName": "dba",
```

```
        "unixGroupId": 12345,
        "unixUname": "oracle",
        "unixUid": 12345,
        "oracleHome": "/home/oracle/oracle.12.2.0.1",
        "oracleBase": "/home/oracle"
    }
}
```

Example Exemplo de manifesto de CEV para Oracle Database 18c

No exemplo a seguir para a PSU de outubro de 2021 do Oracle Database 18c, o RDS Custom aplica p32126855, depois p28730253, depois p27539475 e assim por diante. O exemplo define novos valores para o usuário e o grupo do UNIX, o Oracle home e a base do Oracle.

```
{
    "mediaImportTemplateVersion":"2020-08-14",
    "databaseInstallationFileNames":[
        "V978967-01.zip"
    ],
    "opatchFileNames":[
        "p6880880_180000_Linux-x86-64.zip"
    ],
    "psuRuPatchFileNames":[
        "p32126855_180000_Linux-x86-64.zip"
    ],
    "otherPatchFileNames":[
        "p28730253_180000_Linux-x86-64.zip",
        "p27539475_1813000DBRU_Linux-x86-64.zip",
        "p29213893_180000_Generic.zip",
        "p29374604_1813000DBRU_Linux-x86-64.zip",
        "p29782284_180000_Generic.zip",
        "p28125601_180000_Linux-x86-64.zip",
        "p28852325_180000_Linux-x86-64.zip",
        "p29997937_180000_Linux-x86-64.zip",
        "p31335037_180000_Linux-x86-64.zip",
        "p31335142_180000_Generic.zip"
    ]
    "installationParameters": {
        "unixGroupName": "dba",
        "unixGroupId": 12345,
        "unixUname": "oracle",
        "unixUid": 12345,
        "oracleHome": "/home/oracle/18.0.0.0.ru-2020-10.rur-2020-10.r1",
```

```
        "oracleBase": "/home/oracle/"
    }
}
```

Example Exemplo de manifesto de CEV para Oracle Database 19c

No exemplo a seguir do Oracle Database 19c, o RDS Custom aplica p32126828, depois p29213893 e depois p29782284 e assim por diante. O exemplo define novos valores para o usuário e o grupo do UNIX, o Oracle home e a base do Oracle.

```
{
    "mediaImportTemplateVersion": "2020-08-14",
    "databaseInstallationFileNames": [
        "V982063-01.zip"
    ],
    "opatchFileNames": [
        "p6880880_190000_Linux-x86-64.zip"
    ],
    "psuRuPatchFileNames": [
        "p32126828_190000_Linux-x86-64.zip"
    ],
    "otherPatchFileNames": [
        "p29213893_1910000DBRU_Generic.zip",
        "p29782284_1910000DBRU_Generic.zip",
        "p28730253_190000_Linux-x86-64.zip",
        "p29374604_1910000DBRU_Linux-x86-64.zip",
        "p28852325_190000_Linux-x86-64.zip",
        "p29997937_190000_Linux-x86-64.zip",
        "p31335037_190000_Linux-x86-64.zip",
        "p31335142_190000_Generic.zip"
    ],
    "installationParameters": {
        "unixGroupName": "dba",
        "unixGroupId": 12345,
        "unixUname": "oracle",
        "unixUid": 12345,
        "oracleHome": "/home/oracle/oracle.19.0.0.0.ru-2020-04.rur-2020-04.r1.EE.1",
        "oracleBase": "/home/oracle"
    }
}
```

Etapa 6 (opcional): Validar o manifesto CEV

Opcionalmente, verifique se o manifesto é um arquivo JSON válido executando o script Python `json.tool`. Por exemplo, se você mudar para o diretório que contém um manifesto CEV chamado `manifest.json`, execute o comando a seguir.

```
python -m json.tool < manifest.json
```

Etapa 7: Adicionar permissões do IAM necessárias

Certifique-se de que o elemento principal do IAM que cria o CEV tenha as políticas necessárias descritas em Etapa 5: Conceder as permissões necessárias ao usuário ou ao perfil do IAM.

Criar um CEV

É possível criar uma CEV usando o AWS Management Console ou a AWS CLI. Especifique a arquitetura multilocatário ou não multilocatário. Para ter mais informações, consulte Considerações sobre arquitetura multilocatário.

Em geral, criar um CEV demora cerca de duas horas. Depois que a CEV é criada, você pode usá-la para criar ou atualizar uma instância de banco de dados do RDS Custom. Para ter mais informações, consulte Criar uma instância de banco de dados do RDS Custom for Oracle e Fazer upgrade de uma instância de banco de dados para o RDS Custom for Oracle.

> **Note**
>
> Se sua instância de banco de dados atualmente usa o Oracle Linux 7.9, crie um novo CEV que use a AMI mais recente, que utiliza o Oracle Linux 8. Em seguida, modifique sua instância para usar o novo CEV.

Anote os seguintes requisitos e limitações para criação de uma CEV:

- O bucket do Amazon S3 que contém os arquivos de instalação deve estar na mesma Região da AWS que a CEV. Caso contrário, o processo de criação falhará.
- O nome da CEV deve estar no formato *`major-engine-version`*`.`*`customized_string`*, como em `19.cdb_cev1`.
- O nome da CEV deve conter de 1 a 50 caracteres alfanuméricos, sublinhados, traços ou pontos.

- O nome da CEV não pode conter pontos consecutivos, como em `19..cdb_cev1`.

Console

Para criar um CEV

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Custom engine versions (Versões de mecanismos personalizadas).

   A página Custom engine versions (Versões de mecanismo personalizadas) mostra todas as CEVs existentes. Se você não tiver criado uma CEV, a página estará vazia.
3. Escolha Create custom engine version (Criar versão de mecanismo personalizada).
4. Em Opções de mecanismo, faça o seguinte:

   a. Para Engine type (Tipo de mecanismo), escolha Oracle.

   b. Em Configurações de arquitetura, selecione Arquitetura multilocatário para criar uma CEV multilocatária, que usa o mecanismo de banco de dados `custom-oracle-ee-cdb` ou `custom-oracle-se2-cdb`. Você pode criar um RDS Custom para Oracle CDB somente com uma CEV multilocatária. Se você não escolher essa opção, sua CEV não será de CDB, que usa o mecanismo `custom-oracle-ee` ou `custom-oracle-se2`.

   > **Note**
   >
   > A arquitetura escolhida será uma característica permanente da CEV. Você não pode modificar sua CEV para usar uma arquitetura diferente posteriormente.

   c. Escolha uma das seguintes opções:

   - Crie um novo CEV — Crie um CEV do zero. Nesse caso, você deve especificar um manifesto do JSON especificando os binários do banco de dados.
   - Criar CEV a partir da fonte — Em Especificar o CEV que você deseja copiar, escolha um CEV existente para usar como o CEV de origem. Nesse caso, você pode especificar uma nova Amazon Machine Image (AMI), mas não pode especificar diferentes binários de banco de dados.

   d. Em Versão principal, selecione a versão principal do mecanismo.

5. Em Detalhes da versão, conclua o seguinte:

   a. Insira um nome válido em Nome da versão de mecanismo personalizada. Por exemplo, é possível inserir o nome **`19.cdb_cev1`**.

   b. (Opcional) Insira uma descrição para sua CEV.

6. Em Mídia de instalação, faça o seguinte:

   a. (Opcional) Para ID da AMI, deixe o campo em branco para usar a AMI mais recente do serviço ou insira uma AMI que você usou anteriormente para criar uma CEV. Para obter IDs de AMI usando uma das seguintes técnicas:

      - No console, selecione Versões personalizadas do mecanismo no painel de navegação esquerdo e escolha o nome de uma CEV. O ID da AMI usado pela CEV é exibido na guia Configuração.
      - Na AWS CLI, use o comando `describe-db-engine-versions`. Pesquise a saída de `ImageID`.

   b. Para S3 location of manifest files (Localização dos arquivos de manifesto no S3), insira o local do bucket do Amazon S3 que você especificou em Etapa 3: Fazer upload de seus arquivos de instalação no Amazon S3. Por exemplo, digite **`s3://my-custom-installation-files/123456789012/cev1/`**.

      > **Note**
      >
      > A região da na qual você executa deve estar na mesma região da que o bucket.

   c. Em Manifesto de CEV, insira o manifesto JSON que você criou em Criar o manifesto de CEV.

7. Na seção Chave do KMS, selecione Inserir um ARN de chave para listar as chaves do AWS KMS disponíveis. Em seguida, selecione sua chave do KMS na lista.

   Uma chave do AWS KMS é necessária para o RDS Custom. Para ter mais informações, consulte Etapa 1: Criar ou reutilizar uma chave AWS KMS de criptografia simétrica.

8. (Opcional) Selecione Adicionar nova tag para criar um par de chave/valor para sua CEV.

9. Escolha Create custom engine version (Criar versão de mecanismo personalizada).

   Se o manifesto CEV tiver um formulário inválido, o console exibirá Error validating the CEV manifest. Corrija os problemas e tente novamente.

A página Custom engine versions (Versões de mecanismo personalizadas) é exibida. Sua CEV é mostrada com o status Creating (Criando). O processo para criar o CEV demora aproximadamente duas horas.

AWS CLI

Para criar uma CEV utilizando a AWS CLI, execute o comando create-custom-db-engine-version.

São necessárias as seguintes opções:

- `--engine`: especifique o tipo de mecanismo. Para um que seja de CDB, especifique `custom-oracle-ee-cdb` ou `custom-oracle-se2-cdb`. Para um que não seja de CDB, especifique `custom-oracle-ee` ou `custom-oracle-se2`. É possível criar CDBs somente com base em uma CEV criada com `custom-oracle-ee-cdb` ou `custom-oracle-se2-cdb`. É possível criar não CDBs somente com base em uma CEV criada com `custom-oracle-ee` ou `custom-oracle-se2`.
- `--engine-version`— Especifique a versão do motor. O formato é a versão do *`major-engine-version.customized_string`*. O nome da CEV deve conter de 1 a 50 caracteres alfanuméricos, sublinhados, traços ou pontos. O nome da CEV não pode conter pontos consecutivos, como em `19..cdb_cev1`.
- `--kms-key-id` – Especifique um AWS KMS key.
- `--manifest` – Especifique *`manifest_json_string`* ou `--manifest file:`*`file_name`*. Caracteres de nova linha não são permitidos em *`manifest_json_string`*. Certifique-se de fazer o escape das aspas duplas (“) no código JSON, prefixando-as com uma barra invertida (\).

  O exemplo a seguir mostra o *`manifest_json_string`* para 19c de Etapa 5: Preparar o manifesto CEV. O exemplo define novos valores para a base Oracle, o Oracle Home e o ID e o nome do usuário e do grupo do UNIX/Linux. Se você copiar essa string, remova todos os caracteres de nova linha antes de colá-la no comando.

```
"{\"mediaImportTemplateVersion\": \"2020-08-14\",
\"databaseInstallationFileNames\": [\"V982063-01.zip\"],
\"opatchFileNames\": [\"p6880880_190000_Linux-x86-64.zip\"],
\"psuRuPatchFileNames\": [\"p32126828_190000_Linux-x86-64.zip\"],
\"otherPatchFileNames\": [\"p29213893_1910000DBRU_Generic.zip\",
\"p29782284_1910000DBRU_Generic.zip\",\"p28730253_190000_Linux-
x86-64.zip\",\"p29374604_1910000DBRU_Linux-x86-64.zip\",
\"p28852325_190000_Linux-x86-64.zip\",\"p29997937_190000_Linux-x86-64.zip
```

```
\",\"p31335037_190000_Linux-x86-64.zip\",\"p31335142_190000_Generic.zip
\"]\"installationParameters\":{ \"unixGroupName\":\"dba\",
\ \"unixUname\":\"oracle\", \ \"oracleHome\":\"/home/oracle/
oracle.19.0.0.0.ru-2020-04.rur-2020-04.r1.EE.1\", \ \"oracleBase\":\"/
home/oracle/\"}}"
```

- `--database-installation-files-s3-bucket-name`— Especifique o mesmo nome de bucket que você especificou emEtapa 3: Fazer upload de seus arquivos de instalação no Amazon S3. A Região da AWS na qual você executa o `create-custom-db-engine-version` deve ser a mesma de seu bucket do Amazon S3.

Também é possível especificar as seguintes opções:

- `--description`— Especifique uma descrição do seu CEV.
- `--database-installation-files-s3-prefix`— Especifique o nome da pasta que você especificou emEtapa 3: Fazer upload de seus arquivos de instalação no Amazon S3.
- `--image-id`— Especifique uma ID de AMI que deseja reutilizar. Para encontrar IDs válidos, execute o comando `describe-db-engine-versions` e, depois, pesquise a saída de `ImageID`. Por padrão, o RDS Custom para Oracle usa a AMI mais recente disponível.

O exemplo a seguir cria uma CEV multilocatária chamada `19.cdb_cev1`. O exemplo reutiliza uma AMI existente em vez de usar a AMI mais recente disponível. Certifique-se de que o nome do seu CEV comece com o número da versão principal do mecanismo.

Example

Para Linux, macOS ou Unix:

```
aws rds create-custom-db-engine-version \
    --engine custom-oracle-se2-cdb \
    --engine-version 19.cdb_cev1 \
    --database-installation-files-s3-bucket-name us-east-1-123456789012-custom-
installation-files \
    --database-installation-files-s3-prefix 123456789012/cev1 \
    --kms-key-id my-kms-key \
    --description "test cev" \
    --manifest manifest_string \
    --image-id ami-012a345678901bcde
```

Para Windows:

```
aws rds create-custom-db-engine-version ^
    --engine custom-oracle-se2-cdb ^
    --engine-version 19.cdb_cev1 ^
    --database-installation-files-s3-bucket-name us-east-1-123456789012-custom-
installation-files ^
    --database-installation-files-s3-prefix 123456789012/cev1 ^
    --kms-key-id my-kms-key ^
    --description "test cev" ^
    --manifest manifest_string ^
    --image-id ami-012a345678901bcde
```

Example

Obtenha detalhes sobre a sua CEV com o comando `describe-db-engine-versions`.

```
aws rds describe-db-engine-versions \
    --engine custom-oracle-se2-cdb \
    --include-all
```

A saída de exemplo parcial a seguir mostra o mecanismo, os grupos de parâmetros, o manifesto e outras informações.

```
{
    "DBEngineVersions": [
        {
            "Engine": "custom-oracle-se2-cdb",
            "EngineVersion": "19.cdb_cev1",
            "DBParameterGroupFamily": "custom-oracle-se2-cdb-19",
            "DBEngineDescription": "Containerized Database for Oracle Custom SE2",
            "DBEngineVersionDescription": "test cev",
            "Image": {
                "ImageId": "ami-012a345678901bcde",
                "Status": "active"
            },
            "ValidUpgradeTarget": [],
            "SupportsLogExportsToCloudwatchLogs": false,
            "SupportsReadReplica": true,
            "SupportedFeatureNames": [],
            "Status": "available",
            "SupportsParallelQuery": false,
            "SupportsGlobalDatabases": false,
```

```
            "MajorEngineVersion": "19",
            "DatabaseInstallationFilesS3BucketName": "us-east-1-123456789012-custom-
installation-files",
            "DatabaseInstallationFilesS3Prefix": "123456789012/cev1",
            "DBEngineVersionArn": "arn:aws:rds:us-east-1:123456789012:cev:custom-
oracle-se2-cdb/19.cdb_cev1/abcd12e3-4f5g-67h8-i9j0-k1234l56m789",
            "KMSKeyId": "arn:aws:kms:us-
east-1:732027699161:key/1ab2345c-6d78-9ef0-1gh2-3456i7j89k01",
            "CreateTime": "2023-03-07T19:47:58.131000+00:00",
            "TagList": [],
            "SupportsBabelfish": false,
...
```

Falha ao criar um CEV

Se o processo de criação do CEV falhar, o RDS Custom emitirá `RDS-EVENT-0198` com a mensagem `Creation failed for custom engine version` *`major-engine-version.cev_name`* e incluirá detalhes sobre a falha. Por exemplo, o evento imprime arquivos ausentes.

Não é possível modificar uma CEV com falha. Você pode apenas excluí-la e tentar novamente criar uma CEV depois de corrigir as causas da falha. Para obter informações sobre como solucionar os motivos da falha na criação da CEV, consulte Solucionar problemas com a criação de uma versão de mecanismo personalizado para o RDS Custom for Oracle.

Modificar o status da CEV

Você pode modificar uma CEV utilizando a AWS Management Console ou a AWS CLI. Você pode modificar a descrição da CEV ou seu status de disponibilidade. Sua CEV tem um dos seguintes valores de status:

- `available` – Você pode utilizar essa CEV para criar uma nova instância de banco de dados do RDS Custom ou fazer upgrade de uma instância de banco de dados. Esse é o status padrão para uma CEV recém-criada.
- `inactive` – Você não pode criar ou fazer upgrade de uma instância do RDS Custom com essa CEV. Não é possível restaurar um DB snapshot para criar uma nova instância de banco de dados do RDS Custom com essa CEV.

Você pode alterar a CEV de qualquer status com suporte para qualquer outro status com suporte. Você pode modificar o status para evitar o uso acidental de uma CEV ou tornar uma CEV

descontinuada qualificada para uso novamente. Por exemplo, é possível alterar o status da CEV de `available` para `inactive` e de `inactive` de volta para `available`.

Console

Para modificar uma CEV

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Custom engine versions (Versões de mecanismos personalizadas).
3. Escolha uma CEV cuja descrição ou status você queira modificar.
4. Para Actions (Ações), escolha Modify (Modificar).
5. Faça uma ou todas as alterações a seguir:
   - Para CEV status settings (Configurações de status da CEV), escolha um novo status de disponibilidade.
   - Para Version description (Descrição da versão), insira uma nova descrição.
6. Selecione Modify CEV (Modificar CEV).

   Se a CEV estiver em uso, o console exibirá You can't modify the CEV status (Não é possível modificar o status da CEV). Corrija os problemas e tente novamente.

A página Custom engine versions (Versões de mecanismo personalizadas) é exibida.

AWS CLI

Para modificar uma CEV utilizando a AWS CLI, execute o comando modify-custom-db-engine-version. Você pode encontrar CEVs para modificar executando o comando describe-db-engine-versions.

São necessárias as seguintes opções:

- `--engine` *`engine-type`*, em que o *`tipo de mecanismo`* é `custom-oracle-ee`, `custom-oracle-se2`, `custom-oracle-ee-cdb` ou `custom-oracle-se2-cdb`.
- `--engine-version` *`cev`*, em que *`cev`* é o nome da versão de mecanismo personalizada que você deseja modificar

- `--status` *`status`*, em que *`status`* é o status de disponibilidade que você deseja atribuir à CEV

O exemplo a seguir altera uma CEV chamada `19.my_cev1` do status atual para `inactive`.

Example

Para Linux, macOS ou Unix:

```
aws rds modify-custom-db-engine-version \
    --engine custom-oracle-se2 \
    --engine-version 19.my_cev1 \
    --status inactive
```

Para Windows:

```
aws rds modify-custom-db-engine-version ^
    --engine custom-oracle-se2 ^
    --engine-version 19.my_cev1 ^
    --status inactive
```

Visualizar detalhes da CEV com o Amazon RDS Custom para Oracle

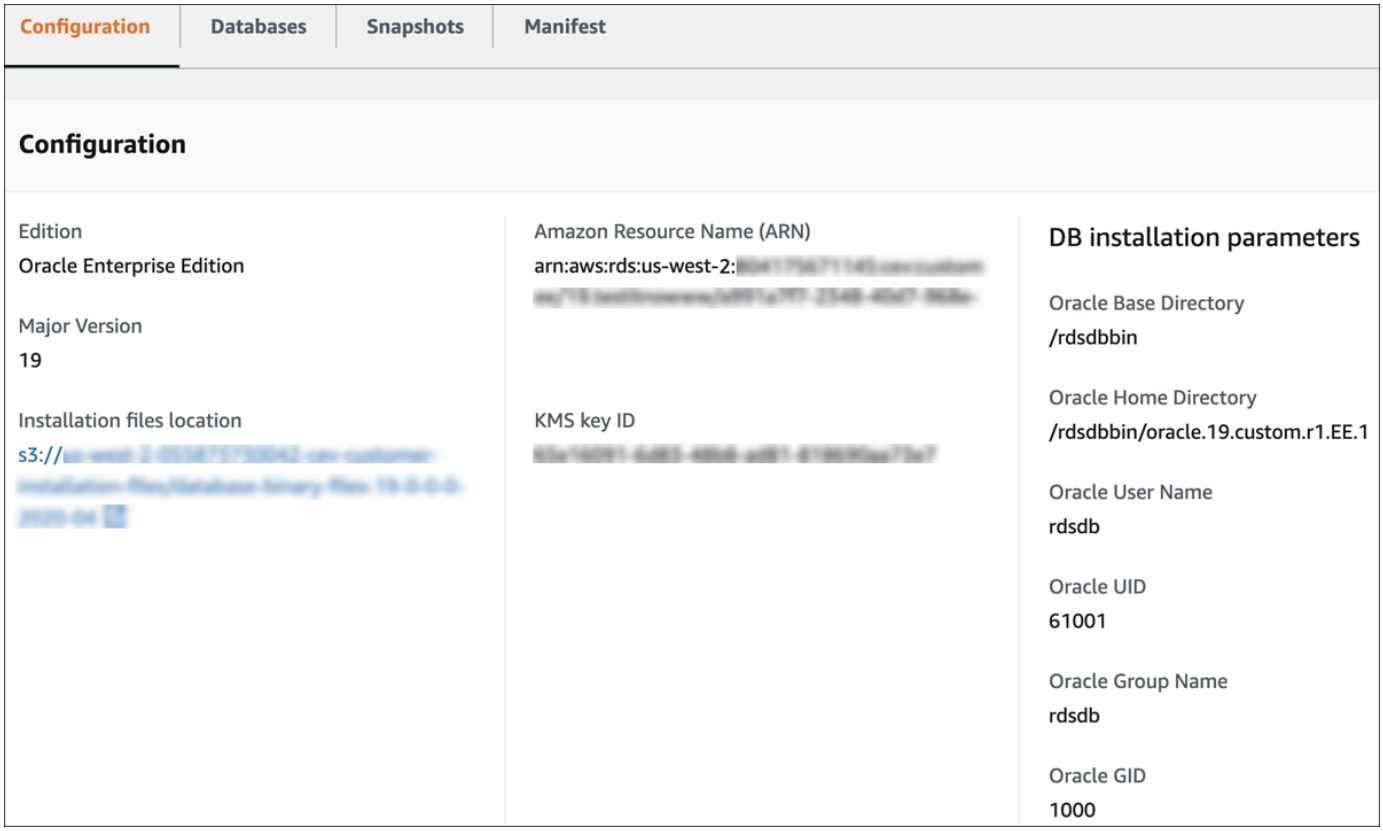
Você pode ver detalhes sobre seu manifesto da CEV e o comando usado para criar sua CEV com o uso do AWS Management Console ou da AWS CLI.

Console

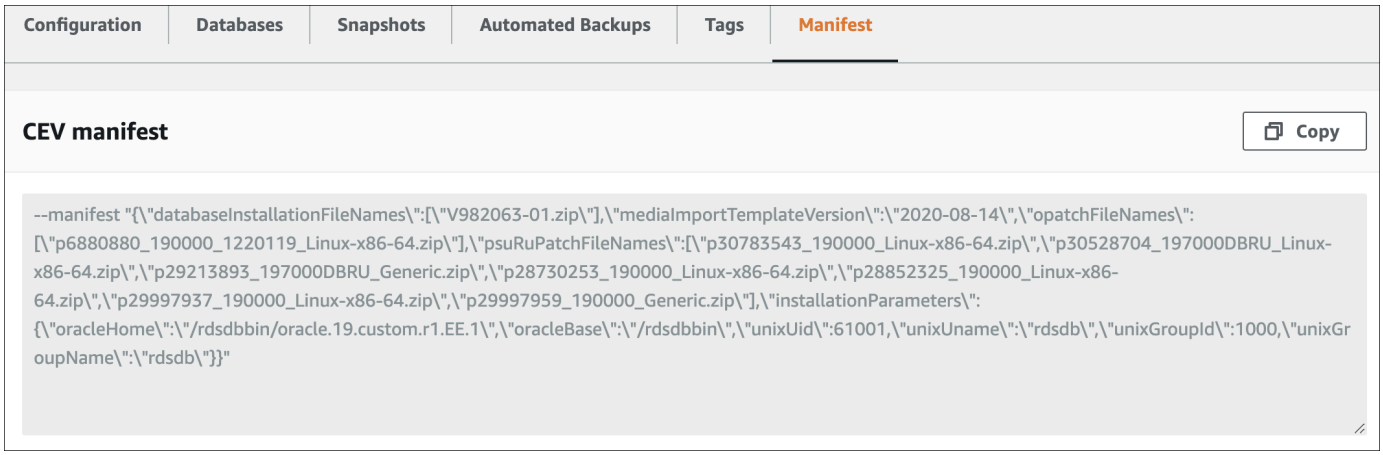
Como visualizar detalhes da CEV

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Custom engine versions (Versões de mecanismos personalizadas).

   A página Custom engine versions (Versões de mecanismo personalizadas) mostra todas as CEVs existentes. Se você não tiver criado uma CEV, a página estará vazia.
3. Selecione o nome da CEV que você deseja visualizar.
4. Selecione Configuration (Configuração) para visualizar os parâmetros de instalação especificados em seu manifesto.



5. Selecione Manifest (Manifesto) para visualizar os parâmetros de instalação especificados na opção `--manifest` do comando `create-custom-db-engine-version`. Você pode copiar esse texto, substituir valores conforme necessário e usá-los em um novo comando.



AWS CLI

Para ver detalhes sobre uma CEV com o uso da AWS CLI, execute o comando describe-db-engine-versions.

São necessárias as seguintes opções:

- `--engine` *`engine-type`*, em que o *`tipo de mecanismo`* é `custom-oracle-ee`, `custom-oracle-se2`, `custom-oracle-ee-cdb` ou `custom-oracle-se2-cdb`.
- `--engine-version` *`major-engine-version.customized_string`*

O exemplo a seguir cria uma CEV que não seja de CDB que usa a Enterprise Edition. O nome `19.my_cev1` da CVE começa com o número da versão principal do mecanismo, que é obrigatório.

Example

Para Linux, macOS ou Unix:

```
aws rds describe-db-engine-versions \
    --engine custom-oracle-ee \
    --engine-version 19.my_cev1
```

Para Windows:

```
aws rds describe-db-engine-versions ^
    --engine custom-oracle-ee ^
    --engine-version 19.my_cev1
```

A saída de exemplo parcial a seguir mostra o mecanismo, os grupos de parâmetros, o manifesto e outras informações.

```
"DBEngineVersions": [
    {
        "Engine": "custom-oracle-ee",
        "MajorEngineVersion": "19",
        "EngineVersion": "19.my_cev1",
        "DatabaseInstallationFilesS3BucketName": "us-east-1-123456789012-cev-customer-
installation-files",
        "DatabaseInstallationFilesS3Prefix": "123456789012/cev1",
        "CustomDBEngineVersionManifest": "{\n\"mediaImportTemplateVersion\":
 \"2020-08-14\",\n\"databaseInstallationFileNames\": [\n\"V982063-01.zip\"\n],
\n\"installationParameters\": {\n\"oracleBase\":\"/tmp\",\n\"oracleHome\":\"/
tmp/Oracle\"\n},\n\"opatchFileNames\": [\n\"p6880880_190000_Linux-x86-64.zip
\"\n],\n\"psuRuPatchFileNames\": [\n\"p32126828_190000_Linux-x86-64.zip
\"\n],\n\"otherPatchFileNames\": [\n\"p29213893_1910000DBRU_Generic.zip\",\n
```

```
\"p29782284_1910000DBRU_Generic.zip\",\n\"p28730253_190000_Linux-x86-64.zip\",\n
\"p29374604_1910000DBRU_Linux-x86-64.zip\",\n\"p28852325_190000_Linux-x86-64.zip\",
\n\"p29997937_190000_Linux-x86-64.zip\",\n\"p31335037_190000_Linux-x86-64.zip\",\n
\"p31335142_190000_Generic.zip\"\n]\n}\n",
        "DBParameterGroupFamily": "custom-oracle-ee-19",
        "DBEngineDescription": "Oracle Database server EE for RDS Custom",
        "DBEngineVersionArn": "arn:aws:rds:us-west-2:123456789012:cev:custom-oracle-
ee/19.my_cev1/0a123b45-6c78-901d-23e4-5678f901fg23",
        "DBEngineVersionDescription": "test",
        "KMSKeyId": "arn:aws:kms:us-east-1:123456789012:key/ab1c2de3-f4g5-6789-h012-
h3ijk4567l89",
        "CreateTime": "2022-11-18T09:17:07.693000+00:00",
        "ValidUpgradeTarget": [
        {
            "Engine": "custom-oracle-ee",
            "EngineVersion": "19.cev.2021-01.09",
            "Description": "test",
            "AutoUpgrade": false,
            "IsMajorVersionUpgrade": false
        }
]
```


Excluir uma CEV

É possível excluir uma CEV usando o AWS Management Console ou a AWS CLI. Normalmente, a exclusão demora alguns minutos.

Para excluir uma CEV, ela não pode estar em uso por nenhum dos seguintes procedimentos:

- Uma instância de banco de dados do RDS Custom
- Um snapshot de uma instância de banco de dados do RDS Custom
- Um backup automatizado da sua instância de banco de dados do RDS Custom

Console

Para excluir uma CEV

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Custom engine versions (Versões de mecanismos personalizadas).

3. Escolha uma CEV cuja descrição ou status você deseja excluir.
4. Em Ações, escolha Excluir.

   A caixa de diálogo Delete ***cev_name***? (Excluir cev_name?) é exibida.
5. Insira **`delete me`** e escolha Delete (Excluir).

   Na página Custom engine versions (Versões de mecanismos personalizadas), o banner mostra que sua CEV está sendo excluída.

AWS CLI

Para excluir uma CEV utilizando a AWS CLI, execute o comando delete-custom-db-engine-version.

São necessárias as seguintes opções:

- `--engine` *`engine-type`*, em que o *`tipo de mecanismo`* é `custom-oracle-ee`, `custom-oracle-se2`, `custom-oracle-ee-cdb` ou `custom-oracle-se2-cdb`.
- `--engine-version` *`cev`*, em que *`cev`* é o nome da versão de mecanismo personalizada a ser excluída

O exemplo a seguir exclui uma CEV chamada `19.my_cev1`.

Example

Para Linux, macOS ou Unix:

```
aws rds delete-custom-db-engine-version \
    --engine custom-oracle-ee \
    --engine-version 19.my_cev1
```

Para Windows:

```
aws rds delete-custom-db-engine-version ^
    --engine custom-oracle-ee ^
    --engine-version 19.my_cev1
```

Configurar uma instância de banco de dados para o Amazon RDS Custom para Oracle

Você pode criar uma instância de banco de dados do RDS Custom e, em seguida, conectar-se a ela via Secure Shell (SSH) ou AWS Systems Manager.

Consulte mais informações sobre como se conectar e fazer login em instância de banco de dados do RDS Custom para Oracle nos tópicos a seguir.

- Conectar-se à sua instância de banco de dados do RDS Custom usando o Session Manager
- Conectar-se à sua instância de banco de dados do RDS Custom via SSH
- Fazer login no banco de dados do RDS Custom para Oracle como SYS

Criar uma instância de banco de dados do RDS Custom for Oracle

Crie uma instância de banco de dados do Amazon RDS Custom for Oracle usando o AWS Management Console ou a AWS CLI. O procedimento é semelhante ao de criação de uma instância de banco de dados do Amazon RDS. Para ter mais informações, consulte Criar uma instância de banco de dados do Amazon RDS.

Se você incluiu parâmetros de instalação em seu manifesto da CEV, sua instância de banco de dados usará a base do Oracle, o Oracle home e o ID e o nome do usuário e do grupo do UNIX/Linux que você especificou. O arquivo `oratab`, criado pelo banco de dados Oracle durante a instalação, aponta para o local real da instalação em vez de para um link simbólico. Quando o RDS Custom para Oracle executa comandos, ele o faz como o usuário configurado do sistema operacional e não como o usuário padrão `rdsdb`. Para ter mais informações, consulte Etapa 5: Preparar o manifesto CEV.

Antes de tentar criar ou conectar-se a uma instância de banco de dados do RDS Custom, conclua as tarefas em Configurar seu ambiente para o Amazon RDS Custom for Oracle.

Console

Para criar uma instância de banco de dados do RDS Custom for Oracle

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Bancos de dados.
3. Escolha Create database (Criar banco de dados).

4. Em Choose a database creation method (Escolher um método de criação de banco de dados), selecione Standard Create (Criação padrão).
5. Na seção Opções de mecanismo, faça o seguinte:

   a. Para Engine type (Tipo de mecanismo), escolha Oracle.

   b. Para Database management type (Tipo de gerenciamento de banco de dados), escolha Amazon RDS Custom.

   c. Em Configurações da arquitetura, faça um dos seguintes procedimentos:

      - Selecione Arquitetura multilocatário para criar um banco de dados de contêiner (CDB). Na criação, o CDB contém uma semente de PDB e um PDB inicial.

        > **Note**
        >
        > A configuração Arquitetura multilocatário é compatível somente com o Oracle Database 19c.

      - Desmarque Arquitetura multilocatário para criar um não CDB. Um não CDB não pode conter PDBs.

   d. Em Edição, selecione Oracle Enterprise Edition ou Oracle Standard Edition 2.

   e. Para Versão de mecanismo personalizada, selecione uma versão de mecanismo personalizada do RDS Custom (CEV) existente. A CEV tem o seguinte formato: *`major-engine-version.customized_string`*. Um exemplo de identificador é `19.cdb_cev1`.

      Se você selecionou Arquitetura multilocatário na etapa anterior, só poderá especificar uma CEV que use o tipo de mecanismo `custom-oracle-ee-cdb` ou `custom-oracle-se2-cdb`. O console filtra as CEVs que foram criadas com diferentes tipos de mecanismo.
6. Em Templates, escolha Production (Produção).
7. Na seção Configurações, faça o seguinte:

   a. Em Identificador da instância de banco de dados, insira um nome exclusivo de sua instância de banco de dados.

   b. Em Nome de usuário principal, insira um nome de usuário. Você pode recuperar esse valor do console posteriormente.

      Quando você se conecta a um não CDB, o usuário principal é o usuário do não CDB. Quando você se conecta a um CDB, o usuário principal é o usuário do CDB. Para se

conectar à raiz do CDB, faça login no host, inicie um cliente SQL e crie um usuário administrativo com comandos SQL.

c. Desmarque a opção Gerar automaticamente uma senha.

8. Selecione uma classe de instância de banco de dados.

   Para conhecer as classes com suporte, consulte Suporte a classes de instâncias de banco de dados do RDS Custom for Oracle.

9. Na seção Storage (Armazenamento), faça o seguinte:

   a. Em Tipo de armazenamento, selecione um tipo de SSD: io1, gp2 ou gp3. Você tem as seguintes opções adicionais:

      - Para io1 ou gp3, selecione uma taxa para IOPS provisionadas. O padrão é 1.000 para io1 e 12.000 para gp3.
      - Para gp3, selecione uma taxa para Throughput de armazenamento. O padrão é 500 MiBps.

   b. Em Armazenamento alocado, selecione um tamanho de armazenamento. O padrão é de 40 GiB.

10. Em Conectividade, especifique sua nuvem privada virtual (VPC), grupo de sub-rede de banco de dados e grupo de segurança VPC (firewall).

11. Para RDS Custom security (Segurança do RDS Custom), faça o seguinte:

    a. Para IAM instance profile (Perfil da instância do IAM), selecione o perfil da instância de banco de dados do RDS Custom for Oracle.

       O perfil da instância do IAM deve começar com `AWSRDSCustom`, por exemplo *`AWSRDSCustomInstanceProfileForRdsCustomInstance`*.

    b. Para Encryption (Criptografia), escolha Enter a key ARN (Insira um ARN chave) para listar as chaves do AWS KMS disponíveis. Em seguida, escolha sua chave na lista.

       Uma chave do AWS KMS é necessária para o RDS Custom. Para ter mais informações, consulte Etapa 1: Criar ou reutilizar uma chave AWS KMS de criptografia simétrica.

12. Em Opções do banco de dados, faça o seguinte:

    a. (Opcional) Em ID do sistema (SID), insira um valor para o Oracle SID, que também é o nome do CDB. O SID é o nome da instância do banco de dados do Oracle que gerencia seus arquivos de banco de dados. Nesse contexto, o termo “instância de banco de dados

Oracle" se refere exclusivamente à área global do sistema (SGA) e aos processos em segundo plano da Oracle. Se você não especificar um SID, o valor assumirá **`RDSCDB`** como padrão.

b. (Opcional) Em Nome do banco de dados inicial, insira um nome. O valor padrão é **`ORCL`**. Na arquitetura multilocatário, o nome inicial do banco de dados é o nome do PDB.

> **Note**
>
> O SID e o nome do PDB devem ser diferentes.

c. Em Grupo de opções, selecione um grupo de opções ou aceite o padrão.

> **Note**
>
> A única opção compatível com o RDS Custom para Oracle é `Timezone`. Para ter mais informações, consulte Fuso horário da Oracle.

d. Em Período de retenção de backup, selecione um valor. Você não pode selecionar 0 dias.

e. Nas seções restantes, especifique suas configurações preferenciais de instâncias de banco de dados do RDS Custom. Para obter informações sobre cada configuração, consulte Configurações para instâncias de banco de dados. As configurações a seguir não têm suporte e não aparecem no console:

- Processor features (Recursos do processador)
- Storage autoscaling (Autoscaling do armazenamento)
- Opção Password and Kerberos authentication (Senha e autenticação Kerberos) em Database authentication (Autenticação do banco de dados) (somente a autenticação com senha tem suporte)
- Performance Insights
- Exportações de log
- Habilitar a atualização automática da versão secundária
- Deletion protection (Proteção contra exclusão)

13. Selecione Criar banco de dados.

> **Important**
>
> Ao criar uma instância de banco de dados do RDS Custom para Oracle, você pode receber o seguinte erro: The service-linked role is in the process of being created (A função vinculada ao serviço está no processo de criação). Tente novamente mais tarde. Se você fizer isso, aguarde alguns minutos e tente recriar a instância de banco de dados.

O botão View credential details (Visualizar detalhes de credenciais) aparece na página Databases (Bancos de dados).

Para visualizar o nome de usuário primário e a senha da instância de banco de dados do RDS Custom, escolha View credential details (Visualizar detalhes de credenciais) .

Para se conectar à instância de banco de dados como o usuário mestre, use o nome de usuário e a senha exibidos.

> **Important**
>
> Você não pode visualizar a senha do usuário principal novamente no console. Caso você não a registre, talvez seja necessário alterá-la. Para alterar a senha do usuário principal depois que a instância de banco de dados do RDS Custom estiver disponível, faça login no banco de dados e execute um comando `ALTER USER`. Você pode redefinir a senha usando a opção Modificar no console.

14. Escolha Databases (Bancos de dados) para visualizar a lista de instâncias de banco de dados do RDS Custom.
15. Escola a instância de banco de dados do RDS Custom que você acabou de criar.

    No console do RDS, os detalhes da nova instância de banco de dados do RDS Custom são exibidos:

    - A instância de banco de dados do RDS Custom apresenta o status Creating (Criando) até que esteja criada e pronta para uso. Quando o status muda para available (disponível), você pode se conectar à instância de banco de dados. Dependendo da classe da instância e do

armazenamento alocado, pode levar alguns minutos até que a nova instância de banco de dados fique disponível.

- Role (Função) tem o valor Instância (RDS Custom).
- RDS Custom automation mode (Modo de automação do RDS Custom) tem o valor Full automation (Automação total). Essa configuração significa que a instância de banco de dados fornece monitoramento automático e recuperação de instâncias.

AWS CLI

Você cria uma instância de banco de dados do RDS Custom utilizando o comando create-db-instance da AWS CLI.

São necessárias as seguintes opções:

- `--db-instance-identifier`
- `--db-instance-class` (para obter uma lista de classes de instância de banco de dados com suporte, consulte Suporte a classes de instâncias de banco de dados do RDS Custom for Oracle)
- `--engine` *`engine-type`*, em que o *`tipo de mecanismo`* é `custom-oracle-ee`, `custom-oracle-se2`, `custom-oracle-ee-cdb` ou `custom-oracle-se2-cdb`.
- `--engine-version` *`cev`* (em que *`cev`* é o nome da versão de mecanismo personalizada que você especificou em Criar um CEV)
- `--kms-key-id` *`my-kms-key`*
- `--backup-retention-period` *`days`* (em que *`days`* é um valor maior que `0`)
- `--no-auto-minor-version-upgrade`
- `--custom-iam-instance-profile AWSRDSCustomInstanceProfile-`*`us-east-1`* (em que *`region`* é a Região da AWS onde você está criando sua instância de banco de dados)

O exemplo a seguir cria uma instância de banco de dados do RDS Custom chamada `my-cfo-cdb-instance`. O banco de dados é um CDB com o nome não padrão *`MYCDB`*. O nome não padrão do PDB é *`MYPDB`*. Defina o período de retenção de backup como três dias.

Example

Para Linux, macOS ou Unix:

```
aws rds create-db-instance \
    --engine custom-oracle-ee-cdb \
```

```
    --db-instance-identifier my-cfo-cdb-instance \
    --engine-version 19.cdb_cev1 \
    --db-name MYPDB \
    --db-system-id MYCDB \
    --allocated-storage 250 \
    --db-instance-class db.m5.xlarge \
    --db-subnet-group mydbsubnetgroup \
    --master-username myuser \
    --master-user-password mypassword \
    --backup-retention-period 3 \
    --port 8200 \
    --kms-key-id my-kms-key \
    --no-auto-minor-version-upgrade \
    --custom-iam-instance-profile AWSRDSCustomInstanceProfile-us-east-1
```

Para Windows:

```
aws rds create-db-instance ^
    --engine custom-oracle-ee-cdb ^
    --db-instance-identifier my-cfo-cdb-instance ^
    --engine-version 19.cdb_cev1 ^
    --db-name MYPDB ^
    --db-system-id MYCDB ^
    --allocated-storage 250 ^
    --db-instance-class db.m5.xlarge ^
    --db-subnet-group mydbsubnetgroup ^
    --master-username myuser ^
    --master-user-password mypassword ^
    --backup-retention-period 3 ^
    --port 8200 ^
    --kms-key-id my-kms-key ^
    --no-auto-minor-version-upgrade ^
    --custom-iam-instance-profile AWSRDSCustomInstanceProfile-us-east-1
```

> **Note**
>
> Especifique uma senha diferente do prompt mostrado aqui como prática recomendada de segurança.

Obtenha detalhes sobre a sua instância com o comando da `describe-db-instances`.

Example

```
aws rds describe-db-instances --db-instance-identifier my-cfo-cdb-instance
```

A saída parcial a seguir mostra o mecanismo, os grupos de parâmetros e outras informações.

```
        {
            "DBInstanceIdentifier": "my-cfo-cdb-instance",
            "DBInstanceClass": "db.m5.xlarge",
            "Engine": "custom-oracle-ee-cdb",
            "DBInstanceStatus": "available",
            "MasterUsername": "admin",
            "DBName": "MYPDB",
            "DBSystemID": "MYCDB",
            "Endpoint": {
                "Address": "my-cfo-cdb-instance.abcdefghijkl.us-
east-1.rds.amazonaws.com",
                "Port": 1521,
                "HostedZoneId": "A1B2CDEFGH34IJ"
            },
            "AllocatedStorage": 100,
            "InstanceCreateTime": "2023-04-12T18:52:16.353000+00:00",
            "PreferredBackupWindow": "08:46-09:16",
            "BackupRetentionPeriod": 7,
            "DBSecurityGroups": [],
            "VpcSecurityGroups": [
                {
                    "VpcSecurityGroupId": "sg-0a1bcd2e",
                    "Status": "active"
                }
            ],
            "DBParameterGroups": [
                {
                    "DBParameterGroupName": "default.custom-oracle-ee-cdb-19",
                    "ParameterApplyStatus": "in-sync"
                }
            ],
...
```

Considerações sobre arquitetura multilocatário

Se você criar uma instância de banco de dados do Amazon RDS Custom para Oracle com a arquitetura multilocatário Oracle (tipo de mecanismo `custom-oracle-ee-cdb` ou `custom-oracle-se2-cdb`), seu banco de dados será um banco de dados de contêiner (CDB). Se você não especificar a arquitetura multilocatário Oracle, seu banco de dados será não CDB tradicional que usa o tipo de mecanismo `custom-oracle-ee` ou `custom-oracle-se2`. Um não CDB não pode conter bancos de dados conectáveis (PDBs). Para ter mais informações, consulte Arquitetura de banco de dados para o Amazon RDS Custom para Oracle.

Ao criar uma instância de CDB do RDS Custom para Oracle, considere o seguinte:

- Só é possível criar um banco de dados multilocatário usando uma CEV do Oracle Database 19c.
- Só será possível criar uma instância de CDB se a CEV usar o tipo de mecanismo `custom-oracle-ee-cdb` ou `custom-oracle-se2-cdb`.
- Se você criar uma instância de CDB usando a Standard Edition 2, o CDB poderá conter no máximo 3 PDBs.
- Por padrão, seu CDB recebe o nome de `RDSCDB`, que também é o nome do ID do Oracle System (Oracle SID). Você pode selecionar um nome diferente.
- O CDB contém somente um PDB inicial. O nome padrão do PDB é `ORCL`. Você pode selecionar um nome diferente para o PDB inicial, mas o Oracle SID e o nome do PDB não podem ser os mesmos.
- O RDS Custom para Oracle não fornece APIs para PDBs. Para criar PDBs adicionais, use o comando `CREATE PLUGGABLE DATABASE` do Oracle SQL. O RDS Custom para Oracle não restringe o número de PDBs que você pode criar. Em geral, você é responsável por criar e gerenciar PDBs, como em uma implantação on-premises.
- Não é possível usar as APIs do RDS para criar, modificar e excluir PDBs: é necessário usar as instruções SQL do Oracle. Se você criar um PDB usando o Oracle SQL, recomendamos gerar um snapshot manual posteriormente em caso de precisar fazer uma recuperação para um ponto no tempo (PITR).
- Você não pode renomear PDBs existentes usando as APIs do Amazon RDS. Você também não pode renomear o CDB usando o comando `modify-db-instance`.
- O modo aberto para a raiz do CDB é `READ WRITE` no banco de dados primário e `MOUNTED` em um banco de dados standby montado. O RDS Custom para Oracle tenta abrir todos os PDBs ao abrir o CDB. Se o RDS Custom para Oracle não conseguir abrir todos os PDBs, ele emitirá o evento `tenant database shutdown`.

Função vinculada ao serviço do RDS Custom

Uma função vinculada ao serviço concede ao Amazon RDS Custom acesso a recursos em sua Conta da AWS. Isso facilita a configuração do RDS Custom, pois você não precisa adicionar manualmente as permissões necessárias. O RDS Custom define as permissões de suas funções vinculadas a serviços e, exceto se definido de outra maneira, somente o RDS Custom pode assumir suas funções. As permissões definidas incluem a política de confiança e a política de permissões, e essa política de permissões não pode ser anexada a nenhuma outra entidade do IAM.

Quando você cria uma instância de banco de dados do RDS Custom, as funções vinculadas a serviços do Amazon RDS e do RDS Custom são criadas (se ainda não existirem) e utilizadas. Para ter mais informações, consulte Usar funções vinculadas ao serviço do Amazon RDS.

Ao criar uma instância de banco de dados do RDS Custom for Oracle pela primeira vez, você pode receber o seguinte erro: The service-linked role is in the process of being created (A função vinculada ao serviço está no processo de criação). Tente novamente mais tarde. Se você fizer isso, aguarde alguns minutos e tente recriar a instância de banco de dados.

Instalar componentes de software adicionais em uma instância de banco de dados do RDS Custom para Oracle

Em uma instância de banco de dados recém-criada, seu ambiente de banco de dados inclui binários Oracle, um banco de dados e um receptor de banco de dados. Talvez você queira instalar software adicional no sistema operacional host da instância de banco de dados. Por exemplo, talvez você queira instalar o Oracle Application Express (APEX), o agente do Oracle Enterprise Manager (OEM) ou o agente do Guardium S-TAP. Para obter diretrizes e instruções de alto nível, consulte a publicação detalhada no blog da AWS Instalar componentes de software adicionais no Amazon RDS Custom para Oracle.

Conectar-se à sua instância de banco de dados do RDS Custom usando o Session Manager

Depois de criar sua instância de banco de dados do RDS Custom, você pode conectar-se a ela usando o AWS Systems Manager Session Manager. Essa é a técnica preferida quando sua instância de banco de dados não está acessível publicamente.

O Session Manager possibilita que você acesse instâncias do Amazon EC2 por meio de um shell baseado no navegador ou viaAWS CLI. Para ter mais informações, consulte o AWS Systems Manager Session Manager.

Console

Para conectar-se à sua instância de banco de dados usando o Session Manager

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Databases (Bancos de dados) e a instância de banco de dados do RDS Custom à qual você deseja se conectar.
3. Escolher configuração.
4. Observe o Resource ID (ID do recurso) da instância de banco de dados. Por exemplo, o ID do recurso pode ser `db-ABCDEFGHIJKLMNOPQRS0123456`.
5. Abra o console do Amazon EC2 em https://console.aws.amazon.com/ec2/.
6. No painel de navegação, escolha Instances (Instâncias).
7. Procure o nome da sua instância do EC2 e depois clique no ID de instância associado a ela. Por exemplo, o ID da instância pode ser `i-abcdefghijklm01234`.
8. Selecione Conectar.
9. Escolha Session Manager.
10. Selecione Conectar.

    Uma janela é aberta para a sua sessão.

AWS CLI

Você pode conectar-se à sua instância de banco de dados do RDS Custom usando a AWS CLI. Essa técnica requer o plugin Session Manager para a AWS CLI. Para saber como instalar o plugin, consulte Instalar o plugin Session Manager para a AWS CLI.

Para encontrar o ID de recurso de banco de dados da sua instância de banco de dados do RDS Custom, use `aws rds describe-db-instances`.

```
aws rds describe-db-instances \
    --query 'DBInstances[*].[DBInstanceIdentifier,DbiResourceId]' \
    --output text
```

A saída de exemplo a seguir mostra o ID de recurso da sua instância do RDS Custom. O prefixo é `db-`.

```
db-ABCDEFGHIJKLMNOPQRS0123456
```

Para encontrar o ID de instância do EC2 da sua instância de banco de dados, use `aws ec2 describe-instances`. O exemplo a seguir usa `db-ABCDEFGHIJKLMNOPQRS0123456` para o ID do recurso.

```
aws ec2 describe-instances \
    --filters "Name=tag:Name,Values=db-ABCDEFGHIJKLMNOPQRS0123456" \
    --output text \
    --query 'Reservations[*].Instances[*].InstanceId'
```

O exemplo de saída a seguir mostra o ID da instância do EC2.

```
i-abcdefghijklm01234
```

Use o comando `aws ssm start-session`, fornecendo o ID da instância do EC2 no parâmetro `--target`.

```
aws ssm start-session --target "i-abcdefghijklm01234"
```

Um resultado bem-sucedido tem a seguinte aparência.

```
Starting session with SessionId: yourid-abcdefghijklm1234
[ssm-user@ip-123-45-67-89 bin]$
```

Conectar-se à sua instância de banco de dados do RDS Custom via SSH

O Secure Shell Protocol (SSH) é um protocolo de rede compatível com a comunicação criptografada em uma rede não segura. Depois de criar sua instância de banco de dados do RDS Custom, você pode se conectar a ela usando um cliente ssh. Para ter mais informações, consulte Conexão à sua instância do Linux utilizando SSH.

A técnica de conexão SSH depende de a instância de banco de dados ser privada, o que significa que ela não aceita conexões da Internet pública. Nesse caso, você deve usar o tunelamento SSH para conectar o utilitário ssh à sua instância. Essa técnica transporta dados com um fluxo de dados dedicado (túnel) dentro de uma sessão SSH existente. Você pode configurar o tunelamento SSH usando AWS Systems Manager.

> **Note**
>
> Várias estratégias são compatíveis para acessar instâncias privadas. Para saber como conectar um cliente ssh a instâncias privadas usando bastion hosts, consulte Linux Bastion Hosts na AWS. Para saber como configurar o encaminhamento de portas, consulte Encaminhamento de portas usando o AWS Systems Manager Session Manager.

Se sua instância de banco de dados estiver em uma sub-rede pública e tiver a configuração disponível publicamente, nenhum tunelamento SSH será necessário. Você pode se conectar a SSH da mesma forma que faria com uma instância pública do Amazon EC2.

Para conectar um cliente ssh à sua instância de banco de dados, conclua as seguintes etapas:

1. Etapa 1: Configurar sua instância de banco de dados para permitir conexões SSH
2. Etapa 2: Recuperar sua chave secreta SSH e o ID da instância do EC2
3. Etapa 3: Conectar-se à sua instância do EC2 usando o utilitário ssh

Etapa 1: Configurar sua instância de banco de dados para permitir conexões SSH

Para garantir que sua instância de banco de dados possa aceitar conexões SSH, faça o seguinte:

- Certifique-se de que o grupo de segurança da instância de banco de dados permita conexões de entrada na porta 22 para TCP.

  Para saber mais sobre como configurar o grupo de segurança de sua instância de banco de dados, consulte Controlar acesso com grupos de segurança.
- Se você não planeja usar o tunelamento SSH, sua instância de banco de dados deve residir em uma sub-rede pública e estar acessível publicamente.

  No console, o campo relevante é Acessível publicamente na guia Conectividade e segurança da página de detalhes do banco de dados. Para conferir suas configurações na CLI, execute o seguinte comando:

```
aws rds describe-db-instances \
--query 'DBInstances[*].
{DBInstanceIdentifier:DBInstanceIdentifier,PubliclyAccessible:PubliclyAccessible}' \
--output table
```

Para alterar as configurações de acessibilidade de sua instância de banco de dados, consulte nModificar uma instância de banco de dados do Amazon RDS.

Etapa 2: Recuperar sua chave secreta SSH e o ID da instância do EC2

Para conectar-se à instância de banco de dados usando SSH, você precisa do par de chaves associado a essa instância. O RDS Custom cria o par de chaves SSH em seu nome, usando a convenção de nomenclatura `do-not-delete-rds-custom-ssh-privatekey-`*`resource_id`*`-`*`uuid`* ou `rds-custom!oracle-do-not-delete-`*`resource_id`*`-`*`uuid`*`-ssh-privatekey`. O AWS Secrets Manager armazena sua chave privada SSH como um segredo.

Recupere sua chave secreta SSH usando o AWS Management Console ou a AWS CLI. Se sua instância tem um DNS público e você não pretende usar o tunelamento SSH, recupere também o nome do DNS. Você especifica o nome DNS para conexões públicas.

Console

Como recuperar a chave SSH secreta

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Databases (Bancos de dados) e a instância de banco de dados do RDS Custom à qual você deseja se conectar.
3. Escolher configuração.
4. Observe o valor de Resource ID (ID do recurso). Por exemplo, o ID do recurso de instância de banco de dados pode ser `db-ABCDEFGHIJKLMNOPQRS0123456`.
5. Abra o console do Amazon EC2 em https://console.aws.amazon.com/ec2/.
6. No painel de navegação, escolha Instances (Instâncias).
7. Localize o nome da sua instância do EC2 e escolha o ID de instância associado a ela. Por exemplo, o ID da instância do EC2 pode ser `i-abcdefghijklm01234`.
8. Em Details (Detalhes), localize Key pair name (Nome do par de chaves). O nome do par inclui o ID de recurso de instância de banco de dados. Por exemplo, o nome do par pode ser `do-not-delete-rds-custom-ssh-privatekey-db-ABCDEFGHIJKLMNOPQRS0123456-0d726c` ou `rds-custom!oracle-do-not-delete-db-ABCDEFGHIJKLMNOPQRS0123456-0d726c-ssh-privatekey`.

9. Se sua instância do EC2 for pública, observe o DNS IPv4 público. Para o exemplo, o endereço do Sistema de Nomes de Domínio (DNS) público pode ser `ec2-12-345-678-901.us-east-2.compute.amazonaws.com`.
10. Abra o console do AWS Secrets Manager em https://console.aws.amazon.com/secretsmanager/.
11. Escolha o segredo com nome idêntico ao seu par de chaves.
12. Escolha Recuperar valor do segredo.
13. Copie a chave privada SSH em um arquivo de texto e salve o arquivo com a extensão `.pem`. Por exemplo, salve o arquivo como `/tmp/do-not-delete-rds-custom-ssh-privatekey-db-ABCDEFGHIJKLMNOPQRS0123456-0d726c.pem` ou `/tmp/rds-custom!oracle-do-not-delete-db-ABCDEFGHIJKLMNOPQRS0123456-0d726c-ssh-privatekey.pem`.

AWS CLI

Para recuperar a chave privada SSH e salvá-la em um arquivo .pem, você pode usar a AWS CLI.

1. Encontre o ID de recurso de banco de dados de sua instância de banco de dados do RDS Custom usando `aws rds describe-db-instances`.

```
aws rds describe-db-instances \
    --query 'DBInstances[*].[DBInstanceIdentifier,DbiResourceId]' \
    --output text
```

   A saída de exemplo a seguir mostra o ID de recurso da sua instância do RDS Custom. O prefixo é `db-`.

```
db-ABCDEFGHIJKLMNOPQRS0123456
```

2. Encontre o ID de instância do EC2 da sua instância de banco de dados usando `aws ec2 describe-instances`. O exemplo a seguir usa `db-ABCDEFGHIJKLMNOPQRS0123456` para o ID do recurso.

```
aws ec2 describe-instances \
    --filters "Name=tag:Name,Values=db-ABCDEFGHIJKLMNOPQRS0123456" \
    --output text \
    --query 'Reservations[*].Instances[*].InstanceId'
```

   O exemplo de saída a seguir mostra o ID da instância do EC2.

```
i-abcdefghijklm01234
```

3. Para encontrar o nome da chave, especifique o ID da instância do EC2. O exemplo a seguir descreve a instância do EC2 `i-0bdc4219e66944afa`.

```
aws ec2 describe-instances \
    --instance-ids i-0bdc4219e66944afa \
    --output text \
    --query 'Reservations[*].Instances[*].KeyName'
```

A saída de exemplo a seguir mostra o nome da chave, que usa o formato de nomenclatura `do-not-delete-rds-custom-ssh-privatekey-`*`resource_id`*`-`*`uuid`* ou `rds-custom!oracle-do-not-delete-`*`resource_id`*`-`*`uuid`*`-ssh-privatekey`.

```
do-not-delete-rds-custom-ssh-privatekey-db-ABCDEFGHIJKLMNOPQRS0123456-0d726c
rds-custom!oracle-do-not-delete-db-ABCDEFGHIJKLMNOPQRS0123456-0d726c-ssh-privatekey
```

4. Salve a chave privada em um arquivo .pem com o nome dessa chave usando `aws secretsmanager`.

O exemplo a seguir salva a chave `do-not-delete-rds-custom-ssh-privatekey-db-ABCDEFGHIJKLMNOPQRS0123456-0d726c` em um arquivo no diretório `/tmp`.

```
aws secretsmanager get-secret-value \
    --secret-id do-not-delete-rds-custom-ssh-privatekey-db-
ABCDEFGHIJKLMNOPQRS0123456-0d726c \
    --query SecretString \
    --output text >/tmp/do-not-delete-rds-custom-ssh-privatekey-db-
ABCDEFGHIJKLMNOPQRS0123456-0d726c.pem
```

O exemplo a seguir salva a chave `rds-custom!oracle-do-not-delete-db-ABCDEFGHIJKLMNOPQRS0123456-0d726c-ssh-privatekey` em um arquivo no diretório `/tmp`.

```
aws secretsmanager get-secret-value \
    --secret-id rds-custom!oracle-do-not-delete-db-
ABCDEFGHIJKLMNOPQRS0123456-0d726c-ssh-privatekey \
    --query SecretString \
```

```
    --output text >/tmp/rds-custom!oracle-do-not-delete-db-
ABCDEFGHIJKLMNOPQRS0123456-0d726c-ssh-privatekey.pem
```

Etapa 3: Conectar-se à sua instância do EC2 usando o utilitário ssh

Sua técnica de conexão depende do fato de você estar se conectando a uma instância de banco de dados privada ou a uma instância pública. Uma conexão privada exige que você configure o tunelamento SSH por meio de AWS Systems Manager.

Como se conectar à sua instância do EC2 usando o utilitário ssh

1. Para conexões privadas, modifique seu arquivo de configuração SSH para comandos proxy para AWS Systems Manager Session Manager. Para conexões públicas, vá para a etapa 2.

   Adicione as linhas a seguir a `~/.ssh/config`. Essas linhas substituem comandos SSH para hosts cujos nomes começam com `i-` ou `mi-`.

   ```
   Host i-* mi-*
       ProxyCommand sh -c "aws ssm start-session --target %h --document-name AWS-
   StartSSHSession --parameters 'portNumber=%p'"
   ```

2. Mude para o diretório que contém seu arquivo .pem. Usando `chmod`, defina as permissões como `400`.

   O exemplo a seguir muda para o diretório `/tmp` e define permissões para o arquivo .pem `do-not-delete-rds-custom-ssh-privatekey-db-ABCDEFGHIJKLMNOPQRS0123456-0d726c.pem`.

   ```
   cd /tmp
   chmod 400 do-not-delete-rds-custom-ssh-privatekey-db-
   ABCDEFGHIJKLMNOPQRS0123456-0d726c.pem
   ```

   O exemplo a seguir muda para o diretório `/tmp` e define permissões para o arquivo .pem `rds-custom!oracle-do-not-delete-db-ABCDEFGHIJKLMNOPQRS0123456-0d726c-ssh-privatekey.pem`.

   ```
   cd /tmp
   chmod 400 rds-custom!oracle-do-not-delete-db-ABCDEFGHIJKLMNOPQRS0123456-0d726c-ssh-
   privatekey.pem
   ```

3. Execute o utilitário ssh, especificando o arquivo .pem e o nome do DNS público (para conexões públicas) ou o ID da instância do EC2 (para conexões privadas). Faça login como usuário `ec2-user`.

   O exemplo a seguir estabelece conexão com uma instância pública usando o nome DNS `ec2-12-345-678-901.us-east-2.compute.amazonaws.com`.

   ```
   # .pem file using naming prefix do-not-delete
   ssh -i \
     "do-not-delete-rds-custom-ssh-privatekey-db-
   ABCDEFGHIJKLMNOPQRS0123456-0d726c.pem" \
     ec2-user@ec2-12-345-678-901.us-east-2.compute.amazonaws.com

   # .pem file using naming prefix rds-custom!oracle-do-not-delete
   ssh -i \
     "rds-custom!oracle-do-not-delete-db-ABCDEFGHIJKLMNOPQRS0123456-0d726c-ssh-
   privatekey.pem" \
     ec2-user@ec2-12-345-678-901.us-east-2.compute.amazonaws.com
   ```

   O exemplo a seguir estabelece conexão com uma instância privada usando o ID de instância do EC2 `i-0bdc4219e66944afa`.

   ```
   # .pem file using naming prefix do-not-delete
   ssh -i \
     "do-not-delete-rds-custom-ssh-privatekey-db-
   ABCDEFGHIJKLMNOPQRS0123456-0d726c.pem" \
     ec2-user@i-0bdc4219e66944afa

   # .pem file using naming prefix rds-custom!oracle-do-not-delete
   ssh -i \
     "rds-custom!oracle-do-not-delete-db-ABCDEFGHIJKLMNOPQRS0123456-0d726c-ssh-
   privatekey.pem" \
     ec2-user@i-0bdc4219e66944afa
   ```

Fazer login no banco de dados do RDS Custom para Oracle como SYS

Depois de criar sua instância de banco de dados do RDS Custom, você pode fazer login no banco de dados do Oracle como usuário`SYS`, o que confere a você privilégios `SYSDBA`. Você tem as seguintes opções de login:

- Obtenha a senha SYS do Secrets Manager e especifique essa senha no seu cliente SQL.
- Use a autenticação do sistema operacional para fazer login em seu banco de dados. Nesse caso, você não precisa de uma senha.

Encontrar a senha do SYS para o banco de dados do RDS Custom para Oracle

Você pode fazer login em seu banco de dados Oracle como SYS ou SYSTEM ou especificando o nome de usuário principal em uma chamada de API. A senha para SYS e SYSTEM é armazenada no Secrets Manager.

O segredo usa o formato de nomenclatura do-not-delete-rds-custom-*resource_id*-*uuid* ou rds-custom!oracle-do-not-delete-*resource_id*-*uuid*. Você pode encontrar a senha usando o AWS Management Console.

Console

Como encontrar a senha SYS para seu banco de dados no Secrets Manager

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No console do RDS, conclua as seguintes etapas:
   a. No painel de navegação, escolha Bancos de dados.
   b. Selecione o nome da sua instância de banco de dados do RDS Custom para Oracle.
   c. Escolher configuração.
   d. Copie o valor abaixo do ID do recurso. Por exemplo, seu ID de recurso pode ser db-ABC12CDE3FGH4I5JKLMNO6PQR7.
3. Abra o console do Secrets Manager em https://console.aws.amazon.com/secretsmanager/.
4. No console do Secrets Manager, conclua as seguintes etapas:
   a. No painel de navegação à esquerda, selecione Segredos.
   b. Filtre os segredos pelo ID do recurso que você copiou na Etapa 2.d.
   c. Escolha o segredo que usa o formato de nomenclatura do-not-delete-rds-custom-***resource_id*-*uuid*** ou rds-custom!oracle-do-not-delete-***resource_id*-*uuid***. O *resource_id* é o ID do recurso que você copiou na etapa 2.d.

Por exemplo, se seu ID de recurso for db-ABC12CDE3FGH4I5JKLMNO6PQR7 e seu UUID for 1234ab, seu segredo será denominado do-not-delete-rds-custom-db-ABC12CDE3FGH4I5JKLMNO6PQR7-1234ab ou rds-custom!oracle-do-not-delete-db-ABC12CDE3FGH4I5JKLMNO6PQR7-1234ab.

d. Na seção Valor de segredo, selecione Recuperar o valor de segredo.

e. Em Chave/valor, copie o valor da senha.

5. Instale o SQL*Plus em sua instância de banco de dados e faça login em seu banco de dados como SYS. Para ter mais informações, consulte Etapa 3: Conectar seu cliente SQL a uma instância de banco de dados Oracle..

Fazer login no banco de dados do RDS Custom para Oracle usando a autenticação do sistema operacional

O usuário do sistema operacional `rdsdb` é responsável pelos binários do banco de dados Oracle. Você pode alternar para o usuário `rdsdb` e fazer login no seu banco de dados do RDS Custom para Oracle sem uma senha.

1. Conecte-se à sua instância de banco de dados com AWS Systems Manager. Para ter mais informações, consulte Conectar-se à sua instância de banco de dados do RDS Custom usando o Session Manager.
2. Em um navegador da Web, acesse https://www.oracle.com/database/technologies/instant-client/linux-x86-64-downloads.html.
3. Para obter a versão mais recente do banco de dados que aparece na página da Web, copie os links .rpm (não os links .zip) para o Instant Client Basic Package e o SQL*Plus Package. Por exemplo, os links a seguir são para o Oracle Database versão 21.9:

   - https://download.oracle.com/otn_software/linux/instantclient/219000/oracle-instantclient-basic-21.9.0.0.0-1.el8.x86_64.rpm
   - https://download.oracle.com/otn_software/linux/instantclient/219000/oracle-instantclient-sqlplus-21.9.0.0.0-1.el8.x86_64.rpm

4. Em sua sessão SSH, execute o comando `wget` para baixar os arquivos .rpm dos links que você obteve na etapa anterior. O exemplo a seguir baixa os arquivos .rpm para a versão 21.9 do Oracle Database:

```
wget https://download.oracle.com/otn_software/linux/instantclient/219000/oracle-
instantclient-basic-21.9.0.0.0-1.el8.x86_64.rpm
wget https://download.oracle.com/otn_software/linux/instantclient/219000/oracle-
instantclient-sqlplus-21.9.0.0.0-1.el8.x86_64.rpm
```

5. Instale os pacotes executando o comando `yum` da seguinte forma:

```
sudo yum install oracle-instantclient-*.rpm
```

6. Alterne para o usuário `rdsdb`.

```
sudo su - rdsdb
```

7. Faça login em seu banco de dados usando a autenticação do sistema operacional.

```
$ sqlplus / as sysdba

SQL*Plus: Release 21.0.0.0.0 - Production on Wed Apr 12 20:11:08 2023
Version 21.9.0.0.0

Copyright (c) 1982, 2020, Oracle.  All rights reserved.


Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.10.0.0.0
```

Gerenciar uma instância de banco de dados do Amazon RDS Custom for Oracle

O Amazon RDS Custom oferece suporte a um subconjunto das tarefas de gerenciamento habituais de instâncias de banco de dados Amazon RDS. A seguir, você encontra instruções para as tarefas de gerenciamento compatíveis do RDS Custom for Oracle utilizando o AWS Management Console e a AWS CLI.

Tópicos

- Trabalhar com bancos de dados de contêineres (CDBs) no RDS Custom para Oracle
- Trabalhar com recursos de alta disponibilidade para o RDS Custom for Oracle
- Personalizar seu ambiente do RDS Custom
- Modificar sua instância de banco de dados do RDS Custom para Oracle
- Alterar o conjunto de caracteres de uma instância de banco de dados do RDS Custom para Oracle
- Definir o valor NLS_LANG no RDS Custom para Oracle
- Suporte para o Transparent Data Encryption
- Marcar recursos do RDS Custom for Oracle
- Excluir uma instância de banco de dados do RDS Custom for Oracle

Trabalhar com bancos de dados de contêineres (CDBs) no RDS Custom para Oracle

É possível criar sua instância de banco de dados do RDS Custom para Oracle com a arquitetura de multilocação Oracle (tipo de mecanismo `custom-oracle-ee-cdb` ou `custom-oracle-se2-cdb`) ou com a arquitetura tradicional que não é de CDB (tipo de mecanismo `custom-oracle-ee` ou `custom-oracle-se2`). Quando você cria um banco de dados de contêiner (CDB), ele contém um banco de dados conectável (PDB) e uma semente de PDB. Você pode criar PDBs adicionais manualmente utilizando o Oracle SQL.

Nomes do PDB e do CDB

Ao criar uma instância de CDB do RDS Custom para Oracle, especifique um nome para o PDB inicial. Por padrão, seu PDB inicial é denominado `ORCL`. Você pode selecionar um nome diferente.

Por padrão, o CDB é denominado `RDSCDB`. Você pode selecionar um nome diferente. O nome do CDB também é o nome de seu identificador de sistema (SID) Oracle, que identifica de forma

exclusiva a memória e os processos que gerenciam seu CDB. Para ter mais informações sobre o Oracle SID, consulte Oracle System Identifier (SID) em Oracle Database Concepts.

Você não pode renomear PDBs existentes usando as APIs do Amazon RDS. Você também não pode renomear o CDB usando o comando `modify-db-instance`.

Gerenciamento de PDB

No modelo de responsabilidade compartilhada do RDS Custom para Oracle, você é responsável por gerenciar PDBs e criar quaisquer PDBs adicionais. O RDS Custom não restringe o número de PDBs. Você pode criar, modificar e excluir PDBs manualmente conectando-se à raiz do CDB e executando uma instrução SQL. Crie PDBs em um volume de dados do Amazon EBS para evitar que a instância de banco de dados saia do perímetro de suporte.

Para modificar seus CDBs ou PDBs, conclua estas etapas:

1. Pause a automação para evitar interferências nas ações do RDS Custom.
2. Modifique seu CDB ou PDBs.
3. Faça backup de todos os PDBs modificados.
4. Retome a automação do RDS Custom.

Recuperação automática da raiz do CDB

O RDS Custom mantém a raiz do CDB aberta da mesma forma que mantém uma raiz não CDB aberta. Se o estado da raiz do CDB mudar, a automação de monitoramento e recuperação tentará recuperar a raiz do CDB para o estado desejado. Você recebe notificações de eventos do RDS quando o CDB raiz é desligado (`RDS-EVENT-0004`) ou reiniciado (`RDS-EVENT-0006`), semelhante à arquitetura não CDB. O RDS Custom tenta abrir todos os PDBs no modo `READ WRITE` na inicialização da instância de banco de dados. Se não for possível abrir alguns PDBs, o RDS Custom publicará o seguinte evento: `tenant database shutdown`.

Trabalhar com recursos de alta disponibilidade para o RDS Custom for Oracle

O RDS Custom para Oracle oferece alta disponibilidade integrada por meio de implantações multi-AZ. Como alternativa, você pode usar o Oracle Data Guard como uma opção gerenciada pelo cliente com base em seus casos de uso.

Implantações multi-AZ (totalmente gerenciadas)

Em implantações multi-AZ para o RDS Custom para Oracle, o Amazon RDS provisiona e mantém automaticamente uma réplica em espera síncrona em outra zona de disponibilidade (AZ). A instância de banco de dados primária é replicada de forma síncrona nas AZs para uma réplica em espera com o objetivo de oferecer redundância de dados. Tanto a Enterprise Edition quanto a Standard Edition 2 permitem implantações multi-AZ. Para mais detalhes, consulte the section called “Gerenciar uma implantação multi-AZ para o RDS Custom para Oracle”.

Oracle Data Guard (gerenciado pelo cliente)

Como alternativa, você pode obter alta disponibilidade configurando manualmente o Oracle Data Guard para replicar dados entre instâncias de banco de dados do RDS Custom para Oracle. A instância de banco de dados primário sincroniza os dados automaticamente com as instâncias em espera. O Oracle Data Guard pode ser usado somente na Enterprise Edition.

É possível configurar o ambiente de alta disponibilidade das seguintes maneiras:

- Configure instâncias em espera em diferentes AZs para que elas sejam resilientes a falhas de AZ.
- Coloque seus bancos de dados em espera no modo somente leitura ou montado. O modo somente leitura requer uma licença do Oracle Active Data Guard.
- Failover ou alternância do banco de dados primário para um banco de dados em espera sem perda de dados.
- Migre dados configurando a alta disponibilidade para a sua instância on-premises e, em seguida, fazendo failover ou mudando para o banco de dados em espera do RDS Custom para Oracle.

Para saber como configurar o Oracle Data Guard para oferecer alta disponibilidade, consulte o Blog da AWS Build high availability for RDS Custom for Oracle using read replicas. Você pode executar as seguintes tarefas:

- Utilize um túnel de rede privada virtual (VPN) para criptografar dados em trânsito para as suas instâncias com alta disponibilidade. A criptografia em trânsito não é configurada automaticamente pelo RDS Custom para Oracle.
- Configure o Oracle Fast-Failover Observer (FSFO) para monitorar suas instâncias de alta disponibilidade.
- Permita que o observador realize o failover automático quando as condições necessárias forem atendidas.

Personalizar seu ambiente do RDS Custom

O RDS Custom para Oracle inclui recursos integrados que possibilitam personalizar seu ambiente de instância de banco de dados sem pausar a automação. Por exemplo, você pode usar as APIs do RDS para personalizar seu ambiente da seguinte forma:

- Crie e restaure snapshots de banco de dados para criar um ambiente de clonagem.
- Crie réplicas de leitura.
- Modifique as configurações de armazenamento.
- Altere a CEV para aplicar atualizações de versão

Para algumas personalizações, como alterar o conjunto de caracteres, não é possível usar as APIs do RDS. Nesses casos, você precisa alterar o ambiente manualmente acessando sua instância do Amazon EC2 como usuário raiz ou fazendo login em seu banco de dados Oracle como `SYSDBA`.

Para personalizar sua instância manualmente, você deve pausar e retomar a automação do RDS Custom. A pausa garante que suas personalizações não interfiram na automação do RDS Custom. Dessa forma, você evita quebrar o perímetro de suporte, o que coloca a instância no estado `unsupported-configuration` até que você corrija os problemas subjacentes. Pausar e retomar a automação são as únicas tarefas de automação compatíveis ao modificar uma instância de banco de dados do RDS Custom para Oracle.

Etapas gerais para personalizar seu ambiente do RDS Custom

Para personalizar sua instância de banco de dados do RDS Custom, conclua as seguintes etapas:

1. Pause a automação do RDS Custom por um período específico usando o console ou a CLI.
2. Identifique sua instância do Amazon EC2 subjacente.
3. Conecte-se à sua instância do Amazon EC2 usando chaves SSH ou AWS Systems Manager.
4. Verifique suas configurações atuais no banco de dados ou na camada do sistema operacional.

   Você pode validar suas alterações comparando a configuração inicial com a configuração alterada. Dependendo do tipo de personalização, use ferramentas do sistema operacional ou consultas ao banco de dados.
5. Personalize a instância de banco de dados do RDS Custom para Oracle, conforme necessário.
6. Reinicialize sua instância ou banco de dados, se necessário.

> **Note**
>
> Em um Oracle CDB on-premises, você pode preservar um modo aberto especificado para PDBs usando um comando integrado ou após um gatilho de inicialização. Esse mecanismo leva os PDBs a um estado especificado quando o CDB é reiniciado. Ao abrir seu CDB, a automação do RDS Custom sempre descarta todos os estados preservados especificados pelo usuário e tenta abrir todos os PDBs. Se o RDS Custom não conseguir abrir todos os PDBs, o seguinte evento será emitido: `The following PDBs failed to open: list-of-PDBs`.

7. Verifique suas novas configurações comparando-as com as configurações anteriores.
8. Retome a automação do RDS Custom das seguintes formas:
   - Retome a automação manualmente.
   - Aguarde o período de pausa terminar. Nesse caso, o RDS Custom retoma o monitoramento e a recuperação de instâncias automaticamente.
9. Verificar a framework de automação do RDS Custom

   Se você seguiu as etapas anteriores corretamente, o RDS Custom iniciará um backup automatizado. O status da instância no console mostra Disponível.

Para saber as práticas recomendadas e instruções detalhadas, consulte as publicações do blog AWS Fazer alterações na configuração de uma instância do Amazon RDS Custom para Oracle: Parte 1 e Recriar um banco de dados do Amazon RDS Custom para Oracle: Parte 2.

Pausar e retomar sua instância de banco de dados do RDS Custom

Você pode pausar e retomar a sua instância de banco de dados usando o console ou a CLI.

Console

Para pausar ou retomar a automação do RDS Custom

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Databases (Bancos de dados) e a instância de banco de dados do RDS Custom que você deseja modificar.

3. Selecione Modify. A página Modify DB instance (Modificar instância de banco de dados) será exibida.
4. Para RDS Custom automation mode (Modo de automação do RDS Custom), selecione uma das seguintes opções:

   - Paused (Pausado) pausa o monitoramento e a recuperação da instância de banco de dados do RDS Custom. Insira a duração da pausa que você deseja (em minutos) para Automation mode duration (Duração do modo de automação). O valor mínimo é 60 minutos (padrão). O valor máximo é 1.440 minutos.
   - Full automation (Automação total) retoma a automação.
5. Escolha Continue (Continuar) e verifique o resumo de modificações.

   Uma mensagem indica que o RDS Custom aplicará as alterações imediatamente.
6. Se estiverem corretas, escolha Modify DB Instance (Modificar instância de banco de dados). Ou escolha Back (Voltar) para editar as alterações ou Cancel (Cancelar) para cancelar as alterações.

   No console do RDS, são exibidos os detalhes da modificação. Se você pausou a automação, o Status da sua instância de banco de dados do RDS Custom indicará Automation paused (Automação pausada).
7. (Opcional) No painel de navegação, escolha Databases (Bancos de dados) e depois selecione sua instância de banco de dados do RDS Custom.

   No painel Summary (Resumo), RDS Custom automation mode (Modo de automação do RDS Custom) indica o status da automação. Se a automação estiver pausada, o valor será Paused (Pausado). Automação retomada em ***número*** minutos.

AWS CLI

Para pausar ou retomar a automação do RDS Custom, utilize o comando `modify-db-instance` da AWS CLI. Identifique a instância de banco de dados utilizando o parâmetro obrigatório `--db-instance-identifier`. Controle o modo de automação com os parâmetros a seguir:

- `--automation-mode` especifica o estado de pausa da instância de banco de dados. Os valores válidos são `all-paused`, que pausa a automação, e `full`, que retoma a automação.
- `--resume-full-automation-mode-minutes` especifica a duração da pausa. O valor padrão são 60 minutos.

> **Note**
>
> Independentemente de você especificar `--no-apply-immediately` ou `--apply-immediately`, o RDS Custom aplica modificações de maneira assíncrona logo que possível.

Na resposta do comando, `ResumeFullAutomationModeTime` indica o tempo de retomada como um carimbo de data/hora UTC. Quando o modo de automação é `all-paused`, é possível utilizar `modify-db-instance` para retomar o modo de automação ou estender o período de pausa. Nenhuma outra opção `modify-db-instance` tem suporte.

O exemplo a seguir pausa a automação de `my-custom-instance` por 90 minutos.

Example

Para Linux, macOS ou Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier my-custom-instance \
    --automation-mode all-paused \
    --resume-full-automation-mode-minutes 90
```

Para Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier my-custom-instance ^
    --automation-mode all-paused ^
    --resume-full-automation-mode-minutes 90
```

O exemplo a seguir estende a duração da pausa por mais 30 minutos. Os 30 minutos são adicionados à hora original mostrada em `ResumeFullAutomationModeTime`.

Example

Para Linux, macOS ou Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier my-custom-instance \
    --automation-mode all-paused \
    --resume-full-automation-mode-minutes 30
```

Para Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier my-custom-instance ^
    --automation-mode all-paused ^
    --resume-full-automation-mode-minutes 30
```

O exemplo a seguir retoma a automação completa de `my-custom-instance`.

Example

Para Linux, macOS ou Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier my-custom-instance \
    --automation-mode full \
```

Para Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier my-custom-instance ^
    --automation-mode full
```

Na seguinte saída parcial da amostra, o valor pendente de `AutomationMode` é `full`.

```
{
    "DBInstance": {
        "PubliclyAccessible": true,
        "MasterUsername": "admin",
        "MonitoringInterval": 0,
        "LicenseModel": "bring-your-own-license",
        "VpcSecurityGroups": [
            {
                "Status": "active",
                "VpcSecurityGroupId": "0123456789abcdefg"
            }
        ],
        "InstanceCreateTime": "2020-11-07T19:50:06.193Z",
        "CopyTagsToSnapshot": false,
        "OptionGroupMemberships": [
            {
                "Status": "in-sync",
```

```
                "OptionGroupName": "default:custom-oracle-ee-19"
            }
        ],
        "PendingModifiedValues": {
            "AutomationMode": "full"
        },
        "Engine": "custom-oracle-ee",
        "MultiAZ": false,
        "DBSecurityGroups": [],
        "DBParameterGroups": [
            {
                "DBParameterGroupName": "default.custom-oracle-ee-19",
                "ParameterApplyStatus": "in-sync"
            }
        ],
        ...
        "ReadReplicaDBInstanceIdentifiers": [],
        "AllocatedStorage": 250,
        "DBInstanceArn": "arn:aws:rds:us-west-2:012345678912:db:my-custom-instance",
        "BackupRetentionPeriod": 3,
        "DBName": "ORCL",
        "PreferredMaintenanceWindow": "fri:10:56-fri:11:26",
        "Endpoint": {
            "HostedZoneId": "ABCDEFGHIJKLMNO",
            "Port": 8200,
            "Address": "my-custom-instance.abcdefghijk.us-west-2.rds.amazonaws.com"
        },
        "DBInstanceStatus": "automation-paused",
        "IAMDatabaseAuthenticationEnabled": false,
        "AutomationMode": "all-paused",
        "EngineVersion": "19.my_cev1",
        "DeletionProtection": false,
        "AvailabilityZone": "us-west-2a",
        "DomainMemberships": [],
        "StorageType": "gp2",
        "DbiResourceId": "db-ABCDEFGHIJKLMNOPQRSTUVW",
        "ResumeFullAutomationModeTime": "2020-11-07T20:56:50.565Z",
        "KmsKeyId": "arn:aws:kms:us-west-2:012345678912:key/
aa111a11-111a-11a1-1a11-1111a11a1a1a",
        "StorageEncrypted": false,
        "AssociatedRoles": [],
        "DBInstanceClass": "db.m5.xlarge",
        "DbInstancePort": 0,
        "DBInstanceIdentifier": "my-custom-instance",
```

```
        "TagList": []
    }
```

Modificar sua instância de banco de dados do RDS Custom para Oracle

O procedimento para modificar uma instância de banco de dados do RDS Custom para Oracle é semelhante ao procedimento de modificação de uma instância do Amazon RDS. É possível alterar configurações como as seguintes:

- Classe de instância de banco de dados
- Alocação e tipo de armazenamento
- Backup retention period (Período de retenção de backup)
- Proteção contra exclusão
- Option group
- CEV (consulte Fazer upgrade de uma instância de banco de dados para o RDS Custom for Oracle)
- Port (Porta)

Tópicos

- Requisitos e limitações ao modificar o armazenamento de sua instância de banco de dados
- Requisitos e limitações ao modificar sua classe de instância de banco de dados
- Como o RDS Custom cria sua instância de banco de dados quando você modifica a classe da instância
- Modificar sua instância de banco de dados do RDS Custom para Oracle

Requisitos e limitações ao modificar o armazenamento de sua instância de banco de dados

Considere os requisitos e as limitações a seguir ao modificar o armazenamento de uma instância de banco de dados do RDS Custom para Oracle:

- O armazenamento mínimo alocado para o RDS Custom for Oracle é 40 GiB, e o máximo é 64 TiB.
- Como acontece com o Amazon RDS, não é possível diminuir o armazenamento alocado. Essa é uma limitação de volumes do Amazon EBS.
- A autoescalabilidade de armazenamento não tem suporte com instâncias de banco de dados RDS Custom.

- Todos os volumes de armazenamento anexados manualmente à sua instância de banco de dados do RDS Custom estão fora do perímetro de suporte.

  Para ter mais informações, consulte Perímetro de suporte do RDS Custom.
- O armazenamento magnético (padrão) do Amazon EBS não é compatível com o RDS Custom. Você pode selecionar somente os tipos de armazenamento SSD io1, gp2 ou gp3.

Para ter mais informações sobre o armazenamento do Amazon EBS, consulte Armazenamento de instâncias de banco de dados do Amazon RDS. Para obter informações gerais sobre a modificação do armazenamento, consulte Trabalhar com armazenamento para instâncias de banco de dados do Amazon RDS.

Requisitos e limitações ao modificar sua classe de instância de banco de dados

Considere os requisitos e as limitações a seguir ao modificar a classe de uma instância de banco de dados do RDS Custom para Oracle:

- Sua instância de banco de dados deve estar no estado `available`.
- Sua instância de banco de dados deve ter no mínimo 100 MiB de espaço livre no volume raiz, no volume de dados e no volume binário.
- Você pode atribuir somente um único IP elástico (EIP) à sua instância de banco de dados do RDS Custom para Oracle ao usar a interface de rede elástica (ENI) padrão. Se você anexar várias ENIs à sua instância de banco de dados, ocorrerá uma falha na operação de modificação.
- Todas as tags do RDS Custom para Oracle devem estar presentes.
- Se você usar a replicação do RDS Custom para Oracle, observe os seguintes requisitos e limitações:
  - Para instâncias de banco de dados primárias e réplicas de leitura, você pode alterar a classe da instância para somente uma instância de banco de dados por vez.
  - Se sua instância de banco de dados do RDS Custom para Oracle tiver um banco de dados primário ou réplica on-premises, atualize manualmente os endereços IP privados na instância de banco de dados on-premises após a conclusão da modificação. Essa ação é necessária para preservar a funcionalidade do Oracle DataGuard. O RDS Custom para Oracle publica um evento quando a modificação é bem-sucedida.
  - Você não pode modificar sua classe de instância de banco de dados do RDS Custom para Oracle quando as instâncias de banco de dados primárias ou de réplica de leitura têm o FSFO (Fast-Start Failover) configurado.

Como o RDS Custom cria sua instância de banco de dados quando você modifica a classe da instância

Como você modifica sua classe de instância, o RDS Custom cria sua instância de banco de dados da seguinte forma:

- Cria a instância do Amazon EC2.
- Cria um volume raiz do snapshot de banco de dados mais recente. O RDS Custom para Oracle não retém as informações adicionadas ao volume raiz após o último snapshot de banco de dados.
- Cria alarmes do Amazon CloudWatch.
- Criará um par de chaves SSH do Amazon EC2 se você tiver excluído o par de chaves original. Caso contrário, o RDS Custom para Oracle reterá o par de chaves original.
- Cria recursos usando as tags anexadas à sua instância de banco de dados quando você inicia a modificação. O RDS Custom não transfere tags para os novos recursos quando eles são anexados diretamente aos recursos subjacentes.
- Transfere os volumes binários e de dados com as modificações mais recentes na nova instância de banco de dados.
- Transfere o endereço IP elástico (EIP). Se a instância de banco de dados estiver acessível publicamente, o RDS Custom anexará temporariamente um endereço IP público à nova instância de banco de dados antes de transferir o EIP. Se a instância de banco de dados não estiver acessível ao público, o RDS Custom não criará endereços IP públicos.

Modificar sua instância de banco de dados do RDS Custom para Oracle

Você pode modificar a classe de instância de banco de dados ou o armazenamento usando o console, a AWS CLI ou a API do RDS.

Console

Como modificar uma instância de banco de dados do RDS Custom para Oracle

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Databases (Bancos de dados).
3. Escolha a instância de banco de dados que você deseja modificar.
4. Selecione Modify.

5. (Opcional) Em Configuração da instância, selecione um valor para Classe da instância de banco de dados. Para conhecer as classes com suporte, consulte Suporte a classes de instâncias de banco de dados do RDS Custom for Oracle.
6. (Opcional) Em Armazenamento, faça as seguintes alterações conforme necessário:
   a. Digite um novo valor em Allocated Storage (Armazenamento alocado). Ele deve ser maior que o valor atual e variar entre 40 GiB e 64 TiB.
   b. Altere o valor de Tipo de armazenamento para SSD de uso geral (gp2), SSD de uso geral (gp3) ou IOPS provisionadas (io1).
   c. Se você usar IOPS provisionadas (io1) ou SSD de uso geral (gp3), poderá alterar o valor de IOPS provisionadas.
7. (Opcional) Em Configuração adicional, faça o seguinte, conforme necessário:
   - Em Grupo de opções, selecione um novo grupo de opções. Para ter mais informações, consulte Trabalhar com grupos de opções no RDS Custom para Oracle.
8. Escolha Continue.
9. Escolha Apply immediately (Aplicar imediatamente) ou Apply during the next scheduled maintenance window (Aplicar durante a próxima janela de manutenção programada).
10. Selecione Modify DB instance (Modificar instância de banco de dados).

AWS CLI

Para modificar o armazenamento de uma instância de banco de dados do RDS Custom for Oracle, utilize o comando modify-db-instance da AWS CLI. Defina os seguintes parâmetros, conforme necessário:

- `--db-instance-class`: uma nova classe de instância. Para conhecer as classes com suporte, consulte Suporte a classes de instâncias de banco de dados do RDS Custom for Oracle.
- `--allocated-storage`: quantidade de armazenamento a ser alocada para a instância de banco de dados, em gibibytes. Ele deve ser maior que o valor atual e variar entre 40 GiB e 65.536 GiB.
- `--storage-type`: o tipo de armazenamento: gp2, gp3 ou io1.
- `--iops`: IOPS provisionadas para a instância de banco de dados, se você estiver usando os tipos de armazenamento io1 ou gp3.
- `--apply-immediately`: use `--apply-immediately` para aplicar as alterações de armazenamento de forma imediata.

Ou use `--no-apply-immediately` (o padrão) para aplicar as alterações durante a próxima janela de manutenção.

O exemplo a seguir altera a classe de instância de banco de dados de my-cfo-instance para db.m5.16xlarge. O comando também altera o tamanho do armazenamento para 1 TiB, o tipo de armazenamento para io1, as IOPS provisionadas para 3.000 e o grupo de opções para cfo-ee-19-mt.

Example

Para Linux, macOS ou Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier my-cfo-instance \
    --db-instance-class db.m5.16xlarge \
    --storage-type io1 \
    --iops 3000 \
    --allocated-storage 1024 \
    --option-group cfo-ee-19-mt \
    --apply-immediately
```

Para Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier my-cfo-instance ^
    --db-instance-class db.m5.16xlarge ^
    --storage-type io1 ^
    --iops 3000 ^
    --allocated-storage 1024 ^
    --option-group cfo-ee-19-mt ^
    --apply-immediately
```

Alterar o conjunto de caracteres de uma instância de banco de dados do RDS Custom para Oracle

O RDS Custom para Oracle usa como padrão o conjunto de caracteres US7ASCII. Talvez você queira especificar conjuntos de caracteres diferentes para atender aos requisitos de caracteres de idiomas ou multibyte. Ao usar o RDS Custom para Oracle, você pode pausar a automação e, depois, alterar o conjunto de caracteres do banco de dados manualmente.

Para alterar o conjunto de caracteres de uma instância de banco de dados do RDS Custom para Oracle, é necessário cumprir os seguintes requisitos:

- Você só pode alterar o caractere em uma instância do RDS Custom recém-provisionada que tenha um banco de dados vazio ou inicial sem dados da aplicação. Para todos os outros cenários, altere o conjunto de caracteres usando DMU (Assistente de migração de banco de dados para Unicode).
- Você só pode mudar para um conjunto de caracteres compatível com o RDS para Oracle. Para ter mais informações, consulte Conjuntos de caracteres de banco de dados compatíveis.

Como alterar o conjunto de caracteres de uma instância de banco de dados do RDS Custom para Oracle

1. Pause a automação do RDS Custom. Para ter mais informações, consulte Pausar e retomar sua instância de banco de dados do RDS Custom.
2. Faça login em seu banco de dados como usuário com privilégios SYSDBA.
3. Reinicie o banco de dados no modo restrito, altere o conjunto de caracteres e reinicie o banco de dados no modo normal.

   Execute o seguinte script no cliente SQL:

   ```
   SHUTDOWN IMMEDIATE;
   STARTUP RESTRICT;
   ALTER DATABASE CHARACTER SET INTERNAL_CONVERT AL32UTF8;
   SHUTDOWN IMMEDIATE;
   STARTUP;
   SELECT VALUE FROM NLS_DATABASE_PARAMETERS WHERE PARAMETER = 'NLS_CHARACTERSET';
   ```

   Verifique se a saída mostra o conjunto de caracteres correto:

   ```
   VALUE
   --------
   AL32UTF8
   ```
4. Retome a automação do RDS Custom. Para ter mais informações, consulte Pausar e retomar sua instância de banco de dados do RDS Custom.

Definir o valor NLS_LANG no RDS Custom para Oracle

Um local é um conjunto de informações que abordam requisitos linguísticos e culturais que correspondem a um determinado idioma e país. Para especificar o comportamento local do software Oracle, defina a variável de ambiente `NLS_LANG` no host do cliente. Essa variável define o idioma, o território e o conjunto de caracteres usados pela aplicação cliente e pelo servidor de banco de dados.

Para o RDS Custom para Oracle, você pode definir somente o idioma na variável `NLS_LANG`: os padrões de uso do território e de caractere. O idioma é usado para mensagens do banco de dados Oracle, agrupamento, nomes de dias e nomes de meses. Cada idioma compatível tem um nome exclusivo, por exemplo, americano, francês ou alemão. Se o idioma não for especificado, o valor padrão é americano.

Depois de criar seu banco de dados do RDS Custom para Oracle, você pode definir `NLS_LANG` no host do cliente um idioma diferente do inglês. Para ver uma lista de idiomas compatíveis com o Oracle Database, faça login em seu banco de dados do RDS Custom para Oracle e execute a seguinte consulta:

```
SELECT VALUE FROM V$NLS_VALID_VALUES WHERE PARAMETER='LANGUAGE' ORDER BY VALUE;
```

Você pode definir `NLS_LANG` na linha de comando do host. O exemplo a seguir define o idioma como alemão para sua aplicação cliente usando o shell Z no Linux.

```
export NLS_LANG=German
```

Sua aplicação lê o valor `NLS_LANG` quando é iniciado e, depois, o comunica ao banco de dados quando se conecta.

Para ter mais informações, consulte Selecionar uma localidade com a variável de ambiente NLS_LANG no Guia de suporte de globalização do Oracle Database.

Suporte para o Transparent Data Encryption

O RDS Custom é compatível com a Transparent Data Encryption (TDE) para instâncias de banco de dados do RDS Custom for Oracle.

Porém, você não pode habilitar a TDE utilizando uma opção em um grupo de opções personalizado como é possível fazer no RDS for Oracle. Você ativa o TDE manualmente. Para obter informações

sobre como usar o Oracle Transparent Data Encryption, consulte o tópico Securing stored data using Transparent Data Encryption (Proteger dados armazenados usando o Transparent Data Encryption) na documentação do Oracle.

Marcar recursos do RDS Custom for Oracle

Você pode marcar recursos do RDS Custom como faz com recursos do Amazon RDS, mas com algumas diferenças importantes:

- Não crie ou modifique a etiqueta `AWSRDSCustom` necessária para a automação do RDS Custom. Se você fizer isso, talvez interrompa a automação.
- A tag `Name` é adicionada aos recursos personalizados do RDS com o valor de prefixo `do-not-delete-rds-custom` ou `rds-custom!oracle-do-not-delete`. Qualquer valor passado pelo cliente para a chave é sobrescrito.
- As etiquetas adicionadas às instâncias de banco de dados RDS Custom durante a criação são propagadas para todos os outros recursos relacionados do RDS Custom.
- As etiquetas não são propagadas quando você as adiciona a recursos do RDS Custom após a criação da instância de banco de dados.

Para obter informações gerais sobre a marcação de recursos, consulte Marcar recursos do Amazon RDS.

Excluir uma instância de banco de dados do RDS Custom for Oracle

Para excluir uma instância de banco de dados do RDS Custom, faça o seguinte:

- Forneça o nome da instância de banco de dados.
- Desmarque a opção de obter um snapshot de banco de dados final da instância de banco de dados.
- Marque ou desmarque a opção para reter backups automatizados.

É possível excluir uma instância de banco de dados do RDS Custom utilizando o console ou a CLI. O tempo necessário para excluir uma instância de banco de dados pode variar dependendo do período de retenção do backup (ou seja, quantos backups serão excluídos) e da quantidade de dados excluídos.

Console

Para excluir uma instância de banco de dados do RDS Custom

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Databases (Bancos de dados) e a instância de banco de dados do RDS Custom que você deseja excluir. As instâncias de banco de dados RDS Custom mostram a função Instância (RDS Custom).
3. Em Ações, escolha Excluir.
4. Para reter backups automatizados, escolha Retain automated backups (Reter backups automatizados).
5. Digite **`delete me`** na caixa.
6. Escolha Excluir.

AWS CLI

Você exclui uma instância de banco de dados do RDS Custom utilizando o comando delete-db-instance da AWS CLI. Identifique a instância de banco de dados utilizando o parâmetro obrigatório `--db-instance-identifier`. Os parâmetros restantes são os mesmos de uma instância de banco de dados Amazon RDS, com as seguintes exceções:

- `--skip-final-snapshot` é necessário.
- `--no-skip-final-snapshot` não é compatível.
- `--final-db-snapshot-identifier` não é compatível.

O exemplo a seguir exclui a instância de banco de dados do RDS Custom chamada `my-custom-instance` e retém backups automatizados.

Example

Para Linux, macOS ou Unix:

```
aws rds delete-db-instance \
    --db-instance-identifier my-custom-instance \
    --skip-final-snapshot \
    --no-delete-automated-backups
```

Para Windows:

```
aws rds delete-db-instance ^
    --db-instance-identifier my-custom-instance ^
    --skip-final-snapshot ^
    --no-delete-automated-backups
```

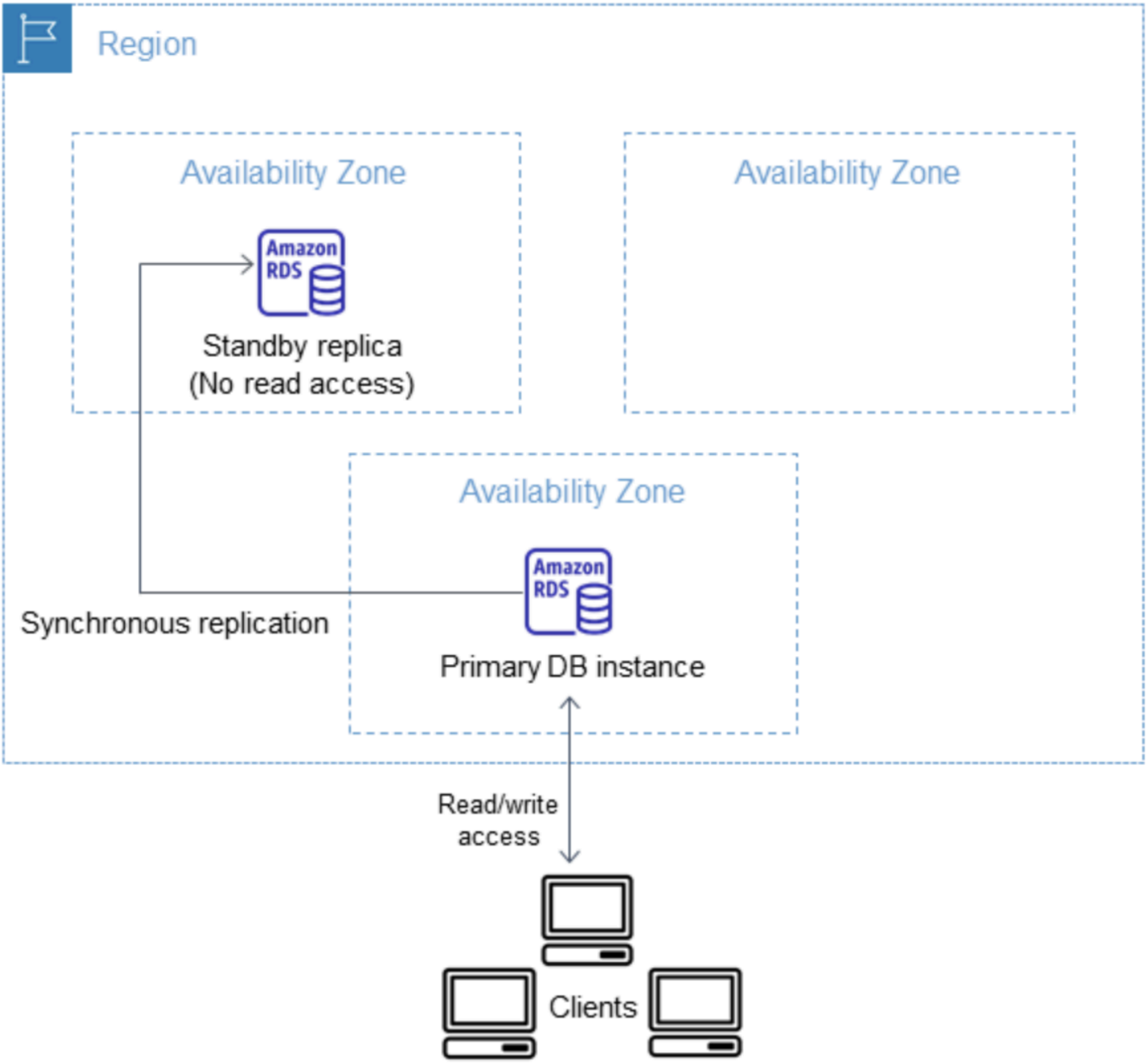
Gerenciar uma implantação multi-AZ para o RDS Custom para Oracle

Em uma implantação de instância de banco de dados multi-AZ para o RDS para Oracle, o Amazon RDS provisiona e mantém automaticamente uma réplica em espera síncrona em outra zona de disponibilidade (AZ). A instância de banco de dados primária é sincronicamente replicada nas zonas de disponibilidade para uma réplica em espera a fim de oferecer redundância de dados.

> **Important**
>
> Uma implantação multi-AZ para o RDS Custom para Oracle é diferente da multi-AZ para o RDS para Oracle. Ao contrário da multi-AZ para o RDS para Oracle, você deve configurar pré-requisitos para o RDS Custom para Oracle antes de criar sua instância de banco de dados multi-AZ, pois o RDS Custom é executado em sua própria conta, o que exige permissões. Além disso, você precisa criar a instância multi-AZ na CEV com a AMI mais recente fornecida pelo serviço que permita implantações multi-AZ.
> Se você não atender aos pré-requisitos, poderá ocorrer uma falha na execução de sua instância de banco de dados multi-AZ ou ela poderá ser revertida automaticamente para uma instância de banco de dados single-AZ. Para ter mais informações sobre pré-requisitos, consulte the section called “Pré-requisitos”.

Executar uma instância de banco de dados com alta disponibilidade pode aumentar a disponibilidade durante a manutenção planejada do sistema. No caso de uma manutenção planejada do banco de dados ou de uma interrupção não planejada do serviço, o Amazon RDS faz failover automático para a instância de banco de dados em espera atualizada. Essa funcionalidade permite que as operações do banco de dados sejam retomadas rapidamente sem intervenção manual. As instâncias primária e em espera usam o mesmo endpoint, cujo endereço de rede física faz a transição para a réplica em espera como parte do processo de failover. Não é necessário reconfigurar seu aplicativo quando ocorre um failover.

O seguinte diagrama mostra a arquitetura multi-AZ do RDS Custom para Oracle:



Você pode criar uma implantação multi-AZ do RDS Custom para Oracle especificando multi-AZ ao criar uma instância de banco de dados do RDS Custom. As instâncias de banco de dados do RDS Custom para Oracle criadas antes de 30 de junho de 2025 não podem ser convertidas em implantações multi-AZ modificando a instância de banco de dados e especificando a opção multi-AZ. Você precisa criar outra instância de banco de dados do RDS Custom para Oracle especificando a opção multi-AZ. As novas instâncias de banco de dados criadas usando a AMI fornecida pelo serviço após 30 de junho de 2025 oferecem a opção multi-AZ. Em seguida, você precisa migrar seu banco

de dados existente do RDS Custom para Oracle para a instância de banco de dados recém-criada. Para obter detalhes, consulte the section called “Migração para CEVs antes de 30/6/2025”.

O console do Amazon RDS mostra a zona de disponibilidade da réplica em espera (a AZ secundária). Você também pode usar o comando `describe-db-instances` da CLI ou a operação `DescribeDBInstances` da API para localizar a AZ secundária.

As instâncias de banco de dados do RDS Custom para Oracle com implantação multi-AZ podem ter maior latência de gravação e confirmação em comparação com uma implantação single-AZ. Esse aumento pode acontecer devido à replicação de dados síncrona entre as instâncias de banco de dados. É possível ter uma alteração na latência se sua implantação falhar na réplica em espera, ainda que o AWS seja desenvolvido com conectividade de rede de baixa latência entre zonas de disponibilidade.

Tópicos

- Disponibilidade de regiões e versões
- Limitações de uma implantação multi-AZ com o RDS Custom para Oracle
- Criar uma instância de banco de dados do RDS Custom para Oracle com uma implantação multi-AZ
- Pré-requisitos para uma implantação multi-AZ com o RDS Custom para Oracle
- Modificar uma implantação single-AZ do RDS Custom para Oracle para multi-AZ
- Modificar uma implantação multi-AZ do RDS Custom para Oracle para single-AZ
- Processo de failover para uma implantação multi-AZ do RDS Custom para Oracle
- Implantar o RDS Custom para Oracle com o AWS CloudFormation

Disponibilidade de regiões e versões

As seguintes edições do Oracle Database permitem implantações multi-AZ para o RDS Custom para Oracle:

- Oracle Database 12c Versão 1 (12.1)
- Oracle Database 12c Versão 2 (12.2)
- Oracle Database 19c
- Enterprise Edition (EE)
- Standard Edition 2 (SE2)

> **Note**
>
> O Oracle Database 18c não permite implantações multi-AZ para o RDS Custom para Oracle.

As implantações multi-AZ para o RDS Custom para Oracle estão disponíveis em todas as regiões em que o RDS Custom para Oracle está disponível. Para ter mais informações sobre a disponibilidade de regiões de implantações multi-AZ para o RDS Custom para Oracle, consulte the section called "Regiões e mecanismos de banco de dados compatíveis com o RDS Custom para Oracle".

Limitações de uma implantação multi-AZ com o RDS Custom para Oracle

As implantações multi-AZ do RDS Custom para Oracle têm as seguintes limitações:

- Implantações multi-AZ entre regiões não são compatíveis.
- Não é possível configurar a instância de banco de dado secundária para aceitar a atividade de leitura de banco de dados.
- Quando você usa uma versão personalizada do mecanismo (CEV) com uma implantação multi-AZ, a instância de banco de dados em espera também usa a mesma CEV. A instância de banco de dados em espera não pode usar uma CEV diferente.
- Não é possível criar uma réplica de leitura em uma instância de implantação multi-AZ nem modificar a instância primária da réplica de leitura para ter uma implantação multi-AZ.

Criar uma instância de banco de dados do RDS Custom para Oracle com uma implantação multi-AZ

Para criar uma instância de banco de dados do RDS Custom para Oracle com uma implantação multi-AZ e configurar seu ambiente com os pré-requisitos, siga as etapas em the section called "Configurar seu ambiente RDS Custom for Oracle".

> **Important**
>
> Para simplificar a configuração, recomendamos que você use o arquivo de modelo AWS CloudFormation mais recente fornecido nas instruções de configuração da rede. Para obter mais informações, consulte the section called "Implantação com AWS CloudFormation".

Você pode criar uma instância do RDS Custom para Oracle com uma implantação multi-AZ ao escolher a opção multi-AZ durante a criação da instância de banco de dados no console do Amazon RDS. Como alternativa, você pode especificar o parâmetro `--multi-az` no comando `create-db-instance` do Amazon RDS na AWS CLI.

Pré-requisitos para uma implantação multi-AZ com o RDS Custom para Oracle

> **Note**
>
> Se você criou sua instância RDS Custom para Oracle antes de 30 de junho de 2025, não poderá modificá-la de uma implantação single-AZ para multi-AZ. A versão personalizada do mecanismo (CEV) subjacente foi criada com uma AMI mais antiga fornecida pelo serviço que não permite a implantação multi-AZ.
> Você deve migrar seu banco de dados para uma nova instância usando uma CEV criada com uma AMI fornecida pelo serviço após 30 de junho de 2025.
> Consulte the section called “Migração para CEVs antes de 30/6/2025”.

O RDS Custom para Oracle tem pré-requisitos específicos para converter de uma implantação single-AZ para multi-AZ. Pré-requisitos não atendidos causarão uma falha na configuração da multi-AZ. Use a configuração manual ou o arquivo do modelo do AWS CloudFormation mais recente fornecido nas instruções de configuração da rede. Para obter mais informações, consulte the section called “Etapa 3: Extrair os modelos do CloudFormation para RDS Custom para Oracle”.

Para atender aos pré-requisitos manualmente, siga as etapas em the section called “Single-AZ para multi-AZ” e observe o seguinte:

- Garanta que a instância de banco de dados do RDS Custom para Oracle use uma CEV criada após 30 de junho de 2025 com a AMI mais recente fornecida pelo serviço.
- Atualize as regras de entrada e saída do grupo de segurança do Amazon RDS para permitir a porta 1120.
- Crie endpoints da VPC do Amazon SQS que permitam que a instância de banco de dados do RDS Custom para Oracle se comunique com o Amazon SQS.
- Atualize as permissões do Amazon SQS na função do perfil de instância.

> **Important**
>
> Não reinicie sua instância de banco de dados primária multi-AZ manualmente fazendo login na instância sem pausar a automação do RDS Custom para Oracle.

Etapas de migração para instâncias de banco de dados usando CEVs criadas antes de 30 de junho de 2025

Se você criou sua instância de banco de dados do RDS Custom para Oracle antes de 30 de junho de 2025, não poderá modificá-la diretamente para a implantação multi-AZ porque a CEV subjacente foi criada com uma AMI mais antiga fornecida pelo serviço que não permite a implantação multi-AZ. Você deve migrar seu banco de dados para uma nova instância usando uma CEV criada após 30 de junho de 2025. Faça o seguinte para fazer a migração:

1. Crie uma CEV da origem com a AMI mais recente fornecida pelo serviço (após 30 de junho de 2025).
2. Inicie uma nova instância de banco de dados com a nova CEV.
3. Migre seu banco de dados da instância de banco de dados existente que não permite a implantação multi-AZ para a instância recém-criada usando um dos seguintes métodos:
   - Migração física de bancos de dados Oracle para o Amazon RDS Custom usando o Data Guard
   - Migração física de bancos de dados Oracle para o Amazon RDS Custom usando a duplicação do RMAN

Modificar uma implantação single-AZ do RDS Custom para Oracle para multi-AZ

É possível modificar uma instância de banco de dados existente do RDS Custom para Oracle de uma implantação single-AZ para multi-AZ. Quando você modifica a instância de banco de dados, o Amazon RDS realiza várias ações:

- Tira um snapshot da instância de banco de dados primária.
- Cria volumes para a réplica em espera baseados no snapshot. Esses volumes são inicializados em segundo plano e a performance máxima do volume é alcançada depois que os dados são totalmente inicializados.
- Ativa a replicação síncrona em nível de bloco entre as instâncias de banco de dados primária e em espera.

> **Important**
>
> Recomendamos que você evite modificar sua instância de banco de dados do RDS Custom para Oracle de uma implantação single-AZ para multi-AZ em uma instância de banco de dados de produção durante períodos de pico de atividade.

A AWS usa um snapshot para criar a instância em espera para evitar tempo de inatividade ao converter da implantação single-AZ em multi-AZ. No entanto, poderá haver um impacto na performance durante e após a conversão em multi-AZ. Esse impacto pode ser significativo para workloads sensíveis à latência de gravação. Embora esse recurso permita que grandes volumes sejam restaurados rapidamente de snapshots, ele pode causar aumento na latência das operações de E/S devido à replicação síncrona. Essa latência pode afetar a performance do seu banco de dados.

Configurar pré-requisitos para modificar uma implantação single-AZ para multi-AZ usando o CloudFormation

Siga the section called “Etapa 3: Extrair os modelos do CloudFormation para RDS Custom para Oracle” para configurar novamente sua VPC e o perfil do IAM e adicionar o endpoint da VPC e a permissão do SQS no perfil do IAM.

Configurar pré-requisitos para modificar uma implantação single-AZ para multi-AZ manualmente

Se você optar por configurar recursos manualmente, realize as tarefas a seguir.

1. Abra o console da Amazon VPC, em https://console.aws.amazon.com/vpc/.
2. Selecione Endpoint. Aparecerá a página Create Endpoint (Criar endpoint).
3. Em Categoria do serviço, selecione Serviços da AWS.
4. Em Serviços, pesquise SQS.
5. Em VPC, escolha a VPC em que sua instância de banco de dados do RDS Custom para Oracle está implantada.
6. Em Sub-redes, escolha as sub-redes em que sua instância de banco de dados do RDS Custom para Oracle está implantada.
7. Em Grupos de segurança, escolha o grupo de segurança em que sua instância de banco de dados do RDS Custom para Oracle está implantada.
8. Em Política, selecione Personalizada

9. Em sua política personalizada, substitua `AWS partition`, `Region`, `accountId` e `IAM-Instance-role` por seus próprios valores.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Condition": {
                "StringLike": {
                    "aws:ResourceTag/AWSRDSCustom": "custom-oracle"
                }
            },
            "Action": [
                "SQS:SendMessage",
                "SQS:ReceiveMessage",
                "SQS:DeleteMessage",
                "SQS:GetQueueUrl"
            ],
            "Resource": "arn:${AWS::Partition}:sqs:${AWS::Region}:${AWS::AccountId}:do-
not-delete-rds-custom-*",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:${AWS::Partition}:iam::${AWS::AccountId}:role/{IAM-
Instance-role}"
            }
        }
    ]
}
```

Atualize o Perfil da instância com permissão para acessar o Amazon SQS. Substitua `AWS partition`, `Region` e `accountId` pelos seus próprios valores.

```
{
    "Sid": "13",
    "Effect": "Allow",
    "Action": [
        "SQS:SendMessage",
        "SQS:ReceiveMessage",
        "SQS:DeleteMessage",
        "SQS:GetQueueUrl"
    ],
    "Resource": [
```

```
        {
            "Fn::Sub": "arn:${AWS::Partition}:sqs:${AWS::Region}:${AWS::AccountId}:do-
not-delete-rds-custom-*"
        }
    ],
    "Condition": {
        "StringLike": {
            "aws:ResourceTag/AWSRDSCustom": "custom-oracle"
        }
    }
}
```

Atualize as regras de entrada e saída do grupo de segurança do Amazon RDS para permitir a porta 1120.

- Em Grupos de segurança, escolha o grupo em que sua instância de banco de dados do RDS Custom para Oracle está implantada.
- Em Regras de entrada, crie uma regra TCP personalizada para permitir a porta `1120` do grupo de origem.
- Em Regras de saída, crie uma regra TCP personalizada a fim de permitir a porta `1120` ao grupo de destino.

Modificar usando o console do RDS, a AWS CLI ou a API do RDS

Depois de atender aos pré-requisitos, você pode modificar uma instância de banco de dados do RDS Custom para Oracle de uma implantação single-AZ para multi-AZ usando o console do Amazon RDS, a AWS CLI ou a API do RDS.

Console

Como modificar uma implantação do RDS Custom para Oracle de single-AZ para multi-AZ

1. Faça login no Console de Gerenciamento da AWS e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No console do Amazon RDS, escolha Databases (Bancos de dados). O painel Databases (Bancos de dados) é exibido.
3. Selecione a instância de banco de dados do RDS Custom para Oracle que você deseja modificar.
4. Em Ações, selecione Converter em implantação multi-AZ.

5. Na página Confirmação, selecione Aplicar imediatamente para aplicar as alterações imediatamente. A escolha dessa opção não causa tempo de inatividade, mas pode causar um possível impacto na performance. Você também pode optar por aplicar a atualização durante a próxima janela de manutenção. Para obter mais informações, consulte the section called “Programação de modificações”.
6. Na página Confirmação, selecione Converter em multi-AZ.

AWS CLI

Para converter em uma implantação de instância de banco de dados multi-AZ usando oAWS CLI, chame o comando modify-db-instance e defina a opção `--multi-az`. Especifique o identificador da instância de banco de dados e os valores para as outras opções que deseja modificar. Para ter mais informações sobre cada opção, consulte the section called “Configurações disponíveis”.

Example Exemplo

O código a seguir modifica `mycustomdbinstance` incluindo a opção `--multi-az`. As alterações serão aplicadas durante a janela de manutenção usando `--no-apply-immediately`. Use `--apply-immediately` para aplicar as alterações imediatamente. Para obter mais informações, consulte the section called “Programação de modificações”.

Para Linux, macOS ou Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier mycustomdbinstance \
    --multi-az \
    [--no-apply-immediately | --apply-immediately]
```

Para Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mycustomdbinstance ^
    --multi-az ^
    [--no-apply-immediately | --apply-immediately]
```

API do RDS

Para converter uma implantação de instância de banco de dados em multi-AZ com a API do Amazon RDS, chame a operação ModifyDBInstance e defina o parâmetro `MultiAZ` como verdadeiro.

Modificar uma implantação multi-AZ do RDS Custom para Oracle para single-AZ

É possível modificar uma instância de banco de dados do RDS Custom para Oracle de uma implantação multi-AZ para single-AZ.

Console

Como modificar uma instância de banco de dados do RDS Custom para Oracle de uma implantação multi-AZ para single-AZ

1. Faça login no Console de Gerenciamento da AWS e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No console do Amazon RDS, escolha Databases (Bancos de dados). O painel Databases (Bancos de dados) é exibido.
3. Selecione a instância de banco de dados do RDS Custom para Oracle que você deseja modificar.
4. Em Implantação multi-AZ, selecione Não.
5. Na página Confirmação, selecione Aplicar imediatamente para aplicar as alterações imediatamente. A escolha dessa opção não causa tempo de inatividade, mas pode causar um possível impacto na performance. Você também pode optar por aplicar a atualização durante a próxima janela de manutenção. Para obter mais informações, consulte the section called “Programação de modificações”.
6. Na página Confirmação, selecione Modificar instância de banco de dados.

AWS CLI

Para modificar uma implantação multi-AZ para single-AZ usando oAWS CLI, chame o comando modify-db-instance e inclua a opção `--no-multi-az`. Especifique o identificador da instância de banco de dados e os valores para as outras opções que deseja modificar. Para ter mais informações sobre cada opção, consulte the section called “Configurações disponíveis”.

Example Exemplo

O código a seguir modifica `mycustomdbinstance` incluindo a opção `--no-multi-az`. As alterações serão aplicadas durante a janela de manutenção usando `--no-apply-immediately`. Use `--apply-immediately` para aplicar as alterações imediatamente. Para obter mais informações, consulte the section called “Programação de modificações”.

Para Linux, macOS ou Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier mycustomdbinstance \
    --no-multi-az \
    [--no-apply-immediately | --apply-immediately]
```

Para Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mycustomdbinstance ^
    --no-multi-az ^
    [--no-apply-immediately | --apply-immediately]
```

API do RDS

Para modificar uma implantação multi-AZ para single-AZ usando a API do Amazon RDS, chame a operação ModifyDBInstance e defina o parâmetro `MultiAZ` como `false`.

Processo de failover para uma implantação multi-AZ do RDS Custom para Oracle

Se uma interrupção planejada ou não planejada da sua instância de banco de dados for o resultado de um defeito de infraestrutura, o Amazon RDS alternará automaticamente para uma réplica em espera em outra zona de disponibilidade se você tiver ativado o multi-AZ. O tempo de conclusão do failover depende da atividade do banco de dados e de outras condições no momento em que a instância de banco de dados primária se tornou indisponível. Em geral, os tempos de failover variam de 60 a 120 segundos. No entanto, transações grandes ou um processo de recuperação longo podem aumentar o tempo de failover. Quando o failover é concluído, o console do Amazon RDS pode levar mais um tempo para mostrar a nova zona de disponibilidade.

> **Note**
>
> Você pode forçar um failover manualmente ao interromper e iniciar o host primário do EC2 enquanto a instância de banco de dados estiver disponível.

O Amazon RDS processa os failovers automaticamente para que você possa retomar as operações de banco de dados o mais rápido possível e sem intervenção administrativa. A instância de banco de dados principal muda automaticamente para a réplica em espera se alguma das condições descritas

na tabela a seguir ocorrer. Os motivos do failover podem ser visualizados no log de eventos do Amazon RDS.

Motivo do failover	Descrição
O sistema operacional subjacente à instância de banco de dados do RDS está sendo corrigido em uma operação offline.	Um failover foi acionado durante a janela de manutenção para um patch de SO ou uma atualização de segurança. Para obter mais informações, consulte the section called “Manutenção de uma instância de banco de dados”.
O host principal da instância RDS multi-AZ não está íntegro.	A implantação de instância de banco de dados multi-AZ detectou uma instância de banco de dados primária danificada e executou failover.
O host principal da instância RDS multi-AZ está inacessível devido à perda de conectividade de rede.	O monitoramento do RDS detectou uma falha de alcançabilidade de rede na instância de banco de dados principal e acionou um failover.
A instância do RDS foi modificada pelo cliente.	Uma modificação da instância de banco de dados do RDS acionou um failover. Para obter mais informações, consulte the section called “Modificar sua instância de banco de dados”.
O volume de armazenamento subjacente ao host principal da instância multi-AZ do RDS sofreu uma falha.	A implantação de instância de banco de dados multi-AZ detectou um problema de armazenamento na instância de banco de dados primária e executou o failover.
A instância primária do RDS multi-AZ está ocupada e não responde.	A instância de banco de dados principal não responde. Recomendamos que você faça o seguinte: examine o evento e os logs do CloudWatch quanto ao uso excessivo de CPU, memória ou espaço de troca. Para ter mais informações, consulte the section called “Trabalhar com a notificação de eventos do Amazon RDS” e the section called “Criar uma regra que é acionada em um evento do Amazon RDS”. Avalie sua workload para determinar se você está usando a classe de instância de banco de dados apropriada. Para obter mais informações, consulte

Motivo do failover	Descrição
	the section called “Classes da instância de banco de dados”.

Para determinar se ocorreu failover na instância de banco de dados multi-AZ, faça o seguinte:

- Configure assinaturas de eventos de banco de dados para notificar você por e-mail ou SMS de que um failover foi iniciado. Para ter mais informações sobre eventos do , consulte the section called “Trabalhar com a notificação de eventos do Amazon RDS”.
- Visualize seus eventos de banco de dados usando o console do Amazon RDS ou operações de API.
- Visualize o estado atual da implantação da instância de banco de dados multi-AZ do RDS Custom para Oracle usando o console do Amazon RDS, a CLI ou operações de API.

Configurações de vida útil (TTL) com aplicações usando uma implantação multi-AZ do RDS Custom para Oracle

O mecanismo de failover modifica automaticamente o registro de Domain Name System (DNS) da instância de banco de dados para apontar para a instância de banco de dados em espera. Como resultado, você precisará restabelecer todas as conexões existentes para sua instância de banco de dados. Garanta que qualquer valor de configuração de tempo de vida útil do cache DNS (TTL) seja baixo e confirme que sua aplicação não armazene o DNS em cache por um período prolongado. Um valor alto de TTL pode impedir que sua aplicação se reconecte rapidamente à instância de banco de dados após o failover.

Implantar o RDS Custom para Oracle com o AWS CloudFormation

Automatize a implantação do RDS Custom para Oracle usando o modelo do AWS CloudFormation fornecido. Atenda aos pré-requisitos a seguir antes de implantar os recursos.

Pré-requisitos

Baixar os arquivos necessários do Oracle

Você precisa de arquivos de instalação específicos do Oracle para criar o modelo do AWS CloudFormation. Baixe esses arquivos antes da implantação.

1. Navegue até Oracle Database 19c (19.3).

2. Localize e baixe o arquivo `LINUX.X64_193000_db_home.zip`.
3. Renomeie o arquivo para `V982063-01.zip`.
4. Baixe os patches restantes, selecionando Plataforma ou idioma como `Linux x86-64`.

Utilitário OPatch mais recente

Patch 6880880

Patches da PSU de janeiro de 2023

Patches da PSU e da RU do banco de dados

- Patch 34765931
- Patch 34786990

Patches adicionais necessários

- Patch 35099667
- Patch 35099674
- Patch 28730253
- Patch 29213893
- Patch 35012866

Patches da PSU de abril de 2023

Patches da PSU e da RU do banco de dados

- Patch 35042068
- Patch 35050341

Patches adicionais necessários

- Patch 28730253
- Patch 29213893
- Patch 33125873

- Patch 35220732
- Patch 35239280

Configuração de bucket do Amazon S3

1. Crie um bucket do Amazon S3 no sua Conta da AWS ou escolha um bucket existente.
2. Crie uma estrutura de pastas no bucket semelhante ao exemplo abaixo.

```
<bucket-name>/
### oracle_cev/
    ### V982063-01.zip
    ### p6880880_190000_Linux-x86-64.zip
    ### p34765931_190000_Linux-x86-64.zip
    ### p34786990_190000_Linux-x86-64.zip
    ### p35099667_190000_Linux-x86-64.zip
    ### p35099674_190000_Generic.zip
    ### p28730253_190000_Linux-x86-64.zip
    ### p29213893_1918000DBRU_Generic.zip
    ### p35012866_1918000DBRU_Linux-x86-64.zip
    ### p35042068_190000_Linux-x86-64.zip
    ### p35050341_190000_Linux-x86-64.zip
    ### p29213893_1919000DBRU_Generic.zip
    ### p33125873_1919000DBRU_Linux-x86-64.zip
    ### p35220732_190000_Linux-x86-64.zip
    ### p35239280_190000_Generic.zip
```

3. Faça upload de todos os arquivos do Oracle que você baixou anteriormente nas pastas apropriadas.

Implantar o RDS Custom para Oracle usando o AWS CloudFormation

Etapa 1: preparar o modelo do AWS CloudFormation

Antes de implantar o RDS Custom para Oracle, você precisa baixar e configurar o modelo do AWS CloudFormation que cria os pré-requisitos que devem ser atendidos.

Copie e salve o modelo

1. Acesse Deploying RDS Custom for Oracle with single and multiple Availability Zones.
2. Copie o modelo no formato de sua preferência (YAML ou JSON).

3. Salve o arquivo no formato YAML ou JSON. Por exemplo, `rds-custom-oracle-prereqs.yaml`.

Inicie a pilha no Console da AWS

1. Abra o Console da AWS e navegue até o AWS CloudFormation.
2. Escolha Criar pilha > Com novos recursos (padrão).
3. Selecione Escolher um modelo existente.
4. Selecione Fazer upload de um arquivo de modelo > Escolher arquivo.
5. Selecione o arquivo de modelo que você baixou anteriormente.
6. Mantenha os valores padrão dos parâmetros.
7. Selecione Próximo para criar a pilha.

Alternativa: usar a AWS CLI

Em vez de usar o console, você pode criar a pilha usando a AWS CLI:

```
aws cloudformation create-stack \
  --stack-name rds-custom-oracle \
  --template-body file://rds-custom-oracle-prereqs.yaml \
  --capabilities CAPABILITY_NAMED_IAM
```

Etapa 2: criar as versões personalizadas do mecanismo (CEVs) e as instâncias do Amazon RDS

Copie e salve o modelo

1. Acesse Deploying RDS Custom for Oracle with single and multiple Availability Zones.
2. Copie o modelo no formato de sua preferência (YAML ou JSON).
3. Atualize os seguintes parâmetros no modelo, se necessário:
   - `BucketName`
   - `CEVS3Prefix`
   - Senha principal do banco de dados (substitua **********).
4. Salve o arquivo no formato YAML ou JSON.

Etapa 3: implantar usando o Console da AWS

1. Abra o Console da AWS e navegue até o AWS CloudFormation.
2. Escolha Criar pilha > Com novos recursos (padrão).
3. Selecione Escolher um modelo existente.
4. Selecione Fazer upload de um arquivo de modelo > Escolher arquivo.
5. Selecione o arquivo de modelo que você baixou anteriormente.
6. Deixe os parâmetros como valores padrão.
7. Preencha os parâmetros da seguinte forma:

```
BucketName: rds-custom-id
CEVS3Prefix: oracle_cev
CEVCreation: Yes
```

8. Analise a configuração e selecione Próximo para criar a pilha.

Opcional: implantar usando a AWS CLI

```
aws cloudformation create-stack \
  --stack-name rds-custom-oracle \
  --template-body file://rds-custom-oracle.yaml \
  --parameters \
    ParameterKey=BucketName,ParameterValue=rds-custom-id \
    ParameterKey=CEVS3Prefix,ParameterValue=oracle_cev \
    ParameterKey=CEVCreation,ParameterValue=Yes \
  --capabilities CAPABILITY_NAMED_IAM
```

Recursos de implantação criados

O modelo de cria os seguintes recursos:

- Amazon VPC com sub-redes públicas e privadas
- Grupos de segurança
- Endpoints da Amazon VPC
- Perfis e políticas do IAM
- Chave do KMS para criptografia
- Versões personalizadas do mecanismo

- Instâncias do RDS Custom para Oracle para configurações single-AZ e multi-AZ

Monitorar o andamento da implantação

Depois de criar a pilha do AWS CloudFormation, monitore o andamento para garantir uma implantação bem-sucedida. O processo de implantação inclui a criação de versões personalizadas do mecanismo (CEVs) e instâncias do RDS.

Para monitorar o andamento da implantação:

1. Abra o console do AWS CloudFormation.
2. Escolha o nome da pilha.
3. Escolha a guia Eventos para visualizar o andamento e identificar erros.

> **Note**
>
> A criação de uma CEV normalmente requer de duas a três horas. Após a conclusão bem-sucedida da criação da CEV, o Amazon RDS começa automaticamente a criar a instância do Amazon RDS.

Pós-implantação

Após a conclusão do processo de criação da pilha, execute as seguintes etapas de verificação e configuração pós-implantação:

1. Na página do console do Amazon RDS, navegue até Versões personalizadas do mecanismo para verificar a criação da CEV.
2. Confirme se as instâncias do Amazon RDS foram criadas e estão disponíveis.
3. Teste a conectividade com as instâncias do Amazon RDS.
4. Configure estratégias de monitoramento e backup conforme necessário.

Limpeza

Para remover todos os recursos, execute o seguinte comando da AWS CLI:

```
aws cloudformation delete-stack --stack-name rds-custom-oracle
```

Solução de problemas

Se você encontrar problemas durante a implantação, use as soluções a seguir para resolver problemas comuns.

Falha na criação da CEV

- Verifique se todos os patches necessários foram enviados ao Amazon S3.
- Confira as suas permissões do IAM.
- Verifique se as versões do patch estão corretas. Consulte os the section called “Pré-requisitos” para ver a lista dos patches necessários.

Falha na criação da instância do Amazon RDS

- Confira as configurações de VPC/sub-rede.
- Verifique as regras de grupo de segurança.
- Confirme se a CEV está disponível.

Trabalhar com réplicas do Oracle para o RDS Custom para Oracle

É possível criar réplicas do Oracle para instâncias de banco de dados do RDS Custom para Oracle que executam o Oracle Enterprise Edition. Há suporte para bancos de dados de contêiner (CDBs) e não CDBs. A Standard Edition 2 não é compatível com o Oracle Data Guard.

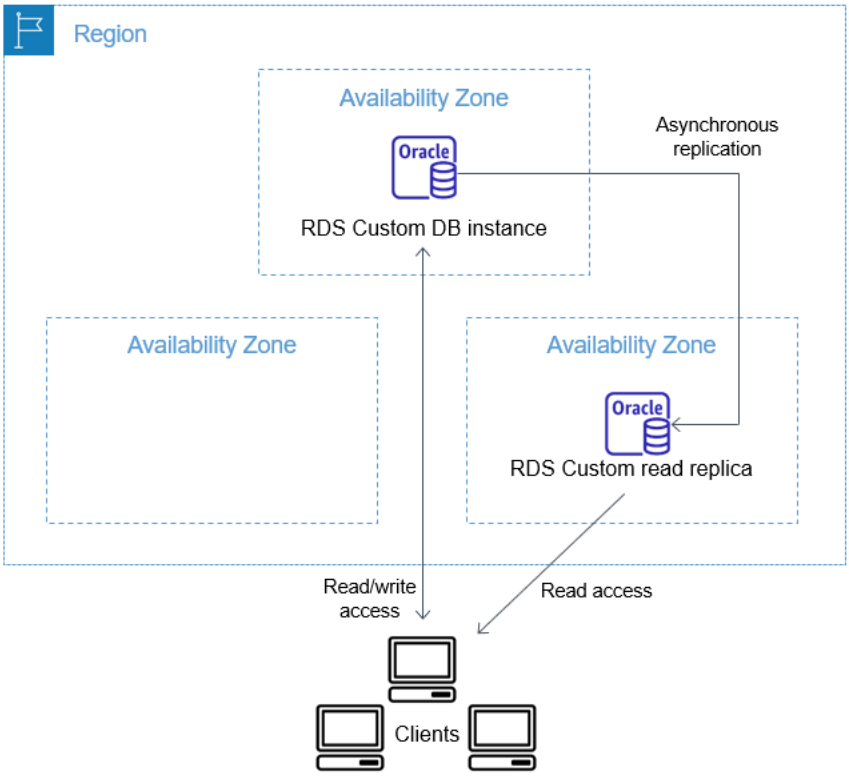
A criação de réplicas do RDS Custom para Oracle é semelhante à criação de réplicas do RDS para Oracle, mas apresenta diferenças importantes. Para obter informações gerais sobre como criar e gerenciar réplicas do Oracle, consulte Trabalhar com réplicas de leitura de instância de banco de dados e Trabalhar com réplicas de leitura do Amazon RDS para Oracle.

Tópicos

- Visão geral da replicação do RDS Custom para Oracle
- Diretrizes e limitações da replicação do RDS Custom para Oracle
- Promover uma réplica do RDS Custom para Oracle a uma instância de banco de dados autônoma
- Configuração de um túnel VPN entre as instâncias primária e de réplica do RDS Custom para Oracle

Visão geral da replicação do RDS Custom para Oracle

A arquitetura da replicação do RDS Custom para Oracle é análoga à replicação do RDS para Oracle. Uma instância de banco de dados primária replica de forma assíncrona para uma ou mais réplicas da Oracle.



Número máximo de réplicas

Como acontece com o RDS Custom para Oracle, você pode criar até cinco réplicas de banco de dados gerenciadas da sua instância de banco de dados primária do RDS Custom para Oracle. Você também pode criar suas próprias réplicas do Oracle configuradas manualmente (externas). As réplicas externas não contam em relação ao limite de instâncias de banco de dados. Elas também estão fora do perímetro de suporte do RDS Custom. Para ter mais informações sobre o perímetro de suporte, consulte Perímetro de suporte do RDS Custom.

Convenção de nomenclatura de réplicas

Os nomes das réplicas da Oracle são baseados no nome exclusivo do banco de dados. O formato é *DB_UNIQUE_NAME_X*, com letras anexadas em sequência. Por exemplo, se o nome exclusivo do seu banco de dados for `ORCL`, as duas primeiras réplicas são chamadas `ORCL_A` e `ORCL_B`. As primeiras seis letras, A-F, são reservadas para o RDS Custom. O RDS Custom copia parâmetros de banco de dados da instância de banco de dados primária para as réplicas. Para ter mais informações, consulte DB_UNIQUE_NAME na documentação do Oracle.

Retenção de backup de réplicas

Por padrão, réplicas do RDS Custom Oracle usam o mesmo período de retenção de backup que a sua instância de banco de dados primária. Você pode modificar o período de retenção de backup

para 1-35 dias. O RDS Custom oferece suporte para backup, restauração e recuperação a um ponto anterior no tempo (PITR). Para ter mais informações sobre como fazer backup e restaurar instâncias de banco de dados do RDS Custom, consulte Fazer o backup e a restauração de uma instância de banco de dados do Amazon RDS Custom for Oracle.

> **Note**
>
> Ao criar uma réplica do Oracle, o RDS Custom pausa temporariamente a limpeza de logs de redo. Dessa forma, o RDS Custom garante que possa aplicar esses logs à nova réplica do Oracle depois que ela for disponibilizada.

Promoção de réplicas

Você pode promover réplicas gerenciadas do Oracle no RDS Custom para Oracle usando o console, o comando `promote-read-replica` do AWS CLI ou a API do `PromoteReadReplica`. Se você excluir sua instância de banco de dados primária e todas as réplicas estiverem íntegras, o RDS Custom para Oracle promoverá automaticamente suas réplicas gerenciadas a instâncias autônomas. Se uma réplica pausou a automação ou estiver fora do perímetro de suporte, você deverá corrigir a réplica antes que o RDS Custom possa promovê-la automaticamente. Você só pode promover réplicas externas do Oracle manualmente.

Diretrizes e limitações da replicação do RDS Custom para Oracle

Quando você cria réplicas de leitura do RDS Custom para Oracle, nem todas as opções de réplica do RDS Oracle são compatíveis.

Tópicos

- Diretrizes gerais da replicação do RDS Custom para Oracle
- Limitações gerais da replicação do RDS Custom para Oracle
- Requisitos e limitações da rede para a replicação do RDS Custom para Oracle
- Limitações da réplica externa para o RDS Custom para Oracle

Diretrizes gerais da replicação do RDS Custom para Oracle

Ao trabalhar com o RDS Custom para Oracle, siga estas diretrizes:

- É possível usar a replicação do RDS Custom para Oracle somente no Oracle Enterprise Edition. A Standard Edition 2 não é compatível.
- Recomendamos fortemente que você implemente um túnel VPN para criptografar a comunicação entre suas instâncias primária e de espera. Para ter mais informações, consulte Configuração de um túnel VPN entre as instâncias primária e de réplica do RDS Custom para Oracle.
- Não modifique o usuário `RDS_DATAGUARD`. Esse usuário está reservado para a automação do RDS Custom para Oracle. Modificar esse usuário pode gerar resultados indesejados, como a incapacidade de criar réplicas do Oracle para sua instância de banco de dados do RDS Custom para Oracle.
- Não altere a senha do usuário da replicação. Ela é necessária para administrar a configuração do Oracle Data Guard no host do RDS Custom. Se você alterar a senha, o RDS Custom para Oracle pode colocar sua réplica do Oracle fora do perímetro de suporte. Para ter mais informações, consulte Perímetro de suporte do RDS Custom.

  A senha é armazenada no AWS Secrets Manager, marcada com o ID do recurso de banco de dados. Cada réplica do Oracle possui seu próprio segredo no Secrets Manager. O segredo usa um dos formatos de nomenclatura a seguir.

  ```
  do-not-delete-rds-custom-db-DB_resource_id-uuid-dg
  rds-custom!oracle-do-not-delete-DB_resource_id-uuid-dg
  ```

- Não altere o `DB_UNIQUE_NAME` da instância de banco de dados primária. Alterar o nome faz com que qualquer operação de restauração fique travada.
- Não especifique a cláusula `STANDBYS=NONE` em um comando `CREATE PLUGGABLE DATABASE` em um CDB do RDS Custom. Dessa forma, se ocorrer um failover, seu CDB de espera é composto por todos os PDBs.

Limitações gerais da replicação do RDS Custom para Oracle

As réplicas do RDS Custom para Oracle têm as seguintes limitações:

- Você não pode criar réplicas do RDS Custom para Oracle no modo somente leitura. No entanto, você pode alterar manualmente o modo montado de réplicas para somente leitura e de somente leitura para montado. Para ter mais informações, consulte a documentação do comando create-db-instance-read-replica do AWS CLI.
- Não é possível criar réplicas do RDS Custom para Oracle entre regiões.

- Você não pode alterar o valor do parâmetro `CommunicationTimeout` do Oracle Data Guard. Esse parâmetro é definido como 15 segundos para instâncias de banco de dados do RDS Custom para Oracle.

Requisitos e limitações da rede para a replicação do RDS Custom para Oracle

Verifique se a sua configuração de rede oferece suporte para réplicas do RDS Custom para Oracle. Considere o seguinte:

- Habilite a porta 1140 para comunicação de entrada e saída dentro da sua nuvem privada virtual (VPC) para a instância de banco de dados primária e todas as réplicas. Isso é necessário para a comunicação do Oracle Data Guard entre as réplicas de leitura.
- O RDS Custom para Oracle valida a rede ao criar uma réplica do Oracle. Se a instância de banco de dados primária e a nova réplica não conseguirem se conectar pela rede, o RDS Custom para Oracle não criará a réplica e a colocará no estado `INCOMPATIBLE_NETWORK`.
- Para réplicas externas do Oracle, como aquelas criadas no Amazon EC2 ou on-premises, use outra porta e ouvinte para replicação do Oracle Data Guard. Tentar usar a porta 1140 pode causar conflitos com a automação do RDS Custom.
- O arquivo `/rdsdbdata/config/tnsnames.ora` contém nomes de serviços de rede mapeados para endereços de protocolo de ouvinte. Observe os seguintes requisitos e recomendações:
  - Entradas em `tnsnames.ora` com o prefixo `rds_custom_` são reservadas para RDS Custom ao lidar com operações de réplica do Oracle.

    Ao criar entradas manuais em `tnsnames.ora`, não use esse prefixo.
  - Em alguns casos, talvez você queira alternar ou fazer failover manualmente ou usar tecnologias de failover, como o Fast-Start Failover (FSFO). Nesse caso, sincronize manualmente as entradas `tnsnames.ora` da instância de banco de dados primária com todas as instâncias em espera. Essa recomendação aplica-se a réplicas do Oracle gerenciadas pelo RDS Custom e a réplicas externas do Oracle.

    A automação do RDS Custom atualiza as entradas `tnsnames.ora` somente na instância de banco de dados primária. Também sincronize ao adicionar ou remover uma réplica do Oracle.

    Se você não sincronizar os arquivos `tnsnames.ora` e alternar ou failover manualmente, o Oracle Data Guard na instância de banco de dados primária talvez não consiga se comunicar com as réplicas do Oracle.

Limitações da réplica externa para o RDS Custom para Oracle

As réplicas externas do RDS Custom para Oracle, que incluem réplicas on-premises, têm as seguintes limitações:

- O RDS Custom para Oracle detecta alterações no perfil da instância após o failover manual, como o FSFO, para réplicas externas do Oracle.

  O RDS Custom para Oracle detecta alterações em réplicas gerenciadas. A alteração da função é anotada no log de eventos. Também é possível ver o novo estado usando o comando describe-db-instances da AWS CLI.
- O RDS Custom para Oracle, não detecta um alto atraso de replicação para réplicas externas do Oracle.

  O RDS Custom para Oracle detecta atrasos para réplicas gerenciadas. O alto atraso de replicação gera o evento `Replication has stopped`. Você também pode ver o status da replicação usando o comando describe-db-instances da AWS CLI, mas pode haver um atraso para que ele seja atualizado.
- O RDS Custom para Oracle não promoverá réplicas externas do Oracle automaticamente se você excluir sua instância de banco de dados primária.

  O recurso de promoção automática está disponível somente para réplicas gerenciadas do Oracle. Para obter informações sobre como promover réplicas do Oracle manualmente, consulte o whitepaper "Enabling high availability with Data Guard on Amazon RDS Custom for Oracle" (Habilitar a alta disponibilidade com o Data Guard no Amazon RDS Custom para Oracle).

Promover uma réplica do RDS Custom para Oracle a uma instância de banco de dados autônoma

Assim como com o RDS para Oracle, você pode promover uma réplica do RDS Custom para Oracle a uma instância de banco de dados autônoma. Ao promover uma réplica do Oracle, a instância de banco de dados é reiniciada antes que ela seja disponibilizada. Para ter mais informações sobre como promover réplicas do Oracle, consulte Promoção de uma réplica de leitura a uma instância de banco de dados autônoma.

Ao promover uma réplica, observe as seguintes diretrizes:

- Não inicie um failover enquanto o RDS Custom para Oracle estiver promovendo sua réplica. Caso contrário, o fluxo de trabalho da promoção poderá travar.
- Não alterne sua instância de banco de dados primária enquanto o RDS Custom para Oracle estiver promovendo sua réplica do Oracle. Caso contrário, o fluxo de trabalho da promoção poderá travar.
- Não interrompa sua instância de banco de dados primária enquanto o RDS Custom para Oracle estiver promovendo sua réplica do Oracle. Caso contrário, o fluxo de trabalho da promoção poderá travar.
- Não tente reiniciar a replicação com sua instância de banco de dados recém-promovida como destino. Depois que o RDS Custom para Oracle promover sua réplica do Oracle, ela se tornará uma instância de banco de dados autônoma e não terá mais a função de réplica.

Observe as seguintes limitações da promoção de réplicas do RDS Custom para Oracle:

- Você não pode promover uma réplica enquanto o RDS Custom para Oracle estiver fazendo backup dela.
- Ao promover sua réplica do Oracle, você não pode alterar o período de retenção de backup para `0`.
- Você não pode promover sua réplica quando ela não está em um estado íntegro.

  Se você chamar `delete-db-instance` na instância de banco de dados primária, o RDS Custom para Oracle valida que cada réplica gerenciada do Oracle está íntegra e disponível para promoção. Uma réplica pode não estar qualificada para promoção porque a automação está pausada ou está fora do perímetro de suporte. Nesses casos, o RDS Custom para Oracle publica um evento explicando o problema para que você possa reparar sua réplica do Oracle manualmente.

As etapas a seguir mostram o processo geral para promover uma réplica do Oracle a uma instância de banco de dados:

1. Pare a gravação de transações na instância de banco de dados primária.
2. Aguarde até que o RDS Custom para Oracle aplique todas as atualizações à sua réplica do Oracle.
3. Promova a réplica do Oracle escolhendo a opção Promote (Promover) no console do Amazon RDS, o comando `promote-read-replica` da AWS CLI ou a operação da API `PromoteReadReplica` do Amazon RDS.

A promoção de uma réplica do Oracle leva alguns minutos para ser concluída. Durante o processo, o RDS Custom para Oracle interrompe a replicação e reinicializa sua réplica. Quando a reinicialização for concluída, a réplica do Oracle estará disponível como uma nova instância de banco de dados. Consulte mais informações sobre como solucionar problemas de promoção de réplicas em Solucionar problemas de promoção de réplicas no RDS Custom para Oracle.

Console

Para promover uma réplica do RDS Custom para Oracle a uma instância de banco de dados autônoma

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No console do Amazon RDS, escolha Databases (Bancos de dados).

   O painel Databases (Bancos de dados) é exibido. Todas as réplicas do Oracle exibem Replica (Réplica) na coluna Role (Função).
3. Escolha a réplica do RDS Custom para Oracle que você deseja promover.
4. Em Actions (Ações), selecione Promote (Promover).
5. Na página Promote Oracle replica (Promover réplica do Oracle), insira o período de retenção de backup e a janela de backup para a instância de banco de dados recém-promovida. Você não pode definir esse valor como 0.
6. Quando estiver satisfeito com as configurações, escolha Promote Oracle replica (Promover réplica do Oracle).

AWS CLI

Para promover sua réplica do RDS Custom para Oracle a uma instância de banco de dados autônoma, use o comando `promote-read-replica` da AWS CLI.

Example

Para Linux, macOS ou Unix:

```
aws rds promote-read-replica \
--db-instance-identifier my-custom-read-replica \
--backup-retention-period 2 \
--preferred-backup-window 23:00-24:00
```

Para Windows:

```
aws rds promote-read-replica ^
--db-instance-identifier my-custom-read-replica ^
--backup-retention-period 2 ^
--preferred-backup-window 23:00-24:00
```

API do RDS

Para promover sua réplica do RDS Custom para Oracle a uma instância de banco de dados autônoma, chame a operação da API `PromoteReadReplica` do Amazon RDS com o parâmetro obrigatório `DBInstanceIdentifier`.

Configuração de um túnel VPN entre as instâncias primária e de réplica do RDS Custom para Oracle

Um túnel VPN é uma conexão criptografada entre dois ou mais dispositivos em uma rede. Para garantir o mais alto nível de segurança para suas instâncias do Oracle Data Guard no RDS Custom para Oracle, é altamente recomendável que você implemente um túnel VPN para criptografar a comunicação entre suas instâncias primária e em espera. O túnel serve como uma proteção para dados confidenciais enquanto eles percorrem a rede entre as instâncias. Embora essa configuração seja opcional, recomendamos como uma prática recomendada para alcançar a segurança dos dados e a conformidade regulatória.

Satisfaça os seguintes pré-requisitos:

- Você tem acesso raiz aos hosts primário e em espera.
- Você tem o conhecimento técnico para executar o comando `ipsec`.

Como configurar um túnel VPN entre uma instância primária e uma réplica no RDS Custom para Oracle

1. Adicione os grupos de segurança da instância primária e da instância em espera à lista de permissões usando as seguintes regras:

   ```
   ACTION FLOW SOURCE PROTO PORT

   ALLOW ingress this-SG 50 (ESP) all (N/A)
   ALLOW egress this-SG 50 (ESP) all (N/A)
   ```

```
ALLOW ingress this-SG 17 (UDP) 500 (IKE)
ALLOW egress this-SG 17 (UDP) 500 (IKE)
```

2. Mude para o usuário raiz.

```
$ sudo su - root
```

3. Execute os comandos a seguir na instância primária e na instância em espera para inicializar o banco de dados do Network Security Services (NSS) sob o usuário `root`.

```
ipsec initnss --nssdir /etc/ipsec.d
```

4. Gere chaves RSA da seguinte forma:

   a. Na instância primária, gere as chaves usando um dos comandos `ipsec` a seguir, dependendo da versão do seu sistema operacional.

   ```
   ipsec newhostkey --nssdir /etc/ipsec.d       ## for Oracle Linux Version 8
   ipsec newhostkey --output /etc/ipsec.secrets ## for Oracle Linux version 7.9
   ```

   b. Obtenha a chave pública, de que você precisa para criar a configuração. No exemplo a seguir, a instância primária é `left` porque na linguagem `ipsec`, `left` refere-se ao dispositivo que você está configurando no momento e `right` refere-se ao dispositivo na outra extremidade do túnel.

   ```
   ipsec showhostkey --left --ckaid ckaid-returned-in-last-statement
   ```

   c. Na instância em espera, gere chaves para a instância em espera.

   ```
   ipsec newhostkey --nssdir /etc/ipsec.d       ## for Oracle Linux Version 8
   ipsec newhostkey --output /etc/ipsec.secrets ## for Oracle Linux version 7.9
   ```

   d. Obtenha a chave pública para a instância em espera, de que você precisa para criar a configuração. No exemplo a seguir, a instância em espera é `right` porque se refere ao dispositivo na outra extremidade do túnel.

   ```
   ipsec showhostkey --right --ckaid ckaid-returned-in-last-statement
   ```

5. Com base nas chaves RSA que você obteve, gere a configuração. A configuração é idêntica tanto para a instância primária quanto para a instância em espera. Você pode encontrar o

endereço IPv4 da instância primária e o endereço IPv4 da instância em espera no console da AWS.

Tanto na instância primária quanto na instância em espera, salve a configuração a seguir no arquivo `/etc/ipsec.d/custom-fb-tunnel.conf`.

```
conn custom-db-tunnel
 type=transport
 auto=add
 authby=rsasig
 left=IPV4-for-primary
 leftrsasigkey=RSA-key-generated-on-primary
 right=IPV4-for-standby
 rightrsasigkey=RSA-key-generated-on-standby
```

6. Tanto na instância primária quanto na instância em espera, inicie o daemon `ipsec` nos dois hosts.

```
ipsec setup start
```

7. Inicie o túnel na instância primária ou na instância em espera. A saída deve ser semelhante à seguinte.

```
[root@ip-172-31-6-81 ~]# ipsec auto --up custom-db-tunnel
181 "custom-db-tunnel" #1: initiating IKEv2 connection
181 "custom-db-tunnel" #1: sent IKE_SA_INIT request to 172.31.32.196:500
182 "custom-db-tunnel" #1: sent IKE_AUTH request {cipher=AES_GCM_16_256 integ=n/a
 prf=HMAC_SHA2_512 group=DH19}
003 "custom-db-tunnel" #1: initiator established IKE SA; authenticated peer '3584-
bit PKCS#1 1.5 RSA with SHA1' signature using preloaded certificate '172.31.32.196'
004 "custom-db-tunnel" #2: initiator established Child SA using #1; IPsec transport
 [172.31.6.81-172.31.6.81:0-65535 0] -> [172.31.32.196-172.31.32.196:0-65535 0]
 {ESP/ESN=>0xda9c4815 <0xb742ca42 xfrm=AES_GCM_16_256-NONE DPD=passive}
[root@ip-172-31-6-81 ~]#
```

Fazer o backup e a restauração de uma instância de banco de dados do Amazon RDS Custom for Oracle

Como o Amazon RDS, o RDS Custom cria e salva backups automatizados da sua instância de banco de dados do RDS Custom for Oracle durante a janela de backup de sua instância de banco de dados. Também é possível fazer backup da instância de banco de dados manualmente.

O procedimento é idêntico a obter um snapshot de uma instância de banco de dados do Amazon RDS. O primeiro snapshot de uma instância de banco de dados do RDS Custom os dados da instância de banco de dados completa. Os snapshots subsequentes são incrementais.

Restaure snapshots de banco de dados utilizando o AWS Management Console ou a AWS CLI.

Tópicos

- Criar um snapshot do RDS Custom for Oracle
- Restaurar um snapshot de banco de dados do RDS Custom for Oracle
- Restaurar uma instância do RDS Custom for Oracle para um ponto no tempo
- Excluir um snapshot do RDS Custom for Oracle
- Excluir backups automatizados do RDS Custom for Oracle

Criar um snapshot do RDS Custom for Oracle

O RDS Custom for Oracle cria um snapshot do volume de armazenamento da instância de banco de dados, fazendo o backup de toda a instância de banco de dados e não apenas dos bancos de dados individuais. Quando sua instância de banco de dados contiver um banco de dados de contêiner (CDB), o snapshot da instância incluirá o CDB raiz e todos os PDBs.

Ao criar um snapshot do RDS Custom for Oracle, especifique qual instância de banco de dados do RDS Custom deve receber o backup. Atribua um nome ao snapshot para que ele possa ser restaurado posteriormente.

Quando você cria um snapshot, o RDS Custom for Oracle cria um snapshot do Amazon EBS para cada volume anexado à instância de banco de dados. O RDS Custom for Oracle usa o snapshot do EBS do volume raiz para registrar uma nova imagem de máquina da Amazon (AMI). Para facilitar a associação de snapshots a uma instância de banco de dados específica, eles são marcados com `DBSnapshotIdentifier`, `DbiResourceId` e `VolumeType`.

A criação de um snapshot de banco de dados resulta em uma breve suspensão de E/S. Essa suspensão pode durar desde alguns segundos até alguns minutos, dependendo do tamanho e da classe da sua instância de banco de dados. O tempo de criação do snapshot varia dependendo do tamanho do banco de dados. Como o snapshot inclui todo o volume de armazenamento, o tamanho de arquivos, como arquivos temporários, também afeta o tempo necessário para criar esse snapshot. Para saber mais sobre como criar snapshots, consulte Criar um snapshot de banco de dados para uma instância de banco de dados single-AZ para o Amazon RDS.

Crie um snapshot do RDS Custom for Oracle utilizando o console ou a AWS CLI.

Console

Para criar um snapshot do RDS Custom

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Bancos de dados.
3. Na lista de instâncias de banco de dados do RDS Custom, escolha a instância da qual você deseja obter um snapshot.
4. Em Actions (Ações), escolha Take snapshot (Fazer snapshot).

   A janela Take snapshot de banco de dados (Fazer snapshot de banco de dados) é exibida.
5. Para Snapshot name (Nome do snapshot), insira o nome do snapshot.
6. Selecione Take Snapshot (Fazer snapshot).

AWS CLI

Você cria um snapshot de uma instância de banco de dados do RDS Custom utilizando o comando create-db-snapshot da AWS CLI.

Especifique as seguintes opções:

- `--db-instance-identifier` – Identifica de qual instância de banco de dados do RDS Custom você fará backup
- `--db-snapshot-identifier` – Dê um nome para seu snapshot do RDS Custom para que ele possa ser restaurado mais tarde

Neste exemplo, você cria um snapshot de banco de dados chamado *my-custom-snapshot* para uma instância de banco de dados do RDS Custom chamada *my-custom-instance*.

Example

Para Linux, macOS ou Unix:

```
aws rds create-db-snapshot \
    --db-instance-identifier my-custom-instance \
    --db-snapshot-identifier my-custom-snapshot
```

Para Windows:

```
aws rds create-db-snapshot ^
    --db-instance-identifier my-custom-instance ^
    --db-snapshot-identifier my-custom-snapshot
```

Restaurar um snapshot de banco de dados do RDS Custom for Oracle

Ao restaurar uma instância de banco de dados do RDS Custom for Oracle, você fornece o nome do snapshot de banco de dados e um nome para a nova instância. Não é possível restaurar de um snapshot para uma instância de banco de dados do RDS Custom existente. Uma nova instância de banco de dados do RDS Custom é criada quando você realiza a restauração.

O processo de restauração difere das seguintes maneiras da restauração no Amazon RDS:

- Antes de restaurar um snapshot, o RDS Custom for Oracle faz backup dos arquivos de configuração existentes. Esses arquivos estão disponíveis no diretório `/rdsdbdata/config/backup` da instância restaurada. O RDS Custom for Oracle restaura o snapshot de banco de dados com parâmetros padrão e substitui os arquivos de configuração de banco de dados anteriores pelos existentes. Portanto, a instância restaurada não preserva parâmetros personalizados e alterações em arquivos de configuração do banco de dados.
- O banco de dados restaurado tem o mesmo nome que no snapshot. Se desejar, especifique um nome diferente. (Para o RDS Custom for Oracle, o padrão é `ORCL`.)

Console

Para restaurar uma instância de banco de dados do RDS Custom a partir de um snapshot de banco de dados

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Snapshots.
3. Escolha o DB snapshot do qual você deseja restaurar.
4. Em Actions (Ações), escolha Restore snapshot (Restaurar snapshot).
5. Na página Restore DB instance (Restaurar instância de banco de dados), para DB Instance Identifier (Identificador da instância de banco de dados), insira o nome da instância de banco de dados do RDS Custom restaurada.
6. Escolha Restore DB Instance.

AWS CLI

Você restaura um snapshot de banco de dados do RDS Custom utilizando o comando restore-db-instance-from-db-snapshot da AWS CLI.

Se o snapshot do qual você está restaurando for para uma instância de banco de dados privada, certifique-se de especificar `db-subnet-group-name` e `no-publicly-accessible`, ambos corretos. Caso contrário, a instância de banco de dados assumirá como padrão o estado de acesso público. São necessárias as seguintes opções:

- `db-snapshot-identifier` – Identifica o snapshot do qual restaurar
- `db-instance-identifier` – Especifica o nome da instância de banco de dados do RDS Custom a ser criada a partir do snapshot de banco de dados
- `custom-iam-instance-profile`: especifica o perfil da instância associado à instância subjacente do Amazon EC2 de uma instância de banco de dados do RDS Custom.

O código a seguir restaura o snapshot chamado `my-custom-snapshot` para `my-custom-instance`.

Example

Para Linux, macOS ou Unix:

```
aws rds restore-db-instance-from-db-snapshot \
  --db-snapshot-identifier my-custom-snapshot \
  --db-instance-identifier my-custom-instance \
  --custom-iam-instance-profile AWSRDSCustomInstanceProfileForRdsCustomInstance \
  --no-publicly-accessible
```

Para Windows:

```
aws rds restore-db-instance-from-db-snapshot ^
  --db-snapshot-identifier my-custom-snapshot ^
  --db-instance-identifier my-custom-instance ^
  --custom-iam-instance-profile AWSRDSCustomInstanceProfileForRdsCustomInstance ^
  --no-publicly-accessible
```

Restaurar uma instância do RDS Custom for Oracle para um ponto no tempo

É possível restaurar uma instância de banco de dados para um ponto anterior no tempo (PITR) criando uma nova instância de banco de dados. Para dar suporte ao PITR, suas instâncias de banco de dados devem ter a retenção de backup definida como diferente de zero.

O tempo de restauração mais recente de uma instância de banco de dados do RDS Custom for Oracle depende de vários fatores, mas em geral é de até cinco minutos do horário atual. Para visualizar o tempo restaurável mais recente para uma instância de banco de dado, use o comando AWS CLI describe-db-instances e confira o valor retornado no campo `LatestRestorableTime` para a instância de banco de dados. Para ver o tempo de restauração mais recente para cada instância de banco de dados no console Amazon RDS, selecione Backups automatizados.

É possível fazer a restauração para qualquer momento dentro do período de retenção de backup. Para ver o tempo de restauração mais antigo para cada instância de banco de dados, selecione Backups automatizados no console do Amazon RDS.

Para obter informações gerais sobre PITR, consulte Restaurar uma instância de banco de dados para um momento especificado no Amazon RDS.

Tópicos

- Considerações sobre o PITR para o RDS Custom for Oracle

Considerações sobre o PITR para o RDS Custom for Oracle

No RDS Custom for Oracle, o PITR difere das seguintes maneiras importantes do PITR no Amazon RDS:

- O banco de dados restaurado tem o mesmo nome da instância de banco de dados de origem. Se desejar, especifique um nome diferente. O padrão é `ORCL`.
- `AWSRDSCustomIamRolePolicy` requer novas permissões. Para obter mais informações, consulte Etapa 2: Adicionar uma política de acesso a AWSRDSCustomInstanceRoleForRdsCustomInstance.
- Todas as instâncias de banco de dados do RDS Custom for Oracle devem ter a retenção de backup definida como um valor diferente de zero.
- Se você alterar o fuso horário do sistema operacional ou da instância de banco de dados, a PITR talvez não funcione. Para obter informações sobre como alterar fusos horários, consulte Fuso horário da Oracle.
- Se você definir a automação como `ALL_PAUSED`, o RDS Custom pausará o upload dos arquivos de log redo arquivados, incluindo os logs criados antes da hora da última restauração possível (LRT). Convém pausar a automação por um breve período.

  Para ilustrar, suponha que sua LRT seja 10 minutos atrás. Você pausa a automação. Durante a pausa, o RDS Custom não carrega logs de redo arquivados. Se a sua instância de banco de dados falhar, apenas será possível recuperar em um momento antes da LRT que existia quando você pausou. Quando você retomar a automação, o RDS Custom retomará o carregamento de logs. A LRT avançará. Regras de PITR normais são aplicáveis.
- No RDS Custom, é possível especificar manualmente um número arbitrário de horas para reter logs de redo arquivados antes que o RDS Custom os exclua após o upload. Especifique o número de horas da seguinte forma:

  1. Crie um arquivo de texto chamado `/opt/aws/rdscustomagent/config/redo_logs_custom_configuration.json`.
  2. Adicione um objeto JSON no seguinte formato: `{"archivedLogRetentionHours" : "num_of_hours"}`. O número deve ser um número inteiro no intervalo de 1 a 840.
- Suponha que você conecte um não CDB a um banco de dados de contêiner (CDB) como um PDB e, depois, tente o PITR. A operação só será bem-sucedida se você tiver feito backup do PDB anteriormente. Depois de criar ou modificar um PDB, recomendamos que você sempre faça backup dele.

- Convém não personalizar parâmetros de inicialização do banco de dados. Por exemplo, a modificação dos seguintes parâmetros afeta a PITR:
  - `CONTROL_FILE_RECORD_KEEP_TIME` afeta as regras para carregar e excluir logs.
  - `LOG_ARCHIVE_DEST_`*n* não oferece suporte a vários destinos.
  - `ARCHIVE_LAG_TARGET` afeta a última restauração possível. `ARCHIVE_LAG_TARGET` está definido como `300` porque o objetivo de ponto de recuperação (RPO) é de cinco minutos. Para honrar esse objetivo, o RDS troca o log redo on-line a cada cinco minutos e o armazena em um bucket do Amazon S3. Se a frequência da troca de log causar um problema de desempenho em seu banco de dados do RDS Custom para Oracle, você poderá escalar a instância de banco de dados e o armazenamento para um com maior IOPS e throughput. Se necessário para seu plano de recuperação, é possível ajustar a configuração do parâmetro de inicialização `ARCHIVE_LAG_TARGET` para um valor de 60 a 7.200.
- Se você personalizar os parâmetros de inicialização do banco de dados, é altamente recomendável personalizar apenas o seguinte:
  - `COMPATIBLE`
  - `MAX_STRING_SIZE`
  - `DB_FILES`
  - `UNDO_TABLESPACE`
  - `ENABLE_PLUGGABLE_DATABASE`
  - `CONTROL_FILES`
  - `AUDIT_TRAIL`
  - `AUDIT_TRAIL_DEST`

  Para todos os outros parâmetros de inicialização, o RDS Custom restaura os valores padrão. Se você modificar um parâmetro que não esteja na lista anterior, ele talvez tenha um efeito adverso na PITR, levando a resultados imprevisíveis. Por exemplo, `CONTROL_FILE_RECORD_KEEP_TIME` afeta as regras para carregar e excluir logs.

Você pode restaurar uma instância de banco de dados do RDS Custom em um ponto anterior utilizando o AWS Management Console, a AWS CLI ou a API do RDS.

Console

Para restaurar uma instância de banco de dados do RDS Custom para um ponto anterior especificado

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Automated backups (Backups automatizados).
3. Escolha a instância de banco de dados do RDS Custom que você deseja restaurar.
4. Em Actions (Ações), escolha Restore to point in time (Restaurar para point-in-time).

   A janela Restore to point in time (Restaurar para point-in-time) é exibida.
5. Escolha Latest restorable time (Hora da última restauração) para restaurar no último horário possível ou escolha Custom (Personalizar) para escolher um horário.

   Se você escolher Custom (Personalizado), insira a data e a hora para a qual deseja restaurar a instância.

   Os horários são mostrados no fuso horário local, que é indicado por um deslocamento do Tempo Universal Coordenado (UTC). Por exemplo, UTC-5 é a Hora Padrão do Leste dos EUA/Horário de Verão Central.
6. Para DB instance identifier (Identificador de instância de banco de dados), insira o nome da instância de banco de dados do RDS Custom restaurada de destino. O nome deve ser exclusivo.
7. Escolha outras opções conforme necessário, como a classe da instância de banco de dados.
8. Escolha Restore to point in time (Restaurar para point-in-time).

AWS CLI

Você restaura uma instância de banco de dados em um horário especificado utilizando o comando restore-db-instance-to-point-in-time da AWS CLI para criar uma nova instância de banco de dados do RDS Custom.

Utilize uma das seguintes opções para especificar o backup a ser restaurado:

- `--source-db-instance-identifier` *`mysourcedbinstance`*
- `--source-dbi-resource-id` *`dbinstanceresourceID`*

- `--source-db-instance-automated-backups-arn backupARN`

A opção `custom-iam-instance-profile` é obrigatória.

O exemplo a seguir restaura `my-custom-db-instance` para uma nova instância de banco de dados denominada `my-restored-custom-db-instance`, a partir do ponto anterior especificado.

Example

Para Linux, macOS ou Unix:

```
aws rds restore-db-instance-to-point-in-time \
    --source-db-instance-identifier my-custom-db-instance\
    --target-db-instance-identifier my-restored-custom-db-instance \
    --custom-iam-instance-profile AWSRDSCustomInstanceProfileForRdsCustomInstance \
    --restore-time 2022-10-14T23:45:00.000Z
```

Para Windows:

```
aws rds restore-db-instance-to-point-in-time ^
    --source-db-instance-identifier my-custom-db-instance ^
    --target-db-instance-identifier my-restored-custom-db-instance ^
    --custom-iam-instance-profile AWSRDSCustomInstanceProfileForRdsCustomInstance ^
    --restore-time 2022-10-14T23:45:00.000Z
```

Excluir um snapshot do RDS Custom for Oracle

É possível excluir snapshots de banco de dados gerenciados pelo RDS Custom for Oracle quando eles não são mais necessários. O procedimento de exclusão é o mesmo para instâncias de banco de dados Amazon RDS e RDS Custom.

Os snapshots do Amazon EBS para os volumes binário e raiz permanecem na sua conta por mais tempo, pois podem estar vinculados a algumas instâncias em execução na sua conta ou a outros snapshots do RDS Custom for Oracle. Esses snapshots do EBS serão excluídos automaticamente quando não estiverem mais relacionados a nenhum recurso existente do RDS Custom for Oracle (instâncias de banco de dados ou backups).

Console

Para excluir um snapshot de uma instância de banco de dados do RDS Custom

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Snapshots.
3. Escolha o snapshot de banco de dados que você deseja excluir.
4. Em Actions (Ações), selecione Delete Snapshot (Excluir snapshot).
5. Escolha Delete (Excluir) na página de confirmação.

AWS CLI

Para excluir um snapshot do RDS Custom, utilize o comando da AWS CLI delete-db-snapshot.

A seguinte opção é necessária:

- `--db-snapshot-identifier` # – o snapshot a ser excluído

O exemplo a seguir exclui o snapshot de banco de dados `my-custom-snapshot`.

Example

Para Linux, macOS ou Unix:

```
aws rds delete-db-snapshot \
  --db-snapshot-identifier my-custom-snapshot
```

Para Windows:

```
aws rds delete-db-snapshot ^
  --db-snapshot-identifier my-custom-snapshot
```

Excluir backups automatizados do RDS Custom for Oracle

Você pode excluir backups automatizados retidos para o RDS Custom for Oracle quando eles não são mais necessários. O procedimento é idêntico ao de exclusão de backups do Amazon RDS.

Console

Como excluir um backup automatizado retido

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Automated backups (Backups automatizados).
3. Escolha Retained (Retido).
4. Escolha o backup automatizado retido que você deseja excluir.
5. Em Actions, selecione Delete.
6. Na página de confirmação, insira **`delete me`** e escolha Delete (Excluir).

AWS CLI

Você pode excluir um backup automatizado retido usando o comando da AWS CLI delete-db-instance-automated-backup.

A seguinte opção é usada para excluir um backup automatizado retido:

- `--dbi-resource-id` – o identificador de recurso da instância de banco de dados do RDS Custom de origem.

  Você pode encontrar o identificador de recurso da instância de banco de dados de origem de um backup automatizado retido usando o comando da AWS CLI describe-db-instance-automated-backups.

O exemplo a seguir exclui o backup automatizado retido com o identificador de recurso da instância de banco de dados de origem `custom-db-123ABCEXAMPLE`.

Example

Para Linux, macOS ou Unix:

```
aws rds delete-db-instance-automated-backup \
    --dbi-resource-id custom-db-123ABCEXAMPLE
```

Para Windows:

```
aws rds delete-db-instance-automated-backup ^
    --dbi-resource-id custom-db-123ABCEXAMPLE
```

Trabalhar com grupos de opções no RDS Custom para Oracle

O RDS Custom usa grupos de opções para habilitar e configurar recursos adicionais. Um grupo de opções especifica recursos, chamados de opções, que estão disponíveis para uma instância de banco de dados do RDS para Oracle. Opções podem ter configurações que especificam como a opção funciona. Ao associar uma instância de banco de dados do RDS Custom para Oracle a um grupo de opções, as opções e as configurações de opções especificadas são habilitadas para essa instância. Para ter informações gerais sobre grupos de opções no Amazon RDS, consulte Trabalhar com grupos de opções.

Tópicos

- Visão geral de grupos de opções no RDS Custom para Oracle
- Fuso horário da Oracle

Visão geral de grupos de opções no RDS Custom para Oracle

Para habilitar opções para seu banco de dados Oracle, adicione-as a um grupo de opções e, em seguida, associe o grupo de opções à sua instância de banco de dados. Para ter mais informações, consulte Trabalhar com grupos de opções.

Tópicos

- Resumo das opções do RDS Custom para Oracle
- Etapas básicas para adicionar uma opção a uma instância de banco de dados do RDS Custom para Oracle
- Criar um grupo de opções no RDS Custom para Oracle
- Associar um grupo de opções a uma instância de banco de dados do RDS Custom para Oracle

Resumo das opções do RDS Custom para Oracle

O RDS Custom para Oracle é compatível com as opções de instância de um banco de dados do Oracle a seguir.

Opção	ID da opção	Descrição
Fuso horário da Oracle	`Timezone`	O fuso horário usado pela instância de banco de dados do RDS Custom para Oracle.

Etapas básicas para adicionar uma opção a uma instância de banco de dados do RDS Custom para Oracle

O procedimento geral para adicionar uma opção à instância de banco de dados do RDS Custom for Oracle é o seguinte:

1. Crie um novo grupo de opções, ou copie ou modifique um existente.
2. Adicione a opção ao grupo de opções.
3. Associe o novo grupo de opções à instância de banco de dados ao criá-la ou modificá-la.

Criar um grupo de opções no RDS Custom para Oracle

Você pode criar um grupo de opções que obtenha suas configurações do grupo de opções padrão. Então, você adiciona uma ou mais opções ao novo grupo de opções. Ou, se você já possui um grupo de opções, pode copiar esse grupo com todas as suas opções em um novo grupo de opções. Para saber como copiar um grupo de opções, consulte Copiar um grupo de opções.

Os grupos de opções padrão do RDS Custom para Oracle são os seguintes:

- `default:custom-oracle-ee`
- `default:custom-oracle-se2`
- `default:custom-oracle-ee-cdb`
- `default:custom-oracle-se2-cdb`

Ao criar um grupo de opções, as configurações são derivadas do grupo de opções padrão. Depois de adicionar a opção `TIME_ZONE`, será possível associar o grupo de opções à instância de banco de dados.

Console

Uma maneira de criar um grupo de opções é usando o AWS Management Console.

Para criar um novo grupo de opções usando o console

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Option groups (Grupos de opções).
3. Escolha Create group (Criar grupo).
4. Na janela Create option group (Criar grupo de opções), faça o seguinte:
   a. Em Name (Nome), digite um nome para o grupo de opções que seja exclusivo na sua conta da AWS. O nome pode conter somente letras, números e hifens.
   b. Em Description (Descrição), digite uma breve descrição do grupo de opções. A descrição é usada para fins de exibição.
   c. Em Mecanismo, selecione qualquer um dos seguintes mecanismos de banco de dados do RDS Custom para Oracle:
      - custom-oracle-ee
      - custom-oracle-se2
      - custom-oracle-ee-cdb
      - custom-oracle-se2-cdb
   d. Em Versão do mecanismo principal, selecione uma versão do mecanismo principal compatível com o RDS Custom para Oracle. Para ter mais informações, consulte Regiões e mecanismos de banco de dados compatíveis com o RDS Custom para Oracle.
5. Escolha Create (Criar) para continuar. Para cancelar a operação, escolha Cancel (Cancelar).

AWS CLI

Para criar um grupo de opções, use o comando AWS CLI da `create-option-group` com os seguintes parâmetros obrigatórios.

- `--option-group-name`
- `--engine-name`
- `--major-engine-version`
- `--option-group-description`

Example

O exemplo a seguir cria um grupo de opções denominado `testoptiongroup`, que está associado ao mecanismo de banco de dados Oracle Enterprise Edition. A descrição está entre aspas.

Para Linux, macOS ou Unix:

```
aws rds create-option-group \
    --option-group-name testoptiongroup \
    --engine-name custom-oracle-ee-cdb \
    --major-engine-version 19 \
    --option-group-description "Test option group for a Custom Oracle CDB"
```

Para Windows:

```
aws rds create-option-group ^
    --option-group-name testoptiongroup ^
    --engine-name custom-oracle-ee-cdb ^
    --major-engine-version 19 ^
    --option-group-description "Test option group for a Custom Oracle CDB"
```

API do RDS

Para criar um grupo de opções, chame a operação `CreateOptionGroup` da API do Amazon RDS.

Associar um grupo de opções a uma instância de banco de dados do RDS Custom para Oracle

É possível associar o grupo de opções a uma instância de banco de dados nova ou existente:

- Para uma nova instância de banco de dados, aplique o grupo de opções ao criar a instância. Para ter mais informações, consulte Criar uma instância de banco de dados do RDS Custom for Oracle.
- Para uma instância de banco de dados existente, aplique o grupo de opções modificando a instância e anexando o novo grupo de opções. Para ter mais informações, consulte Modificar sua instância de banco de dados do RDS Custom para Oracle.

Fuso horário da Oracle

Para alterar o fuso horário do sistema usado pela instância de banco de dados do RDS Custom para Oracle, use a opção de fuso horário. Por exemplo, é possível alterar o fuso horário de uma instância de banco de dados para que ele seja compatível com um ambiente local ou um aplicativo herdado.

A opção de fuso horário altera o fuso horário no nível do host. Alterar o fuso horário afeta todas as colunas e valores de data, incluindo `SYSDATE` e `SYSTIMESTAMP`.

Tópicos

- Configurações de opções de fuso horário no RDS Custom para Oracle
- Fusos horários disponíveis no RDS Custom para Oracle
- Considerações sobre a configuração de fuso horário no RDS Custom para Oracle
- Limitações da configuração de fuso horário no RDS Custom para Oracle
- Adicionar a opção de fuso horário a um grupo de opções
- Remover opção de fuso horário

Configurações de opções de fuso horário no RDS Custom para Oracle

O Amazon RDS oferece suporte para as seguintes configurações da opção de fuso horário.

Configuração da opção	Valores válidos	Descrição
`TIME_ZONE`	Um dos fusos horários disponíveis. Para ver a lista completa, consulte Fusos horários disponíveis no RDS Custom para Oracle.	O novo fuso horário da sua instância de banco de dados.

Fusos horários disponíveis no RDS Custom para Oracle

Os valores a seguir podem ser usados para a opção de fuso horário.

Zona	Time zone (Fuso horário)
África	África/Cairo, África/Casablanca, África/Harare, África/Lagos, África/Luanda, África/Monrovia, África/Nairobi, África/Tripoli, África/Windhoek
América	América/Araguaina, América/Argentina/Buenos_Aires, América/Asuncion, América/Bogota, América/Caracas, América/Chicago, América/Chihuahua, América/Cuiaba, América/Denver, América/Detroit, América/Fortaleza, América/Godthab, América/Guatemala, América/Halifax, América/Lima,

Zona	Time zone (Fuso horário)
	América/Los_Angeles, América/Manaus, América/Matamoros, América/Mexico_City, América/Monterrey, América/Montevideo, América/New_York, América/Phoenix, América/Santiago, América/Sao_Paulo, América/Tijuana, América/Toronto
Ásia	Ásia/Amman, Ásia/Ashgabat, Ásia/Baghdad, Ásia/Baku, Ásia/Bangkok, Ásia/Beirut, Ásia/Calcutta, Ásia/Damascus, Ásia/Dhaka, Ásia/Hong_Kong, Ásia/Irkutsk, Ásia/Jakarta, Ásia/Jerusalem, Ásia/Kabul, Ásia/Karachi, Ásia/Kathmandu, Ásia/Kolkata, Ásia/Krasnoyarsk, Ásia/Magadan, Ásia/Manila, Ásia/Muscat, Ásia/Novosibirsk, Ásia/Rangoon, Ásia/Riyadh, Ásia/Seoul, Ásia/Shanghai, Ásia/Singapore, Ásia/Taipei, Ásia/Tehran, Ásia/Tokyo, Ásia/Ulaanbaatar, Ásia/Vladivostok, Ásia/Yakutsk, Ásia/Yerevan
Atlântico	Atlântico/Açores, Atlântico/Cape_Verde
Austrália	Austrália/Adelaide, Austrália/Brisbane, Austrália/Darwin, Austrália/Eucla, Austrália/Hobart, Austrália/Lord_Howe, Austrália/Perth, Austrália/Sydney
Brasil	Brasil/DeNoronha, Brasil/Leste
Canadá	Canadá/Newfoundland, Canadá/Saskatchewan
Etc	Etc/GMT-3
Europa	Europa/Amsterdam, Europa/Athens, Europa/Berlin, Europa/Dublin, Europa/Helsinki, Europa/Kaliningrad, Europa/London, Europa/Madrid, Europa/Moscow, Europa/Paris, Europa/Prague, Europa/Rome, Europa/Sarajevo
Pacífico	Pacífico/Apia, Pacífico/Auckland, Pacífico/Chatham, Pacífico/Fiji, Pacífico/Guam, Pacífico/Honolulu, Pacífico/Kiritimati, Pacífico/Marquesas, Pacífico/Samoa, Pacífico/Tongatapu, Pacífico/Wake
EUA	EUA/Alasca, EUA/Central, EUA/East-Indiana, EUA/Eastern, EUA/Pacífico
UTC	UTC

Considerações sobre a configuração de fuso horário no RDS Custom para Oracle

Se você optar por definir o fuso horário da instância de banco de dados, pense no seguinte:

- Quando você adiciona a opção de fuso horário, ocorre uma breve interrupção enquanto sua instância de banco de dados é reiniciada automaticamente.
- Se você definir o fuso horário incorretamente por engano, será necessário recuperar a instância de banco de dados para sua configuração de fuso horário anterior. Por esse motivo, sugerimos fortemente que você use uma das seguintes estratégias antes de adicionar a opção de fuso horário à instância:
  - Se a instância de banco de dados do RDS Custom para Oracle usar o grupo de opções padrão, crie um snapshot da instância de banco de dados. Para ter mais informações, consulte Criar um snapshot do RDS Custom for Oracle.
  - Se a instância de banco de dados atualmente usa um grupo de opções não padrão, crie um snapshot da instância de banco de dados e, depois, crie o grupo de opções com a opção de fuso horário.
- É altamente recomendável fazer backup da instância de banco de dados manualmente depois de aplicar a opção `Timezone`.
- É altamente recomendável testar a opção de fuso horário em uma instância de banco de dados de teste antes de adicioná-la a uma instância de banco de dados de produção. Adicionar a opção de fuso horário pode causar problemas com tabelas que usam a data do sistema para adicionar datas ou horários. Recomendamos que você analise seus dados e aplicações para avaliar o impacto da mudança do fuso horário.

Limitações da configuração de fuso horário no RDS Custom para Oracle

Observe as seguintes limitações:

- Não é possível alterar o fuso horário diretamente no host sem movê-lo para fora do perímetro de suporte. Para alterar o fuso horário do banco de dados, é necessário criar um grupo de opções.
- Como a opção de fuso horário é uma opção persistente (mas não permanente), não é possível fazer o seguinte:
  - Remover a opção de um grupo de opções depois de adicioná-la.
  - Modificar a configuração de fuso horário da opção para um fuso horário diferente.
- Não é possível associar vários grupos de opções à instância de banco de dados do RDS Custom para Oracle.

- Não é possível definir o fuso horário para PDBs individuais em um CDB.

Adicionar a opção de fuso horário a um grupo de opções

Os grupos de opções padrão do RDS Custom para Oracle são os seguintes:

- `default:custom-oracle-ee`
- `default:custom-oracle-se2`
- `default:custom-oracle-ee-cdb`
- `default:custom-oracle-se2-cdb`

Ao criar um grupo de opções, as configurações são derivadas do grupo de opções padrão. Para ter informações gerais sobre grupos de opções no Amazon RDS, consulte Trabalhar com grupos de opções.

Console

Como adicionar a opção de fuso horário a um grupo de opções

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Option groups (Grupos de opções).
3. Selecione o grupo de opções que você deseja modificar e escolha Add Option (Adicionar opção).
4. Na janela Add option (Adicionar opção), faça o seguinte:
   a. Selecione Fuso horário.
   b. Em Configurações de opção, selecione um fuso horário.
   c. Para habilitar a opção em todas as instâncias de banco de dados associadas do RDS Custom para Oracle assim que a adicionar, em Aplicar imediatamente, selecione Sim. Se você selecionar Não (o padrão), a opção será habilitada para cada instância de banco de dados associada durante a próxima janela de manutenção.
   d.

> **Important**
>
> Se você adicionar a opção de fuso horário a um grupo de opções existente que já está anexado a uma ou mais instâncias de banco de dados, ocorre uma breve

interrupção enquanto todas as instâncias de banco de dados são reiniciadas automaticamente.

5. Quando estiver satisfeito com as configurações, escolha Add Option (Adicionar opção).
6. Faça backup das instâncias de banco de dados do RDS Custom para Oracle cujos fusos horários foram atualizados. Para ter mais informações, consulte Criar um snapshot do RDS Custom for Oracle.

AWS CLI

O exemplo a seguir usa o comando add-option-to-option-group da AWS CLI para adicionar a opção `Timezone` e a configuração da opção `TIME_ZONE` a um grupo de opções chamado de `testoptiongroup`. O fuso horário é definido como `America/Los_Angeles`.

Para Linux, macOS ou Unix:

```
aws rds add-option-to-option-group \
    --option-group-name "testoptiongroup" \
    --options "OptionName=Timezone,OptionSettings=[{Name=TIME_ZONE,Value=America/
Los_Angeles}]" \
    --apply-immediately
```

Para Windows:

```
aws rds add-option-to-option-group ^
    --option-group-name "testoptiongroup" ^
    --options "OptionName=Timezone,OptionSettings=[{Name=TIME_ZONE,Value=America/
Los_Angeles}]" ^
    --apply-immediately
```

Remover opção de fuso horário

A opção de fuso horário é uma opção persistente, mas não permanente. Você não poderá remover a opção de um grupo de opções depois de adicioná-la. Para dissociar o grupo de opções antigo da instância de banco de dados:

1. Crie um grupo de opções com uma opção `Timezone` atualizada.
2. Associe o novo grupo de opções à instância de banco de dados ao modificar a instância.

Migrar um banco de dados on-premises para o RDS Custom para Oracle

Antes de migrar um banco de dados Oracle on-premises para o RDS Custom para Oracle, considere os seguintes fatores:

- A quantidade de tempo de inatividade compatível com a aplicação
- O tamanho do banco de dados de origem
- Conectividade de rede
- Um requisito para um plano de fallback
- A versão do banco de dados Oracle de origem e de destino e os tipos de sistema operacional da instância de banco de dados
- Ferramentas de replicação disponíveis, como AWS Database Migration Service, Oracle GoldenGate ou ferramentas de replicação de terceiros

Com base nesses fatores, você pode selecionar migração física, migração lógica ou uma combinação. Se você selecionar a migração física, poderá usar as seguintes técnicas:

Duplicação de RMAN

A duplicação ativa do banco de dados não exige um backup do banco de dados de origem. Ele duplica o banco de dados de origem ativo para o host de destino copiando os arquivos do banco de dados pela rede para a instância auxiliar. O comando `DUPLICATE` RMAN copia os arquivos necessários como cópias de imagem ou conjuntos de backup. Para aprender essa técnica, consulte a publicação do blog AWS Migração física de bancos de dados Oracle para o Amazon RDS Custom usando a duplicação RMAN.

Oracle Data Guard

Nessa técnica, é possível fazer backup de um banco de dados on-premises e copiar os backups para um bucket do Amazon S3. Depois, é possível copiar os backups para a sua instância de banco de dados do RDS Custom para Oracle. Depois de realizar a configuração necessária, você alterna manualmente o banco de dados principal para o banco de dados em espera do RDS Custom para Oracle. Para aprender essa técnica, consulte a publicação do blog AWS Migração física de bancos de dados Oracle para o Amazon RDS Custom usando o Data Guard.

Para ter informações gerais sobre a importação de dados de forma lógica para o RDS para Oracle, consulte Importar dados para o Oracle no Amazon RDS.

Fazer upgrade de uma instância de banco de dados para o Amazon RDS Custom for Oracle

Você pode fazer upgrade de uma instância de banco de dados do Amazon RDS Custom modificando-a para utilizar uma nova versão de mecanismo personalizada (CEV). Para obter informações gerais sobre upgrades, consulte Atualizar a versão de mecanismo de uma instância de banco de dados.

Tópicos

- Visão geral das atualizações do mecanismo RDS para Oracle
- Requisitos para atualizações do RDS Custom para Oracle
- Considerações sobre atualizações do banco de dados do RDS Custom para Oracle
- Considerações sobre atualizações do sistema operacional do RDS Custom para Oracle
- Visualizar destinos de upgrade válidos para instâncias de banco de dados do RDS Custom for Oracle
- Fazer upgrade de uma instância de banco de dados para o RDS Custom for Oracle
- Visualizar upgrades pendentes para instâncias de banco de dados do RDS Custom
- Solucionar falha no upgrade de uma instância de banco de dados do RDS Custom para Oracle

Visão geral das atualizações do mecanismo RDS para Oracle

Com o RDS Custom for Oracle, você pode corrigir seu banco de dados Oracle ou seu sistema operacional (OS) de instância de banco de dados criando novos CEVs e modificando sua instância para usar o novo CEV.

Tópicos

- Opções de atualização de CEV
- Aplicação de patches sem CEVs
- Etapas gerais para corrigir sua instância de banco de dados com um CEV

Opções de atualização de CEV

Ao criar uma CEV para uma atualização, existem as seguintes opções mutuamente exclusivas:

Somente banco de dados

Reutilize a Amazon Machine Image (AMI) atualmente em uso pela sua instância de banco de dados, mas especifique diferentes binários de banco de dados. O RDS Custom aloca um novo volume binário e o anexa à instância existente do Amazon EC2. O RDS Custom substituem o volume completo do banco de dados por um novo volume que usa a versão do banco de dados de destino.

Somente sistema operacional

Reutilize os binários do banco de dados atualmente em uso pela sua instância de banco de dados, mas especifique uma AMI diferente. O RDS Custom aloca uma nova instância do Amazon EC2 e, em seguida, anexa o volume binário existente à nova instância. O volume do banco de dados existente é mantido.

Se você quiser atualizar o sistema operacional e o banco de dados, deverá atualizar o CEV duas vezes. Você pode atualizar o sistema operacional e depois o banco de dados ou atualizar o banco de dados e depois o sistema operacional.

> **Warning**
>
> Ao corrigir seu sistema operacional, você perde os dados do volume raiz e qualquer personalização existente do sistema operacional. Portanto, é altamente recomendável não usar o volume para instalações ou para armazenar dados ou arquivos permanentes. Recomendamos também que você faça backup dos dados antes da atualização.

Aplicação de patches sem CEVs

É altamente recomendável que você faça upgrade de sua instância de banco de dados do RDS Custom para Oracle usando CEVs. A automação do RDS Custom para Oracle sincroniza os metadados do patch com o binário do banco de dados na instância de banco de dados

Em circunstâncias especiais, o RDS Custom oferece suporte à aplicação de um patch “único” diretamente na instância subjacente do Amazon EC2 usando OPATCH. Um caso de uso válido pode ser um patch que você queira aplicar imediatamente, mas a equipe do RDS Custom está atualizando o recurso de CEV, causando um atraso. Para aplicar um patch manualmente, realize as seguintes etapas:

1. Pause a automação do RDS Custom.

2. Aplique seu patch nos binários de banco de dados na instância do Amazon EC2.
3. Retome a automação do RDS Custom.

Uma desvantagem da técnica anterior é que você deve aplicar o patch manualmente em cada instância que deseja atualizar. Por outro lado, ao criar uma CEV, você pode criar ou fazer upgrade de várias instâncias de banco de dados usando a mesma CEV.

Etapas gerais para corrigir sua instância de banco de dados com um CEV

Se você aplicar patches no SO ou no banco de dados, execute as seguintes etapas básicas:

1. Crie um CEV que contenha um dos itens a seguir, dependendo se você está corrigindo o banco de dados ou o sistema operacional:
   - A Oracle Database RU que deseja aplicar à instância de banco de dados
   - Uma AMI diferente — a mais recente disponível ou a que você especificar — e um CEV existente para usar como fonte

   Siga as etapas em Criar um CEV.
2. (Opcional para aplicação de patches no banco de dados) Verifique as atualizações disponíveis da versão do mecanismo executando. `describe-db-engine-versions`
3. Inicie o processo de aplicação de patches executando`modify-db-instance`.

   O status da instância que está sendo corrigida difere da seguinte forma:
   - Enquanto o RDS aplica patches no banco de dados, o status da instância de banco de dados muda para Atualização.
   - Enquanto o RDS aplica patches no SO, o status da instância de banco de dados muda para Modificando.

   Quando a instância de banco de dados tem o status Disponível, a correção é concluída.
4. Confirme se sua instância de banco de dados usa o novo CEV executando`describe-db-instances`.

Requisitos para atualizações do RDS Custom para Oracle

Para atualizar a instância de banco de dados do RDS Custom para Oracle, os seguintes requisitos devem ser atendidos:

- A CEV de destino para o qual você está fazendo a atualização deve existir.
- Você deve atualizar o sistema operacional ou o banco de dados em uma única operação. Não há suporte para atualizar o sistema operacional e o banco de dados em uma única chamada de API.
- A CEV de destino deve usar as configurações dos parâmetros de instalação que estão no manifesto da CEV atual. Por exemplo, não é possível atualizar um banco de dados que use o Oracle home padrão para uma CEV que use um Oracle home não padrão.
- A CEV de destino deve usar uma nova versão secundária do banco de dados, não uma nova versão principal. Por exemplo, não é possível fazer atualização de uma CEV do Oracle Database 12c para uma CEV do Oracle Database 19c. Mas é possível atualizar da versão 2.,0.0.0.ru-2023-04.rur-2023-04.r1 para a versão 21.0.0.0.ru-2023-07.rur-2023-07.r1.
- Para atualizações do sistema operacional, o CEV de destino deve usar uma AMI diferente, mas ter a mesma versão principal.

Considerações sobre atualizações do banco de dados do RDS Custom para Oracle

Se você planeja atualizar o banco de dados, pense no seguinte:

- A versão do sistema operacional (SO) atualmente compatível é o Oracle Linux 8. Para continuar recebendo as atualizações e os patches de segurança mais recentes do RDS Custom para Oracle, atualize suas instâncias de banco de dados para o Oracle Linux 8 especificando um CEV com base nesse sistema operacional. O Oracle Database 12c Release 1 (12.1), o Oracle Database 12c Release 2 (12.2) e o Oracle Database 19c são as únicas versões que oferecem suporte ao Oracle Linux 8. Para migrar para a AMI mais recente do Oracle Linux 8, atualize seu sistema operacional para a AMI mais recente. Para ter mais informações, consulte Fazer upgrade de uma instância de banco de dados para o RDS Custom for Oracle.

  O suporte ao Oracle Linux 7.9 terminou em 31 de dezembro de 2024. Para continuar executando o Oracle Linux 7 após o fim do suporte, adquira uma licença do Oracle Extended Support. Você é responsável pelas atualizações de segurança e deve aplicar patches manualmente nas instâncias do RDS Custom para Oracle. Para obter mais informações, consulte Política de suporte durante a vida útil: cobertura para ofertas de serviços de código abertos Oracle.
- Quando você faz upgrade da instância de banco de dados primária, o RDS Custom para Oracle faz upgrade das réplicas de leitura automaticamente. No entanto, ao atualizar o sistema operacional, você deve atualizar as réplicas de leitura manualmente.
- Ao atualizar um banco de dados de contêiner (CDB) para uma nova versão de banco de dados, o RDS Custom para Oracle confere se todos os PDBs estão abertos ou podem ser abertos. Se

essas condições não forem atendidas, o RDS Custom interromperá a verificação e retornará o banco de dados ao seu estado original sem tentar a atualização. Se as condições forem atendidas, o RDS Custom corrigirá primeiro a raiz do CDB e depois corrigirá todos os outros PDBs (inclusive `PDB$SEED`) em paralelo.

Depois que a aplicação de patches for concluída, o RDS Custom tentará abrir todos os PDBs. Se algum PDB não abrir, você receberá o seguinte evento: `The following PDBs failed to open: list-of-PDBs`. Se o RDS Custom não corrigir a raiz do CDB ou qualquer PDB, a instância será colocada no estado `PATCH_DB_FAILED`.

- Talvez você queira realizar uma atualização de versão principal e uma conversão de não CDB em CDB ao mesmo tempo. Nesse caso, recomendamos que você proceda da seguinte forma:
  1. Crie uma instância de banco de dados do RDS Custom para Oracle que use a arquitetura multilocatária da Oracle.
  2. Conecte um não CDB à raiz do CDB, criando-o como um PDB. O não CDB deve ser da mesma versão principal que seu CDB.
  3. Converta o PDB executando o script SQL `noncdb_to_pdb.sql` da Oracle.
  4. Valide sua instância do CDB.
  5. Atualize sua instância do CDB.

Considerações sobre atualizações do sistema operacional do RDS Custom para Oracle

Ao planejar uma atualização do sistema operacional, pense no seguinte:

- Não é possível fornecer a própria AMI para uso em uma CEV do RDS Custom para Oracle. É possível especificar a AMI padrão, que usa o Oracle Linux 8, ou uma AMI que tenha sido usada anteriormente por uma CEV do RDS Custom para Oracle.

> **Note**
>
> O RDS Custom para Oracle lança uma nova AMI padrão quando vulnerabilidades e exposições comuns são descobertas. Nenhum cronograma fixo está disponível ou garantido. O RDS Custom para Oracle tende a publicar uma nova AMI padrão a cada trinta dias.

- Ao atualizar o sistema operacional na instância de banco de dados primária, é necessário atualizar as réplicas de leitura associadas manualmente.
- Reserve capacidade computacional suficiente do Amazon EC2 para o tipo de instância na AZ antes de começar a aplicar patches no sistema operacional.

  Ao criar uma reserva de capacidade, você especifica a AZ, o número de instâncias e os atributos da instância (incluindo o tipo de instância). Por exemplo, se a instância de banco de dados utiliza o tipo de instância de EC2 subjacente r5.large, convém reservar a capacidade do EC2 para r5.large na AZ. Durante a aplicação de patches no sistema operacional, o RDS Custom cria um novo host do tipo db.r5.large, que pode ficar preso se o AZ não tiver capacidade do EC2 para esse tipo de instância. Se você reservar a capacidade do EC2, reduzirá o risco de bloqueios de patches causados por restrições de capacidade. Consulte mais informações em On-Demand Capacity Reservations no Guia do usuário do Amazon EC2.
- Faça backup da instância de banco de dados antes de atualizar o sistema operacional. A atualização remove os dados do volume raiz e todas as personalizações existentes do sistema operacional.
- No modelo de responsabilidade compartilhada, você é responsável por manter o sistema operacional atualizado. O RDS Custom para Oracle não determina quais patches são aplicados ao sistema operacional. Se o RDS Custom para Oracle estiver funcional, será possível usar a AMI associada a essa CEV indefinidamente.

Visualizar destinos de upgrade válidos para instâncias de banco de dados do RDS Custom for Oracle

Você pode visualizar CEVs existentes na página Custom engine versions (Versões de mecanismo personalizadas) no AWS Management Console.

Também pode utilizar o comando describe-db-engine-versions da AWS CLI para encontrar upgrades válidos para suas instâncias de banco de dados, como mostra o exemplo a seguir. Este exemplo pressupõe que você criou uma instância de banco de dados usando a versão `19.my_cev1` do mecanismo e que as versões `19.my_cev2` de upgrade `19.my_cev` existem.

```
aws rds describe-db-engine-versions --engine custom-oracle-ee --engine-version
 19.my_cev1
```

A saída será semelhante à seguinte. O `ImageId` campo mostra o ID da AMI.

```
{
    "DBEngineVersions": [
        {
            "Engine": "custom-oracle-ee",
            "EngineVersion": "19.my_cev1",
            ...
            "Image": {
                "ImageId": "ami-2345",
                "Status": "active"
            },
            "DBEngineVersionArn": "arn:aws:rds:us-west-2:123456789012:cev:custom-
oracle-ee/19.my_cev1/12a34b5c-67d8-90e1-2f34-gh56ijk78lm9"
            "ValidUpgradeTarget": [
                {
                    "Engine": "custom-oracle-ee",
                    "EngineVersion": "19.my_cev2",
                    "Description": "19.my_cev2 description",
                    "AutoUpgrade": false,
                    "IsMajorVersionUpgrade": false
                },
                {
                    "Engine": "custom-oracle-ee",
                    "EngineVersion": "19.my_cev3",
                    "Description": "19.my_cev3 description",
                    "AutoUpgrade": false,
                    "IsMajorVersionUpgrade": false
                }
            ]
            ...
```

Fazer upgrade de uma instância de banco de dados para o RDS Custom for Oracle

Para fazer upgrade da sua instância de banco de dados do RDS Custom, você a modifica para utilizar uma nova CEV. Esse CEV pode conter novos binários de banco de dados ou uma nova AMI. Por exemplo, para atualizar sua instância de banco de dados Oracle Linux 7.9 para o Oracle Linux 8, especifique a AMI mais recente, que usa o Oracle Linux 8. Para atualizar o banco de dados e o sistema operacional, deverá realizar duas atualizações separadas.

> **Note**
>
> Se você atualizar o banco de dados, o RDS Custom atualizará automaticamente as réplicas de leitura após atualizar a instância de banco de dados primária. Se você atualizar o SO, deverá atualizar as réplicas manualmente.

Antes de começar, revise Requisitos para atualizações do RDS Custom para Oracle Considerações sobre atualizações do banco de dados do RDS Custom para Oracle e.

Console

Como modificar uma instância de banco de dados do RDS Custom para Oracle

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Databases (Bancos de dados) e a instância de banco de dados do RDS Custom para SQL Server que você deseja excluir.
3. Selecione Modify. A página Modify DB instance (Modificar instância de banco de dados) será exibida.
4. Em DB engine version (Versão do mecanismo de banco de dados), selecione outra CEV. Faça o seguinte:

   - Se você estiver corrigindo o banco de dados, certifique-se de que o CEV especifique binários de banco de dados diferentes daqueles usados pela sua instância de banco de dados e não especifique uma AMI diferente da AMI usada atualmente pela sua instância de banco de dados.
   - Se você estiver corrigindo o sistema operacional, certifique-se de que o CEV especifique uma AMI diferente da AMI usada atualmente pela sua instância de banco de dados e não especifique binários de banco de dados diferentes.

   > **Warning**
   >
   > Ao corrigir seu sistema operacional, você perde os dados do volume raiz e qualquer personalização existente do sistema operacional.

5. Escolha Continue (Continuar) e verifique o resumo de modificações.

Escolha Apply immediately (Aplicar imediatamente) para aplicar as alterações imediatamente.

6. Se estiverem corretas, escolha Modify DB Instance (Modificar instância de banco de dados). Ou escolha Back (Voltar) para editar as alterações ou Cancel (Cancelar) para cancelar as alterações.

AWS CLI

Os exemplos a seguir mostram possíveis cenários de atualização. Os exemplos pressupõem que você criou uma instância de banco de dados do RDS Custom for Oracle com as seguintes características:

- Nome da instância de banco de dados
- CEV chamado `19.my_cev1`
- Oracle Database 19c usando a arquitetura não CDB
- Oracle Linux 8 usando AMI `ami-1234`

A AMI mais recente fornecida pelo serviço é. `ami-2345` Você pode encontrar o endpoint executando o comando `describe-db-engine-versions` da CLI:

Tópicos

- Atualizando o SO
- Atualizando o banco de dados

Atualizando o SO

Neste exemplo, você deseja fazer o upgrade `ami-1234` para`ami-2345`, que é a AMI mais recente fornecida pelo serviço. Como essa é uma atualização do sistema operacional, os binários do banco de dados para `ami-1234` e `ami-2345` devem ser os mesmos. Você cria um novo CEV chamado `19.my_cev2` com base em. `19.my_cev1`

Example

Para Linux, macOS ou Unix:

```
aws rds create-custom-db-engine-version \
    --engine custom-oracle-ee \
```

```
    --engine-version 19.my_cev2 \
    --description "Non-CDB CEV based on ami-2345" \
    --kms-key-id key-name \
    --source-custom-db-engine-version-identifer arn:aws:rds:us-
west-2:123456789012:cev:custom-oracle-ee/19.my_cev1/12345678-ab12-1234-cde1-
abcde123456789 \
    --image-id ami-2345
```

Para Windows:

```
aws rds create-custom-db-engine-version ^
    --engine custom-oracle-ee ^
    --engine-version 19.my_cev2 ^
    --description "Non-CDB CEV based on ami-2345" ^
    --kms-key-id key-name ^
    --source-custom-db-engine-version-identifer arn:aws:rds:us-
west-2:123456789012:cev:custom-oracle-ee/19.my_cev1/12345678-ab12-1234-cde1-
abcde123456789 ^
    --image-id ami-2345
```

Para fazer upgrade de uma instância de banco de dados do RDS Custom, utilize o comando modify-db-instance da AWS CLI com os seguintes parâmetros:

- `--db-instance-identifier`— Especifique a instância de banco de dados do RDS Custom for Oracle a ser atualizada.
- `--engine-version`— Especifique o CEV que tem a nova AMI.
- `--no-apply-immediately` | `--apply-immediately` – Se você deseja realizar o upgrade imediatamente ou aguardar até a janela de manutenção agendada

O exemplo a seguir faz upgrade do `my-custom-instance` para a versão `19.my_cev2`. Somente o sistema operacional é atualizado.

Example

Para Linux, macOS ou Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier my-custom-instance \
    --engine-version 19.my_cev2 \
    --apply-immediately
```

Para Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier my-custom-instance ^
    --engine-version 19.my_cev2 ^
    --apply-immediately
```

Atualizando o banco de dados

Neste exemplo, convém aplicar o patch Oracle p35042068 à instância de banco de dados do RDS para Oracle. Como você atualizou seu sistema operacional emAtualizando o SO, sua instância de banco de dados está sendo usada atualmente`19.my_cev2`, com base em`ami-2345`. Você cria um novo CEV chamado `19.my_cev3` que também usa`ami-2345`, mas especifica um novo manifesto JSON na variável de `$MANIFEST` ambiente. Portanto, somente os binários do banco de dados são diferentes em seu novo CEV e no CEV que sua instância está usando atualmente.

Example

Para Linux, macOS ou Unix:

```
aws rds create-custom-db-engine-version \
    --engine custom-oracle-ee \
    --engine-version 19.my_cev3 \
    --description "Non-CDB CEV with p35042068 based on ami-2345" \
    --kms-key-id key-name \
    --image-id ami-2345 \
    --manifest $MANIFEST
```

Para Windows:

```
aws rds create-custom-db-engine-version ^
    --engine custom-oracle-ee ^
    --engine-version 19.my_cev3 ^
    --description "Non-CDB CEV with p35042068 based on ami-2345" ^
    --kms-key-id key-name ^
    --image-id ami-2345 ^
    --manifest $MANIFEST
```

O exemplo a seguir faz upgrade do `my-custom-instance` para a versão `19.my_cev3`. Somente o banco de dados é atualizado.

Example

Para Linux, macOS ou Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier my-custom-instance \
    --engine-version 19.my_cev3 \
    --apply-immediately
```

Para Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier my-custom-instance ^
    --engine-version 19.my_cev3 ^
    --apply-immediately
```

Visualizar upgrades pendentes para instâncias de banco de dados do RDS Custom

É possível ver upgrades pendentes para as instâncias de banco de dados do Amazon RDS Custom utilizando o comando describe-db-instances ou describe-pending-maintenance-actions da AWS CLI.

Contudo, essa abordagem não funcionará se você tiver usado a opção `--apply-immediately` ou se o upgrade estiver em andamento.

O comando `describe-db-instances` a seguir mostra upgrades pendentes para `my-custom-instance`.

```
aws rds describe-db-instances --db-instance-identifier my-custom-instance
```

A saída será semelhante à seguinte.

```
{
    "DBInstances": [
        {
           "DBInstanceIdentifier": "my-custom-instance",
            "EngineVersion": "19.my_cev1",
            ...
            "PendingModifiedValues": {
                "EngineVersion": "19.my_cev3"
            ...
            }
```

```
        }
    ]
}
```

Solucionar falha no upgrade de uma instância de banco de dados do RDS Custom para Oracle

Se um upgrade de instância de banco de dados do RDS Custom falhar, um evento do RDS será gerado, e o status da instância de banco de dados se tornará `upgrade-failed`.

Você pode ver esse status utilizando o comando describe-db-instances da AWS CLI, como mostra o exemplo a seguir.

```
aws rds describe-db-instances --db-instance-identifier my-custom-instance
```

A saída será semelhante à seguinte.

```
{
    "DBInstances": [
        {
           "DBInstanceIdentifier": "my-custom-instance",
            "EngineVersion": "19.my_cev1",
            ...
            "PendingModifiedValues": {
                "EngineVersion": "19.my_cev3"
            ...
            }
            "DBInstanceStatus": "upgrade-failed"
        }
    ]
}
```

Após uma falha de upgrade, todas as ações do banco de dados são bloqueadas, exceto para modificar a instância de banco de dados para realizar as seguintes tarefas:

- Nova tentativa de executar o mesmo upgrade
- Pausar e retomar a automação do RDS Custom
- Recuperação em um ponto anterior no tempo (PITR)
- Exclusão de uma instância de banco de dados

> **Note**
>
> Se a automação tiver sido pausada para a instância de banco de dados do RDS Custom, você não poderá repetir o upgrade até retomá-la.
> As mesmas ações são aplicáveis a uma falha de upgrade para uma réplica de leitura gerenciada pelo RDS e a uma instância principal.

Para obter mais informações, consulte Solucionar problemas de upgrade do RDS Custom para Oracle.

Solucionar problemas de banco de dados do Amazon RDS Custom para Oracle

O modelo de responsabilidade compartilhada do RDS Custom fornece acesso ao nível de shell do SO e acesso pelo administrador do banco de dados. O RDS Custom executa recursos na sua conta, ao contrário do Amazon RDS, que executa recursos em uma conta do sistema. A extensão do acesso aumenta a responsabilidade. Nas seções a seguir, você vai aprender a solucionar problemas em instâncias de banco de dados do Amazon RDS Custom.

> **Note**
>
> Esta seção explica como solucionar problemas do RDS Custom para Oracle. Para solucionar problemas do RDS Custom para Oracle, consulte Solucionar problemas de banco de dados do Amazon RDS Custom para SQL Server.

Tópicos

- Visualizar eventos personalizados do RDS Custom
- Assinar eventos do RDS Custom
- Solucionar problemas com a criação de uma versão de mecanismo personalizado para o RDS Custom for Oracle
- Corrigir configurações não compatíveis no RDS Custom para Oracle
- Solucionar problemas de upgrade do RDS Custom para Oracle
- Solucionar problemas de promoção de réplicas no RDS Custom para Oracle

Visualizar eventos personalizados do RDS Custom

O procedimento para visualizar eventos é o mesmo para instâncias de banco de dados do RDS Custom e do Amazon RDS. Para ter mais informações, consulte Visualizar eventos do Amazon RDS.

Para visualizar a notificação de eventos do RDS Custom utilizando a AWS CLI, execute o comando `describe-events`. O RDS Custom introduz vários novos eventos. As categorias dos eventos são as mesmas que as do Amazon RDS. Para visualizar a lista de eventos, consulte Categorias de eventos e mensagens de eventos do Amazon RDS.

O exemplo a seguir recupera detalhes dos eventos que ocorreram para a instância de banco de dados do RDS Custom especificada.

```
aws rds describe-events \
    --source-identifier my-custom-instance \
    --source-type db-instance
```

Assinar eventos do RDS Custom

O procedimento para assinar eventos é o mesmo para instâncias de banco de dados do RDS Custom e do Amazon RDS. Para ter mais informações, consulte Inscrever-se em notificações de eventos do Amazon RDS.

Para assinar a notificação de eventos do RDS Custom usando a CLI, execute o comando `create-event-subscription`. Inclua os seguintes parâmetros necessários:

- `--subscription-name`
- `--sns-topic-arn`

O exemplo a seguir cria uma assinatura para eventos de backup e recuperação para uma instância de banco de dados do RDS Custom na conta atual da AWS. As notificações são enviadas para um tópico do Amazon Simple Notification Service (Amazon SNS), especificado por `--sns-topic-arn`.

```
aws rds create-event-subscription \
    --subscription-name my-instance-events \
    --source-type db-instance \
    --event-categories '["backup","recovery"]' \
    --sns-topic-arn arn:aws:sns:us-east-1:123456789012:interesting-events
```

Solucionar problemas com a criação de uma versão de mecanismo personalizado para o RDS Custom for Oracle

Quando a criação da CEV falha, o RDS Custom emite `RDS-EVENT-0198` com a mensagem `Creation failed for custom engine version` *`major-engine-version.cev_name`* e inclui detalhes sobre a falha. Por exemplo, o evento imprime arquivos ausentes.

A criação da CEV pode falhar devido aos seguintes problemas:

- O bucket do Amazon S3 que contém seus arquivos de instalação não está na mesma região da AWS que a CEV.

- Quando você solicita a criação da CEV em um Região da AWS pela primeira vez, o RDS Custom cria um bucket do S3 para armazenar recursos do RDS Custom (como artefatos de CEV, logs de AWS CloudTrail e logs de transações).

  A criação da CEV falhará se o RDS Custom não conseguir criar o bucket do S3. O autor da chamada não tem permissões do S3, conforme descrito em Etapa 5: Conceder as permissões necessárias ao usuário ou ao perfil do IAM, ou o número de buckets do S3 atingiu o limite.
- O autor da chamada não tem permissões para obter arquivos do bucket do S3 que contém os arquivos da mídia de instalação. Essas permissões estão descritas em Etapa 7: Adicionar permissões do IAM necessárias.
- Sua política do IAM tem uma condição `aws:SourceIp`. Certifique-se de seguir as recomendações em A AWS nega acesso à AWS com base no IP da fonte no Guia do usuário do AWS Identity and Access Management. Certifique-se também de que o autor da chamada tenha as permissões do S3 descritas em Etapa 5: Conceder as permissões necessárias ao usuário ou ao perfil do IAM.
- Os arquivos da mídia de instalação listados no manifesto da CEV não estão no bucket do S3.
- As somas de verificação SHA-256 dos arquivos de instalação são desconhecidas para o RDS Custom.

  Confirme se as somas de verificação SHA-256 dos arquivos fornecidos correspondem à soma de verificação SHA-256 no site da Oracle. Se as somas de verificação corresponderem, entre em contato com o AWS Support e forneça o nome da CEV com falha, o nome do arquivo e a soma de verificação.
- A versão do OPatch é compatível com seus arquivos de patch. Você pode receber a seguinte mensagem: `OPatch is lower than minimum required version. Check that the version meets the requirements for all patches, and try again`. Para aplicar um patch Oracle, você deve usar uma versão compatível do utilitário OPatch. É possível encontrar a versão necessária do utilitário Opatch no arquivo readme do patch. Baixe o utilitário OPatch mais recente do My Oracle Support e tente criar sua CEV novamente.
- Os patches especificados no manifesto da CEV estão na ordem incorreta.

É possível visualizar eventos do RDS no console do RDS (no painel de navegação, escolha Events) ou utilizando o comando AWS CLI da `describe-events`. A duração padrão é de 60 minutos. Se nenhum evento for retornado, especifique uma duração maior, como mostra o exemplo a seguir.

```
aws rds describe-events --duration 360
```

Atualmente, o serviço MediaImport, que importa arquivos do Amazon S3 para criar CEVs, não está integrado ao AWS CloudTrail. Portanto, se você ativar o registro de dados em log para o Amazon RDS no CloudTrail, as chamadas para o serviço MediaImport, como o evento `CreateCustomDbEngineVersion` não serão registradas.

Porém, é possível ver chamadas do gateway de API que acessa seu bucket do Amazon S3. Essas chamadas são provenientes do serviço MediaImport para o evento `CreateCustomDbEngineVersion`.

Corrigir configurações não compatíveis no RDS Custom para Oracle

Devido ao modelo de responsabilidade compartilhada, é sua responsabilidade corrigir problemas de configuração que colocam a instância de banco de dados do RDS Custom para Oracle no estado `unsupported-configuration`. Se o problema for com a infraestrutura da AWS, use o console ou a AWS CLI para corrigi-lo. Se o problema for com o sistema operacional ou a configuração do banco de dados, faça login no host para corrigi-lo.

> **Note**
>
> Esta seção explica como corrigir configurações não compatíveis no RDS Custom para Oracle. Para obter informações sobre o RDS Custom para SQL Server, consulte Corrigir configurações não compatíveis no RDS Custom para SQL Server.

As tabelas a seguir incluem descrições de notificações e eventos que o perímetro de suporte envia e como corrigi-los. Essas notificações e o perímetro de suporte estão sujeitos a alterações. Para obter informações básicas sobre o perímetro de suporte, consultePerímetro de suporte do RDS Custom. Para obter informações sobre descrições de eventos, consulte Categorias de eventos e mensagens de eventos do Amazon RDS.

ID do evento	Configuração	Mensagem de evento do RDS	Ação
SP-00000	Configuração manual	The RDS Custom DB instance status is set to [Unsupported configura	Para resolver esse problema, crie um caso no Suporte.

ID do evento	Configuração	Mensagem de evento do RDS	Ação
	incompatível	tion] because of: *reason*.	

Recurso da AWS (infraestrutura)

ID do evento	Configuração	Mensagem de evento do RDS	Ação
SP-O1001	Volumes do Amazon Elastic Block Store (Amazon EBS)	The following EBS volumes were added to EC2 instance *ec2_id*: *volume_id* . To resolve the issue, detach the specified volumes from the instance.	O RDS Custom cria dois tipos de volume do EBS, além do volume raiz criado com base na imagem de máquina da Amazon (AMI), e os associa à instância do EC2: • O volume binário onde os binários do software de banco de dados estão localizados • Os volumes de dados onde os arquivos do banco de dados estão localizados Ao criar a instância de banco de dados, as configurações de armazenamento que você especifica configuram os volumes de dados. O perímetro de suporte monitora o seguinte: • Os volumes iniciais do EBS criados com a instância de banco de dados ainda estão associados à instância. • Se os volumes iniciais do EBS ainda têm as mesmas configurações inicialmente definidas: tipo de armazenamento, tamanho, IOPS provisionadas e taxa de transferência de armazenamento.

ID do evento	Configuração	Mensagem de evento do RDS	Ação

			• Nenhum outro volume do EBS está anexado à instância de banco de dados.
			
			Use o comando da CLI a seguir para comparar o tipo de volume dos detalhes do volume do EBS e os detalhes da instância de banco de dados do RDS Custom para Oracle:
			
			`aws rds describe-db-instances \`
			`--db-instance-identifier db-instance-name | grep StorageType`

ID do evento	Configuração	Mensagem de evento do RDS	Ação
SP-O1002	Volumes do Amazon Elastic Block Store (Amazon EBS)	EBS volume *volume_id* has been detached from EC2 instance [*ec2_id*]. You can't detach the original volume from this instance. To resolve the issue, re-attach *volume_id* to *ec2_id*.	O RDS Custom cria dois tipos de volume do EBS, além do volume raiz criado com base na imagem de máquina da Amazon (AMI), e os associa à instância do EC2: • O volume binário onde os binários do software de banco de dados estão localizados • Os volumes de dados onde os arquivos do banco de dados estão localizados Ao criar a instância de banco de dados, as configurações de armazenamento que você especifica configuram os volumes de dados. O perímetro de suporte monitora o seguinte: • Os volumes iniciais do EBS criados com a instância de banco de dados ainda estão associados à instância. • Se os volumes iniciais do EBS ainda têm as mesmas configurações inicialmente definidas: tipo de armazenamento, tamanho, IOPS provisionadas e taxa de transferência de armazenamento. • Nenhum outro volume do EBS está anexado à instância de banco de dados. Use o comando da CLI a seguir para comparar o tipo de volume dos detalhes do volume do EBS e os detalhes da instância de banco de dados do RDS Custom para Oracle: `aws rds describe-db-instances \`

ID do evento	Configuração	Mensagem de evento do RDS	Ação
			`--db-instance-identifier db-instance-name | grep StorageType`

ID do evento	Configuração	Mensagem de evento do RDS	Ação
SP-O1003	Volumes do Amazon Elastic Block Store (Amazon EBS)	The original EBS volume *volume_id* attached to EC2 instance *ec2_id* has been modified as follows: size [*X*] to [*Y*], type [*N*] to [*M*], or IOPS [*J*] to [*K*]. To resolve the issue, revert the modification.	O RDS Custom cria dois tipos de volume do EBS, além do volume raiz criado com base na imagem de máquina da Amazon (AMI), e os associa à instância do EC2: • O volume binário onde os binários do software de banco de dados estão localizados • Os volumes de dados onde os arquivos do banco de dados estão localizados Ao criar a instância de banco de dados, as configurações de armazenamento que você especifica configuram os volumes de dados. O perímetro de suporte monitora o seguinte: • Os volumes iniciais do EBS criados com a instância de banco de dados ainda estão associados à instância. • Se os volumes iniciais do EBS ainda têm as mesmas configurações inicialmente definidas: tipo de armazenamento, tamanho, IOPS provisionadas e taxa de transferência de armazenamento. • Nenhum outro volume do EBS está anexado à instância de banco de dados. Use o comando da CLI a seguir para comparar o tipo de volume dos detalhes do volume do EBS e os detalhes da instância de banco de dados do RDS Custom para Oracle: `aws rds describe-db-instances \`

ID do evento	Configuração	Mensagem de evento do RDS	Ação
			`--db-instance-identifier db-instance-name | grep StorageType`
SP-O1004	Status da instância do Amazon EC2	Automated recovery left EC2 instance [*ec2_id*] in an impaired state. To resolve the issue, see Troubleshooting instance recovery failures.	Para conferir o status de uma instância de banco de dados, use o console ou execute o seguinte comando AWS CLI: `aws rds describe-db-instances \ --db-instance-identifier *db-instance-name* |grep DBInstanceStatus`
SP-O1005	Atributos da instância do Amazon EC2	EC2 instance [*ec2_id*] was modified as follows: attribute [*att1*] changed from [*val-old*] to [*val-new*], attribute [*att2*] changed from [*val-old*] to [*val-new*]. To resolve the issue, revert to the original value.	

ID do evento	Configuração	Mensagem de evento do RDS	Ação
SP-O1006	Status da instância do Amazon EC2	EC2 instance [*ec2_id*] was terminated or can't be found. To resolve the issue, delete the RDS Custom DB instance.	O perímetro de suporte monitora notificações de alteração de estado da instância do EC2. A instância do EC2 deve estar sempre em execução. Para excluir a instância de banco de dados 1. Para conferir o status de uma instância de banco de dados, use o console ou execute o seguinte comando AWS CLI: `aws rds describe-db-instances \ --db-instance-identifier db-instance-name |grep DBInstanceStatus` 2. Exclua a instância de banco de dados do RDS Custom para Oracle.
SP-O1007	Status da instância do Amazon EC2	EC2 instance [*ec2_id*] was stopped. To resolve the issue, start the instance.	O perímetro de suporte monitora notificações de alteração de estado da instância do EC2. A instância do EC2 deve estar sempre em execução. Para reiniciar a instância de banco de dados 1. Para conferir o status de uma instância de banco de dados, use o console ou execute o seguinte comando AWS CLI: `aws rds describe-db-instances \ --db-instance-identifier db-instance-name |grep DBInstanceStatus` 2. Inicie a instância de banco de dados. 3. Remonte os volumes binários e de dados.

ID do evento	Configuração	Mensagem de evento do RDS	Ação

SP-10	Permissões do Amazon SQS	Permissions are missing for Amazon SQS. Check the permissions for the IAM instance profile, VPC endpoint policy, and dependent service connections, and then try again.	Você pode resolver esse problema garantindo que o perfil do IAM associado ao host tenha as seguintes permissões:
			`"SQS:SendMessage"`
			`"SQS:ReceiveMessage"`
			`"SQS:DeleteMessage"`
			`"SQS:GetQueueUrl"`

SP-10	Amazon Simple Queue Service (Amazon SQS)	The SQS queue [%s] was deleted and couldn't be recovered. To resolve this issue, recreate the queue.	Crie novamente a fila do Amazon SQS.

Sistema operacional

ID do evento	Configuração	Mensagem de evento do RDS	Ação
SP-O2001	Status do agente do RDS Custom	The RDS Custom agent isn't running on EC2 instance [*ec2_id*]. Make sure the agent is running on [*ec2_id*].	No RDS Custom para Oracle, a instância de banco de dados sairá do perímetro de suporte se o agente do RDS Custom parar. O agente publica a métrica `IamAlive` para o Amazon CloudWatch a cada 30 segundos. Um alarme será acionado se a métrica não for publicada por 30 segundos. O perímetro de

ID do evento	Configuração	Mensagem de evento do RDS	Ação

			suporte também monitora o estado do processo do agente do RDS Custom no host a cada 30 minutos.
			
			Para reiniciar o agente do RDS Custom
			
			1. Faça login no host e verifique se o agente do RDS Custom esteja em execução.
			2. Execute o comando a seguir para encontrar o status do agente.
			
			`service rdscustomagent status`
			
			3. Use o comando a seguir para iniciar o agente.
			
			`service rdscustomagent start`
			
			Quando o agente do RDS Custom estiver sendo executado novamente, a métrica `IamAlive` será publicada no Amazon CloudWatch, e o alarme mudará para o estado `OK`. Essa opção notifica o perímetro de suporte em que o agente está em execução.

ID do evento	Configuração	Mensagem de evento do RDS	Ação
SP-02002	Status do agente do AWS Systems Manager (agente do SSM)	The Systems Manager agent on EC2 instance [*ec2_id*] is unreachable. Make sure that you that have correctly configured the network, agent, and IAM permissions.	O SSM Agent sempre deve estar em execução. O agente do RDS Custom é responsável por garantir que o agente do Systems Manager esteja em execução. Se o SSM Agent foi encerrado e, depois, reiniciado, o agente do RDS Custom publica uma métrica no CloudWatch. O agente do RDS Custom tem um alarme na métrica definido para acionamento quando tiver ocorrido uma reinicialização em cada um dos três minutos anteriores. O perímetro de suporte também monitora o estado do processo do SSM Agent no host a cada 30 minutos. Para ter mais informações, consulte Solucionar problemas com o SSM Agent.
SP-02003	Status do agente do AWS Systems Manager (agente do SSM)	The Systems Manager agent on EC2 instance [*ec2_id*] crashed multiple times. For more information, see the SSM Agent troubleshooting documentation.	Para ter mais informações, consulte Solucionar problemas com o SSM Agent.

ID do evento	Configuração	Mensagem de evento do RDS	Ação
SP-O2004	Fuso horário do SO	The time zone on EC2 instance [*ec2_id*] was changed. To resolve this issue, revert the timezone to the previous setting of [*previous-time-zone*]. Then use an RDS options group to change the time zone.	A automação do RDS detectou que o fuso horário no host foi alterado sem o uso de um grupo de opções. Essa alteração em nível de host pode causar falhas na automação do RDS, portanto, a instância do EC2 é colocada no estado `unsupported-configuration`. Para corrigir a configuração do fuso horário 1. Faça login no host do EC2 e verifique o fuso horário do sistema operacional da seguinte forma: `timedatectl` 2. Pause a automação do RDS Custom. Para obter mais informações, consulte Pausar e retomar sua instância de banco de dados do RDS Custom. 3. Interrompa a instância de banco de dados. 4. Reverta a alteração de fuso horário no sistema operacional. 5. Inicie a instância de banco de dados. 6. Retome a automação do RDS Custom. A instância de banco de dados fica disponível em 30 minutos. Para não sair do perímetro no futuro, modifique o fuso horário por meio de um grupo de opções. Para obter mais informações, consulte Fuso horário da Oracle.

ID do evento	Configuração	Mensagem de evento do RDS	Ação
SP-O2005	Configurações de `sudo`	The sudo configurations on EC2 instance [*ec2_id*] lack necessary permissions. To resolve this issue, revert the recent changes to the sudo configurations.	O perímetro de suporte verifica se certos usuários do sistema operacional têm permissão para executar determinados comandos no host. Ele monitora as configurações `sudo` e as compara com o estado suportado. Se as configurações de `sudo` não tiverem suporte, o RDS Custom tentará substituí-las e voltará ao estado anterior com suporte. Se a tentativa for bem-sucedida, o RDS Custom enviará a seguinte notificação: O RDS Custom substituiu sua configuração com sucesso. Se a substituição não for bem-sucedida, sua instância de banco de dados permanecerá no estado de configuração não suportado. Para resolver esse problema, reverta as alterações no arquivo `sudoers.d/` ou corrija as permissões. Como investigar as alterações nas configurações **`sudo`** 1. Faça login no host. 2. Execute o seguinte comando: `visudo -c -f /etc/sudoers.d/ individual_sudo_files` 3. Modifique as configurações de `sudo` conforme necessário. Depois que o perímetro de suporte determina que as configurações de `sudo` têm suporte, a instância

ID do evento	Configuração	Mensagem de evento do RDS	Ação
			de banco de dados do RDS Custom para Oracle fica disponível em até 30 minutos.
SP-O2006	Acessibilidade do bucket do S3	RDS Custom automation can't download files from the S3 bucket on EC2 instance [*ec2_id*]. Check your networking configuration and make sure the instance allows connections to and from S3.	
SP-20	Versão da solução de software de alta disponibilidade	The HA solution of your instance differs from the expected version. To resolve this issue, create an AWS Support case.	Crie um caso no AWS Support:

Banco de dados

ID do evento	Configuração	Mensagem de evento do RDS	Ação
SP-O3001	Destino de atraso de arquivo	The ARCHIVE_LAG_TARGET parameter on	O perímetro de suporte monitora o parâmetro de banco de dados `ARCHIVE_LAG_TARGET` para

ID do evento	Configuração	Mensagem de evento do RDS	Ação
	do banco de dados	EC2 instance [*ec2_id*] is out of the recommended range *value_range* . To resolve the issue, set the parameter to a value within value_range.	verificar se a hora da última restauração da instância de banco de dados está dentro de limites razoáveis. Para alterar a meta de atraso para logs redo arquivados 1. Faça login no host do EC2. 2. Conectar-se à instância de banco de dados do RDS Custom para Oracle 3. Altere o parâmetro `ARCHIVE_LAG_TARGET` para um valor de 60 a 7.200. Por exemplo, use a instrução SQL a seguir. `ALTER SYSTEM SET ARCHIVE_LAG_TARGET=300 SCOPE=BOTH;` A instância de banco de dados fica disponível em 30 minutos.

ID do evento	Configuração	Mensagem de evento do RDS	Ação

SP-O3002	Função do Oracle Data Guard	The database role [*role_name*] isn't supported for Oracle Data Guard on EC2 instance [*ec2_id*]. To resolve the issue, set the DATABASE_ROLE parameter to either PRIMARY or PHYSICAL STANDBY.	O perímetro de suporte monitora a função de banco de dados atual a cada 15 segundos e envia uma notificação do CloudWatch quando ela tiver sido alterada. O parâmetro `DATABASE_ROLE` do Oracle Data Guard deve ser `PRIMARY` ou `PHYSICAL STANDBY`.
			
			Para restaurar o perfil de banco de dados do Oracle Data Guard para um valor compatível
			
			1. Verifique o perfil do Oracle Data Guard executando a seguinte instrução:
			
			`SELECT DATABASE_ROLE FROM V$DATABASE;`
			
			2. Se a instância de banco de dados for autônoma, use qualquer uma das instruções a seguir para alterá-la de volta para o perfil `PRIMARY`:
			
			`ALTER DATABASE COMMIT TO SWITCHOVER PRIMARY;`
			`ALTER DATABASE ACTIVATE STANDBY DATABASE;`
			
			Se a instância de banco de dados for uma réplica, use a seguinte instrução para alterá-la de volta para o perfil `PHYSICAL STANDBY`:
			
			`ALTER DATABASE CONVERT TO PHYSICAL STANDBY;`
			
			Depois que o perímetro de suporte determinar que a função de banco de dados é compatível, a instância de banco de dados do RDS Custom para Oracle ficará disponível em 15 segundos.

ID do evento	Configuração	Mensagem de evento do RDS	Ação
SP-O3003	Integridade do banco de dados	The SMON process of the Oracle database is in a zombie state. To resolve the issue, manually recover the database on EC2 instance [*ec2_id*], open the database, and then immediately back it up. For more help, contact Suporte.	O perímetro de suporte monitora o estado da instância de banco de dados. Ele também monitora quantas reinicializações ocorreram durante a hora anterior e ao longo do dia. Você é notificado quando a instância está em um estado em que ela ainda existe, mas não é possível interagir com ela. Para fazer com que o perímetro de suporte avalie o estado da instância 1. Faça login no host e determine o estado do banco de dados. `ps -eo pid,state,command | grep smon` 2. Se necessário, reinicie a instância de banco de dados. Se a reinicialização falhar, passe para a próxima etapa. 3. Se necessário, reinicie o host do EC2. Após a reinicialização da instância de banco de dados, o agente do RDS Custom detecta que a instância de banco de dados não está mais em um estado sem resposta. Em seguida, ele notifica o perímetro de suporte para reavaliar o estado dessa instância.

ID do evento	Configuração	Mensagem de evento do RDS	Ação

SP-O3004	Modo de log de banco de dados	The database log mode on EC2 instance [*ec2_id*] was changed to [*value_b*]. To resolve the issue, set the log mode to [*value_a*].	Para alterar o modo de log da instância de banco de dados para **`ARCHIVELOG`**
			1. Faça login no host do EC2.
			2. Conecte-se ao banco de dados e execute a seguinte instrução:
			`SELECT LOG_MODE FROM V$DATABASE;`
			Ou é possível executar o seguinte comando no SQL*Plus:
			`ARCHIVE LOG LIST`
			3. Execute o comando do SQL*Plus a seguir para iniciar um desligamento consistente.
			`SHUTDOWN IMMEDIATE`
			O agente do RDS Custom reinicia automaticamente a instância de banco de dados e define o modo de log como `ARCHIVELOG` . A instância de banco de dados fica disponível em 30 minutos.

SP-O3005	Caminho inicial do Oracle	The Oracle home on EC2 instance [*ec2_id*] was changed to *new_path*. To resolve the issue, revert the setting to *old_path*.	

ID do evento	Configuração	Mensagem de evento do RDS	Ação

SP-O3006	Nome exclusivo do banco de dados	The database unique name on EC2 instance [*ec2_id*] was changed to *new_value* . To resolve the issue, revert the name to *old_value* .	Para alterar o nome exclusivo do banco de dados da instância de banco de dados
			1. Faça login no host do EC2.
			2. Conecte-se ao banco de dados e execute a seguinte instrução:
			`SELECT DB_UNIQUE_NAME FROM V$DATABASE;`
			3. Especifique o nome exclusivo do banco de dados original usando o comando `ALTER SYSTEM SET DB_UNIQUE_NAME` .
			4. Execute a instrução SQL a seguir para iniciar um desligamento consistente.
			`SHUTDOWN IMMEDIATE;`
			O agente do RDS Custom reinicia automaticamente a instância de banco de dados e define o modo de log como `ARCHIVELOG` . A instância de banco de dados fica disponível em 30 minutos.

Solucionar problemas de upgrade do RDS Custom para Oracle

O upgrade de uma instância do RDS Custom para Oracle pode falhar. Veja a seguir técnicas que podem ser utilizadas durante upgrades de instâncias de banco de dados do RDS Custom para Oracle:

- Examine os arquivos de log de saída de atualização no diretório `/tmp` na instância de banco de dados. Os nomes dos logs dependem da versão do mecanismo de banco de dados. Por exemplo, você pode ver logs que contêm as strings `catupgrd` ou `catup`.
- Examine o arquivo `alert.log` localizado no diretório `/rdsdbdata/log/trace`.

- Execute o seguinte comando da `grep` no diretório `root` para acompanhar o processo de upgrade do SO. Esse comando mostra onde os arquivos de log estão sendo gravados e determina o estado do processo de upgrade.

```
ps -aux | grep upg
```

Veja a seguir um exemplo de saída.

```
root     18884  0.0  0.0 235428  8172 ?        S<   17:03   0:00 /usr/bin/
sudo -u rdsdb /rdsdbbin/scripts/oracle-control ORCL op_apply_upgrade_sh RDS-
UPGRADE/2.upgrade.sh
rdsdb    18886  0.0  0.0 153968 12164 ?        S<   17:03   0:00 /usr/bin/perl -T -
w /rdsdbbin/scripts/oracle-control ORCL op_apply_upgrade_sh RDS-UPGRADE/2.upgrade.sh
rdsdb    18887  0.0  0.0 113196  3032 ?        S<   17:03   0:00 /bin/sh /rdsdbbin/
oracle/rdbms/admin/RDS-UPGRADE/2.upgrade.sh
rdsdb    18900  0.0  0.0 113196  1812 ?        S<   17:03   0:00 /bin/sh /rdsdbbin/
oracle/rdbms/admin/RDS-UPGRADE/2.upgrade.sh
rdsdb    18901  0.1  0.0 167652 20620 ?        S<   17:03   0:07 /rdsdbbin/oracle/
perl/bin/perl catctl.pl -n 4 -d /rdsdbbin/oracle/rdbms/admin -l /tmp catupgrd.sql
root     29944  0.0  0.0 112724  2316 pts/0    S+   18:43   0:00 grep --color=auto
 upg
```

- Execute a seguinte consulta SQL para verificar o estado atual dos componentes e localizar a versão do banco de dados e as opções instaladas na instância de banco de dados.

```
SET LINESIZE 180
COLUMN COMP_ID FORMAT A15
COLUMN COMP_NAME FORMAT A40 TRUNC
COLUMN STATUS FORMAT A15 TRUNC
SELECT COMP_ID, COMP_NAME, VERSION, STATUS FROM DBA_REGISTRY ORDER BY 1;
```

A saída será semelhante à seguinte.

```
COMP_NAME                                STATUS               PROCEDURE
---------------------------------------- --------------------
 ------------------------------------------------
Oracle Database Catalog Views            VALID
 DBMS_REGISTRY_SYS.VALIDATE_CATALOG
Oracle Database Packages and Types       VALID
 DBMS_REGISTRY_SYS.VALIDATE_CATPROC
Oracle Text                              VALID                VALIDATE_CONTEXT
```

```
Oracle XML Database                    VALID                DBMS_REGXDB.VALIDATEXDB

4 rows selected.
```

- Execute a seguinte consulta SQL para verificar se há objetos inválidos que possam interferir no processo de upgrade.

```
SET PAGES 1000 LINES 2000
COL OBJECT FOR A40
SELECT SUBSTR(OWNER,1,12) OWNER,
       SUBSTR(OBJECT_NAME,1,30) OBJECT,
       SUBSTR(OBJECT_TYPE,1,30) TYPE, STATUS,
       CREATED
FROM   DBA_OBJECTS
WHERE  STATUS <>'VALID'
AND    OWNER IN ('SYS','SYSTEM','RDSADMIN','XDB');
```

Solucionar problemas de promoção de réplicas no RDS Custom para Oracle

Você pode promover réplicas gerenciadas do Oracle no RDS Custom para Oracle usando o console, o comando `promote-read-replica` do AWS CLI ou a API do `PromoteReadReplica`. Se você excluir sua instância de banco de dados primária e todas as réplicas estiverem íntegras, o RDS Custom para Oracle promoverá automaticamente suas réplicas gerenciadas a instâncias autônomas. Se uma réplica tiver pausado a automação ou estiver fora do perímetro de suporte, corrija a réplica para que o RDS Custom possa promovê-la automaticamente. Para obter mais informações, consulte Promover uma réplica do RDS Custom para Oracle a uma instância de banco de dados autônoma.

O fluxo de trabalho de promoção de réplicas pode ficar travado na seguinte situação:

- A instância de banco de dados primária está no estado `STORAGE_FULL`.
- O banco de dados principal não pode arquivar todos os logs de redo on-line.
- Existe uma lacuna entre os arquivos de log de redo arquivados na réplica do Oracle e no banco de dados primário.

Como responder ao fluxo de trabalho travado

1. Sincronize a lacuna de logs de redo na réplica da instância de banco de dados Oracle.

2. Force a promoção de sua réplica de leitura para o log de redo mais recente aplicado. Execute os seguintes comandos no SQL*Plus:

```
ALTER DATABASE ACTIVATE STANDBY DATABASE;
SHUTDOWN IMMEDIATE
STARTUP
```

3. Entre em contato com o Suporte e peça que a instância de banco de dados passe para o status `available`.

Problemas conhecidos do Amazon RDS Custom para Oracle

Ao trabalhar com o RDS Custom para Oracle, observe os problemas das instâncias de banco de dados a seguir:

- Não há suporte para redimensionar os volumes raiz ou dbbin.

  > **Warning**
  >
  > Recomendamos fortemente que você não redimensione os volumes raiz ou dbbin manualmente. Recomendamos que você armazene todas as configurações no volume de dados, que persiste após a aplicação de patches, e redimensione o volume usando somente a API de armazenamento em escala do RDS.

- Algumas APIs do RDS podem ser bloqueadas quando uma instância de banco de dados está em uma AMI mais antiga, por exemplo, uma AMI que usa o Oracle Linux 7. Para resolver esse problema, corrija a instância de banco de dados para a AMI mais recente usando patches do sistema operacional. Para ter mais informações, consulte Opções de atualização de CEV.
- Antes de realizar operações do RDS, verifique se sua Conta da AWS tem cota suficiente para computação e armazenamento.
- Se o banco de dados estiver no estado de criação e você fizer login ativamente no banco de dados ou no host do Amazon EC2 e executar comandos, a criação do banco de dados poderá não ser concluída.
- Atualmente, a multiplexação de arquivos de controle não é aceita devido a um problema de réplica de leitura. Antes de criar uma réplica de leitura, especifique somente um nome de arquivo no parâmetro de inicialização `CONTROL_FILES` no banco de dados de origem.
- Você não pode alterar o modo do banco de dados de `PHYSICAL STANDBY` (montado ou somente leitura) para `SNAPSHOT STANDBY` (convertendo para leitura/gravação).
- Se uma Conta da AWS fizer parte de uma organização da AWS com uma política de controle de serviços (SCP), e a SCP contiver uma chave de condição, uma instância de banco de dados do RDS Custom para Oracle poderá falhar na criação com o seguinte erro:

```
You can't create the DB instance because of incompatible resources.
The IAM instance profile role [AWSRDSCustomInstanceRole1-us-east-1] is missing the
 following permissions:
EFFECT [Allow] on ACTION(S) [ssm:DescribeAssociation, ssm:DescribeDocument,
 ssm:GetConnectionStatus,
```

```
 ssm:GetDeployablePatchSnapshotForInstance, ssmmessages:OpenControlChannel,
ssm:GetParameters,
 ssm:ListInstanceAssociations, ssm:PutConfigurePackageResult,
ssmmessages:CreateControlChannel,
 ssm:GetParameter, ssm:UpdateAssociationStatus, ssm:GetManifest,
ssmmessages:CreateDataChannel,
 ssm:PutInventory, ssm:UpdateInstanceInformation, ssm:DescribeInstanceInformation,
 ssmmessages:OpenDataChannel, ssm:GetDocument, ssm:ListAssociations,
ssm:PutComplianceItems,
 ssm:UpdateInstanceAssociationStatus] for RESOURCE(S) [], EFFECT [Allow] on
ACTION(S) [ec2messages:DeleteMessage,
 ec2messages:FailMessage, ec2messages:GetEndpoint, ec2messages:AcknowledgeMessage,
ec2messages:GetMessages,
 ec2messages:SendReply] for RESOURCE(S) [], EFFECT [Allow] on ACTION(S)
[logs:CreateLogStream,
 logs:DescribeLogStreams, logs:PutRetentionPolicy, logs:PutLogEvents]
```

Para resolver esse problema, crie um ticket com o Suporte.

Problemas conhecidos com contas de usuário de banco de dados

Observe os seguintes problemas:

- Não remova contas de usuário do banco de dados que comecem com a string `RDS`, como `RDSADMIN` e `RDS_DATAGUARD`. O RDS Custom para Oracle usa a conta do `RDS` para automação. Se você remover essa conta de usuário, o RDS Custom moverá a instância para o estado de configuração sem suporte.
- Não é possível alterar o nome de usuário principal da instância de banco de dados do RDS Custom para Oracle usando a API `ModifyDBInstance`.
- O RDS Custom para Oracle alterna as credenciais da conta do usuário em todas as instâncias de banco de dados. Para ter mais informações, consulte Alternar as credenciais do RDS Custom para Oracle para programas de conformidade. Se você usa uma configuração primária/em espera on-premises, a rotação de credenciais pode afetar os seguintes recursos:
  - Instâncias do RDS Custom para Oracle em espera criadas manualmente

    Para resolver esse problema, elimine os bancos de dados em espera manuais e crie uma réplica de leitura do Oracle usando uma chamada de API. Gerencie manualmente os segredos dos bancos de dados em espera manuais para que eles correspondam à instância de banco de dados de origem.

- Réplicas de leitura entre regiões criadas manualmente

  Para resolver esse problema, mantenha manualmente os segredos para que eles correspondam à instância de banco de dados primária.

Problemas conhecidos com arquivos de parâmetros e configuração

- É necessário configurar o arquivo `crontab` após ajustes de escala de computação, atualizações do sistema operacional e outras operações em que o RDS Custom substitui o volume-raiz. É altamente recomendável que você mantenha um backup do `crontab`.
- Observe as seguintes diretrizes ao configurar o arquivo `listener.ora`:
  - Certifique-se de que cada entrada no arquivo esteja em uma única linha. Essa abordagem evita problemas de recuo durante a criação da instância.
  - Certifique-se de que `GLOBAL_DBNAME` seja igual ao valor de `SID_NAME`.
  - Certifique-se de que o valor de `LISTENER` siga a convenção de nomenclatura `L_`*`dbname`*`_001`.
  - Certifique-se de que o arquivo `listener.ora` mantenha uma conexão com o nome do banco de dados. O RDS Custom usa essa conexão para verificar a inicialização do banco de dados. Se você modificar esse arquivo incorretamente, operações como computação em escala ou aplicação de patches poderão falhar.

    O exemplo a seguir mostra um `listener.ora` configurado corretamente:

    ```
    ADR_BASE_L_ORCL_001=/rdsdbdata/log/
    USE_SID_AS_SERVICE_L_ORCL_001=ON
    SID_LIST_L_ORCL_001=(SID_LIST = (SID_DESC = (SID_NAME = ORCL)(GLOBAL_DBNAME = ORCL)
     (ORACLE_HOME = /rdsdbbin/oracle.19.custom.r1.EE.1)))
    SUBSCRIBE_FOR_NODE_DOWN_EVENT_L_ORCL_001=OFF
    L_ORCL_001=(DESCRIPTION_LIST = (DESCRIPTION = (ADDRESS = (PROTOCOL = TCP)(PORT =
     XXXX)(HOST = x.x.x.x))) (DESCRIPTION = (ADDRESS = (PROTOCOL = TCP)(PORT = XXXX)
    (HOST = 127.0.0.1))))
    ```

- Comentários não são aceitos em um arquivo de parâmetros do servidor ou um arquivo de parâmetros de inicialização.
- É necessário declarar os seguintes parâmetros de inicialização no arquivo de parâmetro do servidor (`/rdsdbdata/config/oracle_pfile`):
  - `MEMORY_MAX_TARGET`
  - `MEMORY_TARGET`

- PGA_AGGREGATE_TARGET
- PROCESSES
- SGA_TARGET
- USE_LARGE_PAGES

Se os parâmetros anteriores não forem declarados em `/rdsdbdata/config/oracle_pfile`, a criação da réplica de leitura e o ajuste de escala de computação poderão falhar.

- Você não pode excluir os links simbólicos para arquivos de configuração, como o arquivo de parâmetros do servidor, arquivos de auditoria, `listener.ora`, `tnsnames.ora` ou `sqlnet.ora`. Também não é possível modificar a estrutura de diretórios desses arquivos. A automação do RDS Custom espera que esses arquivos existam em uma estrutura de diretórios específica.

  Para criar um arquivo de parâmetros do servidor por meio de um arquivo de parâmetros de inicialização, use a sintaxe a seguir.

```
CREATE SPFILE='/rdsdbdata/admin/$ORACLE_SID/pfile/spfile$ORACLE_SID.ora'
    FROM PFILE='/rdsdbdata/config/oracle_pfile';
```

Trabalhar com o RDS Custom for SQL Server

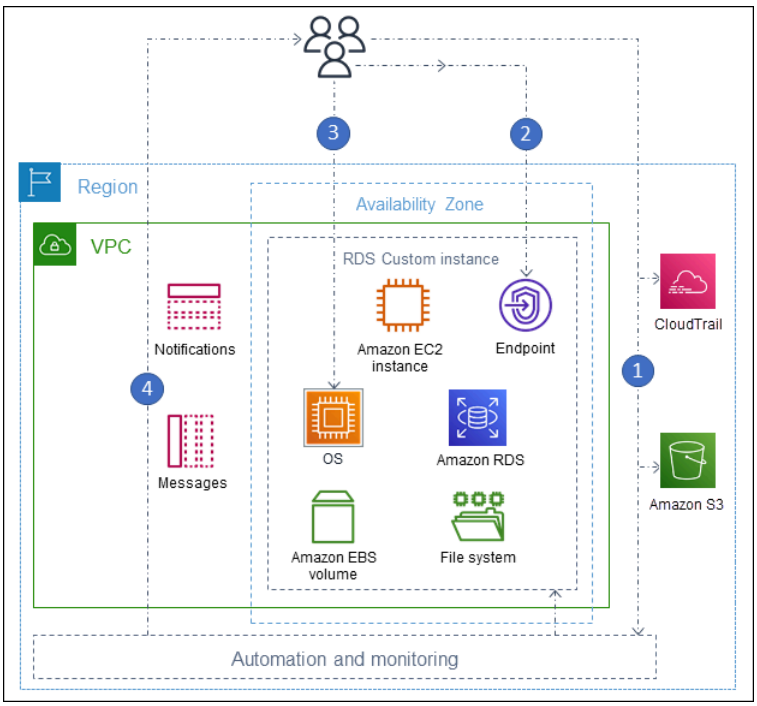
Veja a seguir instruções para criar, gerenciar e manter as instâncias de banco de dados do RDS Custom for SQL Server.

Tópicos

- Fluxo de trabalho do RDS Custom for SQL Server
- Requisitos e limitações do Amazon RDS Custom for SQL Server
- Configurar seu ambiente para o Amazon RDS Custom for SQL Server
- Modelo BYOM (traga sua própria mídia) com o RDS Custom para SQL Server
- Trabalhar com versões de mecanismo personalizadas para o RDS Custom para SQL Server
- Criar e conectar-se a uma instância de banco de dados Amazon RDS Custom for SQL Server
- Gerenciar uma instância de banco de dados do Amazon RDS Custom for SQL Server
- Trabalhar com o Microsoft Active Directory e o RDS Custom para SQL Server
- Gerenciar uma implantação multi-AZ para o RDS Custom para SQL Server
- Fazer backup e restauração de uma instância de banco de dados do Amazon RDS Custom
- Copiar um snapshot de banco de dados do Amazon RDS Custom para SQL Server
- Migrar um banco de dados on-premises para o Amazon RDS Custom for SQL Server
- Fazer atualização de uma instância de banco de dados do Amazon RDS Custom para SQL Server
- Solucionar problemas de banco de dados do Amazon RDS Custom para SQL Server

Fluxo de trabalho do RDS Custom for SQL Server

O diagrama a seguir mostra o fluxo de trabalho típico do RDS Custom for SQL Server.



As etapas são:

1. Crie uma instância de banco de dados do RDS Custom for SQL Server usando uma versão de mecanismo oferecida pelo RDS Custom.

   Para obter mais informações, consulte Criar uma instância de banco de dados do RDS Custom for SQL Server.
2. Conectar sua aplicação ao endpoint da instância de banco de dados do RDS Custom.

   Para ter mais informações, consulte Conectar-se à sua instância de banco de dados do RDS Custom via AWS Systems Manager e Conectar-se à sua instância de banco de dados do RDS Custom via RDP.
3. (Opcional) Acessar o host para personalizar o software.
4. Monitore notificações e mensagens geradas pela automação do RDS Custom.

Criar uma instância de banco de dados do RDS Custom for Oracle

Você cria sua instância de banco de dados do RDS Custom utilizando o comando `create-db-instance`. O procedimento é semelhante ao de criação de uma instância do Amazon RDS. No entanto, alguns dos parâmetros são diferentes. Para obter mais informações, consulte Criar e conectar-se a uma instância de banco de dados Amazon RDS Custom for SQL Server.

Conexão de banco de dados

Como uma instância de banco de dados do Amazon RDS, sua instância de banco de dados do RDS Custom for SQL Server reside em uma VPC. Sua aplicação se conecta à instância personalizada do RDS utilizando um cliente como o SQL Server Management Suite (SSMS), assim como no RDS for SQL Server.

Personalização do RDS Custom

É possível acessar o host do RDS Custom para instalar ou personalizar softwares. Para evitar conflitos entre suas alterações e a automação do RDS Custom, é possível pausar a automação por um período especificado. Durante esse período, o RDS Custom não realiza o monitoramento ou a recuperação de instâncias. No final do período, o RDS Custom retomará a automação completa. Para obter mais informações, consulte Pausar e retomar a automação do RDS Custom.

Requisitos e limitações do Amazon RDS Custom for SQL Server

Veja a seguir um resumo dos requisitos e das limitações do Amazon RDS Custom for SQL Server para referência rápida. Requisitos e limitações também aparecem nas seções relevantes.

Tópicos

- Disponibilidade de região e versão
- Requisitos gerais do RDS Custom for SQL Server
- Compatibilidade com classes de instâncias de banco de dados do RDS Custom for SQL Server
- Limitações do RDS Custom for SQL Server
- Definir conjuntos de caracteres e agrupamentos para instâncias de banco de dados do RDS Custom para SQL Server
- Fuso horário local para instâncias de banco de dados do RDS Custom para SQL Server
- Usar uma chave mestra de serviço com RDS Custom para SQL Server
- Suporte à captura de dados de alteração (CDC) com o RDS Custom para SQL Server

Disponibilidade de região e versão

A disponibilidade e a compatibilidade de recursos variam entre versões específicas de cada mecanismo de banco de dados e entre Regiões da AWS. Para ter mais informações sobre a disponibilidade de versões e regiões do Amazon RDS com o Amazon RDS Custom para SQL Server, consulte Regiões e mecanismos de banco de dados compatíveis com o RDS Custom para SQL Server.

Requisitos gerais do RDS Custom for SQL Server

Certifique-se de seguir estes requisitos para o Amazon RDS Custom for SQL Server:

- Use as classes de instância mostradas em Compatibilidade com classes de instâncias de banco de dados do RDS Custom for SQL Server. Os únicos tipos de armazenamento compatíveis são unidades de estado sólido (SSD) dos tipos gp2, gp3, io1 e io2 Block Express. O limite máximo de armazenamento para io1, gp2 e gp3 é de 16 TiB, enquanto o de io2 é de 64 TiB.
- Verifique se você tem uma chave simétrica do AWS KMS para criar uma instância de banco de dados do RDS Custom. Para ter mais informações, consulte Verifique se você tem uma chave do AWS KMS de criptografia simétrica..

- Certifique-se de criar uma função do AWS Identity and Access Management (IAM) e um perfil de instância. Para obter mais informações, consulte Criar sua função do IAM e seu perfil de instância manualmente e Criação automática de perfis de instância usando o AWS Management Console.
- Forneça uma configuração de rede que o RDS Custom pode utilizar para acessar outros Serviços da AWS. Para conhecer os requisitos específicos, consulte Etapa 2: configurar rede, perfil de instância e criptografia.
- O número combinado de instâncias de banco de dados do RDS Custom e do Amazon RDS não pode exceder seu limite de cota. Por exemplo, se sua cota for de 40 instâncias de banco de dados, você poderá ter 20 instâncias de banco de dados do RDS Custom for SQL Server e 20 instâncias de banco de dados do Amazon RDS.
- O RDS Custom cria automaticamente uma trilha AWS CloudTrail cujo nome começa com `do-not-delete-rds-custom-`. O perímetro de suporte do RDS Custom depende dos eventos do CloudTrail para determinar se suas ações afetam a automação do RDS Custom. O RDS Custom cria a trilha quando você cria sua primeira instância de banco de dados. Para usar um CloudTrail existente, entre em contato com o AWS Support. Para obter mais informações, consulte AWS CloudTrail.

Compatibilidade com classes de instâncias de banco de dados do RDS Custom for SQL Server

Verifique se a classe de instância de bancos de dados é compatível com a região usando o comando describe-orderable-db-instance-options.

O RDS Custom para SQL Server é compatível com as classes de instância de banco de dados mostradas na seguinte tabela:

Edição do SQL Server	Suporte pelo RDS Custom
Edição Enterprise	db.r5.xlarge–db.r5.24xlarge db.r5b.xlarge–db.r5b.24xlarge db.m5.xlarge–db.m5.24xlarge db.r6i.xlarge–db.r6i.32xlarge

Edição do SQL Server	Suporte pelo RDS Custom
	db.m6i.xlarge–db.m6i.32xlarge db.x2iedn.xlarge–db.x2iedn.32xlarge
Edição Standard	db.r5.large–db.r5.24xlarge db.r5b.large–db.r5b.8xlarge db.m5.large – db.m5.24xlarge db.r6i.large–db.r6i.8xlarge db.m6i.large–db.m6i.8xlarge db.x2iedn.xlarge–db.x2iedn.8xlarge
Developer Edition	db.r5.xlarge–db.r5.24xlarge db.r5b.xlarge–db.r5b.24xlarge db.m5.xlarge–db.m5.24xlarge db.r6i.xlarge–db.r6i.32xlarge db.m6i.xlarge–db.m6i.32xlarge db.x2iedn.xlarge–db.x2iedn.32xlarge
Edição Web	db.r5.large–db.r5.4xlarge db.m5.large – db.m5.4xlarge db.r6i.large–db.r6i.4xlarge (somente 19c) db.m6i.large–db.m6i.4xlarge db.r5b.large–db.r5b.4xlarge

As recomendações a seguir se aplicam aos tipos de classe db.x2iedn:

- Na criação, o armazenamento local é um dispositivo bruto e não alocado. Antes de usar uma instância de banco de dados com essa classe de instância, é necessário montar e formatar o armazenamento local. Depois, configure `tempdb` nele para garantir o desempenho ideal. Consulte mais informações em Optimize tempdb performance in Amazon RDS Custom for SQL Server using local instance storage.
- O armazenamento local volta ao estado bruto e não alocado quando você executa operações de instância de banco de dados, como computação em escala, substituição de instâncias, restauração de snapshots ou recuperação para um ponto no tempo (PITR). Nessas situações, é necessário remontar, reformatar e reconfigurar a unidade e `tempdb` para restaurar a funcionalidade.
- Para instâncias multi-AZ, recomendamos que você execute a configuração em uma instância de banco de dados em espera. Dessa forma, se ocorrer um failover, o sistema continuará operando sem problemas porque a configuração já está em vigor na instância em espera.

Limitações do RDS Custom for SQL Server

As limitações a seguir se aplicam ao RDS Custom for SQL Server:

- Não é possível criar réplicas de leitura no Amazon RDS para RDS Custom para instâncias de banco de dados do RDS Custom for SQL Server. No entanto, é possível configurar a alta disponibilidade automaticamente com uma implantação multi-AZ. Para obter mais informações, consulte Gerenciar uma implantação multi-AZ para o RDS Custom para SQL Server.
- Não é possível modificar o identificador da instância de banco de dados existente do RDS Custom para SQL Server.
- Para uma instância de banco de dados do RDS Custom para SQL Server que não foi criada com uma versão de mecanismo personalizada (CEV), não é garantido que as alterações no sistema operacional Microsoft Windows sejam mantidas. Por exemplo, você perderá essas alterações ao iniciar um snapshot ou uma operação de restauração para um ponto no tempo. Se a instância de banco de dados do RDS Custom para SQL Server foi criada com uma CEV, essas alterações serão mantidas.
- Nem todas as opções são compatíveis. Por exemplo, quando você cria uma instância de banco de dados do RDS Custom for SQL Server, não pode fazer o seguinte:
  - Modifique o número de núcleos de CPU e de threads por núcleo na classe da instância de banco de dados.

  - Habilite a escalabilidade automática de armazenamento.
  - Especifique seu próprio grupo de parâmetros de banco de dados, grupo de opções ou conjunto de caracteres.
  - Habilite o Performance Insights.
  - Habilite o upgrade automático de versões secundárias.
- O armazenamento máximo da instância de banco de dados é de 64 TiB.
- Não é possível usar o RDS Proxy com o RDS Custom para SQL Server.
- Não é possível usar a API `describe-reserved-db-instances` para instâncias de banco de dados do RDS Custom para SQL Server.

Definir conjuntos de caracteres e agrupamentos para instâncias de banco de dados do RDS Custom para SQL Server

Visão geral

Com as instâncias de banco de dados do RDS Custom para SQL Server, é possível definir as configurações de conjuntos de caracteres e agrupamentos que determinam como os dados são armazenados e classificados. Os conjuntos de caracteres definem quais caracteres são permitidos, enquanto os agrupamentos especificam as regras para classificar e comparar dados. É importante definir conjuntos de caracteres e agrupamentos apropriados para aplicações que funcionam com dados multilíngues ou têm requisitos específicos de classificação. Por exemplo, talvez você precise lidar com caracteres acentuados e definir regras de classificação específicas do idioma ou manter a integridade dos dados em diferentes localidades. As seções a seguir fornecem informações sobre conjuntos de caracteres e agrupamentos para as instâncias de banco de dados do RDS Custom para SQL Server.

O RDS Custom para SQL Server comporta uma ampla variedade de agrupamentos de servidores, tanto na codificação tradicional quanto na codificação UTF-8, para as localidades SQL_Latin, japonês, alemão e árabe. O agrupamento de servidores padrão é o `SQL_Latin1_General_CP1_CI_AS`, mas não é possível selecionar outro agrupamento compatível para usar. Você pode selecionar um agrupamento usando o mesmo procedimento usado pelo RDS para SQL Server. Para ter mais informações, consulte Gerenciar agrupamentos e conjuntos de caracteres para o Amazon RDS para Microsoft SQL Server.

Considerações

Os seguintes requisitos e limitações aplicam-se ao trabalhar com agrupamentos de servidores no RDS Custom para SQL Server:

- Você pode definir o agrupamento de servidores ao criar uma instância de banco de dados do RDS Custom para SQL Server. Não é possível modificar o agrupamento em nível de servidor após a criação da instância de banco de dados.
- Também não é possível modificar o agrupamento em nível de servidor ao restaurar por meio de um snapshot de banco de dados ou durante uma recuperação para um ponto no tempo (PITR).
- Quando você cria uma instância de banco de dados usando uma CEV do RDS Custom para SQL Server, a instância de banco de dados não herda o agrupamento de servidores da CEV. Em vez disso, o agrupamento de servidores padrão `SQL_Latin1_General_CP1_CI_AS` é usado. Se você tiver configurado um agrupamento de servidores não padrão em uma CEV do RDS Custom para SQL Server e quiser usar esse mesmo agrupamento de servidores em uma nova instância de banco de dados, selecione esse mesmo agrupamento ao criar a instância de banco de dados usando a CEV.

  > **Note**
  >
  > Se o agrupamento selecionado ao criar a instância de banco de dados for diferente do agrupamento da CEV, os bancos de dados do sistema Microsoft SQL Server na nova instância de banco de dados do RDS Custom para SQL Server serão recriados para usar o agrupamento atualizado. O processo de reconstrução é executado somente na instância de banco de dados do RDS Custom para SQL Server e não tem nenhum impacto na CEV em si. Quaisquer modificações anteriores feitas nos bancos de dados do sistema na CEV não serão mantidas na nova instância de banco de dados do RDS Custom para SQL Server depois que os bancos de dados do sistema forem reconstruídos. Exemplos de algumas modificações incluem objetos definidos pelo usuário no banco de dados `master`, trabalhos agendados no banco de dados `msdb` ou alterações nas configurações padrão do banco de dados `model` na CEV. Você pode recriar manualmente as modificações quando a instância de banco de dados do RDS Custom para SQL Server for criada.

- Ao criar uma instância de banco de dados usando uma versão de mecanismo personalizada (CEV) do RDS Custom para SQL Server e selecionar um agrupamento diferente daquele da CEV, sua imagem dourada (AMI) usada para a criação da CEV deve atender aos seguintes requisitos

para que os bancos de dados do sistema Microsoft SQL Server na nova instância de banco de dados possam ser reconstruídos:

- Para o SQL Server 2022, verifique se o arquivo `setup.exe` está localizado no seguinte caminho: `C:\Program Files\Microsoft SQL Server\160\Setup Bootstrap \SQL2022\setup.exe`.
- Para o SQL Server 2019, verifique se o arquivo `setup.exe` está localizado no seguinte caminho: `C:\Program Files\Microsoft SQL Server\150\Setup Bootstrap \SQL2019\setup.exe`.
- Cópias dos dados e modelos de log dos bancos de dados `master`, `model` e `msdb` devem existir nos locais padrão. Para obter mais informações, consulte Reconstruir bancos de dados do sistema.
- O mecanismo de banco de dados do SQL Server deve usar `NT Service\MSSQLSERVER` ou `NT AUTHORITY\NETWORK SERVICE` como a conta de serviço. Qualquer outra conta não terá as permissões necessárias na unidade `C:\` ao configurar um agrupamento de servidores não padrão para a instância de banco de dados.

- Se o agrupamento de servidores selecionado para uma nova instância de banco de dados for o mesmo configurado na CEV, os bancos de dados do sistema Microsoft SQL Server na nova instância de banco de dados do RDS Custom para SQL Server não passarão pelo processo de reconstrução. Todas as modificações anteriores feitas nos bancos de dados do sistema na CEV persistirão automaticamente na nova instância de banco de dados do RDS Custom para SQL Server.

Agrupamentos compatíveis

Você pode definir o agrupamento como um dos valores listados na tabela a seguir.

Collation (Agrupamento)	Descrição
Arabic_100_BIN	Arabic-100, classificação binária
Arabic_100_BIN2	Arabic-100, classificação binária de comparação de ponto de código
Arabic_100_CI_AI	Arabic-100, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, não

Collation (Agrupamento)	Descrição
	diferencia o tipo de kana, não diferencia a largura
Arabic_100_CI_AI_KS	Arabic-100, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, não diferencia a largura
Arabic_100_CI_AI_KS_SC	Arabic-100, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferenci a o tipo de kana, não diferencia a largura, caracteres complementares
Arabic_100_CI_AI_KS_SC_UTF8	Arabic-100, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferenci a o tipo de kana, não diferencia a largura, caracteres complementares, UTF8
Arabic_100_CI_AI_KS_WS	Arabic-100, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, diferencia a largura
Arabic_100_CI_AI_KS_WS_SC	Arabic-100, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, diferencia a largura, caracteres complementares
Arabic_100_CI_AI_KS_WS_SC_UTF8	Arabic-100, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, diferencia a largura, caracteres complementares, UTF8
Arabic_100_CI_AI_SC	Arabic-100, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, não diferencia a largura, caracteres complementares

Collation (Agrupamento)	Descrição
Arabic_100_CI_AI_SC_UTF8	Arabic-100, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, não diferencia a largura, caracteres complementares, UTF8
Arabic_100_CI_AI_WS	Arabic-100, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, diferencia a largura
Arabic_100_CI_AI_WS_SC	Arabic-100, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, diferencia a largura, caracteres complementares
Arabic_100_CI_AI_WS_SC_UTF8	Arabic-100, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, diferencia a largura, caracteres complementares, UTF8
Arabic_100_CI_AS	Arabic-100, não diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura
Arabic_100_CI_AS_KS	Arabic-100, não diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, não diferencia a largura
Arabic_100_CI_AS_KS_SC	Arabic-100, não diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, não diferencia a largura, caractere s complementares
Arabic_100_CI_AS_KS_SC_UTF8	Arabic-100, não diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, não diferencia a largura, caractere s complementares, UTF8

Collation (Agrupamento)	Descrição
Arabic_100_CI_AS_KS_WS	Arabic-100, não diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, diferencia a largura
Arabic_100_CI_AS_KS_WS_SC	Arabic-100, não diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, diferencia a largura, caracteres complementares
Arabic_100_CI_AS_KS_WS_SC_UTF8	Arabic-100, não diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, diferencia a largura, caracteres complementares, UTF8
Arabic_100_CI_AS_SC	Arabic-100, não diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferenci a o tipo de kana, não diferencia a largura, caracteres complementares
Arabic_100_CI_AS_SC_UTF8	Arabic-100, não diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferenci a o tipo de kana, não diferencia a largura, caracteres complementares, UTF8
Arabic_100_CI_AS_WS	Arabic-100, não diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, diferencia a largura
Arabic_100_CI_AS_WS_SC	Arabic-100, não diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, diferencia a largura, caracteres complementares

Collation (Agrupamento)	Descrição
Arabic_100_CI_AS_WS_SC_UTF8	Arabic-100, não diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, diferencia a largura, caracteres complementares, UTF8
Arabic_100_CS_AI	Arabic-100, diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, não diferencia a largura
Arabic_100_CS_AI_KS	Arabic-100, diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, não diferencia a largura
Arabic_100_CS_AI_KS_SC	Arabic-100, diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferenci a o tipo de kana, não diferencia a largura, caracteres complementares
Arabic_100_CS_AI_KS_SC_UTF8	Arabic-100, diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferenci a o tipo de kana, não diferencia a largura, caracteres complementares, UTF8
Arabic_100_CS_AI_KS_WS	Arabic-100, diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, diferencia a largura
Arabic_100_CS_AI_KS_WS_SC	Arabic-100, diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, diferencia a largura, caracteres complementares

Collation (Agrupamento)	Descrição
Arabic_100_CS_AI_KS_WS_SC_UTF8	Arabic-100, diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, diferencia a largura, caracteres complementares, UTF8
Arabic_100_CS_AI_SC	Arabic-100, diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, não diferencia a largura, caracteres complementares
Arabic_100_CS_AI_SC_UTF8	Arabic-100, diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, não diferencia a largura, caracteres complementares, UTF8
Arabic_100_CS_AI_WS	Arabic-100, diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, diferencia a largura
Arabic_100_CS_AI_WS_SC	Arabic-100, diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, diferencia a largura, caracteres complementares
Arabic_100_CS_AI_WS_SC_UTF8	Arabic-100, diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, diferencia a largura, caracteres complementares, UTF8
Arabic_100_CS_AS	Arabic-100, diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura
Arabic_100_CS_AS_KS	Arabic-100, diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, não diferencia a largura

Collation (Agrupamento)	Descrição
Arabic_100_CS_AS_KS_SC	Arabic-100, diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, não diferencia a largura, caractere s complementares
Arabic_100_CS_AS_KS_SC_UTF8	Arabic-100, diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, não diferencia a largura, caractere s complementares, UTF8
Arabic_100_CS_AS_KS_WS	Arabic-100, diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, diferencia a largura
Arabic_100_CS_AS_KS_WS_SC	Arabic-100, diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, diferencia a largura, caracteres complementares
Arabic_100_CS_AS_KS_WS_SC_UTF8	Arabic-100, diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, diferencia a largura, caracteres complementares, UTF8
Arabic_100_CS_AS_SC	Arabic-100, diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferenci a o tipo de kana, não diferencia a largura, caracteres complementares
Arabic_100_CS_AS_SC_UTF8	Arabic-100, diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferenci a o tipo de kana, não diferencia a largura, caracteres complementares, UTF8

Collation (Agrupamento)	Descrição
Arabic_100_CS_AS_WS	Arabic-100, diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, diferencia a largura
Arabic_100_CS_AS_WS_SC	Arabic-100, diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, diferencia a largura, caracteres complementares
Arabic_100_CS_AS_WS_SC_UTF8	Arabic-100, diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, diferencia a largura, caracteres complementares, UTF8
Arabic_BIN	Arabic, classificação binária
Arabic_BIN2	Arabic, classificação binária de comparação de ponto de código
Arabic_CI_AI	Arabic, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, não diferencia a largura
Arabic_CI_AI_KS	Arabic, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, não diferencia a largura
Arabic_CI_AI_KS_WS	Arabic, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, diferencia a largura
Arabic_CI_AI_WS	Arabic, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, diferencia a largura

Collation (Agrupamento)	Descrição
Arabic_CI_AS	Arabic, não diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura
Arabic_CI_AS_KS	Arabic, não diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, não diferencia a largura
Arabic_CI_AS_KS_WS	Arabic, não diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, diferencia a largura
Arabic_CI_AS_WS	Arabic, não diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, diferencia a largura
Arabic_CS_AI	Arabic, diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, não diferencia a largura
Arabic_CS_AI_KS	Arabic, diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, não diferencia a largura
Arabic_CS_AI_KS_WS	Arabic, diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, diferencia a largura
Arabic_CS_AI_WS	Arabic, diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, diferencia a largura
Arabic_CS_AS	Arabic, diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura

Collation (Agrupamento)	Descrição
Arabic_CS_AS_KS	Arabic, diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, não diferencia a largura
Arabic_CS_AS_KS_WS	Arabic, diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, diferencia a largura
Arabic_CS_AS_WS	Arabic, diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, diferencia a largura
Chinese_PRC_BIN2	Chinese-PRC, classificação binária de comparação por pontos de código
Chinese_PRC_CI_AS	Chinese-PRC, não diferencia letras maiúscula s de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura
Chinese_Taiwan_Stroke_CI_AS	Chinese-Taiwan-Stroke, não diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura
Danish_Norwegian_CI_AS	Danish-Norwegian, não diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura
Finnish_Swedish_CI_AS	Finnish-Swedish, não diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura

Collation (Agrupamento)	Descrição
French_CI_AS	French, não diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura
German_PhoneBook_100_BIN	German-PhoneBook-100, classificação binária
German_PhoneBook_100_BIN2	German-PhoneBook-100, classificação binária de comparação de ponto de código
German_PhoneBook_100_CI_AI	German-PhoneBook-100, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, não diferencia a largura
German_PhoneBook_100_CI_AI_KS	German-PhoneBook-100, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, não diferenci a a largura
German_PhoneBook_100_CI_AI_KS_SC	German-PhoneBook-100, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, não diferenci a a largura, caracteres complementares
German_PhoneBook_100_CI_AI_KS_SC_UTF8	German-PhoneBook-100, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, não diferenci a a largura, caracteres complementares, UTF8
German_PhoneBook_100_CI_AI_KS_WS	German-PhoneBook-100, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, diferencia a largura

Collation (Agrupamento)	Descrição
German_PhoneBook_100_CI_AI_KS_WS_SC	German-PhoneBook-100, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, diferencia a largura, caracteres complementares
German_PhoneBook_100_CI_AI_KS_WS_SC_UTF8	German-PhoneBook-100, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, diferencia a largura, caracteres complementares, UTF8
German_PhoneBook_100_CI_AI_SC	German-PhoneBook-100, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, não diferencia a largura, caracteres complemen tares
German_PhoneBook_100_CI_AI_SC_UTF8	German-PhoneBook-100, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, não diferencia a largura, caracteres complemen tares, UTF8
German_PhoneBook_100_CI_AI_WS	German-PhoneBook-100, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, diferenci a a largura
German_PhoneBook_100_CI_AI_WS_SC	German-PhoneBook-100, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, diferenci a a largura, caracteres complementares
German_PhoneBook_100_CI_AI_WS_SC_UTF 8	German-PhoneBook-100, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, diferenci a a largura, caracteres complementares, UTF8

Collation (Agrupamento)	Descrição
German_PhoneBook_100_CI_AS	German-PhoneBook-100, não diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura
German_PhoneBook_100_CI_AS_KS	German-PhoneBook-100, não diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, não diferencia a largura
German_PhoneBook_100_CI_AS_KS_SC	German-PhoneBook-100, não diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, não diferencia a largura, caracteres complementares
German_PhoneBook_100_CI_AS_KS_SC_UTF8	German-PhoneBook-100, não diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, não diferencia a largura, caracteres complementares, UTF8
German_PhoneBook_100_CI_AS_KS_WS	German-PhoneBook-100, não diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, diferencia a largura
German_PhoneBook_100_CI_AS_KS_WS_SC	German-PhoneBook-100, não diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, diferencia a largura, caracteres complementares
German_PhoneBook_100_CI_AS_KS_WS_SC_UTF8	German-PhoneBook-100, não diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, diferencia a largura, caracteres complementares, UTF8

Collation (Agrupamento)	Descrição
German_PhoneBook_100_CI_AS_SC	German-PhoneBook-100, não diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura, caracteres complementares
German_PhoneBook_100_CI_AS_SC_UTF8	German-PhoneBook-100, não diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura, caracteres complementares, UTF8
German_PhoneBook_100_CI_AS_WS	German-PhoneBook-100, não diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, diferencia a largura
German_PhoneBook_100_CI_AS_WS_SC	German-PhoneBook-100, não diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, diferencia a largura, caracteres complementares
German_PhoneBook_100_CI_AS_WS_SC_UTF8	German-PhoneBook-100, não diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, diferencia a largura, caracteres complementares, UTF8
German_PhoneBook_100_CS_AI	German-PhoneBook-100, diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, não diferencia a largura
German_PhoneBook_100_CS_AI_KS	German-PhoneBook-100, diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, não diferenci a a largura

Collation (Agrupamento)	Descrição
German_PhoneBook_100_CS_AI_KS_SC	German-PhoneBook-100, diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, não diferenci a a largura, caracteres complementares
German_PhoneBook_100_CS_AI_KS_SC_UTF 8	German-PhoneBook-100, diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, não diferenci a a largura, caracteres complementares, UTF8
German_PhoneBook_100_CS_AI_KS_WS	German-PhoneBook-100, diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, diferencia a largura
German_PhoneBook_100_CS_AI_KS_WS_SC	German-PhoneBook-100, diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, diferencia a largura, caracteres complementares
German_PhoneBook_100_CS_AI_KS_WS_SC_ UTF8	German-PhoneBook-100, diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, diferencia a largura, caracteres complementares, UTF8
German_PhoneBook_100_CS_AI_SC	German-PhoneBook-100, diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, não diferencia a largura, caracteres complemen tares
German_PhoneBook_100_CS_AI_SC_UTF8	German-PhoneBook-100, diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, não diferencia a largura, caracteres complemen tares, UTF8

Collation (Agrupamento)	Descrição
German_PhoneBook_100_CS_AI_WS	German-PhoneBook-100, diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, diferenci a a largura
German_PhoneBook_100_CS_AI_WS_SC	German-PhoneBook-100, diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, diferenci a a largura, caracteres complementares
German_PhoneBook_100_CS_AI_WS_SC_UTF 8	German-PhoneBook-100, diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, diferenci a a largura, caracteres complementares, UTF8
German_PhoneBook_100_CS_AS	German-PhoneBook-100, diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura
German_PhoneBook_100_CS_AS_KS	German-PhoneBook-100, diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, não diferencia a largura
German_PhoneBook_100_CS_AS_KS_SC	German-PhoneBook-100, diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, não diferencia a largura, caracteres complementares
German_PhoneBook_100_CS_AS_ KS_SC_UTF8	German-PhoneBook-100, diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, não diferencia a largura, caracteres complementares, UTF8

Collation (Agrupamento)	Descrição
German_PhoneBook_100_CS_AS_KS_WS	German-PhoneBook-100, diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, diferencia a largura
German_PhoneBook_100_CS_AS_KS_WS_SC	German-PhoneBook-100, diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, diferencia a largura, caracteres complementares
German_PhoneBook_100_CS_AS_ KS_WS_SC_UTF8	German-PhoneBook-100, diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, diferencia a largura, caracteres complementares, UTF8
German_PhoneBook_BIN	German-PhoneBook, classificação binária
German_PhoneBook_BIN2	German-PhoneBook, classificação binária de comparação de ponto de código
German_PhoneBook_CI_AI	German-PhoneBook, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, não diferencia a largura
German_PhoneBook_CI_AI_KS	German-PhoneBook, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, não diferenci a a largura
German_PhoneBook_CI_AI_KS_WS	German-PhoneBook, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, diferencia a largura

Collation (Agrupamento)	Descrição
German_PhoneBook_CI_AI_WS	German-PhoneBook, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, diferenci a a largura
German_PhoneBook_CI_AS	German-PhoneBook, não diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura
German_PhoneBook_CI_AS_KS	German-PhoneBook, não diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, não diferencia a largura
German_PhoneBook_CI_AS_KS_WS	German-PhoneBook, não diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, diferencia a largura
German_PhoneBook_CI_AS_WS	German-PhoneBook, não diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, diferencia a largura
German_PhoneBook_CS_AI	German-PhoneBook, diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, não diferencia a largura
German_PhoneBook_CS_AI_KS	German-PhoneBook, diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, não diferenci a a largura

Collation (Agrupamento)	Descrição
German_PhoneBook_CS_AI_KS_WS	German-PhoneBook, diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, diferencia a largura
German_PhoneBook_CS_AI_WS	German-PhoneBook, diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, diferenci a a largura
German_PhoneBook_CS_AS	German-PhoneBook, diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura
German_PhoneBook_CS_AS_KS	German-PhoneBook, diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, não diferencia a largura
German_PhoneBook_CS_AS_KS_WS	German-PhoneBook, diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, diferencia a largura
German_PhoneBook_CS_AS_WS	German-PhoneBook, diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, diferencia a largura
Hebrew_BIN	Hebraico, classificação binária
Hebrew_CI_AS	Hebrew, não diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura
Japanese_90_BIN	Japanese-90, classificação binária

Collation (Agrupamento)	Descrição
Japanese_90_BIN2	Japanese-90, classificação binária de comparação de ponto de código
Japanese_90_CI_AI	Japanese-90, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, não diferencia a largura
Japanese_90_CI_AI_KS	Japanese-90, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, não diferencia a largura
Japanese_90_CI_AI_KS_SC	Japanese-90, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, não diferencia a largura, caracteres complementares
Japanese_90_CI_AI_KS_SC_UTF8	Japanese-90, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, não diferencia a largura, caracteres complementares, UTF8
Japanese_90_CI_AI_KS_WS	Japanese-90, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, diferencia a largura
Japanese_90_CI_AI_KS_WS_SC	Japanese-90, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, diferencia a largura, caracteres complementares
Japanese_90_CI_AI_KS_WS_SC_UTF8	Japanese-90, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, diferencia a largura, caracteres complementares, UTF8

Collation (Agrupamento)	Descrição
Japanese_90_CI_AI_SC	Japanese-90, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, não diferencia a largura, caracteres complementares
Japanese_90_CI_AI_SC_UTF8	Japanese-90, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, não diferencia a largura, caracteres complementares, UTF8
Japanese_90_CI_AI_WS	Japanese-90, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, diferencia a largura
Japanese_90_CI_AI_WS_SC	Japanese-90, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, diferencia a largura, caracteres complementares
Japanese_90_CI_AI_WS_SC_UTF8	Japanese-90, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, diferencia a largura, caracteres complementares, UTF8
Japanese_90_CI_AS	Japanese-90, não diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura
Japanese_90_CI_AS_KS	Japanese-90, não diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, não diferencia a largura

Collation (Agrupamento)	Descrição
Japanese_90_CI_AS_KS_SC	Japanese-90, não diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, não diferencia a largura, caractere s complementares
Japanese_90_CI_AS_KS_SC_UTF8	Japanese-90, não diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, não diferencia a largura, caractere s complementares, UTF8
Japanese_90_CI_AS_KS_WS	Japanese-90, não diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, diferencia a largura
Japanese_90_CI_AS_KS_WS_SC	Japanese-90, não diferencia letras maiúscula s de minúsculas, diferencia acentos, diferencia o tipo de kana, diferencia a largura, caracteres complementares
Japanese_90_CI_AS_KS_WS_SC_UTF8	Japanese-90, não diferencia letras maiúscula s de minúsculas, diferencia acentos, diferencia o tipo de kana, diferencia a largura, caracteres complementares, UTF8
Japanese_90_CI_AS_SC	Japanese-90, não diferencia letras maiúscula s de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura, caracteres complementares
Japanese_90_CI_AS_SC_UTF8	Japanese-90, não diferencia letras maiúscula s de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura, caracteres complementares, UTF8

Collation (Agrupamento)	Descrição
Japanese_90_CI_AS_WS	Japanese-90, não diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, diferencia a largura
Japanese_90_CI_AS_WS_SC	Japanese-90, não diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, diferencia a largura, caracteres complementares
Japanese_90_CI_AS_WS_SC_UTF8	Japanese-90, não diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, diferencia a largura, caracteres complementares, UTF8
Japanese_90_CS_AI	Japanese-90, diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, não diferencia a largura
Japanese_90_CS_AI_KS	Japanese-90, diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, não diferencia a largura
Japanese_90_CS_AI_KS_SC	Japanese-90, diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, não diferencia a largura, caracteres complementares
Japanese_90_CS_AI_KS_SC_UTF8	Japanese-90, diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, não diferencia a largura, caracteres complementares, UTF8
Japanese_90_CS_AI_KS_WS	Japanese-90, diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, diferencia a largura

Collation (Agrupamento)	Descrição
Japanese_90_CS_AI_KS_WS_SC	Japanese-90, diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, diferencia a largura, caracteres complementares
Japanese_90_CS_AI_KS_WS_SC_UTF8	Japanese-90, diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, diferencia a largura, caracteres complementares, UTF8
Japanese_90_CS_AI_SC	Japanese-90, diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, não diferencia a largura, caracteres complementares
Japanese_90_CS_AI_SC_UTF8	Japanese-90, diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, não diferencia a largura, caracteres complementares, UTF8
Japanese_90_CS_AI_WS	Japanese-90, diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, diferencia a largura
Japanese_90_CS_AI_WS_SC	Japanese-90, diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, diferencia a largura, caracteres complementares
Japanese_90_CS_AI_WS_SC_UTF8	Japanese-90, diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, diferencia a largura, caracteres complementares, UTF8

Collation (Agrupamento)	Descrição
Japanese_90_CS_AS	Japanese-90, diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura
Japanese_90_CS_AS_KS	Japanese-90, diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, não diferencia a largura
Japanese_90_CS_AS_KS_SC	Japanese-90, diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, não diferencia a largura, caractere s complementares
Japanese_90_CS_AS_KS_SC_UTF8	Japanese-90, diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, não diferencia a largura, caractere s complementares, UTF8
Japanese_90_CS_AS_KS_WS	Japanese-90, diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, diferencia a largura
Japanese_90_CS_AS_KS_WS_SC	Japanese-90, diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, diferencia a largura, caracteres complementares
Japanese_90_CS_AS_KS_WS_SC_UTF8	Japanese-90, diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, diferencia a largura, caracteres complementares, UTF8
Japanese_90_CS_AS_SC	Japanese-90, diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferenci a o tipo de kana, não diferencia a largura, caracteres complementares

Collation (Agrupamento)	Descrição
Japanese_90_CS_AS_SC_UTF8	Japanese-90, diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferenci a o tipo de kana, não diferencia a largura, caracteres complementares, UTF8
Japanese_90_CS_AS_WS	Japanese-90, diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, diferencia a largura
Japanese_90_CS_AS_WS_SC	Japanese-90, diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, diferencia a largura, caracteres complementares
Japanese_90_CS_AS_WS_SC_UTF8	Japanese-90, diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, diferencia a largura, caracteres complementares, UTF8
Japanese_BIN	Japonês, classificação binária
Japanese_BIN2	Japanese, classificação binária de comparação de ponto de código
Japanese_Bushu_Kakusu_100_BIN	Japanese-Bushu-Kakusu-100, classificação binária
Japanese_Bushu_Kakusu_100_BIN2	Japanese-Bushu-Kakusu-100, classificação binária de comparação de ponto de código
Japanese_Bushu_Kakusu_100_CI_AI	Japanese-Bushu-Kakusu-100, não diferencia letras maiúsculas de minúsculas, não diferenci a acentos, não diferencia o tipo de kana, não diferencia a largura

Collation (Agrupamento)	Descrição
Japanese_Bushu_Kakusu_100_CI_AI_KS	Japanese-Bushu-Kakusu-100, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, não diferenci a a largura
Japanese_Bushu_Kakusu_100_CI_AI_KS_SC	Japanese-Bushu-Kakusu-100, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, não diferenci a a largura, caracteres complementares
Japanese_Bushu_Kakusu_100_CI_AI_KS_S C_UTF8	Japanese-Bushu-Kakusu-100, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, não diferenci a a largura, caracteres complementares, UTF8
Japanese_Bushu_Kakusu_100_CI_AI_KS_WS	Japanese-Bushu-Kakusu-100, não diferencia letras maiúsculas de minúsculas, não diferenci a acentos, diferencia o tipo de kana, diferencia a largura
Japanese_Bushu_Kakusu_100_CI_AI_KS_W S_SC	Japanese-Bushu-Kakusu-100, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, diferencia a largura, caracteres complementares
Japanese_Bushu_Kakusu_100_CI_AI_KS_W S_SC_UTF8	Japanese-Bushu-Kakusu-100, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, diferencia a largura, caracteres complementares, UTF8
Japanese_Bushu_Kakusu_100_CI_AI_SC	Japanese-Bushu-Kakusu-100, não diferencia letras maiúsculas de minúsculas, não diferenci a acentos, não diferencia o tipo de kana, não diferencia a largura, caracteres complemen tares

Collation (Agrupamento)	Descrição
Japanese_Bushu_Kakusu_100_CI_AI_SC_UTF8	Japanese-Bushu-Kakusu-100, não diferencia letras maiúsculas de minúsculas, não diferenci a acentos, não diferencia o tipo de kana, não diferencia a largura, caracteres complemen tares, UTF8
Japanese_Bushu_Kakusu_100_CI_AI_WS	Japanese-Bushu-Kakusu-100, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, diferenci a a largura
Japanese_Bushu_Kakusu_100_CI_AI_WS_SC	Japanese-Bushu-Kakusu-100, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, diferenci a a largura, caracteres complementares
Japanese_Bushu_Kakusu_100_CI_AI_WS_SC_UTF8	Japanese-Bushu-Kakusu-100, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, diferenci a a largura, caracteres complementares, UTF8
Japanese_Bushu_Kakusu_100_CI_AS	Japanese-Bushu-Kakusu-100, não diferenci a letras maiúsculas de minúsculas, diferenci a acentos, não diferencia o tipo de kana, não diferencia a largura
Japanese_Bushu_Kakusu_100_CI_AS_KS	Japanese-Bushu-Kakusu-100, não diferenci a letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, não diferenci a a largura
Japanese_Bushu_Kakusu_100_CI_AS_KS_SC	Japanese-Bushu-Kakusu-100, não diferenci a letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, não diferenci a a largura, caracteres complementares

Collation (Agrupamento)	Descrição
Japanese_Bushu_Kakusu_100_CI_AS_KS_SC_UTF8	Japanese-Bushu-Kakusu-100, não diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, não diferencia a largura, caracteres complementares, UTF8
Japanese_Bushu_Kakusu_100_CI_AS_KS_WS	Japanese-Bushu-Kakusu-100, não diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, diferencia a largura
Japanese_Bushu_Kakusu_100_CI_AS_KS_WS_SC	Japanese-Bushu-Kakusu-100, não diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, diferencia a largura, caracteres complementares
Japanese_Bushu_Kakusu_100_CI_AS_KS_WS_SC_UTF8	Japanese-Bushu-Kakusu-100, não diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, diferencia a largura, caracteres complementares, UTF8
Japanese_Bushu_Kakusu_100_CI_AS_SC	Japanese-Bushu-Kakusu-100, não diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura, caracteres complementares
Japanese_Bushu_Kakusu_100_CI_AS_SC_UTF8	Japanese-Bushu-Kakusu-100, não diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura, caracteres complementares, UTF8
Japanese_Bushu_Kakusu_100_CI_AS_WS	Japanese-Bushu-Kakusu-100, não diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, diferencia a largura

Collation (Agrupamento)	Descrição
Japanese_Bushu_Kakusu_100_CI_AS_WS_SC	Japanese-Bushu-Kakusu-100, não diferenci a letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, diferenci a a largura, caracteres complementares
Japanese_Bushu_Kakusu_100_CI_AS_WS_S C_UTF8	Japanese-Bushu-Kakusu-100, não diferenci a letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, diferenci a a largura, caracteres complementares, UTF8
Japanese_Bushu_Kakusu_100_CS_AI	Japanese-Bushu-Kakusu-100, diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, não diferencia a largura
Japanese_Bushu_Kakusu_100_CS_AI_KS	Japanese-Bushu-Kakusu-100, diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, não diferenci a a largura
Japanese_Bushu_Kakusu_100_CS_AI_KS_SC	Japanese-Bushu-Kakusu-100, diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, não diferenci a a largura, caracteres complementares
Japanese_Bushu_Kakusu_100_CS_AI_KS_S C_UTF8	Japanese-Bushu-Kakusu-100, diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, não diferenci a a largura, caracteres complementares, UTF8
Japanese_Bushu_Kakusu_100_CS_AI_KS_WS	Japanese-Bushu-Kakusu-100, diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, diferencia a largura

Collation (Agrupamento)	Descrição
Japanese_Bushu_Kakusu_100_CS_AI_KS_WS_SC	Japanese-Bushu-Kakusu-100, diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, diferencia a largura, caracteres complementares
Japanese_Bushu_Kakusu_100_CS_AI_KS_WS_SC_UTF8	Japanese-Bushu-Kakusu-100, diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, diferencia a largura, caracteres complementares, UTF8
Japanese_Bushu_Kakusu_100_CS_AI_SC	Japanese-Bushu-Kakusu-100, diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, não diferencia a largura, caracteres complemen tares
Japanese_Bushu_Kakusu_100_CS_AI_SC_UTF8	Japanese-Bushu-Kakusu-100, diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, não diferencia a largura, caracteres complemen tares, UTF8
Japanese_Bushu_Kakusu_100_CS_AI_WS	Japanese-Bushu-Kakusu-100, diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, diferenci a a largura
Japanese_Bushu_Kakusu_100_CS_AI_WS_SC	Japanese-Bushu-Kakusu-100, diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, diferenci a a largura, caracteres complementares
Japanese_Bushu_Kakusu_100_CS_AI_WS_SC_UTF8	Japanese-Bushu-Kakusu-100, diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, diferenci a a largura, caracteres complementares, UTF8

Collation (Agrupamento)	Descrição
Japanese_Bushu_Kakusu_100_CS_AS	Japanese-Bushu-Kakusu-100, diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura
Japanese_Bushu_Kakusu_100_CS_AS_KS	Japanese-Bushu-Kakusu-100, diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, não diferencia a largura
Japanese_Bushu_Kakusu_100_CS_AS_KS_SC	Japanese-Bushu-Kakusu-100, diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, não diferencia a largura, caracteres complementares
Japanese_Bushu_Kakusu_100_CS_AS_KS_SC_UTF8	Japanese-Bushu-Kakusu-100, diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, não diferencia a largura, caracteres complementares, UTF8
Japanese_Bushu_Kakusu_100_CS_AS_KS_WS	Japanese-Bushu-Kakusu-100, diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, diferencia a largura
Japanese_Bushu_Kakusu_100_CS_AS_KS_WS_SC	Japanese-Bushu-Kakusu-100, diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, diferencia a largura, caracteres complementares
Japanese_Bushu_Kakusu_100_CS_AS_KS_WS_SC_UTF8	Japanese-Bushu-Kakusu-100, diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, diferencia a largura, caracteres complementares, UTF8

Collation (Agrupamento)	Descrição
Japanese_Bushu_Kakusu_100_CS_AS_SC	Japanese-Bushu-Kakusu-100, diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura, caracteres complementares
Japanese_Bushu_Kakusu_100_CS_AS_SC_UTF8	Japanese-Bushu-Kakusu-100, diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura, caracteres complementares, UTF8
Japanese_Bushu_Kakusu_100_CS_AS_WS	Japanese-Bushu-Kakusu-100, diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, diferencia a largura
Japanese_Bushu_Kakusu_100_CS_AS_WS_SC	Japanese-Bushu-Kakusu-100, diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, diferencia a largura, caracteres complementares
Japanese_Bushu_Kakusu_100_CS_AS_WS_SC_UTF8	Japanese-Bushu-Kakusu-100, diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, diferencia a largura, caracteres complementares, UTF8
Japanese_Bushu_Kakusu_140_BIN	Japanese-Bushu-Kakusu-140, classificação binária
Japanese_Bushu_Kakusu_140_BIN2	Japanese-Bushu-Kakusu-140, classificação binária de comparação de ponto de código
Japanese_Bushu_Kakusu_140_CI_AI	Japanese-Bushu-Kakusu-140, não diferencia letras maiúsculas de minúsculas, não diferenci a acentos, não diferencia o tipo de kana, não diferencia a largura, caracteres complemen tares, não diferencia o seletor de variação

Collation (Agrupamento)	Descrição
Japanese_Bushu_Kakusu_140_CI_AI_KS	Japanese-Bushu-Kakusu-140, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, não diferenci a a largura, caracteres complementares, não diferencia o seletor de variação
Japanese_Bushu_Kakusu_140_CI_AI_KS_U TF8	Japanese-Bushu-Kakusu-140, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, não diferenci a a largura, caracteres complementares, não diferencia o seletor de variação, UTF8
Japanese_Bushu_Kakusu_140_CI_AI_KS_VSS	Japanese-Bushu-Kakusu-140, não diferencia letras maiúsculas de minúsculas, não diferenci a acentos, diferencia o tipo de kana, não diferencia a largura, caracteres complemen tares, diferencia o seletor de variação
Japanese_Bushu_Kakusu_140_CI_AI_KS_V SS_UTF8	Japanese-Bushu-Kakusu-140, não diferencia letras maiúsculas de minúsculas, não diferenci a acentos, diferencia o tipo de kana, não diferencia a largura, caracteres complemen tares, diferencia o seletor de variação, UTF8
Japanese_Bushu_Kakusu_140_CI_AI_KS_WS	Japanese-Bushu-Kakusu-140, não diferencia letras maiúsculas de minúsculas, não diferenci a acentos, diferencia o tipo de kana, diferenci a a largura, caracteres complementares, não diferencia o seletor de variação
Japanese_Bushu_Kakusu_140_CI_AI_KS_W S_UTF8	Japanese-Bushu-Kakusu-140, não diferencia letras maiúsculas de minúsculas, não diferenci a acentos, diferencia o tipo de kana, diferenci a a largura, caracteres complementares, não diferencia o seletor de variação, UTF8

Collation (Agrupamento)	Descrição
Japanese_Bushu_Kakusu_140_CI_AI_KS_WS_VSS	Japanese-Bushu-Kakusu-140, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, diferencia a largura, caracteres complementares, diferencia o seletor de variação
Japanese_Bushu_Kakusu_140_CI_AI_KS_WS_VSS_UTF8	Japanese-Bushu-Kakusu-140, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, diferencia a largura, caracteres complementares, diferencia o seletor de variação, UTF8
Japanese_Bushu_Kakusu_140_CI_AI_UTF8	Japanese-Bushu-Kakusu-140, não diferencia letras maiúsculas de minúsculas, não diferenci a acentos, não diferencia o tipo de kana, não diferencia a largura, caracteres complemen tares, não diferencia o seletor de variação, UTF8
Japanese_Bushu_Kakusu_140_CI_AI_VSS	Japanese-Bushu-Kakusu-140, não diferencia letras maiúsculas de minúsculas, não diferenci a acentos, não diferencia o tipo de kana, não diferencia a largura, caracteres complemen tares, diferencia o seletor de variação
Japanese_Bushu_Kakusu_140_CI_AI_VSS_UTF8	Japanese-Bushu-Kakusu-140, não diferencia letras maiúsculas de minúsculas, não diferenci a acentos, não diferencia o tipo de kana, não diferencia a largura, caracteres complemen tares, diferencia o seletor de variação, UTF8
Japanese_Bushu_Kakusu_140_CI_AI_WS	Japanese-Bushu-Kakusu-140, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, diferenci a a largura, caracteres complementares, não diferencia o seletor de variação

Collation (Agrupamento)	Descrição
Japanese_Bushu_Kakusu_140_CI_AI_WS_UTF8	Japanese-Bushu-Kakusu-140, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, diferenci a a largura, caracteres complementares, não diferencia o seletor de variação, UTF8
Japanese_Bushu_Kakusu_140_CI_AI_WS_VSS	Japanese-Bushu-Kakusu-140, não diferencia letras maiúsculas de minúsculas, não diferenci a acentos, não diferencia o tipo de kana, diferencia a largura, caracteres complemen tares, diferencia o seletor de variação
Japanese_Bushu_Kakusu_140_CI_AI_WS_VSS_UTF8	Japanese-Bushu-Kakusu-140, não diferencia letras maiúsculas de minúsculas, não diferenci a acentos, não diferencia o tipo de kana, diferencia a largura, caracteres complemen tares, diferencia o seletor de variação, UTF8
Japanese_Bushu_Kakusu_140_CI_AS	Japanese-Bushu-Kakusu-140, não diferenci a letras maiúsculas de minúsculas, diferenci a acentos, não diferencia o tipo de kana, não diferencia a largura, caracteres complemen tares, não diferencia o seletor de variação
Japanese_Bushu_Kakusu_140_CI_AS_KS	Japanese-Bushu-Kakusu-140, não diferenci a letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, não diferenci a a largura, caracteres complementares, não diferencia o seletor de variação
Japanese_Bushu_Kakusu_140_CI_AS_KS_UTF8	Japanese-Bushu-Kakusu-140, não diferenci a letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, não diferenci a a largura, caracteres complementares, não diferencia o seletor de variação, UTF8

Collation (Agrupamento)	Descrição
Japanese_Bushu_Kakusu_140_CI_AS_KS_VSS	Japanese-Bushu-Kakusu-140, não diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, não diferencia a largura, caracteres complementares, diferencia o seletor de variação
Japanese_Bushu_Kakusu_140_CI_AS_KS_VSS_UTF8	Japanese-Bushu-Kakusu-140, não diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, não diferencia a largura, caracteres complementares, diferencia o seletor de variação, UTF8
Japanese_Bushu_Kakusu_140_CI_AS_KS_WS	Japanese-Bushu-Kakusu-140, não diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, diferencia a largura, caracteres complementares, não diferencia o seletor de variação
Japanese_Bushu_Kakusu_140_CI_AS_KS_WS_UTF8	Japanese-Bushu-Kakusu-140, não diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, diferencia a largura, caracteres complementares, não diferencia o seletor de variação, UTF8
Japanese_Bushu_Kakusu_140_CI_AS_KS_WS_VSS	Japanese-Bushu-Kakusu-140, não diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, diferencia a largura, caracteres complementares, diferencia o seletor de variação
Japanese_Bushu_Kakusu_140_CI_AS_KS_WS_VSS_UTF8	Japanese-Bushu-Kakusu-140, não diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, diferencia a largura, caracteres complementares, diferencia o seletor de variação, UTF8

Collation (Agrupamento)	Descrição
Japanese_Bushu_Kakusu_140_CI_AS_UTF8	Japanese-Bushu-Kakusu-140, não diferenci a letras maiúsculas de minúsculas, diferenci a acentos, não diferencia o tipo de kana, não diferencia a largura, caracteres complemen tares, não diferencia o seletor de variação, UTF8
Japanese_Bushu_Kakusu_140_CI_AS_VSS	Japanese-Bushu-Kakusu-140, não diferenci a letras maiúsculas de minúsculas, diferenci a acentos, não diferencia o tipo de kana, não diferencia a largura, caracteres complemen tares, diferencia o seletor de variação
Japanese_Bushu_Kakusu_140_CI_AS_VSS_ UTF8	Japanese-Bushu-Kakusu-140, não diferenci a letras maiúsculas de minúsculas, diferenci a acentos, não diferencia o tipo de kana, não diferencia a largura, caracteres complemen tares, diferencia o seletor de variação, UTF8
Japanese_Bushu_Kakusu_140_CI_AS_WS	Japanese-Bushu-Kakusu-140, não diferenci a letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, diferenci a a largura, caracteres complementares, não diferencia o seletor de variação
Japanese_Bushu_Kakusu_140_CI_AS_WS_U TF8	Japanese-Bushu-Kakusu-140, não diferenci a letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, diferenci a a largura, caracteres complementares, não diferencia o seletor de variação, UTF8
Japanese_Bushu_Kakusu_140_CI_AS_WS_V SS	Japanese-Bushu-Kakusu-140, não diferenci a letras maiúsculas de minúsculas, diferenci a acentos, não diferencia o tipo de kana, diferencia a largura, caracteres complemen tares, diferencia o seletor de variação

Collation (Agrupamento)	Descrição
Japanese_Bushu_Kakusu_140_CI_AS_WS_VSS_UTF8	Japanese-Bushu-Kakusu-140, não diferenci a letras maiúsculas de minúsculas, diferenci a acentos, não diferencia o tipo de kana, diferencia a largura, caracteres complemen tares, diferencia o seletor de variação, UTF8
Japanese_Bushu_Kakusu_140_CS_AI	Japanese-Bushu-Kakusu-140, diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, não diferencia a largura, caracteres complemen tares, não diferencia o seletor de variação
Japanese_Bushu_Kakusu_140_CS_AI_KS	Japanese-Bushu-Kakusu-140, diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, não diferenci a a largura, caracteres complementares, não diferencia o seletor de variação
Japanese_Bushu_Kakusu_140_CS_AI_KS_UTF8	Japanese-Bushu-Kakusu-140, diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, não diferenci a a largura, caracteres complementares, não diferencia o seletor de variação, UTF8
Japanese_Bushu_Kakusu_140_CS_AI_KS_VSS	Japanese-Bushu-Kakusu-140, diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, não diferenci a a largura, caracteres complementares, diferencia o seletor de variação
Japanese_Bushu_Kakusu_140_CS_AI_KS_VSS_UTF8	Japanese-Bushu-Kakusu-140, diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, não diferenci a a largura, caracteres complementares, diferencia o seletor de variação, UTF8

Collation (Agrupamento)	Descrição
Japanese_Bushu_Kakusu_140_CS_AI_KS_WS	Japanese-Bushu-Kakusu-140, diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, diferenci a a largura, caracteres complementares, não diferencia o seletor de variação
Japanese_Bushu_Kakusu_140_CS_AI_KS_W S_UTF8	Japanese-Bushu-Kakusu-140, diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, diferenci a a largura, caracteres complementares, não diferencia o seletor de variação, UTF8
Japanese_Bushu_Kakusu_140_CS_AI_KS_W S_VSS	Japanese-Bushu-Kakusu-140, diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, diferencia a largura, caracteres complementares, diferencia o seletor de variação
Japanese_Bushu_Kakusu_140_CS_AI_KS_W S_VSS_UTF8	Japanese-Bushu-Kakusu-140, diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, diferencia a largura, caracteres complementares, diferencia o seletor de variação, UTF8
Japanese_Bushu_Kakusu_140_CS_AI_UTF8	Japanese-Bushu-Kakusu-140, diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, não diferencia a largura, caracteres complemen tares, não diferencia o seletor de variação, UTF8
Japanese_Bushu_Kakusu_140_CS_AI_VSS	Japanese-Bushu-Kakusu-140, diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, não diferencia a largura, caracteres complemen tares, diferencia o seletor de variação

Collation (Agrupamento)	Descrição
Japanese_Bushu_Kakusu_140_CS_AI_VSS_UTF8	Japanese-Bushu-Kakusu-140, diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, não diferencia a largura, caracteres complementares, diferencia o seletor de variação, UTF8
Japanese_Bushu_Kakusu_140_CS_AI_WS	Japanese-Bushu-Kakusu-140, diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, diferencia a largura, caracteres complementares, não diferencia o seletor de variação
Japanese_Bushu_Kakusu_140_CS_AI_WS_UTF8	Japanese-Bushu-Kakusu-140, diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, diferencia a largura, caracteres complementares, não diferencia o seletor de variação, UTF8
Japanese_Bushu_Kakusu_140_CS_AI_WS_VSS	Japanese-Bushu-Kakusu-140, diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, diferencia a largura, caracteres complementares, diferencia o seletor de variação
Japanese_Bushu_Kakusu_140_CS_AI_WS_VSS_UTF8	Japanese-Bushu-Kakusu-140, diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, diferencia a largura, caracteres complementares, diferencia o seletor de variação, UTF8
Japanese_Bushu_Kakusu_140_CS_AS	Japanese-Bushu-Kakusu-140, diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura, caracteres complementares, não diferencia o seletor de variação

Collation (Agrupamento)	Descrição
Japanese_Bushu_Kakusu_140_CS_AS_KS	Japanese-Bushu-Kakusu-140, diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, não diferenci a a largura, caracteres complementares, não diferencia o seletor de variação
Japanese_Bushu_Kakusu_140_CS_AS_KS_UTF8	Japanese-Bushu-Kakusu-140, diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, não diferenci a a largura, caracteres complementares, não diferencia o seletor de variação, UTF8
Japanese_Bushu_Kakusu_140_CS_AS_KS_VSS	Japanese-Bushu-Kakusu-140, diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, não diferencia a largura, caracteres complementares, diferencia o seletor de variação
Japanese_Bushu_Kakusu_140_CS_AS_KS_VSS_UTF8	Japanese-Bushu-Kakusu-140, diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, não diferencia a largura, caracteres complementares, diferencia o seletor de variação, UTF8
Japanese_Bushu_Kakusu_140_CS_AS_KS_WS	Japanese-Bushu-Kakusu-140, diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, diferencia a largura, caracteres complementares, não diferencia o seletor de variação
Japanese_Bushu_Kakusu_140_CS_AS_KS_WS_UTF8	Japanese-Bushu-Kakusu-140, diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, diferencia a largura, caracteres complementares, não diferencia o seletor de variação, UTF8

Collation (Agrupamento)	Descrição
Japanese_Bushu_Kakusu_140_CS_AS_KS_WS_VSS	Japanese-Bushu-Kakusu-140, diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, diferencia a largura, caracteres complementares, diferencia o seletor de variação
Japanese_Bushu_Kakusu_140_CS_AS_KS_WS_VSS_UTF8	Japanese-Bushu-Kakusu-140, diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, diferencia a largura, caracteres complementares, diferencia o seletor de variação, UTF8
Japanese_Bushu_Kakusu_140_CS_AS_UTF8	Japanese-Bushu-Kakusu-140, diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferenci a a largura, caracteres complementares, não diferencia o seletor de variação, UTF8
Japanese_Bushu_Kakusu_140_CS_AS_VSS	Japanese-Bushu-Kakusu-140, diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura, caracteres complementares, diferencia o seletor de variação
Japanese_Bushu_Kakusu_140_CS_AS_VSS_UTF8	Japanese-Bushu-Kakusu-140, diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura, caracteres complementares, diferencia o seletor de variação, UTF8
Japanese_Bushu_Kakusu_140_CS_AS_WS	Japanese-Bushu-Kakusu-140, diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, diferenci a a largura, caracteres complementares, não diferencia o seletor de variação

Collation (Agrupamento)	Descrição
Japanese_Bushu_Kakusu_140_CS_AS_WS_UTF8	Japanese-Bushu-Kakusu-140, diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, diferenci a a largura, caracteres complementares, não diferencia o seletor de variação, UTF8
Japanese_Bushu_Kakusu_140_CS_AS_WS_VSS	Japanese-Bushu-Kakusu-140, diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, diferencia a largura, caracteres complementares, diferencia o seletor de variação
Japanese_Bushu_Kakusu_140_CS_AS_WS_VSS_UTF8	Japanese-Bushu-Kakusu-140, diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, diferencia a largura, caracteres complementares, diferencia o seletor de variação, UTF8
Japanese_CI_AI	Japanese, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, não diferencia a largura
Japanese_CI_AI_KS	Japanese, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, não diferencia a largura
Japanese_CI_AI_KS_WS	Japanese, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, diferencia a largura
Japanese_CI_AI_WS	Japanese, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, diferencia a largura

Collation (Agrupamento)	Descrição
Japanese_CI_AS	Japanese, não diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura
Japanese_CI_AS_KS	Japanese, não diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, não diferencia a largura
Japanese_CI_AS_KS_WS	Japanese, não diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, diferencia a largura
Japanese_CI_AS_WS	Japanese, não diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, diferencia a largura
Japanese_CS_AI	Japanese, diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, não diferencia a largura
Japanese_CS_AI_KS	Japanese, diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, não diferencia a largura
Japanese_CS_AI_KS_WS	Japanese, diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, diferencia a largura
Japanese_CS_AI_WS	Japanese, diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, diferencia a largura
Japanese_CS_AS	Japanese, diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura

Collation (Agrupamento)	Descrição
Japanese_CS_AS_KS	Japanese, diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, não diferencia a largura
Japanese_CS_AS_KS_WS	Japanese, diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, diferencia a largura
Japanese_CS_AS_WS	Japanese, diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, diferencia a largura
Japanese_Unicode_BIN	Japanese-Unicode, classificação binária
Japanese_Unicode_BIN2	Japanese-Unicode, classificação binária de comparação de ponto de código
Japanese_Unicode_CI_AI	Japanese-Unicode, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, não diferencia a largura
Japanese_Unicode_CI_AI_KS	Japanese-Unicode, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, não diferenci a a largura
Japanese_Unicode_CI_AI_KS_WS	Japanese-Unicode, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, diferencia a largura
Japanese_Unicode_CI_AI_WS	Japanese-Unicode, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, diferenci a a largura

Collation (Agrupamento)	Descrição
Japanese_Unicode_CI_AS	Japanese-Unicode, não diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura
Japanese_Unicode_CI_AS_KS	Japanese-Unicode, não diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, não diferencia a largura
Japanese_Unicode_CI_AS_KS_WS	Japanese-Unicode, não diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, diferencia a largura
Japanese_Unicode_CI_AS_WS	Japanese-Unicode, não diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, diferencia a largura
Japanese_Unicode_CS_AI	Japanese-Unicode, diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, não diferencia a largura
Japanese_Unicode_CS_AI_KS	Japanese-Unicode, diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, não diferencia a largura
Japanese_Unicode_CS_AI_KS_WS	Japanese-Unicode, diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, diferencia a largura
Japanese_Unicode_CS_AI_WS	Japanese-Unicode, diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, diferencia a largura

Collation (Agrupamento)	Descrição
Japanese_Unicode_CS_AS	Japanese-Unicode, diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura
Japanese_Unicode_CS_AS_KS	Japanese-Unicode, diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, não diferencia a largura
Japanese_Unicode_CS_AS_KS_WS	Japanese-Unicode, diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, diferencia a largura
Japanese_Unicode_CS_AS_WS	Japanese-Unicode, diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, diferencia a largura
Japanese_XJIS_100_BIN	Japanese-XJIS-100, classificação binária
Japanese_XJIS_100_BIN2	Japanese-XJIS-100, classificação binária de comparação de ponto de código
Japanese_XJIS_100_CI_AI	Japanese-XJIS-100, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, não diferencia a largura
Japanese_XJIS_100_CI_AI_KS	Japanese-XJIS-100, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, não diferencia a largura
Japanese_XJIS_100_CI_AI_KS_SC	Japanese-XJIS-100, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, não diferencia a largura, caracteres complementares

Collation (Agrupamento)	Descrição
Japanese_XJIS_100_CI_AI_KS_SC_UTF8	Japanese-XJIS-100, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, não diferenci a a largura, caracteres complementares, UTF8
Japanese_XJIS_100_CI_AI_KS_WS	Japanese-XJIS-100, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, diferencia a largura
Japanese_XJIS_100_CI_AI_KS_WS_SC	Japanese-XJIS-100, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, diferencia a largura, caracteres complementares
Japanese_XJIS_100_CI_AI_KS_WS_SC_UTF8	Japanese-XJIS-100, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, diferencia a largura, caracteres complementares, UTF8
Japanese_XJIS_100_CI_AI_SC	Japanese-XJIS-100, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, não diferencia a largura, caracteres complemen tares
Japanese_XJIS_100_CI_AI_SC_UTF8	Japanese-XJIS-100, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, não diferencia a largura, caracteres complemen tares, UTF8
Japanese_XJIS_100_CI_AI_WS	Japanese-XJIS-100, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, diferenci a a largura

Collation (Agrupamento)	Descrição
Japanese_XJIS_100_CI_AI_WS_SC	Japanese-XJIS-100, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, diferenci a a largura, caracteres complementares
Japanese_XJIS_100_CI_AI_WS_SC_UTF8	Japanese-XJIS-100, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, diferenci a a largura, caracteres complementares, UTF8
Japanese_XJIS_100_CI_AS	Japanese-XJIS-100, não diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura
Japanese_XJIS_100_CI_AS_KS	Japanese-XJIS-100, não diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, não diferencia a largura
Japanese_XJIS_100_CI_AS_KS_SC	Japanese-XJIS-100, não diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, não diferencia a largura, caracteres complementares
Japanese_XJIS_100_CI_AS_KS_SC_UTF8	Japanese-XJIS-100, não diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, não diferencia a largura, caracteres complementares, UTF8
Japanese_XJIS_100_CI_AS_KS_WS	Japanese-XJIS-100, não diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, diferencia a largura

Collation (Agrupamento)	Descrição
Japanese_XJIS_100_CI_AS_KS_WS_SC	Japanese-XJIS-100, não diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, diferencia a largura, caracteres complementares
Japanese_XJIS_100_CI_AS_KS_WS_SC_UTF8	Japanese-XJIS-100, não diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, diferencia a largura, caracteres complementares, UTF8
Japanese_XJIS_100_CI_AS_SC	Japanese-XJIS-100, não diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura, caracteres complementares
Japanese_XJIS_100_CI_AS_SC_UTF8	Japanese-XJIS-100, não diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura, caracteres complementares, UTF8
Japanese_XJIS_100_CI_AS_WS	Japanese-XJIS-100, não diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, diferencia a largura
Japanese_XJIS_100_CI_AS_WS_SC	Japanese-XJIS-100, não diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, diferencia a largura, caracteres complementares
Japanese_XJIS_100_CI_AS_WS_SC_UTF8	Japanese-XJIS-100, não diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, diferencia a largura, caracteres complementares, UTF8

Collation (Agrupamento)	Descrição
Japanese_XJIS_100_CS_AI	Japanese-XJIS-100, diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, não diferencia a largura
Japanese_XJIS_100_CS_AI_KS	Japanese-XJIS-100, diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, não diferencia a largura
Japanese_XJIS_100_CS_AI_KS_SC	Japanese-XJIS-100, diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, não diferencia a largura, caracteres complementares
Japanese_XJIS_100_CS_AI_KS_SC_UTF8	Japanese-XJIS-100, diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, não diferencia a largura, caracteres complementares, UTF8
Japanese_XJIS_100_CS_AI_KS_WS	Japanese-XJIS-100, diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, diferencia a largura
Japanese_XJIS_100_CS_AI_KS_WS_SC	Japanese-XJIS-100, diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, diferencia a largura, caracteres complementares
Japanese_XJIS_100_CS_AI_KS_WS_SC_UTF8	Japanese-XJIS-100, diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, diferencia a largura, caracteres complementares, UTF8

Collation (Agrupamento)	Descrição
Japanese_XJIS_100_CS_AI_SC	Japanese-XJIS-100, diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, não diferencia a largura, caracteres complementares
Japanese_XJIS_100_CS_AI_SC_UTF8	Japanese-XJIS-100, diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, não diferencia a largura, caracteres complementares, UTF8
Japanese_XJIS_100_CS_AI_WS	Japanese-XJIS-100, diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, diferencia a largura
Japanese_XJIS_100_CS_AI_WS_SC	Japanese-XJIS-100, diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, diferencia a largura, caracteres complementares
Japanese_XJIS_100_CS_AI_WS_SC_UTF8	Japanese-XJIS-100, diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, diferencia a largura, caracteres complementares, UTF8
Japanese_XJIS_100_CS_AS	Japanese-XJIS-100, diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura
Japanese_XJIS_100_CS_AS_KS	Japanese-XJIS-100, diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, não diferencia a largura

Collation (Agrupamento)	Descrição
Japanese_XJIS_100_CS_AS_KS_SC	Japanese-XJIS-100, diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, não diferencia a largura, caracteres complementares
Japanese_XJIS_100_CS_AS_KS_SC_UTF8	Japanese-XJIS-100, diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, não diferencia a largura, caracteres complementares, UTF8
Japanese_XJIS_100_CS_AS_KS_WS	Japanese-XJIS-100, diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, diferencia a largura
Japanese_XJIS_100_CS_AS_KS_WS_SC	Japanese-XJIS-100, diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, diferencia a largura, caracteres complementares
Japanese_XJIS_100_CS_AS_KS_WS_SC_UTF8	Japanese-XJIS-100, diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, diferencia a largura, caracteres complementares, UTF8
Japanese_XJIS_100_CS_AS_SC	Japanese-XJIS-100, diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura, caracteres complementares
Japanese_XJIS_100_CS_AS_SC_UTF8	Japanese-XJIS-100, diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura, caracteres complementares, UTF8

Collation (Agrupamento)	Descrição
Japanese_XJIS_100_CS_AS_WS	Japanese-XJIS-100, diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, diferencia a largura
Japanese_XJIS_140_BIN	Japanese-XJIS-140, classificação binária
Japanese_XJIS_140_BIN2	Japanese-XJIS-140, classificação binária de comparação de ponto de código
Japanese_XJIS_140_CI_AI	Japanese-XJIS-140, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, não diferencia a largura, caracteres complementares, não diferencia o seletor de variação
Japanese_XJIS_140_CI_AI_KS	Japanese-XJIS-140, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, não diferencia a largura, caracteres complementares, não diferencia o seletor de variação
Japanese_XJIS_140_CI_AI_KS_UTF8	Japanese-XJIS-140, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, não diferencia a largura, caracteres complementares, não diferencia o seletor de variação, UTF8
Japanese_XJIS_140_CI_AI_KS_VSS	Japanese-XJIS-140, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, não diferencia a largura, caracteres complementares, diferencia o seletor de variação

Collation (Agrupamento)	Descrição
Japanese_XJIS_140_CI_AI_KS_VSS_UTF8	Japanese-XJIS-140, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, não diferenci a a largura, caracteres complementares, diferencia o seletor de variação, UTF8
Japanese_XJIS_140_CI_AI_KS_WS	Japanese-XJIS-140, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, diferenci a a largura, caracteres complementares, não diferencia o seletor de variação
Japanese_XJIS_140_CI_AI_KS_WS_UTF8	Japanese-XJIS-140, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, diferenci a a largura, caracteres complementares, não diferencia o seletor de variação, UTF8
Japanese_XJIS_140_CI_AI_KS_WS_VSS	Japanese-XJIS-140, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, diferencia a largura, caracteres complementares, diferencia o seletor de variação
Japanese_XJIS_140_CI_AI_KS_WS_VSS_UTF8	Japanese-XJIS-140, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, diferencia a largura, caracteres complementares, diferencia o seletor de variação, UTF8
Japanese_XJIS_140_CI_AI_UTF8	Japanese-XJIS-140, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, não diferencia a largura, caracteres complemen tares, não diferencia o seletor de variação, UTF8

Collation (Agrupamento)	Descrição
Japanese_XJIS_140_CI_AI_VSS	Japanese-XJIS-140, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, não diferencia a largura, caracteres complemen tares, diferencia o seletor de variação
Japanese_XJIS_140_CI_AI_VSS_UTF8	Japanese-XJIS-140, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, não diferencia a largura, caracteres complemen tares, diferencia o seletor de variação, UTF8
Japanese_XJIS_140_CI_AI_WS	Japanese-XJIS-140, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, diferenci a a largura, caracteres complementares, não diferencia o seletor de variação
Japanese_XJIS_140_CI_AI_WS_UTF8	Japanese-XJIS-140, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, diferenci a a largura, caracteres complementares, não diferencia o seletor de variação, UTF8
Japanese_XJIS_140_CI_AI_WS_VSS	Japanese-XJIS-140, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, diferenci a a largura, caracteres complementares, diferencia o seletor de variação
Japanese_XJIS_140_CI_AI_WS_VSS_UTF8	Japanese-XJIS-140, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, diferenci a a largura, caracteres complementares, diferencia o seletor de variação, UTF8

Collation (Agrupamento)	Descrição
Japanese_XJIS_140_CI_AS	Japanese-XJIS-140, não diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferenci a a largura, caracteres complementares, não diferencia o seletor de variação
Japanese_XJIS_140_CI_AS_KS	Japanese-XJIS-140, não diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, não diferencia a largura, caracteres complementares, não diferencia o seletor de variação
Japanese_XJIS_140_CI_AS_KS_UTF8	Japanese-XJIS-140, não diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, não diferencia a largura, caracteres complementares, não diferencia o seletor de variação, UTF8
Japanese_XJIS_140_CI_AS_KS_VSS	Japanese-XJIS-140, não diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, não diferencia a largura, caracteres complementares, diferencia o seletor de variação
Japanese_XJIS_140_CI_AS_KS_VSS_UTF8	Japanese-XJIS-140, não diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, não diferencia a largura, caracteres complementares, diferencia o seletor de variação, UTF8
Japanese_XJIS_140_CI_AS_KS_WS	Japanese-XJIS-140, não diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, diferencia a largura, caracteres complementares, não diferencia o seletor de variação

Collation (Agrupamento)	Descrição
Japanese_XJIS_140_CI_AS_KS_WS_UTF8	Japanese-XJIS-140, não diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, diferencia a largura, caracteres complementares, não diferencia o seletor de variação, UTF8
Japanese_XJIS_140_CI_AS_KS_WS_VSS	Japanese-XJIS-140, não diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, diferencia a largura, caracteres complementares, diferencia o seletor de variação
Japanese_XJIS_140_CI_AS_KS_WS_VSS_UTF8	Japanese-XJIS-140, não diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, diferencia a largura, caracteres complementares, diferencia o seletor de variação, UTF8
Japanese_XJIS_140_CI_AS_UTF8	Japanese-XJIS-140, não diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferenci a a largura, caracteres complementares, não diferencia o seletor de variação, UTF8
Japanese_XJIS_140_CI_AS_VSS	Japanese-XJIS-140, não diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura, caracteres complementares, diferencia o seletor de variação
Japanese_XJIS_140_CI_AS_VSS_UTF8	Japanese-XJIS-140, não diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura, caracteres complementares, diferencia o seletor de variação, UTF8

Collation (Agrupamento)	Descrição
Japanese_XJIS_140_CI_AS_WS	Japanese-XJIS-140, não diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, diferencia a largura, caracteres complementares, não diferencia o seletor de variação
Japanese_XJIS_140_CI_AS_WS_UTF8	Japanese-XJIS-140, não diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, diferencia a largura, caracteres complementares, não diferencia o seletor de variação, UTF8
Japanese_XJIS_140_CI_AS_WS_VSS	Japanese-XJIS-140, não diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, diferencia a largura, caracteres complementares, diferencia o seletor de variação
Japanese_XJIS_140_CI_AS_WS_VSS_UTF8	Japanese-XJIS-140, não diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, diferencia a largura, caracteres complementares, diferencia o seletor de variação, UTF8
Japanese_XJIS_140_CS_AI	Japanese-XJIS-140, diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, não diferencia a largura, caracteres complementares, não diferencia o seletor de variação
Japanese_XJIS_140_CS_AI_KS	Japanese-XJIS-140, diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, não diferencia a largura, caracteres complementares, não diferencia o seletor de variação

Collation (Agrupamento)	Descrição
Japanese_XJIS_140_CS_AI_KS_UTF8	Japanese-XJIS-140, diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, não diferencia a largura, caracteres complementares, não diferencia o seletor de variação, UTF8
Japanese_XJIS_140_CS_AI_KS_VSS	Japanese-XJIS-140, diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, não diferencia a largura, caracteres complementares, diferencia o seletor de variação
Japanese_XJIS_140_CS_AI_KS_VSS_UTF8	Japanese-XJIS-140, diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, não diferencia a largura, caracteres complementares, diferencia o seletor de variação, UTF8
Japanese_XJIS_140_CS_AI_KS_WS	Japanese-XJIS-140, diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, diferencia a largura, caracteres complementares, não diferencia o seletor de variação
Japanese_XJIS_140_CS_AI_KS_WS_UTF8	Japanese-XJIS-140, diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, diferencia a largura, caracteres complementares, não diferencia o seletor de variação, UTF8
Japanese_XJIS_140_CS_AI_KS_WS_VSS	Japanese-XJIS-140, diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, diferencia a largura, caracteres complementares, diferencia o seletor de variação

Collation (Agrupamento)	Descrição
Japanese_XJIS_140_CS_AI_KS_WS_VSS_UTF8	Japanese-XJIS-140, diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, diferencia a largura, caracteres complementares, diferencia o seletor de variação, UTF8
Japanese_XJIS_140_CS_AI_UTF8	Japanese-XJIS-140, diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, não diferencia a largura, caracteres complementares, não diferencia o seletor de variação, UTF8
Japanese_XJIS_140_CS_AI_VSS	Japanese-XJIS-140, diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, não diferencia a largura, caracteres complementares, diferencia o seletor de variação
Japanese_XJIS_140_CS_AI_VSS_UTF8	Japanese-XJIS-140, diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, não diferencia a largura, caracteres complementares, diferencia o seletor de variação, UTF8
Japanese_XJIS_140_CS_AI_WS	Japanese-XJIS-140, diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, diferencia a largura, caracteres complementares, não diferencia o seletor de variação
Japanese_XJIS_140_CS_AI_WS_UTF8	Japanese-XJIS-140, diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, diferencia a largura, caracteres complementares, não diferencia o seletor de variação, UTF8

Collation (Agrupamento)	Descrição
Japanese_XJIS_140_CS_AI_WS_VSS	Japanese-XJIS-140, diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, diferencia a largura, caracteres complementares, diferencia o seletor de variação
Japanese_XJIS_140_CS_AI_WS_VSS_UTF8	Japanese-XJIS-140, diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, diferencia a largura, caracteres complementares, diferencia o seletor de variação, UTF8
Japanese_XJIS_140_CS_AS	Japanese-XJIS-140, diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura, caracteres complementares, não diferencia o seletor de variação
Japanese_XJIS_140_CS_AS_KS	Japanese-XJIS-140, diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, não diferencia a largura, caracteres complementares, não diferencia o seletor de variação
Japanese_XJIS_140_CS_AS_KS_UTF8	Japanese-XJIS-140, diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, não diferencia a largura, caracteres complementares, não diferencia o seletor de variação, UTF8
Japanese_XJIS_140_CS_AS_KS_VSS	Japanese-XJIS-140, diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, não diferencia a largura, caracteres complementares, diferencia o seletor de variação

Collation (Agrupamento)	Descrição
Japanese_XJIS_140_CS_AS_KS_VSS_UTF8	Japanese-XJIS-140, diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, não diferencia a largura, caracteres complementares, diferencia o seletor de variação, UTF8
japonês_XJIS_140_CS_AS_KS_WS	Japanese-XJIS-140, diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, diferencia a largura, caracteres complementares, não diferencia o seletor de variação
Japanese_XJIS_140_CS_AS_KS_WS_UTF8	Japanese-XJIS-140, diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, diferencia a largura, caracteres complementares, não diferencia o seletor de variação, UTF8
Japanese_XJIS_140_CS_AS_KS_WS_VSS	Japanese-XJIS-140, diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, diferencia a largura, caracteres complementares, diferencia o seletor de variação
Japanese_XJIS_140_CS_AS_KS_WS_VSS_UTF8	Japanese-XJIS-140, diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, diferencia a largura, caracteres complementares, diferencia o seletor de variação, UTF8
Japanese_XJIS_140_CS_AS_UTF8	Japanese-XJIS-140, diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura, caracteres complementares, não diferencia o seletor de variação, UTF8

Collation (Agrupamento)	Descrição
Japanese_XJIS_140_CS_AS_VSS	Japanese-XJIS-140, diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura, caracteres complementares, diferencia o seletor de variação
Japanese_XJIS_140_CS_AS_VSS_UTF8	Japanese-XJIS-140, diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura, caracteres complementares, diferencia o seletor de variação, UTF8
Japanese_XJIS_140_CS_AS_WS	Japanese-XJIS-140, diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, diferencia a largura, caracteres complementares, não diferencia o seletor de variação
Japanese_XJIS_140_CS_AS_WS_UTF8	Japanese-XJIS-140, diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, diferencia a largura, caracteres complementares, não diferencia o seletor de variação, UTF8
Japanese_XJIS_140_CS_AS_WS_VSS	Japanese-XJIS-140, diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, diferencia a largura, caracteres complementares, diferencia o seletor de variação
Japanese_XJIS_140_CS_AS_WS_VSS_UTF8	Japanese-XJIS-140, diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, diferencia a largura, caracteres complementares, diferencia o seletor de variação, UTF8

Collation (Agrupamento)	Descrição
Korean_Wansung_CI_AS	Korean-Wansung, não diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura
Latin1_General_100_BIN	Latin1-General-100, classificação binária
Latin1_General_100_BIN2	Latin1-General-100, classificação binária de comparação de ponto de código
Latin1_General_100_BIN2_UTF8	Latin1-General-100, classificação binária de comparação de ponto de código, UTF8
Latin1_General_100_CI_AS	Latin1-General-100, não diferencia maiúscula s de minúsculas, diferenciação de acentos, indiferente ao tipo de kana, indiferente à largura
Latin1_General_100_CI_AS_SC_UTF8	Latin1-General-100, não diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura, caracteres complementares, UTF8
Latin1_General_BIN	Latin1-General, classificação binária
Latin1_General_BIN2	Latin1-General, classificação binária de comparação de ponto de código
Latin1_General_CI_AI	Latin1-General, não diferencia letras maiúscula s de minúsculas, não diferencia acentos, não diferencia o tipo de kana, não diferencia a largura
Latin1_General_CI_AS	Latin1-General, não diferencia letras maiúscula s de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura

Collation (Agrupamento)	Descrição
Latin1_General_CI_AS_KS	Latin1-General, não diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, não diferencia a largura
Latin1_General_CS_AS	Latin1-Geral, diferencia maiúsculas de minúsculas, diferenciação de acentos, indiferente ao tipo de kana, indiferente à largura
Modern_Spanish_CI_AS	Modern-Spanish, não diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura
SQL_1xCompat_CP850_CI_AS	Latin1-Geral, não diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura para dados do Unicode, ordem de classificação do SQL Server 49 na página de código 850 para dados que não são do Unicode
SQL_Latin1_General_CP1_CI_AI	Latin1-Geral, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, não diferencia a largura para dados do Unicode, ordem de classificação do SQL Server 54 na página de código 1252 para dados que não são do Unicode
SQL_Latin1_General_CP1_CI_AS	Latin1-Geral, não diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura para dados do Unicode, ordem de classificação do SQL Server 52 na página de código 1252 para dados que não são do Unicode

Collation (Agrupamento)	Descrição
SQL_Latin1_General_CP1_CS_AS	Latin1-Geral, diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferenci a o tipo de kana, não diferencia a largura para dados do Unicode, ordem de classificação do SQL Server 51 na página de código 1252 para dados que não são do Unicode
SQL_Latin1_General_CP1250_CI_AS	Latin1-Geral, não diferencia letras maiúscula s de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura para dados do Unicode, ordem de classificação do SQL Server 82 na página de código 1250 para dados que não são do Unicode
SQL_Latin1_General_CP1250_CS_AS	Latin1-Geral, diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferenci a o tipo de kana, não diferencia a largura para dados do Unicode, ordem de classificação do SQL Server 81 na página de código 1250 para dados que não são do Unicode
SQL_Latin1_General_CP1251_CI_AS	Latin1-Geral, não diferencia letras maiúscula s de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura para dados do Unicode, ordem de classificação do SQL Server 106 na página de código 1251 para dados que não são do Unicode

Collation (Agrupamento)	Descrição
SQL_Latin1_General_CP1251_CS_AS	Latin1-Geral, diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferenci a o tipo de kana, não diferencia a largura para dados do Unicode, ordem de classificação do SQL Server 105 na página de código 1251 para dados que não são do Unicode
SQL_Latin1_General_CP1253_CI_AI	Latin1-Geral, não diferencia letras maiúscula s de minúsculas, não diferencia acentos, não diferencia o tipo de kana, não diferencia a largura para dados do Unicode, ordem de classificação do SQL Server 124 na página de código 1253 para dados que não são do Unicode
SQL_Latin1_General_CP1253_CI_AS	Latin1-Geral, não diferencia letras maiúscula s de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura para dados do Unicode, ordem de classificação do SQL Server 114 na página de código 1253 para dados que não são do Unicode
SQL_Latin1_General_CP1253_CS_AS	Latin1-Geral, diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferenci a o tipo de kana, não diferencia a largura para dados do Unicode, ordem de classificação do SQL Server 113 na página de código 1253 para dados que não são do Unicode

Collation (Agrupamento)	Descrição
SQL_Latin1_General_CP1254_CI_AS	Turkish, não diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferenci a o tipo de kana, não diferencia a largura para dados do Unicode, ordem de classificação do SQL Server 130 na página de código 1254 para dados que não são do Unicode
SQL_Latin1_General_CP1254_CS_AS	Turkish, diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferenci a o tipo de kana, não diferencia a largura para dados do Unicode, ordem de classificação do SQL Server 129 na página de código 1254 para dados que não são do Unicode
SQL_Latin1_General_CP1255_CI_AS	Latin1-Geral, não diferencia letras maiúscula s de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura para dados do Unicode, ordem de classificação do SQL Server 138 na página de código 1255 para dados que não são do Unicode
SQL_Latin1_General_CP1255_CS_AS	Latin1-Geral, diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferenci a o tipo de kana, não diferencia a largura para dados do Unicode, ordem de classificação do SQL Server 137 na página de código 1255 para dados que não são do Unicode

Collation (Agrupamento)	Descrição
SQL_Latin1_General_CP1256_CI_AS	Latin1-Geral, não diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura para dados do Unicode, ordem de classificação do SQL Server 146 na página de código 1256 para dados que não são do Unicode
SQL_Latin1_General_CP1256_CS_AS	Latin1-Geral, diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura para dados do Unicode, ordem de classificação do SQL Server 145 na página de código 1256 para dados que não são do Unicode
SQL_Latin1_General_CP1257_CI_AS	Latin1-Geral, não diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura para dados do Unicode, ordem de classificação do SQL Server 154 na página de código 1257 para dados que não são do Unicode
SQL_Latin1_General_CP1257_CS_AS	Latin1-Geral, diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura para dados do Unicode, ordem de classificação do SQL Server 153 na página de código 1257 para dados que não são do Unicode
SQL_Latin1_General_CP437_BIN	Latin1-General, classificação binária para dados do Unicode, ordem de classificação do SQL Server 30 na página de código 437 para dados que não são do Unicode

Collation (Agrupamento)	Descrição
SQL_Latin1_General_CP437_BIN2	Latin1-Geral, classificação binária de comparação de ponto de código para dados do Unicode, ordem de classificação do SQL Server 30 na página de código 437 para dados que não são do Unicode
SQL_Latin1_General_CP437_CI_AI	Latin1-Geral, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, não diferencia a largura para dados do Unicode, ordem de classificação do SQL Server 34 na página de código 437 para dados que não são do Unicode
SQL_Latin1_General_CP437_CI_AS	Latin1-Geral, não diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura para dados do Unicode, ordem de classificação do SQL Server 32 na página de código 437 para dados que não são do Unicode
SQL_Latin1_General_CP437_CS_AS	Latin1-Geral, diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura para dados do Unicode, ordem de classificação do SQL Server 31 na página de código 437 para dados que não são do Unicode
SQL_Latin1_General_CP850_bin	Latin1-General, classificação binária para dados do Unicode, ordem de classificação do SQL Server 40 na página de código 850 para dados que não são do Unicode

Collation (Agrupamento)	Descrição
SQL_Latin1_General_CP850_BIN2	Latin1-Geral, classificação binária de comparação de ponto de código para dados do Unicode, ordem de classificação do SQL Server 40 na página de código 850 para dados que não são do Unicode
SQL_Latin1_General_CP850_CI_AI	Latin1-Geral, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, não diferencia a largura para dados do Unicode, ordem de classificação do SQL Server 44 na página de código 850 para dados que não são do Unicode
SQL_Latin1_General_CP850_CI_AS	Latin1-Geral, não diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura para dados do Unicode, ordem de classificação do SQL Server 42 na página de código 850 para dados que não são do Unicode
SQL_Latin1_General_CP850_CS_AS	Latin1-Geral, diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura para dados do Unicode, ordem de classificação do SQL Server 41 na página de código 850 para dados que não são do Unicode

Collation (Agrupamento)	Descrição
SQL_Latin1_General_Pref_CP1_CI_AS	Latin1-Geral, não diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura para dados do Unicode, ordem de classificação do SQL Server 53 na página de código 1252 para dados que não são do Unicode
SQL_Latin1_General_Pref_CP437_CI_AS	Latin1-Geral, não diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura para dados do Unicode, ordem de classificação do SQL Server 33 na página de código 437 para dados que não são do Unicode
SQL_Latin1_General_Pref_CP850_CI_AS	Latin1-Geral, não diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura para dados do Unicode, ordem de classificação do SQL Server 43 na página de código 850 para dados que não são do Unicode
Thai_CI_AS	Thai, não diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura

Fuso horário local para instâncias de banco de dados do RDS Custom para SQL Server

O fuso horário de uma instância de banco de dados do RDS Custom para SQL Server é definido por padrão. O padrão atual é Universal Coordinated Time (UTC). Você pode definir o fuso horário de sua instância de banco de dados como um fuso horário local, para combinar o fuso horário de seus aplicativos.

Você define o fuso horário ao criar sua instância de banco de dados. É possível criar a instância de banco de dados usando o AWS Management Console, a ação CreateDBInstance da API do Amazon RDS ou o comando create-db-instance da AWS CLI.

Se a a instância de banco de dados fizer parte de uma implantação multi-AZ, durante o failover, seu fuso horário permanecerá como o fuso horário local que você definiu.

Ao solicitar uma restauração point-in-time, especifique a hora da restauração. A hora é mostrada no fuso horário local. Para obter mais informações, consulte Restaurar uma instância de banco de dados para um momento especificado no Amazon RDS.

As limitações a seguir se aplicam ao definir o fuso horário local em sua instância de banco de dados:

- Você pode configurar o fuso horário de uma instância de banco de dados durante a criação da instância, mas não pode modificar o fuso horário de uma instância de banco de dados existente do RDS Custom para SQL Server.
- Se o fuso horário for modificado para uma instância de banco de dados existente do RDS Custom para SQL Server, o RDS Custom alterará o status da instância de banco de dados para `unsupported-configuration` e enviará notificações de eventos.
- Não é possível restaurar um snapshot de uma instância de banco de dados em um fuso horário para uma instância de banco de dados em um fuso horário diferente.
- Recomendamos que não restaure um arquivo de backup de um fuso horário em um fuso horário diferente. Se você restaurar um arquivo de backup a partir de um fuso horário para um fuso horário diferente, é preciso auditar as consultas e aplicativos quanto à alteração de fuso horário. Para obter mais informações, consulte Importar e exportar bancos de dados do SQL Server usando backup e restauração nativos.

Fusos horários compatíveis

Você pode definir seu fuso horário local como um dos valores listados na seguinte tabela.

Fuso horário	Compensação de tempo padrão	Descrição	Observações
Hora Padrão do Afeganistão	(UTC+04:30)	Kabul	Este fuso horário não observa o horário de verão.

Fuso horário	Compensação de tempo padrão	Descrição	Observações
Hora Padrão do Alasca	(UTC–09:00)	Alasca	
Hora Padrão Aleutiano	(UTC–10:00)	Ilhas Aleutas	
Hora Padrão de Altai	(UTC+07:00)	Barnaul, Gorno-Altaysk	
Hora Padrão Árabe	(UTC+03:00)	Kuwait, Riade	Este fuso horário não observa o horário de verão.
Hora Padrão da Arábia	(UTC+04:00)	Abu Dhabi, Muscat	
Hora Padrão Árabe	(UTC+03:00)	Bagdá	Este fuso horário não observa o horário de verão.
Hora Padrão da Argentina	(UTC–03:00)	Cidade de Buenos Aires	Este fuso horário não observa o horário de verão.
Hora Padrão de Astracã	(UTC+04:00)	Astracã, Ulianovsk	
Hora Padrão do Atlântico	(UTC–04:00)	Horário do Atlântico (Canadá)	
Hora Padrão Central da Austrália	(UTC+09:30)	Darwin	Este fuso horário não observa o horário de verão.
Hora Padrão Central da Austrália	(UTC+ 08:45)	Eucla	
Hora Padrão do Leste da Austrália	(UTC+10:00)	Canberra, Melbourne, Sydney	
Hora Padrão do Azerbaijão	(UTC+04:00)	Bacu	

Fuso horário	Compensação de tempo padrão	Descrição	Observações
Hora Padrão dos Açores	(UTC–01:00)	Açores	
Hora Padrão da Bahia	(UTC–03:00)	Salvador	
Hora Padrão de Bangladesh	(UTC+06:00)	Daca	Este fuso horário não observa o horário de verão.
Hora Padrão da Bielorrússia	(UTC+03:00)	Minsk	Este fuso horário não observa o horário de verão.
Fuso hora padrão de Bougainville	(UTC+11:00)	Ilha de Bougainville	
Hora Padrão Central do Canadá	(UTC–06:00)	Saskatchewan	Este fuso horário não observa o horário de verão.
Hora Padrão de Cabo Verde	(UTC–01:00)	Ilhas de Cabo Verde	Este fuso horário não observa o horário de verão.
Hora Padrão do Cáucaso	(UTC+04:00)	Erevan	
Hora Padrão da Austrália Central	(UTC+09:30)	Adelaide	
Hora Padrão da América Central	(UTC–06:00)	América Central	Este fuso horário não observa o horário de verão.
Hora Padrão da Ásia Central	(UTC+06:00)	Astana	Este fuso horário não observa o horário de verão.

Fuso horário	Compensação de tempo padrão	Descrição	Observações
Hora Padrão Central do Brasil	(UTC–04:00)	Cuiabá	
Hora Padrão Central da Europa	(UTC+01:00)	Belgrado, Bratislava, Budapeste, Ljubljana, Praga	
Hora Oficial da Europa Central	(UTC+01:00)	Sarajevo, Skopje, Varsóvia, Zagreb	
Hora Padrão Central do Pacífico	(UTC+11:00)	Ilhas Salomão, Nova Caledônia	Este fuso horário não observa o horário de verão.
Hora Padrão Central	(UTC–06:00)	Horário Central (EUA e Canadá)	
Hora Padrão Central (México)	(UTC–06:00)	Guadalajara, Cidade do México, Monterrey	
Fuso horário padrão das Ilhas Chatham	(UTC+ 12:45)	Ilhas Chatham	
Hora Padrão da China	(UTC+08:00)	Pequim, Chongqing, Hong Kong, Urumqi	Este fuso horário não observa o horário de verão.
Hora Padrão de Cuba	(UTC–05:00)	Havana	
Hora Padrão da Linha de Data	(UTC–12:00)	Linha Internacional de Data Oeste	Este fuso horário não observa o horário de verão.
Horário padrão da África Oriental	(UTC+03:00)	Nairóbi	Este fuso horário não observa o horário de verão.

Fuso horário	Compensação de tempo padrão	Descrição	Observações
Horário padrão da Austrália Oriental	(UTC+10:00)	Brisbane	Este fuso horário não observa o horário de verão.
Horário padrão da Europa Oriental	(UTC+02:00)	Chisinau	
Horário padrão da América do Sul Oriental	(UTC–03:00)	Brasília	
Hora Padrão da Ilha de Páscoa	(UTC–06:00)	Ilha de Páscoa	
Hora Oficial do Leste dos EUA	(UTC–05:00)	Hora do Leste (EUA e Canadá)	
Hora Padrão do Leste (México)	(UTC–05:00)	Chetumal	
Hora Padrão do Egito	(UTC+02:00)	Cairo	
Hora Padrão de Ekaterinburg	(UTC+ 05:00)	Ekaterinburg	
Hora Padrão de Fiji	(UTC+12:00)	Fiji	
Hora Padrão de FLE	(UTC+02:00)	Helsinki, Kiev, Riga, Sofia, Talin, Vilnius	
Hora Oficial da Geórgia	(UTC+04:00)	Tbilisi	Este fuso horário não observa o horário de verão.

Fuso horário	Compensação de tempo padrão	Descrição	Observações
Hora Padrão de Greenwich	(UTC)	Dublin, Edimburgo, Lisboa, Londres	Esse fuso horário não é igual ao Horário do Meridiano de Greenwich. Esse fuso horário segue o horário de verão.
Hora Oficial da Groenlândia	(UTC–03:00)	Groenlândia	
Hora Oficial de Greenwich	(UTC)	Monróvia, Reykjavik	Este fuso horário não observa o horário de verão.
Hora Oficial GTB	(UTC+02:00)	Atenas, Bucareste	
Hora Padrão do Haiti	(UTC–05:00)	Haiti	
Hora Oficial do Havaí	(UTC–10:00)	Havaí	
Hora Oficial da Índia	(UTC+05:30)	Chennai, Kolkata, Mumbai, Nova Deli	Este fuso horário não observa o horário de verão.
Hora Padrão do Irã	(UTC+ 03:30)	Teerão	
Hora Padrão de Israel	(UTC+02:00)	Jerusalém	
Hora Padrão da Jordânia	(UTC+02:00)	Amã	
Hora Padrão de Kaliningrado	(UTC+02:00)	Kaliningrado	
Hora Padrão de Kamchatka	(UTC+12:00)	Petropavlovsk-Kamchatsky – antigo	

Fuso horário	Compensação de tempo padrão	Descrição	Observações
Hora Padrão da Coreia	(UTC+09:00)	Seul	Este fuso horário não observa o horário de verão.
Hora Padrão da Líbia	(UTC+02:00)	Trípoli	
Hora Padrão das Ilhas Line	(UTC+ 14:00)	Ilha Kiritimati	
Hora Padrão de Lord Howe	(UTC+ 10:30)	Ilha de Lord Howe	
Hora Padrão de Magadan	(UTC+11:00)	Magadan	Este fuso horário não observa o horário de verão.
Hora Padrão de Magallanes	(UTC–03:00)	Punta Arenas	
Hora Padrão do Marquesas	(UTC–09:30)	Ilhas Marquesas	
Hora Padrão das Ilhas Maurício	(UTC+04:00)	Port Louis	Este fuso horário não observa o horário de verão.
Hora Padrão do Oriente Médio	(UTC+02:00)	Beirute	
Hora Padrão de Montevidéu	(UTC–03:00)	Montevidéu	
Hora Padrão de Marrocos	(UTC+01:00)	Casablanca	
Hora Oficial das Montanhas	(UTC–07:00)	Horário das Montanhas (EUA e Canadá)	
Hora Oficial das Montanhas (México)	(UTC–07:00)	Chihuahua, La Paz, Mazatlan	

Fuso horário	Compensação de tempo padrão	Descrição	Observações
Hora Padrão de Mianmar	(UTC+ 06:30)	Rangum (Rangum)	Este fuso horário não observa o horário de verão.
Horário padrão do Norte da Ásia do Central	(UTC+07:00)	Novosibirsk	
Hora Padrão da Namíbia	(UTC+02:00)	Windhoek	
Hora Padrão do Nepal	(UTC+ 05:45)	Katmandu	Este fuso horário não observa o horário de verão.
Hora Oficial da Nova Zelândia	(UTC+12:00)	Auckland, Wellington	
Hora Oficial de Newfoundland	(UTC–03:30)	Newfoundland	
Hora Padrão de Norfolk	(UTC+11:00)	Ilha Norfolk	
Hora Padrão do Nordeste da Ásia	(UTC+08:00)	Irkutsk	
Hora Padrão do Norte da Ásia	(UTC+07:00)	Krasnoyarsk	
Hora Padrão da Coreia do Norte	(UTC+09:00)	Pyongyang	
Hora Padrão de Omsk	(UTC+06:00)	Omsk	
Hora Padrão do Pacífico SA	(UTC–03:00)	Santiago	
Hora Oficial do Pacífico	(UTC–08:00)	Horário do Pacífico (EUA e Canadá)	
Hora Padrão Pacífico (México)	(UTC–08:00)	Baixa Califórnia	

Fuso horário	Compensação de tempo padrão	Descrição	Observações
Hora Padrão do Paquistão	(UTC+ 05:00)	Islamabad	Este fuso horário não observa o horário de verão.
Hora Padrão do Paraguai	(UTC–04:00)	Assunção	
Hora Padrão de Romance	(UTC+01:00)	Bruxelas, Copenhagen, Madri, Paris	
Fuso Horário da Rússia 10	(UTC+11:00)	Chokurdakh	
Fuso Horário da Rússia 11	(UTC+12:00)	Anadyr, Petropavl ovsk-Kamchatsky	
Fuso Horário da Rússia 3	(UTC+04:00)	Izhevsk, Samara	
Hora Padrão da Rússia	(UTC+03:00)	Moscou, São Petersburgo, Volgogrado	Este fuso horário não observa o horário de verão.
Hora Padrão do Leste de SA	(UTC–03:00)	Cayenne, Fortaleza	Este fuso horário não observa o horário de verão.
Hora Padrão do Pacífico na América do Sul	(UTC–05:00)	Bogotá, Lima, Quito, Rio Branco	Este fuso horário não observa o horário de verão.
Hora Padrão de SA Ocidental	(UTC–04:00)	Georgetown, La Paz, Manaus, San Juan	Este fuso horário não observa o horário de verão.
Hora Padrão de Saint Pierre	(UTC–03:00)	Saint Pierre e Miquelon	

Fuso horário	Compensação de tempo padrão	Descrição	Observações
Hora Padrão de Sacalina	(UTC+11:00)	Sacalina	
Hora Padrão de Samoa	(UTC+ 13:00)	Samoa	
Hora Padrão de São Tomé	(UTC+01:00)	São Tomé	
Hora Padrão de Saratov	(UTC+04:00)	Saratov	
Hora Padrão do Sudeste da Ásia	(UTC+07:00)	Bangkok, Hanoi, Jakarta	Este fuso horário não observa o horário de verão.
Hora Padrão de Cingapura	(UTC+08:00)	Kuala Lumpur, Cingapura	Este fuso horário não observa o horário de verão.
Hora Padrão da África do Sul	(UTC+02:00)	Harare, Pretória	Este fuso horário não observa o horário de verão.
Hora Padrão do Sri Lanka	(UTC+05:30)	Sri Jayawarde nepura	Este fuso horário não observa o horário de verão.
Hora Padrão do Sudão	(UTC+02:00)	Cartum	
Hora Padrão da Síria	(UTC+02:00)	Damasco	
Hora padrão de Taipei	(UTC+08:00)	Taipei	Este fuso horário não observa o horário de verão.
Hora Padrão da Tasmânia	(UTC+10:00)	Hobart	
Hora Padrão do Tocantins	(UTC–03:00)	Araguaina	

Fuso horário	Compensação de tempo padrão	Descrição	Observações
Hora Padrão de Tóquio	(UTC+09:00)	Osaka, Sapporo, Tóquio	Este fuso horário não observa o horário de verão.
Hora Padrão de Tomsk	(UTC+07:00)	Tomsk	
Hora Padrão de Tonga	(UTC+ 13:00)	Nuku'alofa	Este fuso horário não observa o horário de verão.
Hora Padrão do Transbaikal	(UTC+09:00)	Chita	
Hora Padrão da Turquia	(UTC+03:00)	Istambul	
Hora Padrão de Turks e Caicos	(UTC–05:00)	Turks e Caicos	
Hora Padrão de Ulaanbaatar	(UTC+08:00)	Ulaanbaatar	Este fuso horário não observa o horário de verão.
Hora Padrão do Leste dos EUA	(UTC–05:00)	Indiana (Leste)	
Hora Oficial das Montanhas (EUA)	(UTC–07:00)	Arizona	Este fuso horário não observa o horário de verão.
UTC	UTC	Tempo Universal Coordenado	Este fuso horário não observa o horário de verão.
UTC–02	(UTC–02:00)	Tempo Universal Coordenado–02	Este fuso horário não observa o horário de verão.

Fuso horário	Compensação de tempo padrão	Descrição	Observações
UTC–08	(UTC–08:00)	Tempo Universal Coordenado–08	
UTC–09	(UTC–09:00)	Tempo Universal Coordenado–09	
UTC–11	(UTC–11:00)	Tempo Universal Coordenado–11	Este fuso horário não observa o horário de verão.
UTC+12	(UTC+12:00)	Tempo Universal Coordenado+12	Este fuso horário não observa o horário de verão.
UTC+13	(UTC+ 13:00)	Tempo Universal Coordenado+13	
Hora Padrão da Venezuela	(UTC–04:00)	Caracas	Este fuso horário não observa o horário de verão.
Hora Padrão de Vladivostok	(UTC+10:00)	Vladivostok	
Hora Padrão de Volgograd	(UTC+04:00)	Volgograd	
Horário padrão da Austrália Ocidental Ocidental	(UTC+08:00)	Perth	Este fuso horário não observa o horário de verão.
Horário padrão da África Central Ocidental	(UTC+01:00)	Centro-oeste da África	Este fuso horário não observa o horário de verão.
Horário padrão da Europa Ocidental	(UTC+01:00)	Amsterdã, Berlim, Berna, Roma, Estocolmo, Viena	

Fuso horário	Compensação de tempo padrão	Descrição	Observações
Horário Padrão da Mongólia Ocidental	(UTC+07:00)	Hovd	
Hora Padrão do Oeste da Ásia	(UTC+ 05:00)	Ashgabat, Tashkent	Este fuso horário não observa o horário de verão.
Hora Padrão da Cisjordânia	(UTC+02:00)	Gaza, Hebron	
Hora Padrão do Oeste do Pacífico	(UTC+10:00)	Guam, Porto Moresby	Este fuso horário não observa o horário de verão.
Hora Padrão de Yakutsk	(UTC+09:00)	Yakutsk	

Usar uma chave mestra de serviço com RDS Custom para SQL Server

O RDS Custom para SQL Server comporta o uso de uma chave mestra de serviço (SMK). O RDS Custom mantém a mesma SMK durante toda a vida útil da instância de banco de dados do RDS Custom para SQL Server. Ao manter a mesma SMK, a instância de banco de dados pode usar objetos criptografados com a SMK, como senhas e credenciais de servidores vinculados. Se você usa uma implantação multi-AZ, o RDS Custom também sincroniza e mantém a SMK entre as instâncias de banco de dados primária e secundária.

Tópicos

- Disponibilidade de região e versão
- Atributos compatíveis
- Uso de TDE
- Configurar recursos
- Requisitos e limitações

Disponibilidade de região e versão

O uso de uma SMK é compatível em todas as regiões em que o RDS Custom para SQL Server está disponível, para todas as versões do SQL Server disponíveis no RDS Custom. Para ter mais informações sobre a disponibilidade de versões e regiões do Amazon RDS com o RDS Custom para SQL Server, consulte Regiões e mecanismos de banco de dados compatíveis com o RDS Custom para SQL Server.

Atributos compatíveis

Ao usar uma SMK com RDS Custom para SQL Server, os seguintes recursos são compatíveis:

- Transparent Data Encryption (TDE)
- Criptografia em nível de coluna
- Correspondência de banco de dados
- Servidores vinculados
- SQL Server Integration Services (SSIS)

Uso de TDE

Uma SMK permite configurar a Transparent Data Encryption (TDE), que criptografa os dados antes de gravá-los no armazenamento e os descriptografa automaticamente quando os dados são lidos no armazenamento. Ao contrário do RDS para SQL Server, configurar a TDE em uma instância de banco de dados do RDS Custom para SQL Server não exige o uso de grupos de opções. Em vez disso, depois de criar um certificado e uma chave de criptografia de banco de dados, é possível executar o seguinte comando para ativar a TDE em nível de banco de dados:

```
ALTER DATABASE [myDatabase] SET ENCRYPTION ON;
```

Para ter mais informações sobre como usar a TDE com o RDS Custom para SQL Server, consulte Suporte para criptografia de dados transparente no SQL Server.

Para ter informações detalhadas sobre a TDE no Microsoft SQL Server, consulte Transparent data encryption na documentação da Microsoft.

Configurar recursos

Para conhecer as etapas detalhadas da configuração de recursos que usam uma SMK com RDS Custom para SQL Server, é possível usar as seguintes publicações no blog do banco de dados do Amazon RDS:

- Linked servers: Configuring linked servers on RDS Custom for SQL Server.
- SSIS: Migrate SSIS packages to RDS Custom for SQL Server.
- TDE: Secure your data using TDE on RDS Custom for SQL Server.

Requisitos e limitações

Ao usar uma SMK com uma instância de banco de dados do RDS Custom para SQL Server, tenha em mente os seguintes requisitos e limitações:

- Se você gerar novamente a SMK na instância de banco de dados, deverá criar imediatamente um snapshot de banco de dados manual. Recomendamos evitar a nova geração da SMK, se possível.
- É necessário manter backups dos certificados do servidor e das senhas da chave mestra do banco de dados. Se você não mantiver backups deles, poderá haver perda de dados.
- Se você configurar o SSIS, deverá usar um documento SSM para unir a instância de banco de dados do RDS Custom para SQL Server ao domínio no caso de computação em escala ou substituição do host.
- Quando a TDE ou a criptografia de coluna estão habilitadas, os backups do banco de dados são criptografados automaticamente. Ao realizar uma restauração de snapshot ou uma recuperação para um ponto no tempo, a SMK da instância de banco de dados de origem será restaurada para descriptografar os dados para a restauração, e uma nova SMK será gerado para criptografar novamente os dados na instância restaurada.

Para ter mais informações sobre chaves mestras de serviço no Microsoft SQL Server, consulte SQL Server and Database Encryption Keys na documentação da Microsoft.

Suporte à captura de dados de alteração (CDC) com o RDS Custom para SQL Server

Visão geral

O RDS Custom para SQL Server oferece suporte nativo à captura de dados de alteração (CDC), permitindo que você acompanhe e capture modificações nos dados de suas tabelas do SQL Server.

A CDC armazena metadados detalhados sobre essas alterações para recuperação e análise posteriores. Para ter mais informações sobre como a funcionalidade de CDC, consulte Change data capture na documentação da Microsoft.

A operação de CDC no SQL Server requer valores correspondentes entre o servidor local (que tem `server_id` = 0) em `sys.servers` e os identificadores `SERVERPROPERTY('ServerName')`. O RDS Custom para SQL Server mantém automaticamente essa sincronização durante o ciclo de vida da instância para garantir o funcionamento contínuo da CDC, mesmo que os hosts sejam substituídos durante as operações de manutenção ou recuperação.

> **Important**
>
> Após o failover de uma instância multi-AZ, a função `SERVERPROPERTY('Servername')` reflete automaticamente as alterações no nome da rede/computador. No entanto, a função `@@SERVERNAME` mantém o nome antigo do servidor até que o serviço `MSSQLSERVER` seja reiniciado. A consulta de @@SERVERNAME exibe o nome do servidor anterior após um failover. Para obter o nome exato do servidor após um failover, use a seguinte consulta SQL:
>
> ```
> SELECT name FROM sys.servers WHERE server_id=0
> ```
>
> Essa consulta fornece as informações de nome de servidor mais atualizadas sem exigir a reinicialização do serviço.

Disponibilidade de regiões e versões

A funcionalidade de CDC pode ser usada em todas as Regiões da AWS em que o RDS Custom para SQL Server está disponível, para todas as versões do SQL Server compatíveis com o RDS Custom. Para ter mais informações sobre versões compatíveis e disponibilidade de regiões do RDS Custom para SQL Server, consulte Regiões e mecanismos de banco de dados compatíveis com o RDS Custom para SQL Server.

Requisitos e limitações

Ao implementar a CDC no RDS Custom para SQL Server, esteja ciente destas considerações importantes:

- Se você configurar manualmente `@@SERVERNAME` e/ou servidor local em `sys.servers` para usar recursos como o MS Replication, se o valor do servidor local (que tem `server_id = 0`) em

`sys.servers` estiver definido em um formato que corresponda a `*.rds.amazonaws.com` ou `*.awsrds.*.com`, o RDS Custom para SQL Server não tentará modificá-lo para corresponder a `SERVERPROPERTY('ServerName')`.

- O RDS não pode modificar o servidor local (que tem `server_id = 0`) em `sys.servers` para um novo nome de host enquanto logins remotos ou servidores vinculados estiverem usando ativamente o nome de host antigo. Essa limitação se aplica em dois cenários:
  - Quando um servidor vinculado estabelece uma conexão com o servidor local usando um login remoto associado ao nome de host antigo
  - Quando uma instância do RDS Custom para SQL Server atua como publicador ou distribuidor e vincula logins associados ao nome de host antigo às instâncias do assinante.

Solução de problemas

Para identificar logins remotos ou logins vinculados associados ao nome antigo do servidor, use as consultas a seguir. Valide os resultados e remova esses logins para garantir que a CDC funcione adequadamente.

```
SELECT * FROM sys.remote_logins WHERE server_id=0
```

or

```
select sss.srvname,ssp.name,srl.remote_name  from sys.server_principals ssp
inner join sys.remote_logins srl on srl.local_principal_id=ssp.principal_id
inner join sys.sysservers sss  on srl.server_id = sss.srvid
where sss.srvname = @@SERVERNAME
```

Configurar seu ambiente para o Amazon RDS Custom for SQL Server

Antes de criar e gerenciar uma instância de banco de dados para a instância de banco de dados Amazon RDS Custom for SQL Server, certifique-se de realizar as tarefas a seguir.

Sumário

- Pré-requisitos para configurar o RDS Custom for SQL Server
  - Criação automática de perfis de instância usando o AWS Management Console
- Etapa 1: conceder as permissões necessárias à entidade principal do IAM
- Etapa 2: configurar rede, perfil de instância e criptografia
  - Configurar com o AWS CloudFormation
    - Parâmetros exigidos pelo CloudFormation
    - Baixar o arquivo de modelo do AWS CloudFormation
    - Configurar recursos usando o CloudFormation
  - Configurar manualmente
    - Verifique se você tem uma chave do AWS KMS de criptografia simétrica.
    - Criar sua função do IAM e seu perfil de instância manualmente
      - Criar a função do IAM AWSRDSCustomSQLServerInstanceRole
      - Adicionar uma política de acesso a AWSRDSCustomSQLServerInstanceRole
      - Criar seu perfil de instância do RDS Custom for SQL Server
      - Adicionar AWSRDSCustomSQLServerInstanceRole ao seu perfil de instância do RDS Custom for SQL Server
    - Configurar sua VPC manualmente
      - Configurar o grupo de segurança da sua VPC
      - Configurar endpoints para Serviços da AWS dependentes
      - Configurar o serviço de metadados da instância
- Restrição entre instâncias

> **Note**
>
> Para ver um tutorial passo a passo sobre como configurar os pré-requisitos e iniciar o Amazon RDS Custom para SQL Server, consulte Get started with Amazon RDS Custom for

SQL Server using an CloudFormation template (Network setup) e Explore the prerequisites required to create an Amazon RDS Custom for SQL Server instance.

Pré-requisitos para configurar o RDS Custom for SQL Server

Antes de criar uma instância de banco de dados do RDS Custom para SQL Server, seu ambiente deve atender aos requisitos descritos neste tópico. Também é possível usar o modelo do CloudFormation para configurar os pré-requisitos da Conta da AWS. Para obter mais informações, consulte Configurar com o AWS CloudFormation.

O RDS Custom para SQL Server exige que você configure os seguintes pré-requisitos:

- Configure as permissões do AWS Identity and Access Management (IAM) para a criação da instância. Esse é o usuário ou o perfil do AWS Identity and Access Management (IAM) necessário para fazer uma solicitação `create-db-instance` ao RDS.
- Configure os recursos obrigatórios exigidos pela instância de banco de dados do RDS Custom para SQL Server:
  - Configure a chave do AWS KMS necessária para a criptografia da instância do RDS Custom. O RDS Custom exige uma chave gerenciada pelo cliente no momento da criação da instância para criptografia. O ARN da chave do KMS, o ID, o ARN do alias ou o nome do alias é transmitido como o parâmetro `kms-key-id` na solicitação para criar a instância de banco de dados do RDS Custom.
  - Configure as permissões necessárias na instância de banco de dados do RDS Custom para SQL Server. O RDS Custom anexa um perfil à instância de banco de dados no momento da criação e o utiliza para automação dentro da respectiva instância. O nome do perfil da instância está definido como `custom-iam-instance-profile` na solicitação de criação do RDS Custom. É possível criar um perfil de instância por meio do AWS Management Console ou criá-lo manualmente. Para obter mais informações, consulte Criação automática de perfis de instância usando o AWS Management Console e Criar sua função do IAM e seu perfil de instância manualmente.
  - Defina a configuração de rede de acordo com os requisitos do RDS Custom para SQL Server. As instâncias do RDS Custom residem nas sub-redes (configuradas com o grupo de sub-redes de banco de dados) que você fornece na criação da instância. Essas sub-redes devem permitir que as instâncias do RDS Custom comuniquem-se com os serviços necessários para a automação do RDS.

> **Note**
>
> Para atender aos requisitos mencionados acima, verifique se não há nenhuma política de controle de serviços (SCP) restringindo permissões em nível de conta.
> Se a conta que você está usando fizer parte de uma organização da AWS, ela poderá ter políticas de controle de serviço (SCPs) restringindo permissões em nível de conta. Os SCPs não podem restringir as permissões em usuários e perfis criados com os procedimentos a seguir.
> Para ter mais informações sobre SCPs, consulte Políticas de controle de serviço no Guia do usuário do AWS Organizations. Use o comando describe-organization da AWS CLI para conferir se sua conta faz parte de uma organização da AWS.
> Para ter mais informações sobre o AWS Organizations, consulte O que é o AWS Organizations no Guia do usuário do AWS Organizations Organizations.

Para conhecer os requisitos gerais que se aplicam ao RDS Custom for SQL Server, consulte Requisitos gerais do RDS Custom for SQL Server.

Criação automática de perfis de instância usando o AWS Management Console

O RDS Custom exige que você crie e configure um perfil de instância para iniciar qualquer instância de banco de dados do RDS Custom para SQL Server. Use o AWS Management Console para criar um perfil de instância e anexá-lo em uma única etapa. Essa opção está disponível na seção de segurança do RDS Custom nas páginas do console Criar banco de dados, Restaurar snapshot e Restaurar para um momento específico. Selecione Criar um novo perfil da instância para fornecer um sufixo de nome de perfil de instância. O AWS Management Console cria um perfil de instância com as permissões necessárias para tarefas de automação do RDS Custom. Para criar automaticamente perfis de instância, seu usuário conectado ao AWS Management Console deve ter as permissões `iam:CreateInstanceProfile`, `iam:AddRoleToInstanceProfile`, `iam:CreateRole` e `iam:AttachRolePolicy`.

> **Note**
>
> Essa opção só está disponível no AWS Management Console. Se você estiver usando a CLI ou o SDK, use o modelo do CloudFormation fornecido pelo RDS Custom ou crie manualmente um perfil de instância. Para obter mais informações, consulte Criar sua função do IAM e seu perfil de instância manualmente.

Etapa 1: conceder as permissões necessárias à entidade principal do IAM

Verifique se você tem acesso suficiente para criar uma instância do RDS Custom. O perfil ou o usuário do IAM (conhecido como a entidade principal do IAM) para criar uma instância de banco de dados do RDS Custom para SQL Server usando o console ou a CLI deve ter uma das seguintes políticas para a criação bem-sucedida de uma instância de banco de dados:

- A política `AdministratorAccess`
- A política do `AmazonRDSFullAccess` mostra as seguintes permissões adicionais:

```
iam:SimulatePrincipalPolicy
cloudtrail:CreateTrail
cloudtrail:StartLogging
s3:CreateBucket
s3:PutBucketPolicy
s3:PutBucketObjectLockConfiguration
s3:PutBucketVersioning
kms:CreateGrant
kms:DescribeKey
kms:Decrypt
kms:ReEncryptFrom
kms:ReEncryptTo
kms:GenerateDataKeyWithoutPlaintext
kms:GenerateDataKey
ec2:DescribeImages
ec2:RunInstances
ec2:CreateTags
```

O RDS Custom usa essas permissões durante a criação da instância. Essas permissões configuram recursos em sua conta que são necessários para operações do RDS Custom.

Para ter mais informações sobre a permissão `kms:CreateGrant`, consulte Gerenciamento de AWS KMS key.

O exemplo de política JSON a seguir concede as permissões necessárias.

JSON

```
{
    "Version": "2012-10-17",
```

```
    "Statement": [
        {
            "Sid": "ValidateIamRole",
            "Effect": "Allow",
            "Action": "iam:SimulatePrincipalPolicy",
            "Resource": "*"
        },
        {
            "Sid": "CreateCloudTrail",
            "Effect": "Allow",
            "Action": [
                "cloudtrail:CreateTrail",
                "cloudtrail:StartLogging"
            ],
            "Resource": "arn:aws:cloudtrail:*:*:trail/do-not-delete-rds-custom-*"
        },
        {
            "Sid": "CreateS3Bucket",
            "Effect": "Allow",
            "Action": [
                "s3:CreateBucket",
                "s3:PutBucketPolicy",
                "s3:PutBucketObjectLockConfiguration",
                "s3:PutBucketVersioning"
            ],
            "Resource": "arn:aws:s3:::do-not-delete-rds-custom-*"
        },
        {
            "Sid": "CreateKmsGrant",
            "Effect": "Allow",
            "Action": [
                "kms:CreateGrant",
                "kms:DescribeKey"
            ],
            "Resource": "*"
        }
    ]
}
```

A entidade principal do IAM exige as seguintes permissões adicionais para trabalhar com versões personalizadas do mecanismo (CEVs):

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ConfigureKmsKeyEncryptionPermission",
            "Effect": "Allow",
            "Action": [
                "kms:CreateGrant",
                "kms:DescribeKey",
                "kms:Decrypt",
                "kms:ReEncryptFrom",
                "kms:ReEncryptTo",
                "kms:GenerateDataKeyWithoutPlaintext",
                "kms:GenerateDataKey"
            ],
            "Resource": "arn:aws:kms:region:account_id:key/key_id"
        },
        {
            "Sid": "CreateEc2Instance",
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeImages",
                "ec2:RunInstances",
                "ec2:CreateTags"
            ],
            "Resource": "*"
        }
    ]
}
```

Substitua *account_id* pelo ID da conta que você está usando para criar a instância. Substitua *region* pela Região da AWS que você está usando para criar a instância. Substitua *key_id* pelo ID da chave gerenciada pelo cliente. É possível adicionar várias chaves, conforme necessário.

Para ter mais informações sobre as permissões em nível de recurso que são necessárias para iniciar uma instância do EC2, consulte Executar instâncias (RunInstances).

Além disso, a entidade principal do IAM exige a permissão `iam:PassRole` no perfil do IAM. Ela deve ser anexada ao perfil de instância transmitido no parâmetro `custom-iam-instance-`

`profile` na solicitação para criar a instância de banco de dados do RDS Custom. O perfil de instância e sua função anexada são criados posteriormente em Etapa 2: configurar rede, perfil de instância e criptografia.

> **Note**
>
> As permissões listadas anteriormente não devem ser restritas por políticas de controle de serviço (SCPs), limites de permissão nem políticas de sessão associadas à entidade principal do IAM.

Etapa 2: configurar rede, perfil de instância e criptografia

Você pode configurar sua função de perfil de instância do IAM, nuvem privada virtual (VPC) e chave de criptografia simétrica do AWS KMS usando um dos seguintes processos:

- Configurar com o AWS CloudFormation (recomendado)
- Configurar manualmente

> **Note**
>
> Se sua conta fizer parte de qualquer AWS Organizations, as permissões exigidas pelo perfil de instância não deverão ser restritas por políticas de controle de serviços (SCPs).
> As configurações de rede neste tópico funcionam melhor com instâncias de banco de dados que não são acessíveis ao público. Você não pode se conectar diretamente a essas instâncias de banco de dados de fora da VPC.

Configurar com o AWS CloudFormation

Para simplificar a configuração, é possível utilizar um arquivo de modelo do AWS CloudFormation para criar uma pilha do CloudFormation. Um modelo do CloudFormation cria todos os recursos de rede, perfis de instância e criptografia de acordo com os requisitos do RDS Custom.

Para saber como criar pilhas, consulte Criar uma pilha no console do AWS CloudFormation, no Guia do usuário do AWS CloudFormation.

Para ver um tutorial sobre como executar o Amazon RDS Custom para SQL Server usando um modelo do AWS CloudFormation, consulte Começar a usar o Amazon RDS Custom para SQL Server usando um modelo do AWS CloudFormation no blog de banco de dados da AWS.

Tópicos

- Parâmetros exigidos pelo CloudFormation
- Baixar o arquivo de modelo do AWS CloudFormation
- Configurar recursos usando o CloudFormation

Parâmetros exigidos pelo CloudFormation

Os seguintes parâmetros são necessários para configurar os recursos obrigatórios do RDS Custom com o CloudFormation:

Grupo de parâmetros	Nome do parâmetro	Valor padrão	Descrição
Configuração de disponibilidade	Selecionar uma configuração de disponibilidade para configuração de pré-requisitos	Multi-AZ	Especifique se deseja configurar pré-requisitos na configuração single-AZ ou multi-AZ para instâncias do RDS Custom. É necessário usar a configuração multi-AZ caso você precise de pelo menos uma instância de banco de dados multi-AZ nessa configuração.
Configuração de rede	Bloco CIDR IPv4 para VPC	10.0.0.0/16	Especifique um bloco CIDR IPv4 (intervalo de endereços IP) para a VPC. Essa VPC está configurada para criar e trabalhar com a instância de

Grupo de parâmetros	Nome do parâmetro	Valor padrão	Descrição
			banco de dados do RDS Custom.
	Bloco CIDR IPv4 para uma de duas sub-redes privadas	10.0.128.0/20	Especifique um bloco CIDR IPv4 (ou intervalo de endereços IP) para a sua primeira sub-rede privada. Essa é uma das duas sub-redes nas quais a instância de banco de dados do RDS Custom pode ser criada. Essa é uma sub-rede privada sem acesso à internet.
	Bloco CIDR IPv4 para duas de duas sub-redes privadas	10.0.144.0/20	Especifique um bloco CIDR IPv4 (ou intervalo de endereços IP) para a segunda sub-rede privada. Essa é uma das duas sub-redes nas quais a instância de banco de dados do RDS Custom pode ser criada. Essa é uma sub-rede privada sem acesso à internet.

Grupo de parâmetros	Nome do parâmetro	Valor padrão	Descrição
	Bloco CIDR IPv4 da sub-rede pública	10.0.0.0/20	Especifique um bloco CIDR IPv4 (ou intervalo de endereços IP) para a sub-rede pública. Essa é uma das sub-redes nas quais a instância do EC2 pode se conectar com a instância de banco de dados do RDS Custom. Essa é uma sub-rede pública com acesso à internet.
Configuração de acesso RDP	Bloco CIDR IPv4 da origem	-	Especifique um bloco CIDR IPv4 (ou intervalo de endereços IP) da sua origem. Esse é o intervalo de IP do qual você faz a conexão RDP com a instância do EC2 na sub-rede pública. Se não for definida, a conexão RDP com a instância do EC2 não será configurada.

Grupo de parâmetros	Nome do parâmetro	Valor padrão	Descrição
	Configurar o acesso RDP à instância do RDS Custom para SQL Server	Não	Especifique se deseja habilitar a conexão RDP da instância do EC2 com a instância do RDS Custom para SQL Server. Por padrão, a conexão RDP da instância do EC2 com a instância de banco de dados não está configurada.

Recursos criados pelo CloudFormation

A criação bem-sucedida da pilha do CloudFormation usando configurações padrão cria os seguintes recursos na Conta da AWS:

- Chave do KMS de criptografia simétrica para criptografia de dados gerenciados pelo RDS Custom.
- O perfil de instância é associado a um perfil do IAM com `AmazonRDSCustomInstanceProfileRolePolicy` para fornecer as permissões exigidas pelo RDS Custom. Para ter mais informações, consulte AmazonRDSCustomServiceRolePolicy no Guia de referência de políticas gerenciadas da AWS.
- VPC com o intervalo CIDR especificado como o parâmetro CloudFormation. O valor padrão é `10.0.0.0/16`.
- Duas sub-redes privadas com o intervalo CIDR especificado nos parâmetros e duas zonas de disponibilidade diferentes na Região da AWS. Os valores padrão para os CIDRs de sub-rede são `10.0.128.0/20` e `10.0.144.0/20`.
- Uma sub-rede pública com o intervalo CIDR especificado nos parâmetros. O valor padrão para o CIDR da sub-rede é 10.0.0.0/20. A instância do EC2 reside nessa sub-rede e pode ser usada para estabelecer conexão com a instância do RDS Custom.
- Opção DHCP definida para a VPC com resolução de nome de domínio para um servidor do Sistema de Nomes de Domínio da Amazon (DNS).
- Tabela de rota para associar a duas sub-redes privadas e sem acesso à Internet.

- Tabela de rotas para associar à sub-rede pública e ter acesso à internet.
- Gateway da internet associado à VPC para permitir o acesso à internet à sub-rede pública.
- Lista de controle de acesso (ACL) à rede a ser associada a duas sub-redes privadas e acesso restrito a HTTPS e à porta de banco de dados na VPC.
- Grupo de segurança da VPC a ser associado à instância do RDS Custom. Para HTTPS de saída, o acesso é restrito a endpoints de AWS service (Serviço da AWS) que são exigidos pelo RDS Custom e pela porta de banco de dados do grupo de segurança da instância do EC2.
- O grupo de segurança da VPC a ser associado à instância do EC2 na sub-rede pública. Para a porta de banco de dados de saída, o acesso é restrito ao grupo de segurança da instância do RDS Custom.
- Grupo de segurança da VPC a ser associado aos endpoints da VPC criados para endpoints de AWS service (Serviço da AWS) que são exigidos pelo RDS Custom.
- Grupo de sub-redes de banco de dados no qual instâncias do RDS Custom são criadas. Duas sub-redes privadas criadas por esse modelo são adicionadas ao grupo de sub-redes de banco de dados.
- Endpoints da VPC para cada um dos endpoints de AWS service (Serviço da AWS) que são exigidos pelo RDS Custom.

Definir a configuração de disponibilidade como multi-az criará os seguintes recursos, além da lista acima:

- Regras de ACL de rede que permitem a comunicação entre sub-redes privadas.
- Acesso de entrada e de saída à porta multi-AZ dentro do grupo de segurança da VPC associado à instância do RDS Custom.
- Endpoints da VPC para endpoints de serviços da AWS que são necessários para a comunicação multi-AZ.

Além disso, definir a configuração de acesso RDP cria os seguintes recursos:

- Configurar o acesso RDP à sub-rede pública por meio do endereço IP de origem:
  - Regras de ACL de rede que permitem a conexão RDP do IP de origem à sub-rede pública.
  - Acesso de entrada à porta RDP pelo IP de origem ao grupo de segurança da VPC associado à instância do EC2.

- Configurar o acesso RDP da instância do EC2 na sub-rede pública à instância do RDS Custom em sub-redes privadas:
  - Regras de ACL de rede que permitem a conexão RDP da sub-rede pública a sub-redes privadas.
  - Acesso de entrada à porta RDP pelo grupo de segurança da VPC associado à instância do EC2 ao grupo de segurança da VPC associado à instância do RDS Custom.

Use os procedimentos a seguir para criar a pilha do CloudFormation para o RDS Custom para SQL Server.

Baixar o arquivo de modelo do AWS CloudFormation

Como baixar o arquivo de modelo

1. Abra o menu de contexto (clique com o botão direito do mouse) do link custom-sqlserver-onboard.zip e selecione Save Link As (Salvar link como).
2. Salve e extraia o arquivo no computador.

Configurar recursos usando o CloudFormation

Como configurar recursos usando o CloudFormation

1. Abra o console do CloudFormation em https://console.aws.amazon.com/cloudformation.
2. Para iniciar o assistente de criação de pilha, escolha Create Stack (Criar pilha).

   A página Create stack (Criar pilha) é exibida.
3. Em Prerequisite - Prepare template (Pré-requisito - Preparar template), escolha Template is ready (O template está pronto).
4. Na página Specify template (Especificar modelo), faça o seguinte:

   a. Para Template source (Origem do template), escolha Upload a template file (Fazer upload de um arquivo de template).

   b. Em Escolher arquivo, acesse e escolha o arquivo correto.
5. Escolha Próximo.

   A página Specify stack details (Especificar detalhes da pilha) é exibida.
6. Para Stack name (Nome da pilha), insira **`rds-custom-sqlserver`**.

7. Em Parameters (Parâmetros), faça o seguinte:

   a. Para manter as opções padrão, escolha Next (Próximo).

   b. Para alterar as opções, escolha a configuração apropriada de disponibilidade, de rede e de acesso RDP e, depois, selecione Próximo.

      Leia atentamente a descrição de cada parâmetro antes de alterar os parâmetros.

   > **Note**
   >
   > Se você optar por criar pelo menos uma instância multi-AZ nessa pilha do CloudFormation, garanta que o parâmetro da pilha do CloudFormation Selecionar uma configuração de disponibilidade para configuração de pré-requisitos esteja definido como `Multi-AZ`. Se você criar a pilha do CloudFormation como single-AZ, atualize-a para a configuração multi-AZ antes de criar a primeira instância multi-AZ.

8. Na página Configurar opções de pilha, selecione Avançar.
9. Na página Review rds-custom-sqlserver (Revisar rds-custom-sqlserver), faça o seguinte:

   a. Para Capabilities (Recursos), marque a caixa de seleção para confirmar que o AWS CloudFormation pode criar os recursos do IAM com nomes personalizados.

   b. Selecione Criar pilha.

> **Note**
>
> Não atualize os recursos criados com base nessa pilha do AWS CloudFormation diretamente das páginas de recursos. Isso impede que você aplique futuras atualizações a esses recursos usando um modelo do AWS CloudFormation.

O CloudFormation cria os recursos necessários ao RDS Custom para SQL Server. Se ocorrer uma falha na criação da pilha, leia a guia Events (Eventos) para ver em qual recurso a criação falhou e seu motivo de status.

A guia Outputs (Saídas) para essa pilha do CloudFormation no console deve ter informações sobre todos os recursos a serem transmitidos como parâmetros para criar uma instância de banco de dados do RDS Custom para SQL Server. Use o grupo de segurança da VPC e o grupo de sub-

redes de banco de dados criados pelo CloudFormation para instâncias de banco de dados do RDS Custom. Por padrão, o RDS tenta anexar o grupo de segurança da VPC padrão, que pode não ter o acesso necessário.

Se você usou o CloudFormation para criar recursos, você pode ignorar Configurar manualmente.

Atualizar a pilha do CloudFormation

Você também pode atualizar algumas das configurações na pilha do CloudFormation após a criação. As configurações que podem ser atualizadas são:

- Considerações sobre disponibilidade do RDS Custom para SQL Server
  - Selecionar uma configuração de disponibilidade para a configuração de pré-requisitos: atualize esse parâmetro para alternar entre a configuração single-AZ e multi-AZ. Se você estiver usando essa pilha do CloudFormation para pelo menos uma instância multi-AZ, deverá atualizar a pilha para escolher a configuração multi-AZ.
- Configuração do acesso RDP para RDS Custom para SQL Server
  - O bloco CIDR IPv4 da sua origem: é possível atualizar o bloco CIDR IPv4 (ou intervalo de endereços IP) da sua origem atualizando esse parâmetro. Definir esse parâmetro como vazio remove a configuração de acesso RDP do bloco CIDR de origem para a sub-rede pública.
  - Configurar o acesso RDP ao RDS Custom para SQL Server: habilite ou desabilite a conexão RDP da instância do EC2 com a instância do RDS Custom para SQL Server.

Excluir a pilha do CloudFormation

É possível excluir a pilha do CloudFormation depois de excluir todas as instâncias do RDS Custom que usam recursos da pilha. O RDS Custom não monitora a pilha do CloudFormation e, portanto, não bloqueia a exclusão da pilha quando há instâncias de banco de dados que usam recursos da pilha. Garanta que não haja instâncias de banco de dados do RDS Custom que usem os recursos da pilha ao excluí-la.

> **Note**
>
> Quando você exclui a pilha do CloudFormation, todos os recursos criados pela pilha são excluídos, exceto a chave do KMS. A chave do KMS entra em um estado de exclusão pendente e é excluída após trinta dias. Para manter a chave do KMS, execute uma operação CancelKeyDeletion durante o período de carência de 30 dias.

Configurar manualmente

Se você optar por configurar recursos manualmente, realize as tarefas a seguir.

> **Note**
>
> Para simplificar a configuração, é possível usar o arquivo de modelo do AWS CloudFormation para criar uma pilha do CloudFormation em vez de uma configuração manual. Para obter mais informações, consulte Configurar com o AWS CloudFormation. Você também pode usar a AWS CLI para concluir esta seção. Se sim, baixe e instale a CLI mais recente.

Tópicos

- Verifique se você tem uma chave do AWS KMS de criptografia simétrica.
- Criar sua função do IAM e seu perfil de instância manualmente
- Configurar sua VPC manualmente

Verifique se você tem uma chave do AWS KMS de criptografia simétrica.

Uma AWS KMS key de criptografia simétrica é necessária para o RDS Custom. Ao criar uma instância de banco de dados do RDS Custom para SQL Server, forneça o identificador de chave do KMS como o parâmetro `kms-key-id`. Para obter mais informações, consulte Criar e conectar-se a uma instância de banco de dados Amazon RDS Custom for SQL Server.

Você tem as seguintes opções:

- Se você tiver uma chave do KMS gerenciada pelo cliente em sua Conta da AWS, poderá usá-la com o RDS Custom. Nenhuma outra ação é necessária.
- Se você já tiver criado uma chave do KMS de criptografia simétrica gerenciada pelo cliente para outro mecanismo do RDS Custom, poderá reutilizar a mesma chave do KMS. Nenhuma outra ação é necessária.
- Se você não tiver uma chave do KMS de criptografia simétrica gerenciada pelo cliente em sua conta, crie uma seguindo as instruções em Criar chaves no Guia do desenvolvedor do AWS Key Management Service.

- Se você estiver criando uma instância de banco de dados do CEV ou do RDS Custom e sua chave do KMS estiver em outra Conta da AWS, use a AWS CLI. Você não pode usar o console da AWS com chaves do KMS de várias contas.

> **Important**
>
> O RDS Custom não é compatível com chaves do KMS gerenciadas pela AWS.

A chave de criptografia simétrica deve conceder acesso às operações `kms:Decrypt` e `kms:GenerateDataKey` ao perfil do AWS Identity and Access Management (IAM) em seu perfil de instância do IAM. Se você tiver uma nova chave de criptografia simétrica em sua conta, nenhuma alteração será necessária. Do contrário, a política de sua chave de criptografia simétrica precisará conceder acesso a essas operações.

Para ter mais informações, consulte Etapa 4: Configurar o IAM para RDS Custom para Oracle.

Criar sua função do IAM e seu perfil de instância manualmente

É possível criar manualmente um perfil de instância e usá-lo para iniciar instâncias do RDS Custom. Se você planeja criar a instância no AWS Management Console, ignore esta seção. O AWS Management Console permite que você crie e anexe um perfil às suas instâncias do RDS Custom. Para obter mais informações, consulte Criação automática de perfis de instância usando o AWS Management Console.

Ao criar manualmente um perfil de instância, transmita o respectivo nome como o parâmetro `custom-iam-instance-profile` para o comando `create-db-instance` da CLI. O RDS Custom usa o perfil associado a esse perfil de instância a fim de executar a automação para gerenciar a instância.

Como criar o perfil de instância do IAM e perfis do IAM para o RDS Custom para SQL Server

1. Criar a função do IAM chamada `AWSRDSCustomSQLServerInstanceRole` com uma política de confiança que permita ao Amazon EC2 assumir essa função.
2. Adicione a política gerenciada `AmazonRDSCustomInstanceProfileRolePolicy` da AWS a `AWSRDSCustomSQLServerInstanceRole`.
3. Crie um perfil de instância do IAM para o RDS Custom para SQL Server chamado `AWSRDSCustomSQLServerInstanceProfile`.

4. Adicione `AWSRDSCustomSQLServerInstanceRole` ao perfil da instância.

Criar a função do IAM AWSRDSCustomSQLServerInstanceRole

O exemplo a seguir cria a função `AWSRDSCustomSQLServerInstanceRole`. A política de confiança permite que o Amazon EC2 assuma a função.

```
aws iam create-role \
    --role-name AWSRDSCustomSQLServerInstanceRole \
    --assume-role-policy-document '{
        "Version": "2012-10-17",
          "Statement": [
            {
              "Action": "sts:AssumeRole",
              "Effect": "Allow",
              "Principal": {
                  "Service": "ec2.amazonaws.com"
              }
            }
          ]
        }'
```

Adicionar uma política de acesso a AWSRDSCustomSQLServerInstanceRole

Para fornecer as permissões necessárias, anexe a política `AmazonRDSCustomInstanceProfileRolePolicy`gerenciada pela AWS a `AWSRDSCustomSQLServerInstanceRole`. A `AmazonRDSCustomInstanceProfileRolePolicy` permite que as instâncias do RDS Custom enviem e recebam mensagens e realizem várias ações de automação.

> **Note**
>
> As permissões na política de acesso não devem ser restritas por SCPs nem por limites de permissão associados à função de perfil de instância.

O exemplo a seguir anexa a política `AmazonRDSCustomInstanceProfileRolePolicy` gerenciada pela AWS ao perfil `AWSRDSCustomSQLServerInstanceRole`.

```
aws iam attach-role-policy \
```

```
    --role-name AWSRDSCustomSQLServerInstanceRole \
    --policy-arn arn:aws:iam::aws:policy/AmazonRDSCustomInstanceProfileRolePolicy
```

Criar seu perfil de instância do RDS Custom for SQL Server

Um perfil de instância é um contêiner que inclui um único perfil do IAM. O RDS Custom usa o perfil de instância para transmitir o perfil à instância.

Se você usar o AWS Management Console para criar um perfil para o Amazon EC2, o console criará automaticamente um perfil de instância e dará a ele o mesmo nome dado ao perfil quando este foi criado. Crie seu perfil de instância da seguinte maneira, chamando-o de `AWSRDSCustomSQLServerInstanceProfile`.

```
aws iam create-instance-profile \
    --instance-profile-name AWSRDSCustomSQLServerInstanceProfile
```

Adicionar AWSRDSCustomSQLServerInstanceRole ao seu perfil de instância do RDS Custom for SQL Server

Adicione o perfil `AWSRDSCustomInstanceRoleForRdsCustomInstance` ao perfil `AWSRDSCustomSQLServerInstanceProfile` criado anteriormente.

```
aws iam add-role-to-instance-profile \
    --instance-profile-name AWSRDSCustomSQLServerInstanceProfile \
    --role-name AWSRDSCustomSQLServerInstanceRole
```

Configurar sua VPC manualmente

Sua instância de banco de dados do RDS Custom está em uma nuvem privada virtual (VPC) com base no serviço da Amazon VPC, assim como uma instância do Amazon EC2 ou do Amazon RDS. Você fornece e configura sua própria VPC. Assim, você tem controle total sobre a configuração de rede da sua instância.

O RDS Custom envia comunicações de sua instância de banco de dados para outros Serviços da AWS. Garanta que os seguintes serviços estejam acessíveis por meio da sub-rede na qual você cria instâncias de banco de dados do RDS Custom:

- Amazon CloudWatch (`com.amazonaws.`*`region`*`.monitoring`)
- Amazon CloudWatch Logs (`com.amazonaws.`*`region`*`.logs`)

- Amazon CloudWatch Events (`com.amazonaws.region.events`)
- Amazon EC2 (`com.amazonaws.region.ec2` e `com.amazonaws.region.ec2messages`)
- Amazon EventBridge (`com.amazonaws.region.events`)
- Amazon S3 (`com.amazonaws.region.s3`)
- AWS Secrets Manager (`com.amazonaws.region.secretsmanager`)
- AWS Systems Manager (`com.amazonaws.region.ssm` e `com.amazonaws.region.ssmmessages`)

Se você criar implantações multi-AZ

- Amazon Simple Queue Service (`com.amazonaws.region.sqs`)

Se o RDS Custom não conseguir se comunicar com os serviços necessários, ele publicará os seguintes eventos:

```
Database instance in incompatible-network. SSM Agent connection not available. Amazon
 RDS can't connect to the dependent AWS services.
```

```
Database instance in incompatible-network. Amazon RDS can't connect to dependent AWS
 services. Make sure port 443 (HTTPS) allows outbound connections, and try again.
 "Failed to connect to the following services: s3 events"
```

Para evitar erros `incompatible-network`, os componentes da VPC envolvidos na comunicação entre sua instância de banco de dados do RDS Custom e os Serviços da AWS devem cumprir os seguintes requisitos:

- A instância de banco de dados pode fazer conexões de saída na porta 443 para outros Serviços da AWS.
- A VPC permite respostas recebidas a solicitações originadas de sua instância de banco de dados do RDS Custom.
- O RDS Custom pode resolver corretamente os nomes de domínio dos endpoints para cada AWS service (Serviço da AWS).

Se você já configurou uma VPC para outro mecanismo de banco de dados do RDS Custom, poderá reutilizar essa VPC e ignorar esse processo.

Tópicos

- Configurar o grupo de segurança da sua VPC
- Configurar endpoints para Serviços da AWS dependentes
- Configurar o serviço de metadados da instância

Configurar o grupo de segurança da sua VPC

Um grupo de segurança atua como um firewall virtual para uma instância VPC, controlando os tráfegos de entrada e de saída. Uma instância de banco de dados do RDS Custom tem um grupo de segurança anexado à interface de rede que protege a instância. O grupo de segurança deve permitir o tráfego entre o RDS Custom e outros Serviços da AWS por meio de HTTPS. Você transmite esse grupo de segurança como o parâmetro `vpc-security-group-ids` na solicitação de criação da instância.

Para configurar seu grupo de segurança para o RDS Custom

1. Faça login no AWS Management Console e abra o console da Amazon VPC em https://console.aws.amazon.com/vpc.
2. Permita que o RDS Custom use o grupo de segurança padrão ou crie seu próprio grupo de segurança.

   Para obter instruções detalhadas, consulte Fornecer acesso à instância de banco de dados na VPC criando um grupo de segurança.
3. Verifique se o grupo de segurança permite conexões de saída na porta 443. O RDS Custom precisa dessa porta para se comunicar com os Serviços da AWS dependentes.
4. Se você tiver uma VPC privada e utilizar endpoints da VPC, verifique se o grupo de segurança associado à instância de banco de dados permite conexões de saída na porta 443 para endpoints da VPC. Verifique também se o grupo de segurança associado ao endpoint da VPC permite conexões de entrada na porta 443 da instância de banco de dados.

   Se não forem permitidas conexões de entrada, a instância do RDS Custom não poderá se conectar ao AWS Systems Manager e a endpoints do Amazon EC2. Para ter mais informações, consulte Criar um endpoint da nuvem privada virtual no Guia do usuário do AWS Systems Manager.
5. Em relação a instâncias do RDS Custom para SQL Server, verifique se o grupo de segurança associado à instância de banco de dados permite conexões de entrada e de saída na porta 1120 com esse próprio grupo de segurança. Isso é necessário para a conexão de host de

emparelhamento em uma instância multi-AZ de banco de dados do RDS Custom para SQL Server.

Para ter mais informações sobre grupos de segurança, consulte Grupos de segurança para sua VPC no Guia do desenvolvedor da Amazon VPC.

Configurar endpoints para Serviços da AWS dependentes

Convém adicionar endpoints para cada serviço à sua VPC utilizando as instruções a seguir. No entanto, é possível utilizar qualquer solução que permita que a sua VPC se comunique com endpoints de serviços da AWS. Por exemplo, é possível utilizar a Conversão de endereços de rede (NAT) ou AWS Direct Connect.

Como configurar endpoints de Serviços da AWS com os quais o RDS Custom funciona

1. Abra o console da Amazon VPC, em https://console.aws.amazon.com/vpc/.
2. Na barra de navegação, use o seletor de região para escolher a Região da AWS.
3. No painel de navegação, escolha Endpoints. No painel principal, escolha Create Endpoint (Criar endpoint).
4. Em Service category (Categoria de serviço), escolha Serviços da AWS.
5. Para Service Name (Nome do serviço), escolha o endpoint mostrado na tabela.
6. Em VPC, escolha sua VPC.
7. Em Subnets (Sub-redes), escolha uma sub-rede de cada zona de disponibilidade a ser incluída.

   O endpoint da VPC pode abranger várias zonas de disponibilidade. A AWS cria uma interface de rede elástica do endpoint da VPC em cada sub-rede que você escolhe. Cada interface de rede tem um nome de host de Sistema de Nomes de Domínio (DNS) e um endereço IP privado.
8. Para Security group (Grupo de segurança), escolha ou crie um grupo de segurança.

   Você pode usar grupos de segurança para controlar o acesso ao seu endpoint, da mesma forma que usa um firewall. Verifique se o grupo de segurança permite conexões de entrada na porta 443 por meio das instâncias de banco de dados. Para ter mais informações sobre grupos de segurança de VPC, consulte Grupos de segurança para a VPC no Guia do usuário da Amazon VPC.
9. Você também pode anexar uma política ao endpoint da VPC. As políticas de endpoint podem controlar o acesso ao AWS service (Serviço da AWS) ao qual você está se conectando. A política padrão permite que todas as solicitações passem pelo endpoint. Se você estiver usando

uma política personalizada, verifique se as solicitações da instância de banco de dados são permitidas na política.

10. Escolha Criar endpoint.

A tabela a seguir explica como encontrar a lista de endpoints que a sua VPC precisa para comunicações de saída.

Serviço	Formato do endpoint	Observações e links
AWS Systems Manager	Utilize os seguintes formatos de endpoint: • `ssm.region.amazonaws.com` • `ssmmessages. region.amazonaws.com`	Para ver uma lista de todos os endpoints regionais, consulte Cotas e endpoints do AWS Systems Manager no Referência geral da Amazon Web Services.
AWS Secrets Manager	Use o formato de endpoint `secretsmanager. region.amazonaws.com .`	Para ver uma lista de todos os endpoints regionais, consulte Cotas e endpoints do AWS Secrets Manager no Referência geral da Amazon Web Services.
Amazon CloudWatch	Utilize os seguintes formatos de endpoint: • Para métricas do CloudWatch, use `monitoring. region.amazonaws.com` • Para o CloudWatch Events, use `events.region.amazonaws.com` • Para o CloudWatch Logs, use `logs.region.amazonaws.com`	Para ver a lista de endpoints em todas as regiões, consulte: • Endpoints e cotas do Amazon CloudWatch no Referência geral da Amazon Web Services • Endpoints e cotas do Amazon CloudWatch Logs no Referência geral da Amazon Web Services • Endpoints e cotas do Amazon CloudWatch Events no Referência geral da Amazon Web Services

Serviço	Formato do endpoint	Observações e links
Amazon EC2	Utilize os seguintes formatos de endpoint: • `ec2.region.amazonaws.com` • `ec2messages. region.amazonaws.com`	Para ver a lista de endpoints em cada região, consulte Endpoints e cotas do Amazon Elastic Compute Cloud na Referência geral da Amazon Web Services.
Amazon S3	Use o formato de endpoint `s3.region.amazonaws.com` .	Para ver a lista de endpoints em cada região, consulte Endpoints e cotas do Amazon Simple Storage Service na Referência geral da Amazon Web Services. Para saber mais sobre endpoints de gateway para o Amazon S3, consulte Endpoints do Amazon S3, no Guia do desenvolvedor da Amazon VPC. Para saber como criar um ponto de acesso, consulte Criar pontos de acesso no Guia do desenvolvedor do Amazon VPC. Para saber como criar endpoints de gateway para o Amazon S3, consulte Endpoints da VPC do gateway.
Amazon Simple Queue Service	Usar o formato de endpoint `sqs.region.amazonaws.com`	Para ver a lista de endpoints em cada região, consulte Amazon Simple Queue Service endpoints and quotas.

Configurar o serviço de metadados da instância

Verifique se a sua instância pode fazer o seguinte:

- Acesse o serviço de metadados de instância utilizando o Instance Metadata Service versão 2 (IMDSv2).
- Permita comunicações de saída por meio da porta 80 (HTTP) com o endereço IP do link IMDS.
- Solicite metadados da instância de `http://169.254.169.254`, o link IMDSv2.

Consulte mais informações em Usar IMDSv2 no Guia do usuário do Amazon EC2.

Restrição entre instâncias

Quando você cria um perfil de instância seguindo as etapas acima, ele usa a política `AmazonRDSCustomInstanceProfileRolePolicy` gerenciada pela AWS para fornecer as permissões necessárias ao RDS Custom, o que permite o gerenciamento de instâncias e a automação do monitoramento. A política gerenciada garante que as permissões viabilizem acesso somente aos recursos que o RDS Custom exige para executar a automação. Recomendamos usar a política gerenciada para comportar novos recursos e atender aos requisitos de segurança que são aplicados automaticamente aos perfis de instância existentes sem intervenção manual. Para ter mais informações, consulte a política gerenciada pela AWS: AmazonRDSCustomInstanceProfileRolePolicy.

A política gerenciada `AmazonRDSCustomInstanceProfileRolePolicy` impede que o perfil de instância tenha acesso entre contas, mas pode permitir acesso a alguns recursos gerenciados do RDS Custom entre as respectivas instâncias na mesma conta. Com base em suas necessidades, você pode usar limites de permissão para restringir ainda mais o acesso entre instâncias. Os limites de permissões definem o número máximo de permissões que as políticas baseadas em identidade podem conceder a uma entidade, mas, por si sós, não concedem permissões. Para ter mais informações, consulte Avaliar permissões efetivas com limites.

Por exemplo, a política de limite a seguir impede que o perfil de instância acesse uma chave do AWS KMS específica e limita o acesso aos recursos gerenciados do RDS Custom em instâncias que usam chaves do AWS KMS diferentes.

JSON

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "DenyOtherKmsKeyAccess",
            "Effect": "Deny",
            "Action": "kms:*",
            "NotResource": "arn:aws:kms:region:acct_id:key/KMS_key_ID"
        },
        {
            "Sid": "NoBoundarySetByDefault",
            "Effect": "Allow",
            "Action": "*",
            "Resource": "*"
        }
    ]
}
```

> **Note**
>
> Garanta que o limite de permissões não bloqueie nenhuma permissão que a `AmazonRDSCustomInstanceProfileRolePolicy` concede ao RDS Custom.

Modelo BYOM (traga sua própria mídia) com o RDS Custom para SQL Server

O RDS Custom para SQL Server oferece suporte a dois modelos de licenciamento: licença inclusa (LI) e BYOM (traga sua própria mídia).

Com o modelo BYOM, você pode fazer o seguinte:

1. Forneça e instale seus próprios binários do Microsoft SQL Server com atualizações cumulativas (CU) compatíveis em uma AMI do AWS EC2 para Windows.
2. Salve a AMI como uma imagem dourada, que é um modelo que você pode usar para criar uma versão de mecanismo personalizada (CEV).
3. Crie uma CEV com base na imagem dourada.
4. Crie instâncias de banco de dados do RDS Custom para SQL Server usando a CEV.

Depois, o Amazon RDS gerencia suas instâncias de banco de dados para você.

> **Note**
>
> Se você também tiver uma instância de banco de dados do RDS Custom para SQL Server com licença inclusa (LI), não poderá usar o software SQL Server dessa instância de banco de dados com o modelo BYOM. Você deve trazer seus próprios binários do SQL Server para o modelo BYOM.

Requisitos do modelo BYOM para o RDS Custom para SQL Server

Os mesmos requisitos gerais para versões de mecanismo personalizadas com o RDS Custom para SQL Server também se aplicam ao modelo BYOM. Para ter mais informações, consulte Requisitos para CEVs do RDS Custom para SQL Server.

Ao usar o modelo BYOM, certifique-se de atender aos seguintes requisitos adicionais:

- Use uma das seguintes edições aceitas: SQL Server 2022 ou as edições 2019 Enterprise, Standard ou Developer.
- Conceda ao perfil de servidor sysadmin (SA) do SQL Server o privilégio de `NT AUTHORITY\SYSTEM`.

- Mantenha o SO Windows Server do EC2 configurado com horário UTC.

  Por padrão, as instâncias subjacentes do Amazon EC2 para Windows são definidas no fuso horário UTC. Consulte mais informações sobre como visualizar e alterar o horário de uma instância do Windows em Relógio de precisão e sincronização de horas em sua instância do EC2 no Guia do usuário do Amazon EC2.
- Abra a porta TCP 1433 e a porta UDP 1434 para permitir conexões SSM.

Limitações do modelo BYOM para o RDS Custom para SQL Server

As mesmas limitações gerais do RDS Custom para SQL Server também se aplicam ao modelo BYOM. Para ter mais informações, consulte Requisitos e limitações do Amazon RDS Custom for SQL Server.

Com o modelo BYOM, as seguintes limitações adicionais se aplicam:

- Somente a instância padrão do SQL Server (MSSQLSERVER) é compatível. Não há suporte para instâncias nomeadas do SQL Server. O RDS Custom para SQL Server detecta e monitora somente a instância padrão do SQL Server.
- Só há suporte para uma instalação do SQL Server em cada AMI. Não há suporte para várias instalações de diferentes versões do SQL Server.
- O SQL Server Web não é compatível com o modelo BYOM.
- As versões de avaliação das edições do SQL Server não são compatíveis com o modelo BYOM. Ao instalar o SQL Server, não marque a caixa de seleção para usar uma versão de avaliação.
- A disponibilidade e a compatibilidade de recursos variam entre versões específicas de cada mecanismo de banco de dados e entre Regiões da AWS. Para ter mais informações, consulte Disponibilidade de regiões para CEVs do RDS Custom para SQL Server e Versões compatíveis com CEVs do RDS Custom para SQL Server.

Criar uma instância de banco de dados do RDS Custom para SQL Server com o modelo BYOM

Para preparar e criar uma instância de banco de dados do RDS Custom para SQL Server com o modelo BYOM, consulte Preparar uma CEV usando o modelo BYOM (traga sua própria mídia).

Trabalhar com versões de mecanismo personalizadas para o RDS Custom para SQL Server

Uma versão de mecanismo personalizada (CEV) para o RDS Custom para SQL Server é uma imagem de máquina da Amazon (AMI) que inclui o Microsoft SQL Server.

As etapas básicas do fluxo de trabalho da CEV são as seguintes:

1. Escolha uma AMI do AWS EC2 para Windows que será usada como imagem de base para uma CEV. Você tem a opção de usar o Microsoft SQL Server pré-instalado ou usar o modelo BYOM (traga sua própria mídia) para instalar o SQL Server por conta própria.
2. Instale outros softwares no sistema operacional (SO) e personalize a configuração do SO e do SQL Server para atender às necessidades da sua empresa.
3. Salve a AMI como uma imagem dourada.
4. Crie uma versão de mecanismo personalizada (CEV) com base na imagem dourada.
5. Crie instâncias de banco de dados do RDS Custom para SQL Server usando a CEV.

Depois, o Amazon RDS gerencia essas instâncias de banco de dados para você.

Uma CEV possibilita que você mantenha sua configuração de referência preferida do sistema operacional e do banco de dados. O uso de uma CEV garante que a configuração do host, como qualquer instalação de agente de terceiro ou outras personalizações do sistema operacional, permaneça nas instâncias de banco de dados do RDS Custom para SQL Server. Com uma CEV, é possível implantar rapidamente frotas de instâncias de banco de dados do RDS Custom para SQL Server com a mesma configuração.

Tópicos

- Preparar-se para criar uma CEV para RDS Custom para SQL Server
- Criar uma CEV para RDS Custom para SQL Server
- Modificar uma CEV para RDS Custom para SQL Server
- Visualizar detalhes da CEV com o Amazon RDS Custom para SQL Server
- Excluir uma CEV para RDS Custom para SQL Server

Preparar-se para criar uma CEV para RDS Custom para SQL Server

Você pode criar uma CEV usando uma imagem de máquina da Amazon (AMI) que contém o Microsoft SQL Server pré-instalado com licença inclusa (LI), ou com uma AMI na qual você usa sua própria mídia de instalação do SQL Server (BYOM).

Preparar uma CEV

Use os procedimentos a seguir para criar uma CEV usando o modelo traga sua própria mídia (BYOM) ou o Microsoft SQL Server pré-instalado (LI).

Preparar uma CEV usando o modelo BYOM (traga sua própria mídia)

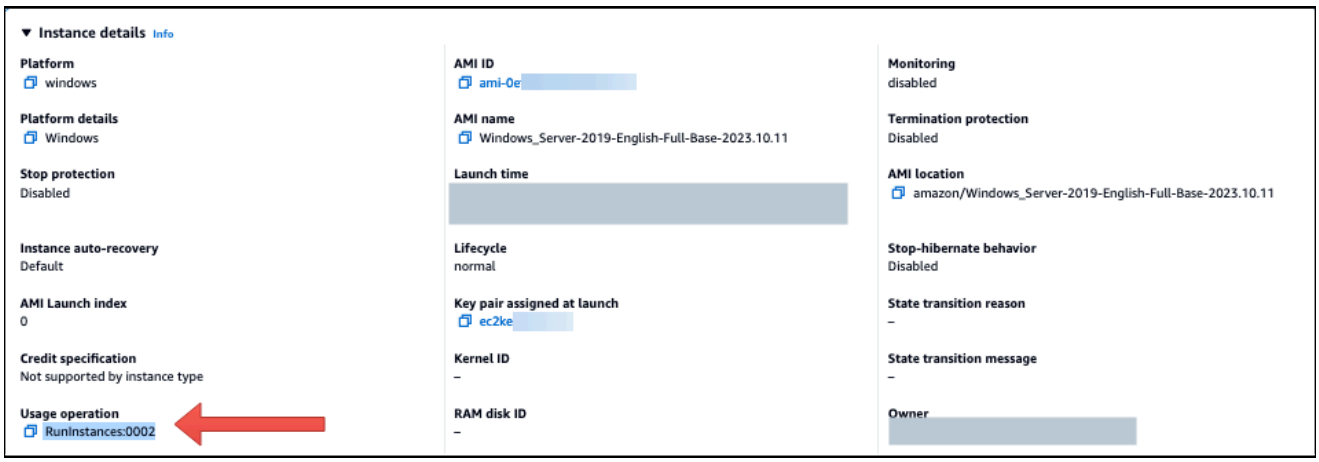
As etapas a seguir usam uma AMI com Windows Server 2019 Base como exemplo.

Como criar uma CEV usando o modelo BYOM

1. No console do Amazon EC2, selecione Iniciar instância.
2. Em Nome, insira o nome da instância.
3. Em Início rápido, selecione Windows.
4. Selecione Microsoft Windows Server 2019 Base.
5. Selecione o tipo de instância, o par de chaves, as configurações de rede e armazenamento apropriados e inicie a instância.
6. Depois de iniciar ou criar a instância do EC2, garanta que a AMI correta do Windows tenha sido selecionada na Etapa 4:

    a. Selecione a instância do EC2 no console do Amazon EC2.

    b. Na seção Detalhes, confira a Operação de uso e garanta que ela esteja definida como RunInstances:0002.



7. Faça login na instância do EC2 e copie sua mídia de instalação do SQL Server para ela.

> **Note**
>
> Se você estiver criando uma CEV usando a edição SQL Server Developer, talvez seja necessário obter a mídia de instalação usando a assinatura do Microsoft Visual Studio.

8. Instale o SQL Server. Faça o seguinte:

   a. Examine Requisitos do modelo BYOM para o RDS Custom para SQL Server e Versões compatíveis com CEVs do RDS Custom para SQL Server.

   b. Defina o diretório-raiz da instância como o `C:\Program Files\Microsoft SQL Server\` padrão. Não altere esse diretório.

   c. Defina o nome da conta do mecanismo de banco de dados do SQL Server como `NT Service\MSSQLSERVER` ou `NT AUTHORITY\NETWORK SERVICE`.

   d. Defina o modo de inicialização do SQL Server como Manual.

   e. Escolha o modo de autenticação do SQL Server como Misto.

   f. Mantenha as configurações atuais para os diretórios de dados padrão e os locais do TempDB.

9. Conceda ao perfil de servidor sysadmin (SA) do SQL Server o privilégio de `NT AUTHORITY\SYSTEM`:

```
USE [master]
GO
EXEC master..sp_addsrvrolemember @loginame = N'NT AUTHORITY\SYSTEM' , @rolename =
 N'sysadmin'
GO
```

10. Instale software adicional ou personalize a configuração do sistema operacional e do banco de dados para atender aos seus requisitos.

11. Execute o Sysprep na instância do EC2. Para ter mais informações, consulte Criar uma AMI do Amazon EC2 usando o Sysprep do Windows.

12. Salve a AMI que contém sua versão instalada do SQL Server, outros softwares e personalizações. Esta será sua imagem dourada.

13. Crie uma CEV fornecendo o ID da AMI da imagem que você criou. Para obter detalhes das etapas, consulte, Criar uma CEV para RDS Custom para SQL Server.

14. Crie uma instância de banco de dados do RDS Custom para SQL Server com o uso da CEV. Para obter detalhes das etapas, consulte, Criar uma instância de banco de dados do RDS Custom para SQL Server a partir de uma CEV.

Preparar uma CEV usando o SQL Server pré-instalado (LI)

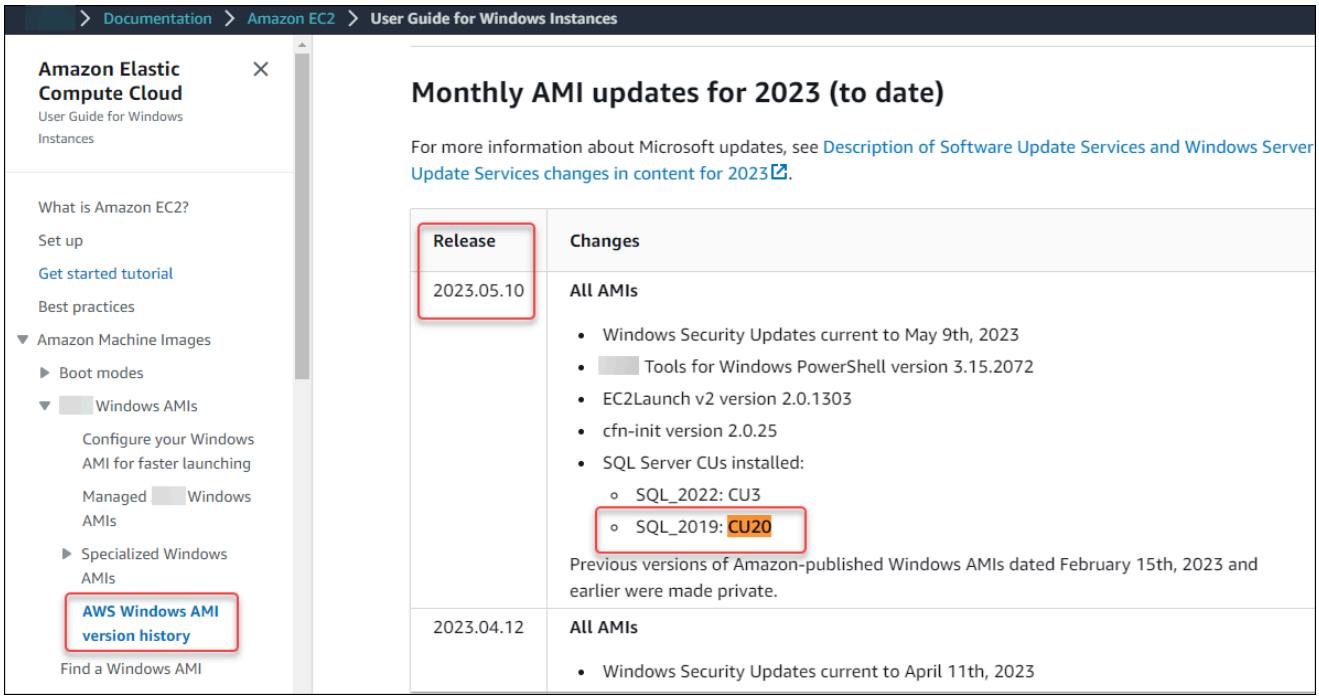
As etapas a seguir para criar uma CEV usando o Microsoft SQL Server pré-instalado (LI) usam uma AMI com o número de versão `2023.05.10` do SQL Server CU20 como exemplo. Ao criar uma CEV, escolha uma AMI com o número de versão mais recente. Isso garante que você esteja usando uma versão compatível do Windows Server e do SQL Server com a atualização cumulativa (CU) mais recente.

Como criar uma CEV usando o Microsoft SQL Server pré-instalado (LI)

1. Escolha a imagem de máquina da Amazon (AMI) do AWS EC2 para Windows mais recente disponível com Microsoft Windows Server e SQL Server com licença inclusa (LI).

   a. Pesquise CU20 no histórico de versões de AMI para Windows.

   b. Anote o número da versão. Para o SQL Server 2019 CU20, o número da versão é `2023.05.10`.



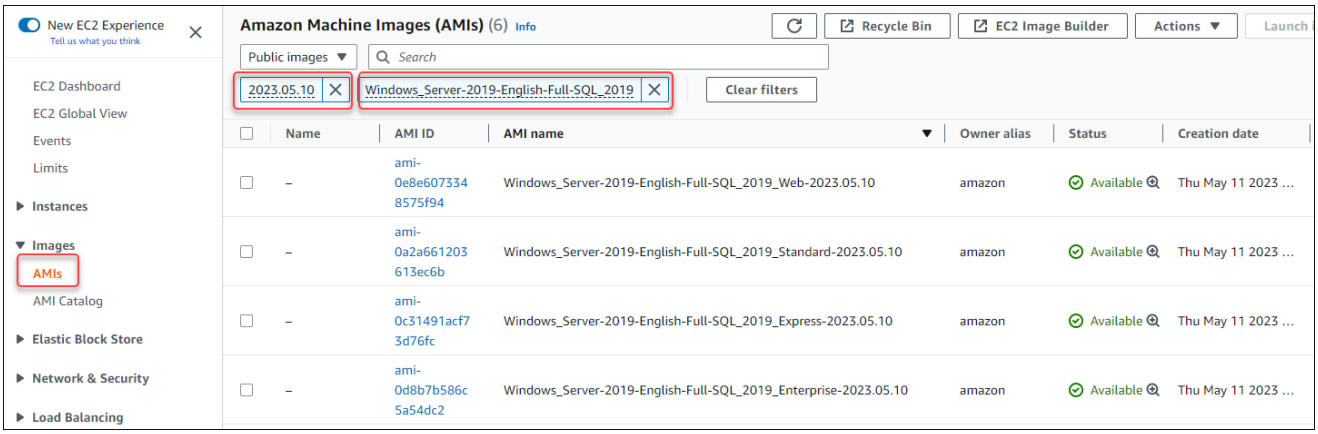
   c. Abra o console do Amazon EC2 em https://console.aws.amazon.com/ec2/.

d. No painel de navegação esquerdo do console do Amazon EC2, selecione Images (Imagens) e, depois, AMIs.

e. Escolha Imagens públicas.

f. Insira `2023.05.10` na caixa de pesquisa. Uma lista de AMIs é exibida.

g. Insira `Windows_Server-2019-English-Full-SQL_2019` na caixa de pesquisa para filtrar os resultados. Os resultados a seguir devem ser exibidos.



h. Selecione a AMI com a edição do SQL Server que deseja usar.

2. Crie ou execute uma instância do EC2 a partir da AMI selecionada.

3. Faça login na instância do EC2 e instale software adicional ou personalize a configuração do sistema operacional e do banco de dados para atender aos seus requisitos.

4. Execute o Sysprep na instância do EC2. Para obter mais informações sobre como preparar uma AMI utilizando o Sysprep, consulte Criar uma imagem de máquina da Amazon (AMI) padronizada usando o Sysprep.

5. Salve a AMI que contém sua versão instalada do SQL Server, outros softwares e personalizações. Esta será sua imagem dourada.

6. Crie uma CEV fornecendo o ID da AMI da imagem que você criou. Para saber as etapas detalhadas sobre a criação de uma CEV, consulte Criar uma CEV para RDS Custom para SQL Server.

7. Crie uma instância de banco de dados do RDS Custom para SQL Server com o uso da CEV. Para obter detalhes das etapas, consulte, Criar uma instância de banco de dados do RDS Custom para SQL Server a partir de uma CEV.

Disponibilidade de regiões para CEVs do RDS Custom para SQL Server

O suporte à versão personalizada do mecanismo (CEV) para o RDS Custom para SQL Server está disponível nas seguintes Regiões da AWS:

- Leste dos EUA (Ohio)
- Leste dos EUA (N. da Virgínia)
- Oeste dos EUA (Oregon)
- Oeste dos EUA (Norte da Califórnia)
- Ásia-Pacífico (Mumbai)
- Ásia-Pacífico (Osaka)
- Ásia-Pacífico (Seul)
- Ásia-Pacífico (Singapura)
- Ásia-Pacífico (Sydney)
- Ásia-Pacífico (Tóquio)
- Canadá (Central)
- Europa (Frankfurt)
- Europa (Irlanda)
- Europa (Londres)
- Europa (Paris)
- Europa (Estocolmo)
- América do Sul (São Paulo)

Versões compatíveis com CEVs do RDS Custom para SQL Server

A criação de CEV para o RDS Custom para SQL Server é compatível com as seguintes AMIs do AWS EC2 para Windows:

- Para CEVs que utilizam mídia pré-instalada, as AMIs do Windows do AWS EC2 com Microsoft Windows Server 2019 (SO) e SQL Server 2022 ou 2019 com licença inclusa (LI).
- Para CEVs que utilizam o modelo BYOM (traga sua própria mídia), as AMIs do Windows do AWS EC2 com Microsoft Windows Server 2019 (SO).

A criação de CEV para o RDS Custom para SQL Server é compatível com as seguintes edições de sistema operacional (SO) e banco de dados:

- Para CEVs que utilizam mídia pré-instalada:
  - SQL Server 2022 Enterprise, Standard ou Web, com CU9, CU13, CU14-GDR, CU15-GDR, CU16, CU17, CU18 e CU19.
  - SQL Server 2019 Enterprise, Standard ou Web, com CU8, CU17, CU18, CU20, CU24, CU26, CU28-GDR, CU29-GDR, CU30 e CU32.
- Para CEVs que utilizam o modelo BYOM (traga sua própria mídia):
  - SQL Server 2022 Enterprise, Standard ou Developer, com CU9, CU13, CU14-GDR, CU15-GDR, CU16, CU17, CU18 e CU19.
  - SQL Server 2019 Enterprise, Standard ou Developer, com CU8, CU17, CU18, CU20, CU24, CU26, CU28-GDR, CU29-GDR, CU30 e CU32.
- Para CEVs que utilizam mídia pré-instalada ou o modelo traga a sua própria mídia (BYOM), o Windows Server 2019 é o único SO compatível.

Consulte mais informações em AWS Windows AMI version history.

Requisitos para CEVs do RDS Custom para SQL Server

Os requisitos a seguir aplicam-se à criação de uma CEV para RDS Custom para SQL Server:

- A AMI utilizada para criar uma CEV deve se basear em uma configuração de SO e banco de dados compatível com o RDS Custom para SQL Server. Para obter mais informações sobre configurações compatíveis, consulte Requisitos e limitações do Amazon RDS Custom for SQL Server.
- A CEV deve ter um nome exclusivo. Não é possível criar uma CEV com o mesmo nome de uma CEV já existente.
- Você deve nomear a CEV utilizando o padrão de nomenclatura exigido do SQL Server versão principal +versão secundária + string personalizada. A versão principal + versão secundária devem corresponder à versão do SQL Server fornecida com a AMI. Por exemplo, você pode nomear uma AMI com SQL Server 2019 CU17 como 15.00.4249.2.my_cevtest.
- Você deve preparar uma AMI com o uso do Sysprep. Para obter mais informações sobre como preparar uma AMI utilizando o Sysprep, consulte Criar uma imagem de máquina da Amazon (AMI) padronizada usando o Sysprep.

- Você é responsável por manter o ciclo de vida da AMI. Uma instância de banco de dados do RDS Custom para SQL Server criada a partir de uma CEV não armazena uma cópia da AMI. Ela mantém um ponteiro para a AMI que você utilizou para criar a CEV. Deve existir uma AMI para que uma instância de banco de dados do RDS Custom para SQL Server permaneça funcional.

Limitações para CEVs do RDS Custom para SQL Server

As limitações a seguir se aplicam às versões de mecanismo personalizadas com RDS Custom para SQL Server:

- Você não poderá excluir uma CEV se houver recursos, como instâncias de banco de dados ou snapshots de banco de dados, associados a ela.
- Para criar uma instância de banco de dados do RDS Custom para SQL Server, uma CEV deve ter o status `pending-validation`, `available`, `failed` ou `validating`. Não é possível criar uma instância de banco de dados do RDS Custom para SQL Server utilizando uma CEV se o status da CEV for `incompatible-image-configuration`.
- Para modificar uma instância de banco de dados do RDS Custom para SQL Server para utilizar uma nova CEV, a CEV deve ter o status `available`.
- Não é possível criar uma AMI ou CEV utilizando uma instância de banco de dados existente do RDS Custom para SQL Server.
- Não é possível modificar uma CEV existente para usar uma AMI diferente. No entanto, é possível modificar uma instância de banco de dados existente do RDS Custom para SQL Server para utilizar uma CEV diferente. Para obter mais informações, consulte Modificar uma instância de banco de dados do RDS Custom for SQL Server.
- Não há suporte para criptografar uma AMI ou uma CEV com uma chave do KMS gerenciada pelo cliente diferente da chave do KMS fornecida durante a criação da instância de banco de dados.
- A cópia entre regiões de CEVs não é compatível.
- A cópia entre contas de CEVs não é compatível.
- Não é possível restaurar ou recuperar uma CEV depois de excluí-la. No entanto, é possível criar uma CEV utilizando a mesma AMI.
- Uma instância de banco de dados do RDS Custom para SQL Server armazena seus arquivos de banco de dados do SQL Server na unidade D:\. A AMI associada a uma CEV deve armazenar os arquivos de banco de dados do sistema Microsoft SQL Server na unidade C:\.
- Uma instância de banco de dados do RDS Custom para SQL Server retém as alterações de configuração feitas no SQL Server. As alterações de configuração feitas no SO em uma instância

de banco de dados do RDS Custom para SQL Server em execução usando uma CEV não são retidas. Se você precisar fazer uma alteração permanente na configuração do sistema operacional e retê-la como sua nova configuração de referência, crie uma CEV e modifique a instância de banco de dados para utilizá-la.

> **Important**
>
> A modificação de uma instância de banco de dados do RDS Custom para SQL Server para utilizar uma nova CEV é uma operação off-line. Você pode realizar a modificação imediatamente ou programá-la para ocorrer durante uma janela de manutenção semanal.

- Quando você modifica uma CEV, o Amazon RDS não envia essas modificações para nenhuma instância de banco de dados associada do RDS Custom para SQL Server. Modifique cada instância de banco de dados do RDS Custom para SQL Server para usar uma CEV nova ou atualizada. Para obter mais informações, consulte Modificar uma instância de banco de dados do RDS Custom for SQL Server.
-

> **Important**
>
> Se uma AMI utilizada por uma CEV for excluída, ocorrerá uma falha em qualquer modificação que possa exigir a substituição do host, por exemplo, escalar a computação. A instância de banco de dados do RDS Custom para SQL Server será então colocada fora do perímetro de suporte do RDS Custom para SQL Server. Recomendamos que você evite excluir qualquer AMI associada a uma CEV.

Criar uma CEV para RDS Custom para SQL Server

Você pode criar uma versão de mecanismo personalizada (CEV) com o uso do AWS Management Console ou da AWS CLI. Depois disso, você pode utilizar a CEV para criar uma instância de banco de dados do RDS Custom para SQL Server.

A imagem de máquina da Amazon (AMI) está na mesma conta e região da AWS que sua CEV. Caso contrário, o processo de criação do CEV falhará.

Para ter mais informações, consulte Criar e conectar-se a uma instância de banco de dados Amazon RDS Custom for SQL Server.

> **Important**
>
> As etapas para criar uma CEV são as mesmas para AMIs criadas com o SQL Server pré-instalado e aquelas criadas usando o modelo BYOM (traga sua própria mídia).

Console

Para criar um CEV

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Custom engine versions (Versões de mecanismos personalizadas).

   A página Custom engine versions (Versões de mecanismo personalizadas) mostra todas as CEVs existentes. Se você não tiver criado nenhuma CEV, a tabela estará em branco.
3. Escolha Create custom engine version (Criar versão de mecanismo personalizada).
4. Em Engine type (Tipo de mecanismo), selecione Microsoft SQL Server.
5. Em Edição, selecione a edição do mecanismo de banco de dados que deseja usar.
6. Em Major version (Versão principal), selecione a versão principal do mecanismo que está instalada em sua AMI.
7. Em Version details (Detalhes da versão), digite um nome válido em Custom engine version name (Nome da versão de mecanismo personalizada).

   O formato do nome é *major-engine-version*.*minor-engine-version*.*customized_string*. Você pode usar de 1 a 50 caracteres alfanuméricos, sublinhados, travessões e pontos. Por exemplo, é possível inserir o nome **15.00.4249.2.my_cevtest**.

   Opcionalmente, insira uma descrição para a CEV.
8. Em Installation Media (Mídia de instalação), procure ou insira o ID da AMI a partir do qual você gostaria de criar a CEV.
9. Na seção Tags, adicione todas as tags para identificar a CEV.
10. Escolha Create custom engine version (Criar versão de mecanismo personalizada).

A página Custom engine versions (Versões de mecanismo personalizadas) é exibida. Sua CEV é mostrada com o status de pending-validation (validação pendente)

AWS CLI

Para criar uma CEV utilizando a AWS CLI, execute o comando create-custom-db-engine-version.

São necessárias as seguintes opções:

- `--engine`
- `--engine-version`
- `--image-id`

Também é possível especificar as seguintes opções:

- `--description`
- `--region`
- `--tags`

O exemplo a seguir cria uma CEV chamada `15.00.4249.2.my_cevtest`. O nome de sua CEV deve começar com o número da versão principal do mecanismo.

Example

Para Linux, macOS ou Unix:

```
aws rds create-custom-db-engine-version \
    --engine custom-sqlserver-ee \
    --engine-version 15.00.4249.2.my_cevtest \
    --image-id ami-0r93cx31t5r596482 \
    --description "Custom SQL Server EE 15.00.4249.2 cev test"
```

A saída parcial a seguir mostra o mecanismo, os grupos de parâmetros e outras informações.

```
"DBEngineVersions": [
    {
    "Engine": "custom-sqlserver-ee",
    "MajorEngineVersion": "15.00",
    "EngineVersion": "15.00.4249.2.my_cevtest",
```

```
    "DBEngineDescription": "Microsoft SQL Server Enterprise Edition for RDS Custom for
 SQL Server",
    "DBEngineVersionArn": "arn:aws:rds:us-east-1:<my-account-id>:cev:custom-sqlserver-
ee/15.00.4249.2.my_cevtest/a1234a1-123c-12rd-bre1-1234567890",
    "DBEngineVersionDescription": "Custom SQL Server EE 15.00.4249.2 cev test",

    "Image": [
        "ImageId": "ami-0r93cx31t5r596482",
        "Status": "pending-validation"
     ],
    "CreateTime": "2022-11-20T19:30:01.831000+00:00",
    "SupportsLogExportsToCloudwatchLogs": false,
    "SupportsReadReplica": false,
    "Status": "pending-validation",
    "SupportsParallelQuery": false,
    "SupportsGlobalDatabases": false,
    "TagList": []
    }
]
```

Se o processo de criação de uma CEV falhar, o RDS Custom para SQL Server emitirá `RDS-EVENT-0198` com a mensagem `Creation failed for custom engine version` *`major-engine-version.cev_name`*. A mensagem inclui detalhes sobre a falha, por exemplo, o evento imprime arquivos ausentes. Para encontrar ideias de solução de problemas na criação da CEV, consulte Solucionar erros de CEV para o RDS Custom para SQL Server.

Criar uma instância de banco de dados do RDS Custom para SQL Server a partir de uma CEV

Depois de criar uma CEV com êxito, o CEV status (Status da CEV) exibe `pending-validation`. Agora você pode criar uma instância de banco de dados do RDS Custom para SQL Server com o uso da CEV. Para criar uma instância de banco de dados do RDS Custom para SQL Server a partir de uma CEV, consulte Criar uma instância de banco de dados do RDS Custom for SQL Server.

Ciclo de vida de uma CEV

O ciclo de vida da CEV inclui os status a seguir.

Status da CEV	Descrição	Sugestões de solução de problemas	
`pending-validation`	Uma CEV foi criada e está	Se não houver nenhuma tarefa existente, crie uma instância de banco de dados	

Status da CEV	Descrição	Sugestões de solução de problemas	
	com a validação pendente da AMI associada. Uma CEV permanecerá em `pending-validation` até que uma instância de banco de dados do RDS Custom para SQL Server seja criada a partir dela.	do RDS Custom para SQL Server a partir da CEV. Ao criar a instância de banco de dados do RDS Custom para SQL Server, o sistema tenta validar a AMI associada de uma CEV.	
`validating`	Uma tarefa de criação da instância de banco de dados do RDS Custom para SQL Server com base em uma nova CEV está em andamento. Ao criar a instância de banco de dados do RDS Custom para SQL Server, o sistema tenta validar a AMI associada de uma CEV.	Aguarde a conclusão da tarefa de criação da instância de banco de dados existente do RDS Custom para SQL Server. Você pode usar o console do RDS EVENTS para revisar mensagens detalhadas de eventos para solução de problemas.	

Status da CEV	Descrição	Sugestões de solução de problemas	
`available`	A CEV foi validada com êxito. Uma CEV entrará no status `available` quando uma instância de banco de dados do RDS Custom para SQL Server tiver sido criada com êxito a partir dela.	A CEV não exige nenhuma validação adicional. Ela pode ser usada para criar instâncias de banco de dados do RDS Custom para SQL Server ou modificar as existentes.	
`inactive`	A CEV foi modificada para um estado inativo.	Você não pode criar nem atualizar uma instância de banco de dados do RDS Custom com essa CEV. Além disso, não é possível restaurar um snapshot de banco de dados para criar uma instância de banco de dados do RDS Custom com essa CEV. Para obter informações sobre como alterar o estado para `ACTIVE`, consulte Modificar uma CEV para RDS Custom para SQL Server.	

Status da CEV	Descrição	Sugestões de solução de problemas	
`failed`	A etapa de criação da instância de banco de dados falhou para essa CEV antes de ela validar a AMI. Como alternativa, a AMI subjacente usada pela CEV não está em um estado disponível.	Solucione a causa raiz do motivo pelo qual o sistema não conseguiu criar a instância de banco de dados. Veja a mensagem de erro detalhada e tente criar uma instância de banco de dados novamente. A AMI subjacente usada pela CEV não deve estar em um estado disponível.	

Status da CEV	Descrição	Sugestões de solução de problemas
`incompatible-image-configuration`	Houve um erro ao validar a AMI.	Veja os detalhes técnicos do erro. Você não pode tentar validar a AMI com essa CEV novamente. Analise o seguinte: recomendações: • Sua CEV deve ser nomeada utilizando o padrão de nomenclatura exigido do SQL Server versão principal +versão secundária + string personalizada. • A versão do SQL Server no nome da CEV deve corresponder à versão fornecida com a AMI. • A versão de compilação do sistema operacional deve atender à versão de compilação mínima exigida. • A versão principal do sistema operacional deve atender à versão mínima exigida. Crie uma CEV com as informações corretas. Se necessário, crie uma instância do EC2 utilizando uma AMI compatível e execute o processo Sysprep nela.

Modificar uma CEV para RDS Custom para SQL Server

Você pode modificar uma CEV utilizando a AWS Management Console ou a AWS CLI. Você pode modificar a descrição da CEV ou seu status de disponibilidade. Sua CEV tem um dos seguintes valores de status:

- `available` – Você pode utilizar essa CEV para criar uma nova instância de banco de dados do RDS Custom ou fazer upgrade de uma instância de banco de dados. Esse é o status padrão para uma CEV recém-criada.

- `inactive`: você não pode criar nem atualizar uma instância de banco de dados do RDS Custom com essa CEV. Não é possível restaurar um DB snapshot para criar uma nova instância de banco de dados do RDS Custom com essa CEV.

Você pode alterar o status da CEV de `available` para `inactive` ou de `inactive` para `available`. Você pode alterar o status para `INACTIVE` para evitar o uso acidental de uma CEV ou tornar uma CEV descontinuada elegível para uso novamente.

Console

Para modificar uma CEV

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Custom engine versions (Versões de mecanismos personalizadas).
3. Escolha uma CEV cuja descrição ou status você queira modificar.
4. Para Actions (Ações), escolha Modify (Modificar).
5. Faça uma ou todas as alterações a seguir:
   - Para CEV status settings (Configurações de status da CEV), escolha um novo status de disponibilidade.
   - Para Version description (Descrição da versão), insira uma nova descrição.
6. Selecione Modify CEV (Modificar CEV).

   Se a CEV estiver em uso, o console exibirá You can't modify the CEV status (Não é possível modificar o status da CEV). Corrija os problemas e tente novamente.

A página Custom engine versions (Versões de mecanismo personalizadas) é exibida.

AWS CLI

Para modificar uma CEV utilizando a AWS CLI, execute o comando modify-custom-db-engine-version. Você pode encontrar CEVs para modificar executando o comando describe-db-engine-versions.

São necessárias as seguintes opções:

- `--engine`
- `--engine-version` *`cev`*, em que *`cev`* é o nome da versão de mecanismo personalizada que você deseja modificar
- `--status` *`status`*, em que *`status`* é o status de disponibilidade que você deseja atribuir à CEV

O exemplo a seguir altera uma CEV chamada `15.00.4249.2.my_cevtest` do status atual para `inactive`.

Example

Para Linux, macOS ou Unix:

```
aws rds modify-custom-db-engine-version \
    --engine custom-sqlserver-ee \
    --engine-version 15.00.4249.2.my_cevtest \
    --status inactive
```

Para Windows:

```
aws rds modify-custom-db-engine-version ^
    --engine custom-sqlserver-ee ^
    --engine-version 15.00.4249.2.my_cevtest ^
    --status inactive
```

Para modificar uma instância de banco de dados do RDS Custom para SQL Server para usar uma nova CEV, consulte Modificar uma instância de banco de dados do RDS Custom para SQL Server para usar uma nova CEV.

Modificar uma instância de banco de dados do RDS Custom para SQL Server para usar uma nova CEV

Não é possível modificar uma instância de banco de dados existente do RDS Custom para SQL Server para usar uma CEV diferente. As alterações que você pode fazer são:

- Alterar a CEV
- Alterar a classe da instância de banco de dados
- Alterar o período de retenção de backup e a janela de backup
- Alterar a janela de manutenção

Console

Para modificar uma instância de banco de dados do RDS Custom for SQL Server

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Databases (Bancos de dados).
3. Escolha a instância de banco de dados que você deseja modificar.
4. Selecione Modify.
5. Faça as seguintes alterações, conforme necessário:
   a. Em DB engine version (Versão do mecanismo de banco de dados), selecione outra CEV.
   b. Altere o valor de DB instance class (Classe da instância de banco de dados). Para conhecer as classes com suporte, consulte Compatibilidade com classes de instâncias de banco de dados do RDS Custom for SQL Server.
   c. Altere o valor de Backup retention period (Período de retenção de backup).
   d. Para Backup window (Janela de backup), defina valores para Start time (Horário de início) e Duration (Duração).
   e. Para DB instance maintenance window (Janela de manutenção da instância de banco de dados), defina valores para Start day (Dia de início), Start time (Horário de início) e Duration (Duração).
6. Escolha Continuar.
7. Escolha Apply immediately (Aplicar imediatamente) ou Apply during the next scheduled maintenance window (Aplicar durante a próxima janela de manutenção programada).
8. Selecione Modify DB instance (Modificar instância de banco de dados).

> **Note**
>
> Ao modificar uma instância de banco de dados de uma CEV para outra CEV, por exemplo, ao atualizar uma versão secundária, os bancos de dados do sistema do SQL Server, inclusive seus dados e configurações, permanecem na instância de banco de dados atual do RDS Custom para SQL Server.

AWS CLI

Para modificar uma instância de banco de dados para usar outra CV utilizando a AWS CLI, execute o comando modify-db-instance.

São necessárias as seguintes opções:

- `--db-instance-identifier`
- `--engine-version` *`cev`*, em que *`cev`* é o nome da versão de mecanismo personalizada que você deseja modificar

O exemplo a seguir modifica uma instância de banco de dados denominada `my-cev-db-instance` para usar uma CEV denominada `15.00.4249.2.my_cevtest_new` e aplica a alteração imediatamente.

Example

Para Linux, macOS ou Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier my-cev-db-instance \
    --engine-version 15.00.4249.2.my_cevtest_new \
    --apply-immediately
```

Para Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier my-cev-db-instance ^
    --engine-version 15.00.4249.2.my_cevtest_new ^
    --apply-immediately
```

Visualizar detalhes da CEV com o Amazon RDS Custom para SQL Server

Você pode ver detalhes sobre sua CEV utilizando o AWS Management Console ou a AWS CLI.

Console

Como visualizar detalhes da CEV

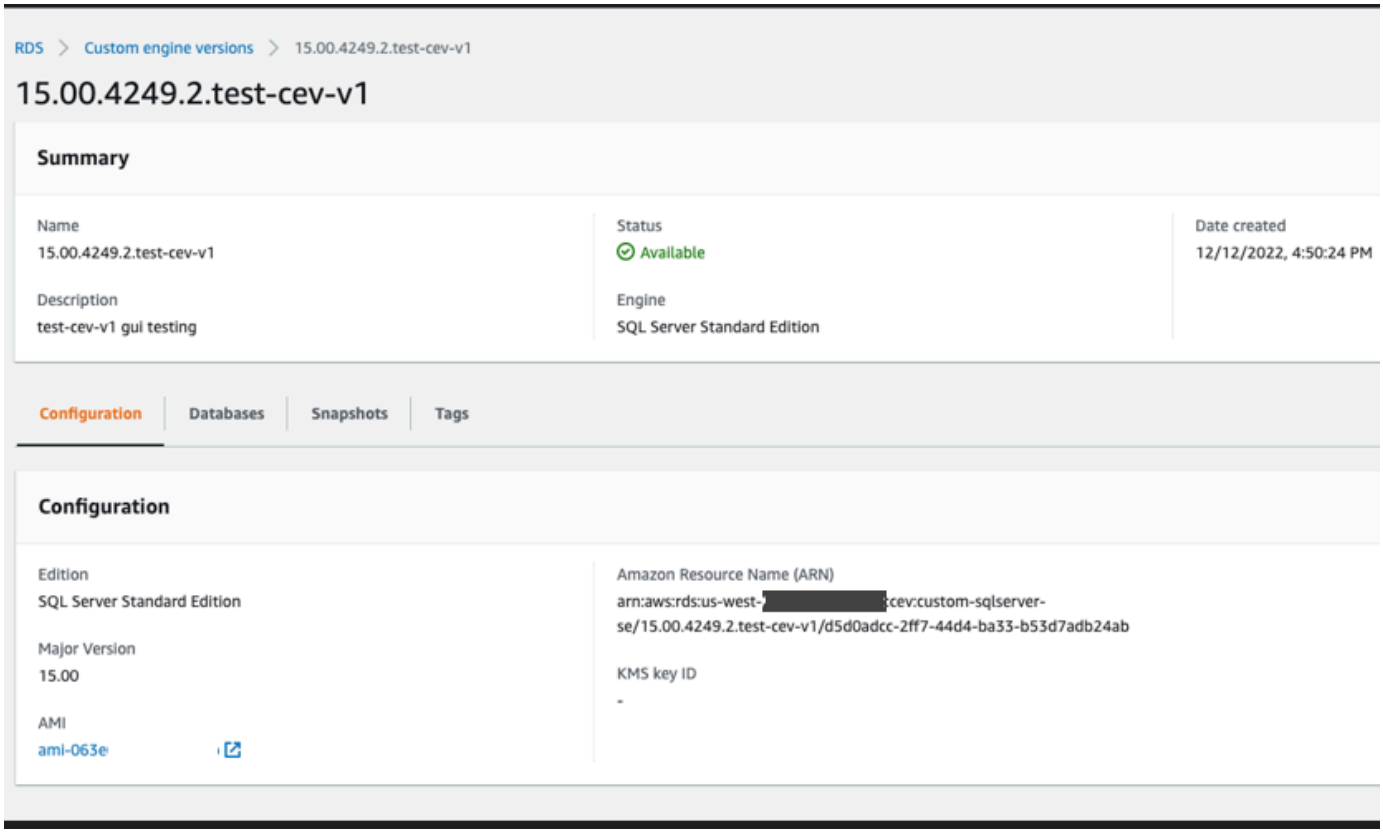
1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.

2. No painel de navegação, escolha Custom engine versions (Versões de mecanismos personalizadas).

   A página Custom engine versions (Versões de mecanismo personalizadas) mostra todas as CEVs existentes. Se você não tiver criado uma CEV, a página estará vazia.

3. Selecione o nome da CEV que você deseja visualizar.

4. Selecione Configuration (Configuração) para visualizar os detalhes.



AWS CLI

Para ver detalhes sobre uma CEV com o uso da AWS CLI, execute o comando describe-db-engine-versions.

Também é possível especificar as seguintes opções:

- `--include-all`, para visualizar todas as CEVs com qualquer estado do ciclo de vida. Sem a opção `--include-all`, somente as CEVs em um estado de ciclo de vida `available` serão retornadas.

```
aws rds describe-db-engine-versions --engine custom-sqlserver-ee --engine-version
 15.00.4249.2.my_cevtest --include-all
{
    "DBEngineVersions": [
        {
            "Engine": "custom-sqlserver-ee",
            "MajorEngineVersion": "15.00",
            "EngineVersion": "15.00.4249.2.my_cevtest",
            "DBParameterGroupFamily": "custom-sqlserver-ee-15.0",
            "DBEngineDescription": "Microsoft SQL Server Enterprise Edition for custom
 RDS",
            "DBEngineVersionArn": "arn:aws:rds:us-east-1:{my-account-id}:cev:custom-
sqlserver-ee/15.00.4249.2.my_cevtest/a1234a1-123c-12rd-bre1-1234567890",
            "DBEngineVersionDescription": "Custom SQL Server EE 15.00.4249.2 cev test",
            "Image": {
                "ImageId": "ami-0r93cx31t5r596482",
                "Status": "pending-validation"
            },
            "DBEngineMediaType": "AWS Provided",
            "CreateTime": "2022-11-20T19:30:01.831000+00:00",
            "ValidUpgradeTarget": [],
            "SupportsLogExportsToCloudwatchLogs": false,
            "SupportsReadReplica": false,
            "SupportedFeatureNames": [],
            "Status": "pending-validation",
            "SupportsParallelQuery": false,
            "SupportsGlobalDatabases": false,
            "TagList": [],
            "SupportsBabelfish": false
        }
    ]
}
```

Você pode utilizar filtros para visualizar CEVs com determinado status de ciclo de vida. Por exemplo, para visualizar CEVs com um status de ciclo de vida de `pending-validation`, `available` ou `failed`:

```
aws rds describe-db-engine-versions engine custom-sqlserver-ee
                region us-west-2 include-all query 'DBEngineVersions[?Status ==
 pending-validation ||
                Status == available || Status == failed]'
```

Excluir uma CEV para RDS Custom para SQL Server

É possível excluir uma CEV usando o AWS Management Console ou a AWS CLI. Normalmente, essa tarefa demora alguns minutos.

Antes de excluir uma CEV, verifique se ela não está em uso por nenhum dos seguintes procedimentos:

- Uma instância de banco de dados do RDS Custom
- Um snapshot de uma instância de banco de dados do RDS Custom
- Um backup automatizado da sua instância de banco de dados do RDS Custom

Console

Para excluir uma CEV

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Custom engine versions (Versões de mecanismos personalizadas).
3. Escolha uma CEV cuja descrição ou status você deseja excluir.
4. Em Ações, escolha Excluir.

   A caixa de diálogo Delete ***cev_name***? (Excluir cev_name?) é exibida.
5. Insira **`delete me`** e escolha Delete (Excluir).

   Na página Custom engine versions (Versões de mecanismos personalizadas), o banner mostra que sua CEV está sendo excluída.

AWS CLI

Para excluir uma CEV utilizando a AWS CLI, execute o comando delete-custom-db-engine-version.

São necessárias as seguintes opções:

- `--engine custom-sqlserver-ee`
- `--engine-version` *`cev`*, em que *`cev`* é o nome da versão de mecanismo personalizada a ser excluída

O exemplo a seguir exclui uma CEV chamada `15.00.4249.2.my_cevtest`.

Example

Para Linux, macOS ou Unix:

```
aws rds delete-custom-db-engine-version \
    --engine custom-sqlserver-ee \
    --engine-version 15.00.4249.2.my_cevtest
```

Para Windows:

```
aws rds delete-custom-db-engine-version ^
    --engine custom-sqlserver-ee ^
    --engine-version 15.00.4249.2.my_cevtest
```

Criar e conectar-se a uma instância de banco de dados Amazon RDS Custom for SQL Server

Você pode criar uma instância de banco de dados do RDS Custom e, em seguida, conectar-se a ela utilizando AWS Systems Manager ou o protocolo RDP.

> **Important**
>
> Antes de criar ou conectar-se a uma instância de banco de dados do RDS Custom for SQL Server, certifique-se de concluir as tarefas em Configurar seu ambiente para o Amazon RDS Custom for SQL Server.
>
> É possível marcar instâncias de banco de dados do RDS Custom ao criá-las, mas não é possível criar ou modificar a etiqueta `AWSRDSCustom` necessária para a automação do RDS Custom. Para ter mais informações, consulte Marcar recursos do RDS Custom for SQL Server.
>
> Ao criar uma instância de banco de dados do RDS Custom for SQL Server pela primeira vez, você pode receber o seguinte erro: The service-linked role is in the process of being created (A função vinculada ao serviço está no processo de criação). Tente novamente mais tarde. Se você fizer isso, aguarde alguns minutos e tente recriar a instância de banco de dados.

Tópicos

- Criar uma instância de banco de dados do RDS Custom for SQL Server
- Função vinculada ao serviço do RDS Custom
- Conectar-se à sua instância de banco de dados do RDS Custom via AWS Systems Manager
- Conectar-se à sua instância de banco de dados do RDS Custom via RDP

Criar uma instância de banco de dados do RDS Custom for SQL Server

Crie uma instância de banco de dados Amazon RDS Custom for SQL Server usando o AWS Management Console ou a AWS CLI. O procedimento é semelhante ao de criação de uma instância de banco de dados do Amazon RDS.

Para obter mais informações, consulte Criar uma instância de banco de dados do Amazon RDS.

Console

Para criar uma instância de banco de dados do RDS Custom for SQL Server

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Bancos de dados.
3. Selecione Criar banco de dados.
4. Escolha Standard create (Criação padrão) para o método de criação do banco de dados.
5. Em Engine options (Opções de mecanismo), escolha Microsoft SQL Server para o tipo de mecanismo.
6. Para Database management type (Tipo de gerenciamento de banco de dados), escolha Amazon RDS Custom.
7. Na seção Edition (Edição), escolha a edição do mecanismo de banco de dados do SQL Server que você deseja usar.
8. (Opcional) Se você pretende criar a instância de banco de dados a partir de uma CEV, marque a caixa de seleção Use custom engine version (CEV) (Usar versão personalizada do mecanismo (CEV)). Selecione sua CEV na lista suspensa.
9. Para Versão do banco de dados, mantenha a versão do valor padrão.
10. Para Templates (Modelos), escolha Production (Produção).
11. Na seção Settings (Configurações), insira um novo nome para DB instance identifier (Identificador da instância de banco de dados).
12. Para inserir sua senha mestre, faça o seguinte:

    a. Na seção Settings (Configurações), abra Credential Settings (Configurações de credencial).

    b. Desmarque a caixa de seleção Auto generate a password (Gerar uma senha automaticamente).

    c. Altere o valor Master username (Nome de usuário primário) e insira a mesma senha em Master password (Senha primária) e Confirm password (Confirmar senha).

    Por padrão, a nova instância de banco de dados do RDS Custom utiliza uma senha gerada automaticamente para o usuário mestre.
13. Na seção DB instance size (Tamanho da instância de banco), escolha um valor para DB instance class (Classe da instância de banco de dados).

Para conhecer as classes com suporte, consulte Compatibilidade com classes de instâncias de banco de dados do RDS Custom for SQL Server.

14. Escolha configurações de Storage (Armazenamento).
15. Para RDS Custom security (Segurança do RDS Custom), faça o seguinte:

    a. Em Perfil de instância do IAM, você tem duas opções para escolher o perfil da instância de banco de dados do RDS Custom para SQL Server.

        1. Selecione Criar um perfil de instância e forneça um sufixo de nome de perfil de instância. Para ter mais informações, consulte Criação automática de perfis de instância usando o AWS Management Console.
        2. Escolha um perfil de instância existente. Na lista suspensa, escolha o perfil de instância que começa com `AWSRDSCustom`.

    b. Para Encryption (Criptografia), escolha Enter a key ARN (Insira um ARN chave) para listar as chaves do AWS KMS disponíveis. Em seguida, escolha sua chave na lista.

        Uma chave do AWS KMS é necessária para o RDS Custom. Para obter mais informações, consulte Verifique se você tem uma chave do AWS KMS de criptografia simétrica..

16. Nas seções restantes, especifique suas configurações preferenciais de instâncias de banco de dados do RDS Custom. Para obter informações sobre cada configuração, consulte Configurações para instâncias de banco de dados. As configurações a seguir não têm suporte e não aparecem no console:

    - Processor features (Recursos do processador)
    - Storage autoscaling (Autoscaling do armazenamento)
    - Disponibilidade e durabilidade
    - Opção Password and Kerberos authentication (Senha e autenticação Kerberos) em Database authentication (Autenticação do banco de dados) (somente a autenticação com senha tem suporte)
    - Grupo Database options (Opções do banco de dados) em Additional configuration (Configuração adicional)
    - Performance Insights
    - Exportações de log
    - Habilitar a atualização automática da versão secundária
    - Deletion protection (Proteção contra exclusão)

Existe suporte para a opção Backup retention period (Período de retenção de backup), mas você não pode escolher 0 dias.

17. Selecione Criar banco de dados.

    O botão View credential details (Visualizar detalhes de credenciais) aparece na página Databases (Bancos de dados).

    Para visualizar o nome de usuário primário e a senha da instância de banco de dados do RDS Custom, escolha View credential details (Visualizar detalhes de credenciais) .

    Para se conectar à instância de banco de dados como o usuário mestre, use o nome de usuário e a senha exibidos.

    > **Important**
    >
    > Você não pode visualizar a senha do usuário principal novamente. Caso você não a registre, talvez seja necessário alterá-la. Para alterar a senha do usuário mestre depois que a instância do banco de dados RDS Custom estiver disponível, modifique a instância de banco de dados. Para ter mais informações sobre a modificação de uma instância de banco de dados, consulte Gerenciar uma instância de banco de dados do Amazon RDS Custom for SQL Server.

18. Escolha Databases (Bancos de dados) para visualizar a lista de instâncias de banco de dados do RDS Custom.
19. Escola a instância de banco de dados do RDS Custom que você acabou de criar.

    No console do RDS, os detalhes da nova instância de banco de dados do RDS Custom são exibidos:

    - A instância de banco de dados do RDS Custom apresenta o status Creating (Criando) até que esteja criada e pronta para uso. Quando o status muda para available (disponível), você pode se conectar à instância de banco de dados. Dependendo da classe da instância e do armazenamento alocado, pode levar alguns minutos até que a nova instância de banco de dados fique disponível.
    - Role (Função) tem o valor Instância (RDS Custom).

- RDS Custom automation mode (Modo de automação do RDS Custom) tem o valor Full automation (Automação total). Essa configuração significa que a instância de banco de dados fornece monitoramento automático e recuperação de instâncias.

AWS CLI

Você cria uma instância de banco de dados do RDS Custom utilizando o comando create-db-instance da AWS CLI.

São necessárias as seguintes opções:

- `--db-instance-identifier`
- `--db-instance-class` (para obter uma lista de classes de instância de banco de dados com suporte, consulte Compatibilidade com classes de instâncias de banco de dados do RDS Custom for SQL Server)
- `--engine` (`custom-sqlserver-ee`, `custom-sqlserver-se` ou `custom-sqlserver-web`)
- `--kms-key-id`
- `--custom-iam-instance-profile`

O exemplo a seguir cria uma instância de banco de dados do RDS Custom for SQL Server chamada `my-custom-instance`. Defina o período de retenção de backup como 3 dias.

> **Note**
>
> Para criar uma instância de banco de dados a partir de uma versão de mecanismo personalizada (CEV), forneça um nome de CEV existente para o parâmetro `--engine-version`. Por exemplo, `--engine-version 15.00.4249.2.my_cevtest`

Example

Para Linux, macOS ou Unix:

```
aws rds create-db-instance \
    --engine custom-sqlserver-ee \
    --engine-version 15.00.4073.23.v1 \
    --db-instance-identifier my-custom-instance \
    --db-instance-class db.m5.xlarge \
```

```
    --allocated-storage 20 \
    --db-subnet-group mydbsubnetgroup \
    --master-username myuser \
    --master-user-password mypassword \
    --backup-retention-period 3 \
    --no-multi-az \
    --port 8200 \
    --kms-key-id mykmskey \
    --custom-iam-instance-profile AWSRDSCustomInstanceProfileForRdsCustomInstance
```

Para Windows:

```
aws rds create-db-instance ^
    --engine custom-sqlserver-ee ^
    --engine-version 15.00.4073.23.v1 ^
    --db-instance-identifier my-custom-instance ^
    --db-instance-class db.m5.xlarge ^
    --allocated-storage 20 ^
    --db-subnet-group mydbsubnetgroup ^
    --master-username myuser ^
    --master-user-password mypassword ^
    --backup-retention-period 3 ^
    --no-multi-az ^
    --port 8200 ^
    --kms-key-id mykmskey ^
    --custom-iam-instance-profile AWSRDSCustomInstanceProfileForRdsCustomInstance
```

> **Note**
>
> Especifique uma senha diferente do prompt mostrado aqui como prática recomendada de segurança.

Obtenha detalhes sobre a sua instância com o comando da `describe-db-instances`.

```
aws rds describe-db-instances --db-instance-identifier my-custom-instance
```

A saída parcial a seguir mostra o mecanismo, os grupos de parâmetros e outras informações.

```
{
    "DBInstances": [
```

```
        {
            "PendingModifiedValues": {},
            "Engine": "custom-sqlserver-ee",
            "MultiAZ": false,
            "DBSecurityGroups": [],
            "DBParameterGroups": [
                {
                    "DBParameterGroupName": "default.custom-sqlserver-ee-15",
                    "ParameterApplyStatus": "in-sync"
                }
            ],
            "AutomationMode": "full",
            "DBInstanceIdentifier": "my-custom-instance",
            "TagList": []
        }
    ]
}
```

Função vinculada ao serviço do RDS Custom

Uma função vinculada ao serviço concede ao Amazon RDS Custom acesso a recursos em sua Conta da AWS. Isso facilita a configuração do RDS Custom, pois você não precisa adicionar manualmente as permissões necessárias. O RDS Custom define as permissões de suas funções vinculadas a serviços e, exceto se definido de outra maneira, somente o RDS Custom pode assumir suas funções. As permissões definidas incluem a política de confiança e a política de permissões, e essa política de permissões não pode ser anexada a nenhuma outra entidade do IAM.

Quando você cria uma instância de banco de dados do RDS Custom, as funções vinculadas a serviços do Amazon RDS e do RDS Custom são criadas (se ainda não existirem) e utilizadas. Para obter mais informações, consulte Usar funções vinculadas ao serviço do Amazon RDS.

Ao criar uma instância de banco de dados do RDS Custom for SQL Server pela primeira vez, você pode receber o seguinte erro: The service-linked role is in the process of being created (A função vinculada ao serviço está no processo de criação). Tente novamente mais tarde. Se você fizer isso, aguarde alguns minutos e tente recriar a instância de banco de dados.

Conectar-se à sua instância de banco de dados do RDS Custom via AWS Systems Manager

Depois de criar sua instância de banco de dados do RDS Custom, você pode conectar-se a ela utilizando o AWS Systems Manager Session Manager. O Session Manager é um recurso do Systems

Manager que pode ser utilizado para gerenciar instâncias do Amazon EC2 por meio de um shell baseado no navegador ou via AWS CLI. Para ter mais informações, consulte o AWS Systems Manager Session Manager.

Console

Para conectar-se à sua instância de banco de dados usando o Session Manager

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Databases (Bancos de dados) e a instância de banco de dados do RDS Custom à qual você deseja se conectar.
3. Escolher configuração.
4. Observe o valor de Resource ID (ID do recurso) da instância de banco de dados. Por exemplo, o ID do recurso pode ser `db-ABCDEFGHIJKLMNOPQRS0123456`.
5. Abra o console do Amazon EC2 em https://console.aws.amazon.com/ec2/.
6. No painel de navegação, escolha Instances (Instâncias).
7. Procure o nome da sua instância do EC2 e depois escolha o ID de instância associado a ela. Por exemplo, o ID da instância pode ser `i-abcdefghijklm01234`.
8. Selecione Conectar.
9. Escolha Session Manager.
10. Selecione Conectar.

    Uma janela é aberta para a sua sessão.

AWS CLI

Você pode conectar-se à sua instância de banco de dados do RDS Custom usando a AWS CLI. Essa técnica requer o plugin Session Manager para a AWS CLI. Para saber como instalar o plugin, consulte Instalar o plugin Session Manager para a AWS CLI.

Para encontrar o ID de recurso de banco de dados da sua instância de banco de dados do RDS Custom, use `describe-db-instances`.

```
aws rds describe-db-instances \
    --query 'DBInstances[*].[DBInstanceIdentifier,DbiResourceId]' \
    --output text
```

A saída de exemplo a seguir mostra o ID de recurso da sua instância do RDS Custom. O prefixo é `db-`.

```
db-ABCDEFGHIJKLMNOPQRS0123456
```

Para encontrar o ID de instância do EC2 da sua instância de banco de dados, use `aws ec2 describe-instances`. O exemplo a seguir usa `db-ABCDEFGHIJKLMNOPQRS0123456` para o ID do recurso.

```
aws ec2 describe-instances \
    --filters "Name=tag:Name,Values=db-ABCDEFGHIJKLMNOPQRS0123456" \
    --output text \
    --query 'Reservations[*].Instances[*].InstanceId'
```

O exemplo de saída a seguir mostra o ID da instância do EC2.

```
i-abcdefghijklm01234
```

Use o comando `aws ssm start-session`, fornecendo o ID da instância do EC2 no parâmetro `--target`.

```
aws ssm start-session --target "i-abcdefghijklm01234"
```

Um resultado bem-sucedido tem a seguinte aparência.

```
Starting session with SessionId: yourid-abcdefghijklm1234
[ssm-user@ip-123-45-67-89 bin]$
```

Conectar-se à sua instância de banco de dados do RDS Custom via RDP

Depois de criar sua instância de banco de dados do RDS Custom, você pode se conectar a ela utilizando um cliente RDP. O procedimento é o mesmo para se conectar a uma instância do Amazon EC2. Para obter mais informações, consulte Conectar-se à sua instância do Windows.

Para conectar-se à instância de banco de dados, você precisa do par de chaves associado a essa instância. O RDS Custom cria o par de chaves para você. O nome do par usa o prefixo `do-not-delete-rds-custom-`*`DBInstanceIdentifier`*. O AWS Secrets Manager armazena sua chave privada como segredo.

Conclua a tarefa nas seguintes etapas:

1. Configurar sua instância de banco de dados para permitir conexões RDP.
2. Recuperar sua chave secreta.
3. Conectar-se à sua instância do EC2 usando o utilitário RDP.

Configurar sua instância de banco de dados para permitir conexões RDP

Para permitir conexões RDP, configure seu grupo de segurança da VPC e defina uma regra de firewall no host.

Configurar o grupo de segurança da sua VPC

Certifique-se de que o grupo de segurança da VPC associado à instância de banco de dados permita conexões de entrada na porta 3389 para o protocolo TCP. Para saber como configurar seu grupo de segurança de VPC, consulte Configurar o grupo de segurança da sua VPC.

Definir a regra de firewall no host

Para permitir conexões de entrada na porta 3389 para TCP, defina uma regra de firewall no host. Os exemplos a seguir mostram como fazer isso.

Recomendamos usar o valor `-Profile` específico: `Public`, `Private`, ou `Domain`. O uso do `Any` refere-se a todos os três valores. Você também pode especificar uma combinação de valores separados por uma vírgula. Para obter mais informações sobre como definir regras de firewall, consulte Set-NetFirewallRule na documentação da Microsoft.

Para utilizar o Systems Manager Session Manager para configurar uma regra de firewall

1. Conecte-se ao Session Manager, conforme mostrado em Conectar-se à sua instância de banco de dados do RDS Custom via AWS Systems Manager.
2. Execute o seguinte comando .

```
Set-NetFirewallRule -DisplayName "Remote Desktop - User Mode (TCP-In)" -Direction
 Inbound -LocalAddress Any -Profile Any
```

Para utilizar comandos da CLI do Systems Manager para configurar uma regra de firewall

1. Utilize o comando a seguir para abrir o RDP no host.

```
OPEN_RDP_COMMAND_ID=$(aws ssm send-command --region $AWS_REGION \
```

```
    --instance-ids $RDS_CUSTOM_INSTANCE_EC2_ID \
    --document-name "AWS-RunPowerShellScript" \
    --parameters '{"commands":["Set-NetFirewallRule -DisplayName \"Remote Desktop -
 User Mode (TCP-In)\" -Direction Inbound -LocalAddress Any -Profile Any"]}' \
    --comment "Open RDP port" | jq -r ".Command.CommandId")
```

2. Utilize o ID do comando retornado na saída para obter o status do comando anterior. Para utilizar a seguinte consulta para retornar o ID do comando, certifique-se de ter o plugin jq instalado.

```
aws ssm list-commands \
    --region $AWS_REGION \
    --command-id $OPEN_RDP_COMMAND_ID
```

Recuperar sua chave secreta

Recupere sua chave secreta utilizando o AWS Management Console ou a AWS CLI.

Console

Para recuperar a chave secreta

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Databases (Bancos de dados) e a instância de banco de dados do RDS Custom à qual você deseja se conectar.
3. Escolha a guia Configuração.
4. Observe o DB instance ID (ID da instância de banco de dados) para sua instância de banco de dados, por exemplo, *my-custom-instance*.
5. Abra o console do Amazon EC2 em https://console.aws.amazon.com/ec2/.
6. No painel de navegação, escolha Instances (Instâncias).
7. Procure o nome da sua instância do EC2 e depois escolha o ID de instância associado a ela.

   Neste exemplo, o ID da instância é `i-abcdefghijklm01234`.
8. Em Details (Detalhes), localize Key pair name (Nome do par de chaves). O nome do par inclui o identificador do banco de dados. Neste exemplo, o nome do par é `do-not-delete-rds-custom-`*`my-custom-instance`*`-`*`0d726c`*.

9. No resumo da instância, localize Public IPv4 DNS (DNS IPv4 público). Para o exemplo, o DNS público pode ser `ec2-12-345-678-901.us-east-2.compute.amazonaws.com`.
10. Abra o console do AWS Secrets Manager em https://console.aws.amazon.com/secretsmanager/.
11. Escolha o segredo com nome idêntico ao seu par de chaves.
12. Escolha Recuperar valor do segredo.

AWS CLI

Para recuperar a chave privada

1. Obtenha a lista das suas instâncias de banco de dados do RDS Custom chamando o comando `aws rds describe-db-instances`.

```
aws rds describe-db-instances \
    --query 'DBInstances[*].[DBInstanceIdentifier,DbiResourceId]' \
    --output text
```

2. Escolha o identificador da instância de banco de dados na saída de amostra, por exemplo, `do-not-delete-rds-custom-`*`my-custom-instance`*.
3. Encontre o ID da instância do EC2 da sua instância de banco de dados chamando o comando `aws ec2 describe-instances`. O exemplo a seguir usa o nome da instância do EC2 para descrever a instância de banco de dados.

```
aws ec2 describe-instances \
    --filters "Name=tag:Name,Values=do-not-delete-rds-custom-my-custom-instance" \
    --output text \
    --query 'Reservations[*].Instances[*].InstanceId'
```

O exemplo de saída a seguir mostra o ID da instância do EC2.

```
i-abcdefghijklm01234
```

4. Localize o nome da chave especificando o ID da instância do EC2, como mostra o exemplo a seguir.

```
aws ec2 describe-instances \
    --instance-ids i-abcdefghijklm01234 \
    --output text \
```

```
    --query 'Reservations[*].Instances[*].KeyName'
```

A saída de exemplo a seguir mostra o nome da chave, que usa o prefixo `do-not-delete-rds-custom-`*`DBInstanceIdentifier`*.

```
do-not-delete-rds-custom-my-custom-instance-0d726c
```

Conectar-se à sua instância do EC2 usando o utilitário RDP

Siga o procedimento em Connect to your Windows instance using RDP no Guia do usuário do Amazon EC2. Esse procedimento pressupõe que você tenha criado um arquivo .pem que contém sua chave privada.

Gerenciar uma instância de banco de dados do Amazon RDS Custom for SQL Server

O Amazon RDS Custom é compatível com um subconjunto das tarefas de gerenciamento habituais de instâncias de banco de dados do Amazon RDS. A seguir, você encontra instruções para as tarefas de gerenciamento compatíveis do RDS Custom for SQL Server usando o AWS Management Console e a AWS CLI.

Tópicos

- Pausar e retomar a automação do RDS Custom
- Modificar uma instância de banco de dados do RDS Custom for SQL Server
- Modificar o armazenamento para uma instância de banco de dados do RDS Custom para SQL Server
- Marcar recursos do RDS Custom for SQL Server
- Excluir uma instância de banco de dados do RDS Custom for SQL Server
- Iniciar e interromper uma instância de banco de dados do RDS Custom para SQL Server

Pausar e retomar a automação do RDS Custom

O RDS Custom fornece automaticamente monitoramento e recuperação de instâncias para uma instância de banco de dados do RDS Custom for SQL Server. Se precisar personalizar essa instância, faça o seguinte:

1. Pause a automação do RDS Custom por um período especificado. A pausa garante que suas personalizações não interfiram na automação do RDS Custom.
2. Personalize a instância de banco de dados do RDS Custom for SQL Server, conforme necessário.
3. Realize um dos procedimentos a seguir:
   - Retome a automação manualmente.
   - Aguarde o período de pausa terminar. Nesse caso, o RDS Custom retoma o monitoramento e a recuperação de instâncias automaticamente.

> **Important**
>
> Pausar e retomar a automação são as únicas tarefas de automação compatíveis ao modificar uma instância de banco de dados do RDS Custom for SQL Server.

Console

Para pausar ou retomar a automação do RDS Custom

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Databases (Bancos de dados) e a instância de banco de dados do RDS Custom que você deseja modificar.
3. Selecione Modify. A página Modify DB instance (Modificar instância de banco de dados) será exibida.
4. Para RDS Custom automation mode (Modo de automação do RDS Custom), selecione uma das seguintes opções:
   - Paused (Pausado) pausa o monitoramento e a recuperação da instância de banco de dados do RDS Custom. Insira a duração da pausa que você deseja (em minutos) para Automation mode duration (Duração do modo de automação). O valor mínimo é 60 minutos (padrão). O valor máximo é 1.440 minutos.
   - Full automation (Automação total) retoma a automação.
5. Escolha Continue (Continuar) e verifique o resumo de modificações.

   Uma mensagem indica que o RDS Custom aplicará as alterações imediatamente.
6. Se estiverem corretas, escolha Modify DB Instance (Modificar instância de banco de dados). Ou escolha Back (Voltar) para editar as alterações ou Cancel (Cancelar) para cancelar as alterações.

   No console do RDS, são exibidos os detalhes da modificação. Se você pausou a automação, o Status da sua instância de banco de dados do RDS Custom indicará Automation paused (Automação pausada).
7. (Opcional) No painel de navegação, escolha Databases (Bancos de dados) e depois selecione sua instância de banco de dados do RDS Custom.

No painel Summary (Resumo), RDS Custom automation mode (Modo de automação do RDS Custom) indica o status da automação. Se a automação estiver pausada, o valor será Paused (Pausado). Automação retomada em ***número*** minutos.

AWS CLI

Para pausar ou retomar a automação do RDS Custom, utilize o comando `modify-db-instance` da AWS CLI. Identifique a instância de banco de dados utilizando o parâmetro obrigatório `--db-instance-identifier`. Controle o modo de automação com os parâmetros a seguir:

- `--automation-mode` especifica o estado de pausa da instância de banco de dados. Os valores válidos são `all-paused`, que pausa a automação, e `full`, que retoma a automação.
- `--resume-full-automation-mode-minutes` especifica a duração da pausa. O valor padrão são 60 minutos.

> **Note**
>
> Independentemente de você especificar `--no-apply-immediately` ou `--apply-immediately`, o RDS Custom aplica modificações de maneira assíncrona logo que possível.

Na resposta do comando, `ResumeFullAutomationModeTime` indica o tempo de retomada como um carimbo de data/hora UTC. Quando o modo de automação é `all-paused`, é possível utilizar `modify-db-instance` para retomar o modo de automação ou estender o período de pausa. Nenhuma outra opção `modify-db-instance` tem suporte.

O exemplo a seguir pausa a automação de `my-custom-instance` por 90 minutos.

Example

Para Linux, macOS ou Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier my-custom-instance \
    --automation-mode all-paused \
    --resume-full-automation-mode-minutes 90
```

Para Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier my-custom-instance ^
    --automation-mode all-paused ^
    --resume-full-automation-mode-minutes 90
```

O exemplo a seguir estende a duração da pausa por mais 30 minutos. Os 30 minutos são adicionados à hora original mostrada em `ResumeFullAutomationModeTime`.

Example

Para Linux, macOS ou Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier my-custom-instance \
    --automation-mode all-paused \
    --resume-full-automation-mode-minutes 30
```

Para Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier my-custom-instance ^
    --automation-mode all-paused ^
    --resume-full-automation-mode-minutes 30
```

O exemplo a seguir retoma a automação completa de `my-custom-instance`.

Example

Para Linux, macOS ou Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier my-custom-instance \
    --automation-mode full \
```

Para Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier my-custom-instance ^
    --automation-mode full
```

Na seguinte saída parcial da amostra, o valor pendente de `AutomationMode` é `full`.


```
{
    "DBInstance": {
        "PubliclyAccessible": true,
        "MasterUsername": "admin",
        "MonitoringInterval": 0,
        "LicenseModel": "bring-your-own-license",
        "VpcSecurityGroups": [
            {
                "Status": "active",
                "VpcSecurityGroupId": "0123456789abcdefg"
            }
        ],
        "InstanceCreateTime": "2020-11-07T19:50:06.193Z",
        "CopyTagsToSnapshot": false,
        "OptionGroupMemberships": [
            {
                "Status": "in-sync",
                "OptionGroupName": "default:custom-oracle-ee-19"
            }
        ],
        "PendingModifiedValues": {
            "AutomationMode": "full"
        },
        "Engine": "custom-oracle-ee",
        "MultiAZ": false,
        "DBSecurityGroups": [],
        "DBParameterGroups": [
            {
                "DBParameterGroupName": "default.custom-oracle-ee-19",
                "ParameterApplyStatus": "in-sync"
            }
        ],
        ...
        "ReadReplicaDBInstanceIdentifiers": [],
        "AllocatedStorage": 250,
        "DBInstanceArn": "arn:aws:rds:us-west-2:012345678912:db:my-custom-instance",
        "BackupRetentionPeriod": 3,
        "DBName": "ORCL",
        "PreferredMaintenanceWindow": "fri:10:56-fri:11:26",
        "Endpoint": {
            "HostedZoneId": "ABCDEFGHIJKLMNO",
            "Port": 8200,
```

```
            "Address": "my-custom-instance.abcdefghijk.us-west-2.rds.amazonaws.com"
        },
        "DBInstanceStatus": "automation-paused",
        "IAMDatabaseAuthenticationEnabled": false,
        "AutomationMode": "all-paused",
        "EngineVersion": "19.my_cev1",
        "DeletionProtection": false,
        "AvailabilityZone": "us-west-2a",
        "DomainMemberships": [],
        "StorageType": "gp2",
        "DbiResourceId": "db-ABCDEFGHIJKLMNOPQRSTUVW",
        "ResumeFullAutomationModeTime": "2020-11-07T20:56:50.565Z",
        "KmsKeyId": "arn:aws:kms:us-west-2:012345678912:key/
aa111a11-111a-11a1-1a11-1111a11a1a1a",
        "StorageEncrypted": false,
        "AssociatedRoles": [],
        "DBInstanceClass": "db.m5.xlarge",
        "DbInstancePort": 0,
        "DBInstanceIdentifier": "my-custom-instance",
        "TagList": []
    }
```

Modificar uma instância de banco de dados do RDS Custom for SQL Server

O procedimento para modificar uma instância de banco de dados do RDS Custom for SQL Server é semelhante ao do Amazon RDS, mas as alterações possíveis estão limitadas ao seguinte:

- Alterar a classe da instância de banco de dados
- Alterar o período de retenção de backup e a janela de backup
- Alterar a janela de manutenção
- Fazer upgrade da versão do mecanismo de banco de dados quando uma nova versão se torna disponível
- Alterar o armazenamento alocado, as IOPS provisionadas e o tipo de armazenamento
- Permitir e remover implantações multi-AZ

As seguintes limitações aplicam-se à modificação de uma instância de banco de dados do RDS Custom for SQL Server:

- Não há suporte para a opção de banco de dados personalizado e grupos de parâmetros.

- Todos os volumes de armazenamento anexados manualmente à sua instância de banco de dados do RDS Custom estão fora do perímetro de suporte.

  Para ter mais informações, consulte Perímetro de suporte do RDS Custom.

Console

Para modificar uma instância de banco de dados do RDS Custom for SQL Server

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Databases (Bancos de dados).
3. Escolha a instância de banco de dados que você deseja modificar.
4. Selecione Modify.
5. Faça as seguintes alterações, conforme necessário:
   a. Em DB engine version (Versão do mecanismo de banco de dados), selecione a nova versão.
   b. Altere o valor de DB instance class (Classe da instância de banco de dados). Para saber as classes compatíveis, consulte Compatibilidade com classes de instâncias de banco de dados do RDS Custom for SQL Server.
   c. Altere o valor de Backup retention period (Período de retenção de backup).
   d. Para Backup window (Janela de backup), defina valores para Start time (Horário de início) e Duration (Duração).
   e. Para DB instance maintenance window (Janela de manutenção da instância de banco de dados), defina valores para Start day (Dia de início), Start time (Horário de início) e Duration (Duração).
6. Escolha Continuar.
7. Escolha Apply immediately (Aplicar imediatamente) ou Apply during the next scheduled maintenance window (Aplicar durante a próxima janela de manutenção programada).
8. Selecione Modify DB instance (Modificar instância de banco de dados).

AWS CLI

Para modificar uma instância de banco de dados do RDS Custom for SQL Server, utilize o comando modify-db-instance da AWS CLI. Defina os seguintes parâmetros, conforme necessário:

- `--db-instance-class`: para saber as classes compatíveis, consulte Compatibilidade com classes de instâncias de banco de dados do RDS Custom for SQL Server.
- `--engine-version` – o número da versão do mecanismo de banco de dados que está recebendo upgrade.
- `--backup-retention-period` – por quanto tempo manter backups automatizados, de 0 a 35 dias.
- `--preferred-backup-window` – o intervalo de tempo diário durante o qual backups automatizados são criados.
- `--preferred-maintenance-window` – o intervalo de tempo semanal (em UTC) durante o qual pode ocorrer manutenção do sistema.
- `--apply-immediately`: use `--apply-immediately` para aplicar as alterações de armazenamento de forma imediata.

  Ou use `--no-apply-immediately` (o padrão) para aplicar as alterações durante a próxima janela de manutenção.

Modificar o armazenamento para uma instância de banco de dados do RDS Custom para SQL Server

O procedimento para modificar o armazenamento de uma instância de banco de dados do RDS Custom para SQL Server é semelhante ao procedimento de modificação de uma instância de banco de dados do Amazon RDS, mas você só pode fazer o seguinte:

- Aumente o tamanho do armazenamento alocado.
- Altere o tipo de armazenamento. Você pode utilizar tipos de armazenamento, como de uso geral ou IOPS provisionadas. As IOPS provisionadas são compatíveis com os tipos de armazenamento gp3, io1 e io2 Block Express.
- Altere as IOPS provisionadas, se você estiver utilizando os tipos de volume compatíveis com as IOPS provisionadas.

As seguintes limitações se aplicam à modificação do armazenamento de uma instância de banco de dados do RDS Custom para SQL Server:

- O tamanho mínimo de armazenamento alocado para o RDS Custom para SQL Server é 20 GiB. O limite máximo de armazenamento para io1, gp2 e gp3 é de 16 TiB, enquanto o de io2 é de 64 TiB.
- Como acontece com o Amazon RDS, não é possível diminuir o armazenamento alocado. Essa é uma limitação dos volumes do Amazon Elastic Block Store (Amazon EBS). Para ter mais informações, consulte Trabalhar com armazenamento para instâncias de banco de dados do Amazon RDS.
- A autoescalabilidade de armazenamento não é compatível com instâncias de banco de dados do RDS Custom para SQL Server.
- Todos os volumes de armazenamento anexados manualmente à sua instância de banco de dados do RDS Custom não são considerados para escalabilidade do armazenamento. Somente os volumes de dados padrão fornecidos pelo RDS, ou seja, a unidade D, são considerados para escalabilidade do armazenamento.

  Para ter mais informações, consulte Perímetro de suporte do RDS Custom.
- A escalabilidade de armazenamento geralmente não gera nenhuma interrupção ou degradação na performance da instância de banco de dados. Depois de modificar o tamanho de armazenamento de uma instância de banco de dados, o status da instância de banco de dados é storage-optimization.
- A otimização de armazenamento pode demorar muitas horas. Não é possível fazer outras modificações de armazenamento por até seis (6) horas ou até a conclusão da otimização de armazenamento na instância, o que acontecer por último. Para ter mais informações, consulte Trabalhar com armazenamento para instâncias de banco de dados do Amazon RDS.

Para ter mais informações sobre armazenamento, consulte Armazenamento de instâncias de banco de dados do Amazon RDS.

Para obter informações gerais sobre a modificação do armazenamento, consulte Trabalhar com armazenamento para instâncias de banco de dados do Amazon RDS.

> **Important**
>
> Não modifique o armazenamento de sua instância de banco de dados do RDS Custom para SQL Server usando o console ou a API do Amazon EC2 ou do Amazon EBS. Modificações

diretas no armazenamento, fora do console ou das APIs do Amazon RDS, geram o estado `unsupported-configuration` para seu banco de dados.
Quando você faz alterações diretas no armazenamento usando o Amazon EC2 ou o Amazon EBS, o Amazon RDS não consegue rastrear ou gerenciar o estado da instância de banco de dados. Isso pode gerar:

- Mecanismos de failover de alta disponibilidade que não funcionam corretamente.
- Configurações de replicação de banco de dados que serão interrompidas.
- Recursos de redundância que apresentarão falha.

Modifique o armazenamento somente por meio do console ou das APIs do Amazon RDS para manter o banco de dados em um estado compatível. Consulte Corrigir configurações não compatíveis no RDS Custom para SQL Server para conhecer as etapas de recuperação.

Console

Como modificar o armazenamento para uma instância de banco de dados do RDS Custom para SQL Server

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Databases (Bancos de dados).
3. Escolha a instância de banco de dados que você deseja modificar.
4. Selecione Modify.
5. Faça as seguintes alterações, conforme necessário:
    a. Digite um novo valor em Allocated Storage (Armazenamento alocado). Ele deve ser maior que o valor atual e variar entre 20 GiB e 16 TiB.
    b. Altere o valor de Storage type (Tipo de armazenamento). É possível escolher entre os tipos de armazenamento de uso geral ou IOPS provisionadas disponíveis. As IOPS provisionadas são compatíveis com os tipos de armazenamento gp3, io1 e io2 Block Express.
    c. Se estiver especificando um tipo de armazenamento que seja compatível com IOPS provisionadas, você poderá definir o valor de IOPS provisionadas.
6. Escolha Continuar.

7. Escolha Apply immediately (Aplicar imediatamente) ou Apply during the next scheduled maintenance window (Aplicar durante a próxima janela de manutenção programada).
8. Selecione Modify DB instance (Modificar instância de banco de dados).

AWS CLI

Para modificar o armazenamento de uma instância de banco de dados do RDS Custom para SQL Server, utilize o comando modify-db-instance da AWS CLI. Defina os seguintes parâmetros, conforme necessário:

- `--allocated-storage`: quantidade de armazenamento a ser alocada para a instância de banco de dados, em gibibytes. Ele deve ser maior que o valor atual e variar entre 20 e 16.384 GiB.
- `--storage-type`: o tipo de armazenamento, por exemplo, gp2, gp3, io1 ou io2.
- `--iops`: IOPS provisionadas para a instância de banco de dados. É possível especificar isso somente para tipos de armazenamento que são compatíveis com IOPS provisionadas (gp3, io1 e io2).
- `--apply-immediately`: use `--apply-immediately` para aplicar as alterações de armazenamento de forma imediata.

  Ou use `--no-apply-immediately` (o padrão) para aplicar as alterações durante a próxima janela de manutenção.

O exemplo a seguir altera o tamanho do armazenamento de my-custom-instance para 200 GiB, o tipo de armazenamento para io1 e as IOPS provisionadas para 3.000.

Example

Para Linux, macOS ou Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier my-custom-instance \
    --storage-type io1 \
    --iops 3000 \
    --allocated-storage 200 \
    --apply-immediately
```

Para Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier my-custom-instance ^
    --storage-type io1 ^
    --iops 3000 ^
    --allocated-storage 200 ^
    --apply-immediately
```

Marcar recursos do RDS Custom for SQL Server

Você pode marcar recursos do RDS Custom como faz com recursos do Amazon RDS, mas com algumas diferenças importantes:

- Não crie ou modifique a etiqueta `AWSRDSCustom` necessária para a automação do RDS Custom. Se você fizer isso, talvez interrompa a automação.
- A tag `Name` é adicionada aos recursos personalizados do RDS com o prefixo `do-not-delete-rds-custom`. Qualquer valor passado pelo cliente para a chave é sobrescrito.
- As etiquetas adicionadas às instâncias de banco de dados RDS Custom durante a criação são propagadas para todos os outros recursos relacionados do RDS Custom.
- As etiquetas não são propagadas quando você as adiciona a recursos do RDS Custom após a criação da instância de banco de dados.

Para obter informações gerais sobre a marcação de recursos, consulte Marcar recursos do Amazon RDS.

Excluir uma instância de banco de dados do RDS Custom for SQL Server

Para excluir uma instância de banco de dados do RDS Custom para SQL Server, faça o seguinte:

- Forneça o nome da instância de banco de dados.
- Marque ou desmarque a opção de obter um snapshot de banco de dados final da instância de banco de dados.
- Marque ou desmarque a opção para reter backups automatizados.

É possível excluir uma instância de banco de dados do RDS Custom para SQL Server utilizando o console ou a CLI. O tempo necessário para excluir a instância de banco de dados pode variar dependendo do período de retenção do backup (ou seja, quantos backups serão excluídos), da quantidade de dados excluídos e se um snapshot final será criado.

> **Warning**
>
> A exclusão de uma instância de banco de dados do RDS Custom para SQL Server excluirá permanentemente a instância do EC2 e os volumes do Amazon EBS associados. Você não deve encerrar ou excluir esses recursos a qualquer momento, caso contrário, a exclusão e a criação final do instantâneo poderão falhar.

> **Note**
>
> Não é possível criar um snapshot de banco de dados final de sua instância de banco de dados se ela tiver o status `creating`, `failed`, `incompatible-create`, `incompatible-restore` ou `incompatible-network`. Para ter mais informações, consulte Visualizar o status de uma instância de banco de dados Amazon RDS em um .

> **Important**
>
> Quando você optar por criar um snapshot final, é recomendável evitar gravar dados em sua instância de banco de dados enquanto a exclusão da instância de banco de dados estiver em andamento. Depois que a exclusão da instância de banco de dados é iniciada, não há garantia de que as alterações de dados sejam capturadas pelo snapshot final.

Console

Para excluir uma instância de banco de dados do RDS Custom

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Databases (Bancos de dados) e a instância de banco de dados do RDS Custom para SQL Server que você deseja excluir. As instâncias de banco de dados do RDS Custom para SQL Server mostram o perfil Instance (RDS Custom for SQL Server) [Instância (RDS Custom para SQL Server)].
3. Em Ações, escolha Excluir.
4. Para criar um snapshot final, escolha Create final snapshot (Criar snapshot final) e forneça um nome no campo Final snapshot name (Nome do snapshot final).

5. Para reter backups automatizados, escolha Retain automated backups (Reter backups automatizados).
6. Digite **`delete me`** na caixa.
7. Escolha Excluir.

AWS CLI

Exclua uma instância de banco de dados do RDS Custom para SQL Server utilizando o comando delete-db-instance da AWS CLI. Identifique a instância de banco de dados utilizando o parâmetro obrigatório `--db-instance-identifier`. Os parâmetros restantes são os mesmos de uma instância de banco de dados Amazon RDS.

O exemplo a seguir exclui a instância de banco de dados do RDS Custom para SQL Server chamada `my-custom-instance`, cria um snapshot final e retém backups automatizados.

Example

Para Linux, macOS ou Unix:

```
aws rds delete-db-instance \
    --db-instance-identifier my-custom-instance \
    --no-skip-final-snapshot \
    --final-db-snapshot-identifier my-custom-instance-final-snapshot \
    --no-delete-automated-backups
```

Para Windows:

```
aws rds delete-db-instance ^
    --db-instance-identifier my-custom-instance ^
    --no-skip-final-snapshot ^
    --final-db-snapshot-identifier my-custom-instance-final-snapshot ^
    --no-delete-automated-backups
```

Para criar um snapshot final, a opção `--final-db-snapshot-identifier` é necessária e deve ser especificada.

Para ignorar o snapshot final, especifique a opção `--skip-final-snapshot` em vez das opções `--no-skip-final-snapshot` e `--final-db-snapshot-identifier` no comando.

Para excluir backups automatizados, especifique a opção `--delete-automated-backups` em vez da opção `--no-delete-automated-backups` no comando.

Iniciar e interromper uma instância de banco de dados do RDS Custom para SQL Server

É possível iniciar e interromper uma instância de banco de dados do RDS Custom para SQL Server. Os mesmos requisitos e limitações gerais para instâncias de banco de dados do RDS para SQL Server aplicam-se à interrupção e inicialização das instâncias de banco de dados do RDS Custom para SQL Server. Para ter mais informações, consulte Parar uma instância de banco de dados do Amazon RDS temporariamente.

As considerações a seguir aplicam-se à inicialização e interrupção da instância de banco de dados do RDS Custom para SQL Server:

- Não é possível modificar um atributo de instância de banco de dados do RDS Custom para SQL Server enquanto o status da respectiva instância for `STOPPED`.
- É possível interromper e iniciar uma instância de banco de dados do RDS Custom para SQL Server somente se ela estiver configurada para uma única zona de disponibilidade. Não é possível interromper uma instância de banco de dados do RDS Custom para SQL Server em uma configuração multi-AZ.
- Um snapshot `SYSTEM` será criado quando você interromper uma instância de banco de dados do RDS Custom para SQL Server. O snapshot será excluído automaticamente quando você iniciar novamente a instância de banco de dados do RDS Custom para SQL Server.
- Se a instância de banco de dados do RDS Custom para SQL Server for excluída enquanto estiver interrompida, a unidade `C:` será substituída quando você iniciar novamente a respectiva instância.
- Desde que você não modifique o tipo de instância, a unidade `C:\`, o nome do host e suas configurações personalizadas persistem quando uma instância de banco de dados do RDS Custom para SQL Server é interrompida.
- Em decorrência das ações a seguir, o RDS Custom colocará a instância de banco de dados fora do perímetro de suporte e você ainda assim receberá cobrança pelas horas da instância de banco de dados:
  - Iniciar a instância do EC2 subjacente enquanto o Amazon RDS estiver parado. Para resolver, você pode chamar a API `start-db-instance` do Amazon RDS ou interromper o EC2 para que a instância do RDS Custom retorne ao estado `STOPPED`.

- Interromper a instância do EC2 subjacente quando o status da instância de banco de dados do RDS Custom para SQL Server for `ACTIVE`.

Para obter mais detalhes sobre como interromper e iniciar instâncias de banco de dados, consulte Parar uma instância de banco de dados do Amazon RDS temporariamente e Iniciar uma instância de banco de dados do Amazon RDS que foi anteriormente interrompida.

Trabalhar com o Microsoft Active Directory e o RDS Custom para SQL Server

O RDS Custom para SQL Server permite associar as instâncias a um Active Directory (AD) autogerenciado ou AWS Managed Microsoft AD. Isso ocorre independentemente de onde o AD esteja hospedado, como em um data center on-premises, no Amazon EC2 ou em qualquer outro provedor de serviços de nuvem.

Em relação à autenticação de usuários e de serviços, é possível usar a autenticação NTLM ou Kerberos na instância de banco de dados do RDS Custom para SQL Server sem usar domínios intermediários e relações de confiança de floresta. Quando um usuário tenta se autenticar na instância de banco de dados do RDS Custom para SQL Server com um Active Directory autoassociado, as solicitações de autenticação são encaminhadas a um AD autogerenciado ou AWS Managed Microsoft AD especificado por você.

Nas seções a seguir, é possível encontrar informações sobre como trabalhar com um Active Directory autogerenciado e com o Active Directory gerenciado pela AWS para o RDS Custom para SQL Server.

Tópicos

- Disponibilidade de regiões e versões
- Configurar o AD autogerenciado ou on-premises
- Configurar o Microsoft Active Directory usando o AWS Directory Service
- Regras de portas para configuração de rede
- Validação da rede
- Configurar a autenticação do Windows para instâncias do RDS Custom para SQL Server
- Gerenciamento de uma instância de banco de dados em um domínio
- Compreensão da associação de domínio
- Solucionar problemas no Active Directory

Disponibilidade de regiões e versões

O RDS Custom para SQL Server aceita o AD autogerenciado e o AWS Managed Microsoft AD usando o NTLM ou o Kerberos em todas as regiões compatíveis com o RDS Custom para SQL Server. Para ter mais informações, consulte Regiões e mecanismos de banco de dados compatíveis com o RDS Custom.

Configurar o AD autogerenciado ou on-premises

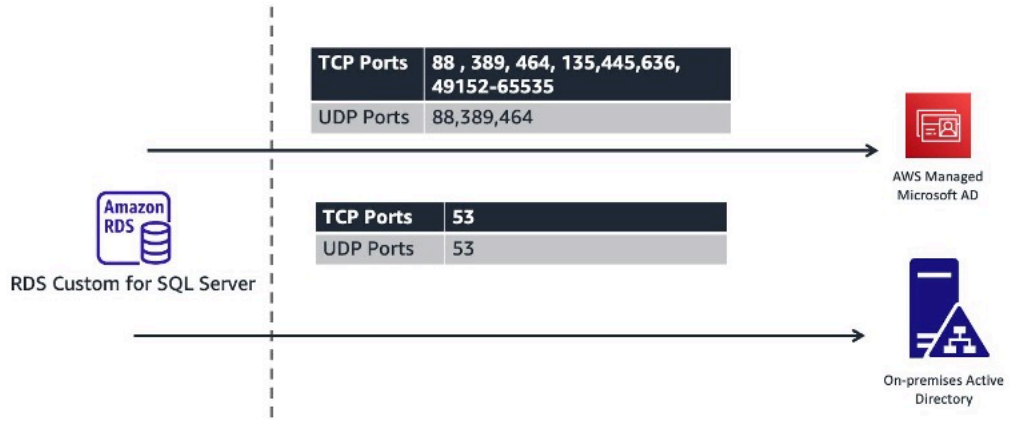
Para associar o Microsoft AD on-premises ou autogerenciado à instância de banco de dados do RDS Custom para SQL Server, o Active Domain deve ser configurado da seguinte maneira:

- Defina as sub-redes na VPC associada à instância de banco de dados do RDS Custom para SQL Server no AD autogerenciado ou on-premises. Confirme se não há nenhum conflito entre as sub-redes na VPC e as sub-redes nos outros sites do AD.
- Seu controlador de domínios de AD tem um nível funcional de domínio do Windows Server 2008 R2 ou posterior.
- O nome de domínio do AD não pode estar no formato de domínio de rótulo único (SLD). O RDS Custom para SQL Server não é compatível com domínios SLD.
- O nome de domínio totalmente qualificado (FQDN) do AD não pode exceder 47 caracteres.

Configurar sua conectividade de rede

Configure a conectividade de rede do AD autogerenciado ou on-premises da seguinte maneira:

- Configure a conectividade entre a Amazon VPC, em que a instância do RDS Custom para SQL Server está em execução, e o AD. Use AWS Direct Connect, AWS VPN, AWS Transit Gateway e emparelhamento da VPC.
- Permita o tráfego nas portas dos grupos de segurança e ACLs de rede do RDS Custom para SQL Server para o AD autogerenciado ou on-premises. Para ter mais informações, consulte Regras de portas para configuração de rede.



Configurar a resolução de DNS

Defina os seguintes requisitos para configurar a resolução de DNS com ADs autogerenciados ou on-premises:

- Configure a resolução de DNS na VPC para resolver um nome de domínio totalmente qualificado (FQDN) do Active Directory auto-hospedado. Um exemplo de FQDN é `corp.example.local`. Para configurar a resolução de DNS, configure o resolvedor de DNS da VPC para encaminhar consultas a determinados domínios com um endpoint de saída do Amazon Route 53 e uma regra de resolvedor. Para ter mais informações, veja como configurar um endpoint de saída do Route 53 Resolver para resolver registros DNS.
- Em relação a workloads que utilizem tanto VPCs quanto recursos on-premises, é necessário resolver os registros DNS hospedados on-premises. Os recursos on-premises talvez precisem resolver nomes hospedados na AWS.

  Para criar uma configuração de nuvem híbrida, use endpoints de resolvedor e regras de encaminhamento condicional para resolver consultas ao DNS entre os recursos on-premises e a VPC personalizada. Para ter mais informações, consulte Resolving DNS queries between VPCs and your network no Guia do desenvolvedor do Amazon Route 53.

> **Important**
>
> A modificação das configurações do resolvedor de DNS da interface de rede no RDS Custom para SQL Server faz com que os endpoints da VPC habilitados para DNS deixem de funcionar corretamente. Os endpoints da VPC habilitados para DNS são necessários para instâncias em sub-redes privadas sem acesso à internet.

Configurar o Microsoft Active Directory usando o AWS Directory Service

O AWS Managed Microsoft AD cria um Microsoft Active Directory totalmente gerenciado na AWS que é habilitado pelo Windows Server 2019 e opera nos níveis funcionais de floresta e de domínio do 2012 R2. O AWS Directory Service cria os controladores de domínio em diferentes sub-redes em uma Amazon VPC, tornando o diretório altamente disponível mesmo em caso de falha.

Para criar um diretório com o AWS Managed Microsoft AD, consulte Getting started with AWS Managed Microsoft AD no Guia de administração do AWS Directory Service.

Configurar sua conectividade de rede

Habilitar o tráfego entre VPCs entre o diretório e a instância de banco de dados

Para localizar o diretório e a instância de banco de dados na mesma VPC, ignore esta etapa e vá para a próxima etapa em Regras de portas para configuração de rede.

Para localizar o diretório e a instância de bancos de dados em VPCs diferentes, configure o tráfego entre VPCs usando o emparelhamento da VPC ou o AWS Transit Gateway. Para ter mais informações sobre emparelhamento da VPC, consulte O que é emparelhamento de VPC? no Guia de emparelhamento da Amazon VPC e What is AWS Transit Gateway? nos Gateways de Trânsito da Amazon VPC.

Habilitar o tráfego entre VPCs usando o emparelhamento da VPC

1. Configure regras apropriadas de roteamento de VPC para garantir que o tráfego de rede possa fluir em ambos os sentidos.
2. Permita que o grupo de segurança da instância de banco de dados possa receber o tráfego de entrada do grupo de segurança do diretório. Para ter mais informações, consulte Regras de portas para configuração de rede.
3. A lista de controle de acesso (ACL) de rede não deve bloquear o tráfego.

Se uma Conta da AWS diferente for proprietária do diretório, será necessário compartilhar o diretório. Para compartilhar o diretório com a Conta da AWS em que a instância do RDS Custom para SQL Server está, siga o Tutorial: Sharing your AWS Managed Microsoft AD for seamless EC2 domain-join no Guia de administração do AWS Directory Service.

Compartilhar um diretório entre Contas da AWS

1. Faça login no console do AWS Directory Service usando a conta para a instância de banco de dados e confira se o domínio tem o status `SHARED` antes de prosseguir.
2. Depois de fazer login no console do AWS Directory Service usando a conta da instância de banco de dados, observe o valor do ID do diretório. Use esse ID para associar a instância de banco de dados ao domínio.

Configurar a resolução de DNS

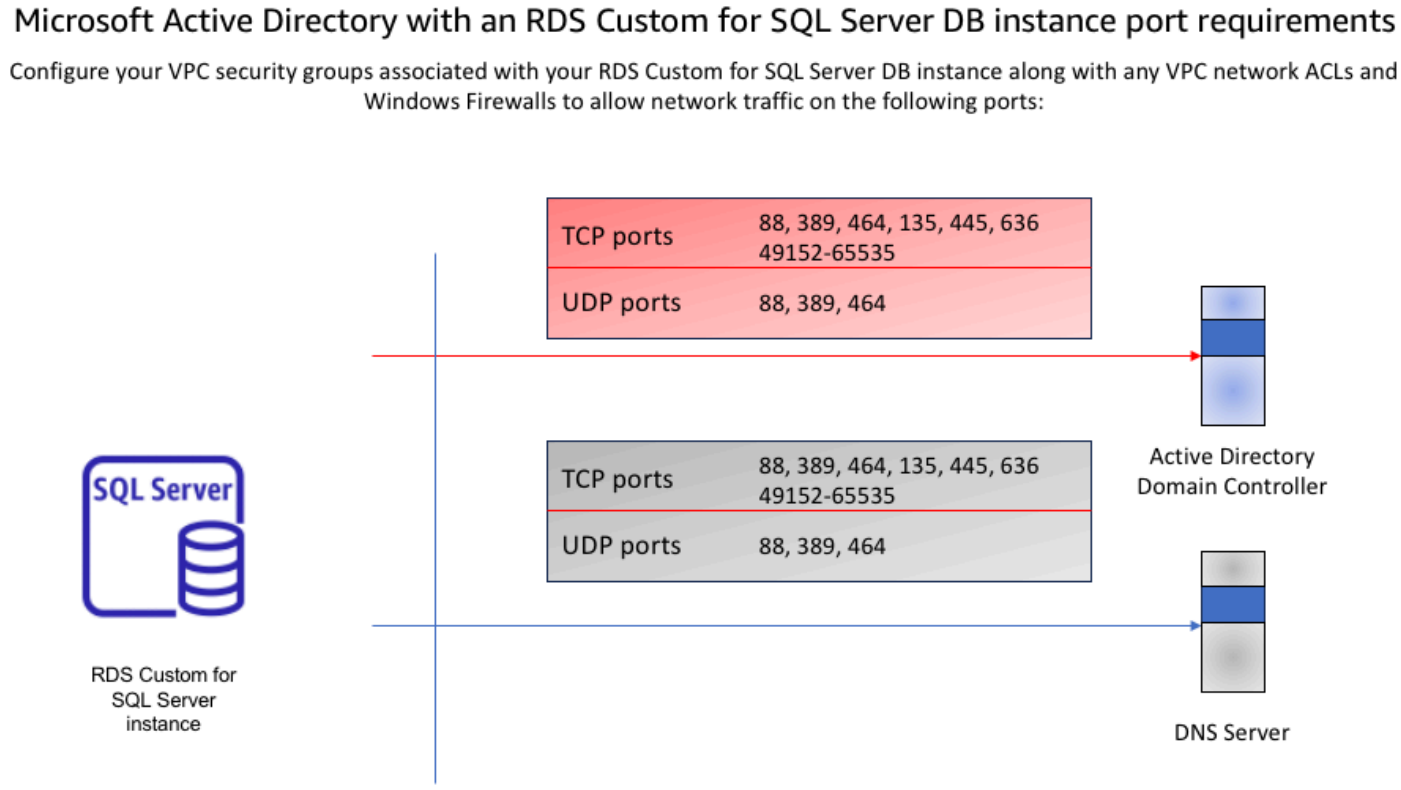
Quando você cria um diretório do AWS Managed Microsoft AD, o AWS Directory Service cria dois controladores de domínio e adiciona o serviço DNS em seu nome.

Se você já tem um AWS Managed Microsoft AD ou planeja inicializá-lo em uma VPC diferente da instância de banco de dados do RDS Custom para SQL Server, configure o resolvedor de DNS da VPC para encaminhar consultas a determinados domínios com uma regra de saída e de resolvedor do Route 53. Veja como configurar um endpoint de saída do Route 53 Resolver para resolver registros DNS.

Regras de portas para configuração de rede

Você precisa atender às seguintes configurações de rede:

- Conectividade configurada entre a Amazon VPC na qual você deseja criar a instância de banco de dados do RDS Custom para SQL Server e o Active Directory autogerenciado ou AWS Managed Microsoft AD. Em relação ao Active Directory autogerenciado, configure a conectividade usando o AWS Direct Connect, a AWS VPN, o emparelhamento da VPC ou o AWS Transit Gateway. Em relação ao AWS Managed Microsoft AD, configure a conectividade utilizando emparelhamento da VPC.
- Verifique se o grupo de segurança e as ACLs de rede da VPC para as sub-redes em que você vai criar a instância de banco de dados do RDS Custom para SQL Server permitem tráfego nas portas e nas direções mostradas no diagrama a seguir.



A tabela a seguir identifica o perfil de cada porta.

Protocolo	Portas	Função
TCP/UDP	53	Domain Name System (DNS)
TCP/UDP	88	Autenticação de Kerberos
TCP/UDP	464	Alterar/definir senha
TCP/UDP	389	Lightweight Directory Access Protocol (LDAP)
TCP	135	Distributed Computing Environment/End Point Mapper (DCE/EPMAP)
TCP	445	Compartilhamento de arquivos de SMB para serviços de diretório

Protocolo	Portas	Função
TCP	636	Lightweight Directory Access Protocol over TLS/SSL (LDAPS)
TCP	49152 – 65535	Portas efêmeras para RPC

- Em geral, os servidores DNS do domínio estão localizados nos controladores do domínio de AD. Você não precisa configurar o conjunto de opções DHCP da VCP para usar esse atributo. Para obter mais informações, consulte Conjuntos de opções DHCP no Guia do usuário do Amazon VPC.

> **Important**
>
> Se você estiver usando ACLs de rede de VPC, também deverá permitir tráfego de saída em portas dinâmicas (49152-65535) da instância de banco de dados do RDS Custom para SQL Server. Confira se essas regras de tráfego também são refletidas nos firewalls que se aplicam a cada um dos controladores do domínio de AD, servidores DNS e instâncias de banco de dados do RDS Custom para SQL Server.
> Embora os grupos de segurança de VPC exijam que as portas sejam abertas somente na direção em que o tráfego de rede é iniciado, a maioria dos firewalls do Windows e das ACLs da rede de VPC exigem que as portas sejam abertas nas duas direções.

Validação da rede

Antes de associar a instância do RDS Custom ao AWS Managed Microsoft AD ou autogerenciado, confira as informações a seguir em uma instância do EC2 na mesma VPC em que você planeja inicializar a instância do RDS Custom para SQL Server.

- Confira se consegue resolver o nome de domínio totalmente qualificado (FQDN) para os IPs do controlador de domínio.

```
nslookup corp.example.com
```

  O comando deve exibir uma saída semelhante:

```
Server:  ip-10-0-0-2.us-west-2.compute.internal
Address:  25.0.0.2

Non-authoritative answer:
Name:    corp.example.com
Addresses:  40.0.9.25 (DC1 IP)
            40.0.50.123 (DC2 IP)
```

- Resolva os serviços da AWS de uma instância do EC2 na VPC em que planeja iniciar a instância do RDS Custom:

```
$region='input-your-aws-region'
$domainFQDN='input-your-domainFQDN'

function Test-DomainPorts {
    param (
        [string]$Domain,
        [array]$Ports
    )

    foreach ($portInfo in $Ports) {
        try {
            $conn = New-Object System.Net.Sockets.TcpClient
            $connectionResult = $conn.BeginConnect($Domain, $portInfo.Port, $null,
 $null)
            $success = $connectionResult.AsyncWaitHandle.WaitOne(1000) # 1 second
 timeout
            if ($success) {
                $conn.EndConnect($connectionResult)
                $result = $true
            } else {
                $result = $false
            }
        }
        catch {
            $result = $false
        }
        finally {
            if ($null -ne $conn) {
                $conn.Close()
            }
        }
```

```
        Write-Host "$($portInfo.Description) port open: $result"
    }
}

# Check if ports can be reached
$ports = @(
    @{Port = 53;   Description = "DNS"},
    @{Port = 88;   Description = "Kerberos"},
    @{Port = 389;  Description = "LDAP"},
    @{Port = 445;  Description = "SMB"},
    @{Port = 5985; Description = "WinRM"},
    @{Port = 636;  Description = "LDAPS"},
    @{Port = 3268; Description = "Global Catalog"},
    @{Port = 3269; Description = "Global Catalog over SSL"},
    @{Port = 9389; Description = "AD DS"}
)

function Test-DomainReachability {
    param (
        [string]$DomainName
    )

    try {
        $dnsResults = Resolve-DnsName -Name $DomainName -ErrorAction Stop
        Write-Host "Domain $DomainName is successfully resolving to following IP
 addresses: $($dnsResults.IpAddress)"
        Write-Host ""
        return $true
    }
    catch {
        Write-Host ""
        Write-Host "Error Message: $($_.Exception.Message)"
        Write-Host "Domain $DomainName reachability check failed, please Configure
 DNS resolution"
        return $false
    }
}

$domain = (Get-WmiObject Win32_ComputerSystem).Domain
if ($domain -eq 'WORKGROUP') {
    Write-Host ""
    Write-Host "Host $env:computername is still part of WORKGROUP and not part of any
 domain"
    }
```

```
else {
    Write-Host ""
    Write-Host "Host $env:computername is joined to $domain domain"
    Write-Host ""
    }


$isReachable = Test-DomainReachability -DomainName $domainFQDN
if ($isReachable) {
    write-Host "Checking if domain $domainFQDN is reachable on required ports  "
    Test-DomainPorts -Domain $domainFQDN -Ports $ports
}
else {
    Write-Host "Port check skipped. Domain not reachable"
}



# Get network adapter configuration
$networkConfig = Get-WmiObject Win32_NetworkAdapterConfiguration |
                 Where-Object { $_.IPEnabled -eq $true } |
                 Select-Object -First 1

# Check DNS server settings
$dnsServers = $networkConfig.DNSServerSearchOrder

if ($dnsServers) {
    Write-Host "`nDNS Server settings:"
    foreach ($server in $dnsServers) {
        Write-Host "  - $server"
    }
} else {
    Write-Host "`nNo DNS servers configured or unable to retrieve DNS server
 information."
}

write-host ""

# Checks reachability to dependent services
$services = "s3", "ec2", "secretsmanager", "logs", "events", "monitoring", "ssm",
 "ec2messages", "ssmmessages"

function Get-TcpConnectionAsync {
    param (
```

```
        $ServicePrefix,
        $region
    )
    $endpoint = "${ServicePrefix}.${region}.amazonaws.com"
    $tcp = New-Object Net.Sockets.TcpClient
    $result = $false

    try {
        $connectTask = $tcp.ConnectAsync($endpoint, 443)
        $timedOut = $connectTask.Wait(3000)
        $result = $tcp.Connected
    }
    catch {
        $result = $false
    }
    return $result
}

foreach ($service in $services) {
    $validationResult = Get-TcpConnectionAsync -ServicePrefix $service -Region
 $region
    Write-Host "Reachability to $service is $validationResult"
}
```

O valor `TcpTestSucceeded` deve retornar `True` para `s3`, `ec2`, `secretsmanager`, `logs`, `events`, `monitoring`, `ssm`, `ec2messages` e `ssmmessages`.

Configurar a autenticação do Windows para instâncias do RDS Custom para SQL Server

Recomendamos criar uma OU dedicada e credenciais de serviço com escopo para essa OU para qualquer Conta da AWS que seja proprietária de uma instância de banco de dados do RDS Custom para SQL Server associada ao domínio do AD. Ao dedicar uma OU e credenciais de serviço, é possível evitar permissões conflitantes e seguir o princípio de privilégio mínimo.

As políticas de grupo ao nível do Active Directory podem entrar em conflito com automações e permissões da AWS. Recomendamos selecionar GPOs que se apliquem somente à OU criados para o RDS Custom para SQL Server.

- Para criar uma OU e um usuário do domínio do AD no AD on-premises ou autogerenciado, é possível conectar o controlador de domínio como administrador de domínio.

- Para criar usuários e grupos em um diretório do AWS Directory Service, é necessário se conectar a uma instância de gerenciamento. Você também precisa fazer login como um usuário com privilégios para criar usuários e grupos. Para ter mais informações, consulte User and group management in AWS Managed Microsoft AD no Guia de administração do AWS Directory Service.
- Para gerenciar o Active Directory na instância do Windows Server do Amazon EC2, é necessário instalar os serviços de domínio do Active Directory e as ferramentas dos serviços do Active Directory Lightweight Directory na instância do EC2. Para ter mais informações, consulte Installing the Active Directory Administration ToolsAWS Managed Microsoft AD no Guia de administração do AWS Directory Service.
- Para facilitar a administração, recomendamos instalar essas ferramentas em uma instância do EC2 separada para administração, e não na instância de banco de dados do RDS Custom para SQL Server.

Veja a seguir os requisitos da conta de serviço do domínio do AD:

- É necessário ter uma conta de serviço no domínio do AD com permissões delegadas para associar computadores ao domínio. Uma conta de serviço de domínio é uma conta de usuário no AD que recebeu permissão para realizar determinadas tarefas.
- Delegue à conta de serviço de domínio as seguintes permissões na unidade organizacional à qual você está associando a instância do RDS Custom para SQL Server:
  - Capacidade validada para gravar no nome do host DNS
  - Capacidade validada para gravar no nome da entidade principal de serviço
  - Criar e excluir objetos de computador
- Em relação ao AD on-premises e autogerenciado, a conta de serviço de domínio deve ser membro do grupo “Administradores do sistema de nomes de domínio delegados da AWS”.
- Em relação ao AWS Managed Microsoft AD, a conta de serviço de domínio deve ser membro do grupo “DnsAdmins”.

Essas permissões representam o conjunto mínimo de permissões necessárias para associar objetos de computador ao AD autogerenciado e ao AWS Managed Microsoft AD. Para ter mais informações, consulte Error: Access is denied when non-administrator users who have been delegated control try to join computers to a domain controller na documentação do Microsoft Windows Server.

> **Important**
>
> Não mova objetos de computador criados pelo RDS Custom para SQL Server na unidade organizacional (UO) depois da criação da instância de banco de dados. Mover os objetos associados fará com que a instância de banco de dados do RDS Custom para SQL Server tenha configurações incorretas. Se você precisar mover os objetos de computador criados pelo Amazon RDS, use a ação ModifyDBInstance para modificar os parâmetros do domínio com a localização desejada dos objetos de computador.

Tópicos

- Etapa 1: criar uma unidade organizacional (UO) no AD
- Etapa 2: criar um usuário do domínio do AD
- Etapa 3: delegar controle ao usuário do AD no AWS Managed Microsoft AD ou autogerenciado
- Etapa 4: criar um segredo
- Etapa 5: criar ou modificar a instância de banco de dados do RDS Custom para SQL Server
- Etapa 6: criar login do SQL Server para autenticação do Windows
- Etapa 7: usar a autenticação do Kerberos ou NTLM

Etapa 1: criar uma unidade organizacional (UO) no AD

Use as seguintes etapas para criar uma unidade organizacional no AD:

Criar uma UO no AD

1. Conecte-se ao domínio do AD como administrador do domínio.
2. Abra Usuários e computadores do Active Directory e selecione o domínio em que deseja criar a OU.
3. Clique com o botão direito do mouse no domínio, escolha Novo e Unidade organizacional.
4. Insira um nome para a OU.

   Habilite a opção Proteger o contêiner contra exclusão acidental.
5. Escolha OK. A nova UO será exibida no domínio.

Em relação ao AWS Managed Microsoft AD, o nome dessa UO é baseado no nome NetBIOS digitado quando você criou o diretório. Essa UO é de propriedade da AWS e contém todos os objetos de diretório relacionados à AWS, dos quais você recebeu controle total. Por padrão, existem duas UOs secundárias sob essa UO, a saber, Computadores e Usuários. As novas UOs criadas pelo RDS Custom são secundárias da OU baseada no NetBIOS.

Etapa 2: criar um usuário do domínio do AD

As credenciais do usuário do domínio são usadas para o segredo no Secrets Manager.

Criar um usuário do domínio no AD

1. Abra Usuários e computadores do Active Directory e selecione o domínio e a UO em que deseja criar o usuário.
2. Clique com o botão direito do mouse no objeto Usuários, escolha Novo e selecione Usuário.
3. Insira um nome, um sobrenome e um nome de login para o usuário. Clique em Next.
4. Insira uma senha para o usuário. Não selecione as opções O usuário deve alterar a senha no próximo login e A conta está desabilitada. Clique em Next.
5. Clique em OK. O novo usuário será exibido no domínio.

Etapa 3: delegar controle ao usuário do AD no AWS Managed Microsoft AD ou autogerenciado

Como delegar controle ao usuário do domínio de AD

1. Abra o snap-in do MMC Usuários e computadores do Active Directory e selecione o domínio.
2. Clique com o botão direito do mouse na UO criada anteriormente e escolha Delegar controle.
3. No Assistente de controle de delegação, clique em Próximo.
4. Na seção Usuários ou grupos, clique em Adicionar.
5. Na seção Selecionar usuários, computadores ou grupos, insira o usuário do AD que você criou e clique em Conferir nomes. Se a verificação de usuário do AD for bem-sucedida, clique em OK.
6. Na seção Usuários ou grupos, confirme se o usuário do AD foi adicionado e clique em Próximo.
7. Na seção Tarefas para delegar, selecione Criar uma tarefa personalizada para delegar e escolha Próximo.
8. Na seção Tipo de objeto do Active Directory:

   Selecione Somente os objetos a seguir na pasta.

Selecione Objetos do computador.

Selecione Criar objetos selecionados nesta pasta.

Selecione Excluir objetos selecionados nesta pasta e clique em Próximo.

9. Na seção Permissões:

   Mantenha a opção Geral selecionada.

   Selecione Gravação validada no nome do host DNS.

   Selecione Gravação validada no nome da entidade principal de serviço e clique em Próximo.

10. Em Concluir o assistente de delegação de controle, confirme as configurações e clique em Concluir.

Etapa 4: criar um segredo

Crie o segredo na mesma Conta da AWS e região que contém a instância de banco de dados do RDS Custom para SQL Server a ser incluída no Active Directory. Armazene as credenciais do usuário do domínio do AD criado em Etapa 2: criar um usuário do domínio do AD.

Console

- No AWS Secrets Manager, selecione Armazenar um novo segredo.
- Em Tipo de segredo, escolha Outro tipo de segredo.
- Em Pares de chave/valor, adicione duas chaves:
  - A primeira chave, `SELF_MANAGED_ACTIVE_DIRECTORY_USERNAME`, e o nome do usuário do AD (sem o prefixo do domínio) para o valor.
  - A segunda chave, `SELF_MANAGED_ACTIVE_DIRECTORY_PASSWORD`, e a senha do usuário do AD no domínio.
- Em Chave de criptografia, insira a mesma chave do AWS KMS que você usou para criar a instância do RDS Custom para SQL Server.
- Em Nome do segredo, escolha o nome do segredo que começa com `do-not-delete-rds-custom-` para permitir que o perfil de instância acesse esse segredo. Se você quiser escolher um nome diferente para o segredo, atualize `RDSCustomInstanceProfile` para acessar o Nome do segredo.

- (Opcional) Em Descrição, insira uma descrição para o nome do segredo.
- Adicione as etiquetas `Key="AWSRDSCustom",Value="custom-sqlserver"`.
- Clique em Salvar, depois em Próximo.
- Em Definir configurações de rotação, mantenha os valores padrão e escolha Próximo.
- Revise as configurações do segredo e clique em Armazenar.
- Escolha o novo segredo e copie o valor do ARN do segredo. Usaremos isso na próxima etapa para configurar o Active Directory.

CLI

Execute o seguinte comando na CLI para criar um serviço:

```
# Linux based
aws secretsmanager create-secret \
--name do-not-delete-rds-custom-DomainUserCredentails \
--description "Active directory user credentials for managing RDS Custom" \
--secret-string "{\"SELF_MANAGED_ACTIVE_DIRECTORY_USERNAME\":\"tester\",
\"SELF_MANAGED_ACTIVE_DIRECTORY_PASSWORD\":\"xxxxxxxx\"}" \
--kms-key-id <RDSCustomKMSKey> \
--tags Key="AWSRDSCustom",Value="custom-sqlserver"

# Windows based
aws secretsmanager create-secret ^
--name do-not-delete-rds-custom-DomainUserCredentails ^
--description "Active directory user credentials for managing RDS Custom" ^
--secret-string "{\"SELF_MANAGED_ACTIVE_DIRECTORY_USERNAME\":\"tester\",
\"SELF_MANAGED_ACTIVE_DIRECTORY_PASSWORD\":\"xxxxxxxx\"}" ^
--kms-key-id <RDSCustomKMSKey> ^
--tags Key="AWSRDSCustom",Value="custom-sqlserver"
```

Etapa 5: criar ou modificar a instância de banco de dados do RDS Custom para SQL Server

Crie ou modifique uma instância de banco de dados do RDS Custom para SQL Server a ser usado com o diretório. É possível usar o console, a CLI ou a API do RDS para associar uma instância de banco de dados a um diretório. Você pode fazer isso por meio de uma das seguintes maneiras:

- Crie uma instância de banco de dados do SQL Server usando o console, o comando create-db-instance da CLI ou a operação da API CreateDBInstance do RDS.

Para obter instruções, consulte Criar uma instância de banco de dados do Amazon RDS.

- Modifique uma instância de banco de dados existente do SQL Server usando o console, o comando modify-db-instance da CLI ou a operação da API ModifyDBInstance do RDS.

  Para obter instruções, consulte Modificar uma instância de banco de dados do Amazon RDS.

- Restaure uma instância de banco de dados do SQL Server de um snapshot de banco de dados usando o console, o comando restore-db-instance-from-db-snapshot da CLI ou a operação da API RestoreDBInstanceFromDBSnapshot do RDS.

  Para obter instruções, consulte Restaurar uma instância de banco de dados.

- Restaure uma instância de banco de dados SQL Server em um determinado momento usando o console, o comando restore-db-instance-to-point-in-time da CLI ou a operação da API RestoreDBInstanceToPointInTime do RDS.

  Para obter instruções, consulte Restaurar uma instância de banco de dados para um momento especificado no Amazon RDS.

> **Note**
>
> Se a instância do RDS Custom para SQL Server já estiver associada manualmente a um AD, confira as configurações de Regras de portas para configuração de rede e Validação da rede, depois conclua as etapas de 1 a 4. Atualize o `--domain-fqdn`, `--domain-ou` e `--domain-auth-secret-arn` para o AD, para que as configurações e as credenciais de ingresso no domínio sejam registradas no RDS Custom para monitorar, registrar o CNAME e realizar ações de recuperação.

Quando você usa a AWS CLI, são necessários os seguintes parâmetros para que a instância de banco de dados possa usar o diretório criado:

- Em relação ao parâmetro `--domain-fqdn`, use o nome de domínio totalmente qualificado do seu AD autogerenciado.
- Para o parâmetro `--domain-ou`, use a OU criada em seu AD autogerenciado.
- Em relação ao parâmetro `--domain-auth-secret-arn`, use o valor do ARN do segredo que você criou.

> **Important**
>
> Se você modificar uma instância de banco de dados para associá-la ou removê-la de um domínio do AD autogerenciado ou do AWS Managed Microsoft AD, será necessária uma reinicialização da instância de banco de dados para que a modificação entre em vigor. Você pode optar por aplicar as alterações imediatamente ou esperar até a próxima janela de manutenção. Escolher a opção Aplicar imediatamente causa tempo de inatividade para uma instância de banco de dados single-AZ. Um cluster de banco de dados multi-AZ realiza um failover antes de concluir a reinicialização. Para obter mais informações, consulte Modificar uma instância de banco de dados do Amazon RDS.

O comando da CLI a seguir cria uma instância de banco de dados do RDS Custom para SQL Server e a associa a um domínio do AWS Managed Microsoft AD ou autogerenciado.

Para Linux, macOS ou Unix:

```
aws rds create-db-instance  \
--engine custom-sqlserver-se \
--engine-version 15.00.4312.2.v1 \
--db-instance-identifier my-custom-instance \
--db-instance-class db.m5.large \
--allocated-storage 100 --storage-type io1 --iops 1000 \
--master-username my-master-username \
--master-user-password my-master-password \
--kms-key-id  my-RDSCustom-key-id \
--custom-iam-instance-profile AWSRDSCustomInstanceProfileForRdsCustomInstance  \
--domain-fqdn "corp.example.com" \
--domain-ou "OU=RDSCustomOU,DC=corp,DC=example,DC=com" \
--domain-auth-secret-arn "arn:aws:secretsmanager:region:account-number:secret:do-not-delete-rds-custom-my-AD-test-secret-123456" \
--db-subnet-group-name my-DB-subnet-grp \
--vpc-security-group-ids  my-securitygroup-id \
--no-publicly-accessible \
--backup-retention-period 3 \
--port 8200 \
--region us-west-2 \
--no-multi-az
```

Para Windows:

```
aws rds create-db-instance  ^
--engine custom-sqlserver-se ^
--engine-version 15.00.4312.2.v1 ^
--db-instance-identifier my-custom-instance ^
--db-instance-class db.m5.large ^
--allocated-storage 100 --storage-type io1 --iops 1000 ^
--master-usernamemy-master-username ^
--master-user-password my-master-password ^
--kms-key-id  my-RDSCustom-key-id ^
--custom-iam-instance-profile AWSRDSCustomInstanceProfileForRdsCustomInstance  ^
--domain-fqdn "corp.example.com" ^
--domain-ou "OU=RDSCustomOU,DC=corp,DC=example,DC=com" ^
--domain-auth-secret-arn "arn:aws:secretsmanager:region:account-number:secret:do-not-
delete-rds-custom-my-AD-test-secret-123456" ^
--db-subnet-group-name my-DB-subnet-grp ^
--vpc-security-group-ids  my-securitygroup-id ^
--no-publicly-accessible ^
--backup-retention-period 3 ^
--port 8200 ^
--region us-west-2 ^
--no-multi-az
```

Important

Se o NetBIOS para AWS Managed Microsoft AD for corpexample, ele aparecerá como uma UO em si. Qualquer nova UO criada anteriormente aparecerá como uma UO aninhada. Para AWS Managed Microsoft AD, defina `--domain-ou` como `"OU=RDSCustomOU,OU=corpexample,DC=corp,DC=example,DC=com"`.

O comando a seguir modifica uma instância de banco de dados do RDS Custom para SQL Server existente para usar um domínio do Active Directory.

Para Linux, macOS ou Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier my-custom-instance \
    --domain-fqdn "corp.example.com" \
    --domain-ou "OU=RDSCustomOU,DC=corp,DC=example,DC=com" \
    --domain-auth-secret-arn "arn:aws:secretsmanager:region:account-number:secret:do-
not-delete-rds-custom-my-AD-test-secret-123456" \
```

Para Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier my-custom-instance ^
    --domain-fqdn "corp.example.com" ^
    --domain-ou "OU=RDSCustomOU,DC=corp,DC=example,DC=com" ^
    --domain-auth-secret-arn "arn:aws:secretsmanager:region:account-number:secret:do-not-delete-rds-custom-my-AD-test-secret-123456" ^
```

O comando da CLI a seguir remove uma instância de banco de dados do RDS Custom para SQL Server de um domínio do Active Directory.

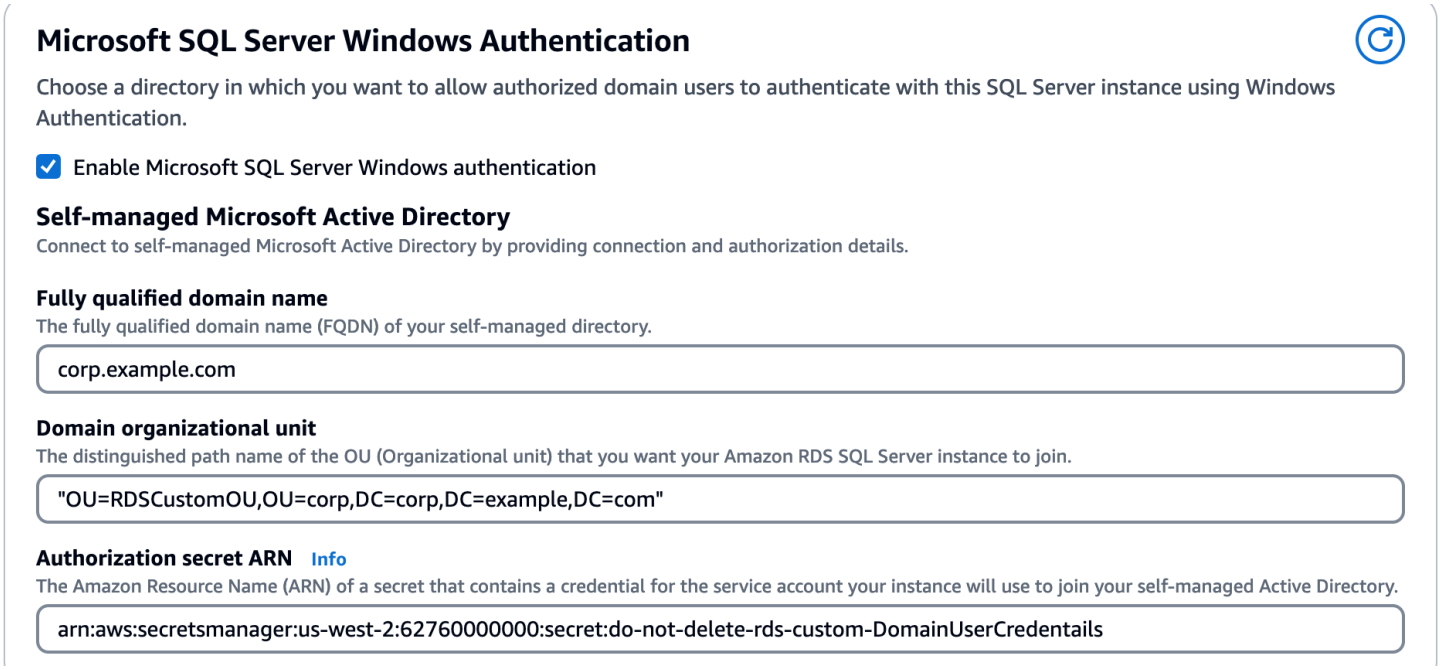
Para Linux, macOS ou Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier my-custom-instance \
    --disable-domain
```

Para Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier my-custom-instance ^
    --disable-domain
```

Ao usar o console para criar ou modificar a instância, clique em Habilitar a autenticação do Microsoft SQL Server Windows para ver as opções a seguir.



Você é responsável por garantir que o FQDN do domínio seja resolvido para os endereços IP do controlador de domínio. Se os IPs do controlador de domínio não estiverem sendo resolvidos, as operações de ingresso no domínio falharão, mas a criação da instância do RDS Custom para SQL Server será bem-sucedida. Para obter informações sobre a solução de problemas, consulte Solucionar problemas no Active Directory.

Etapa 6: criar login do SQL Server para autenticação do Windows

Use as credenciais de usuário mestre do Amazon RDS para se conectar à instância de banco de dados do SQL Server como você faria para qualquer outra instância de banco de dados. Como a instância de banco de dados é associada ao domínio do AD, é possível provisionar logins e usuários do SQL Server. Faça isso usando o utilitário de usuários e grupos do AD no domínio do AD. As permissões de banco de dados são gerenciadas por meio de permissões padrão do SQL Server concedidas e revogadas a esses logins do Windows.

Para que um usuário do AD faça a autenticação com o SQL Server, deve existir um login do Windows para SQL Server para o usuário do AD ou um grupo do Active Directory do qual o usuário seja membro. O controle de acesso refinado é gerenciado por meio da concessão e revogação de permissões nesses logins do SQL Server. Um usuário que não tenha um login do SQL Server nem pertença a um grupo que tenha um login não consegue acessar a instância de banco de dados do SQL Server.

A permissão `ALTER ANY LOGIN` é necessária para criar um login do AD para SQL Server. Se você ainda não criou logins com essa permissão, conecte-se como o usuário principal da instância de

banco de dados usando a autenticação do SQL Server e crie logins do AD para SQL Server no contexto do usuário principal.

É possível executar um comando de linguagem de definição de dados (DDL) como o seguinte para criar um login do SQL Server para um usuário ou um grupo do AD.

```
USE [master]
GO
CREATE LOGIN [mydomain\myuser] FROM WINDOWS WITH DEFAULT_DATABASE = [master],
 DEFAULT_LANGUAGE = [us_english];
GO
```

Os usuários (humanos e aplicações) do domínio agora podem se conectar à instância do RDS Custom para SQL Server por meio de uma máquina cliente conectada ao domínio usando a autenticação do Windows.

Etapa 7: usar a autenticação do Kerberos ou NTLM

Autenticação do NTLM usando um endpoint do RDS

Cada instância de banco de dados do Amazon RDS tem um endpoint, e cada endpoint tem um nome de DNS e um número de porta para a instância de banco de dados. Para se conectar à sua instância de banco de dados usando um aplicativo cliente SQL, você precisa do nome DNS e do número da porta para sua instância de banco de dados. Para se autenticar usando o NTLM, é necessário se conectar ao endpoint do RDS.

Durante uma manutenção planejada do banco de dados ou de uma interrupção do serviço não planejada, o Amazon RDS faz failover automático para o banco de dados secundário atualizado. Dessa maneira, as operações podem ser retomadas rapidamente sem intervenção manual. As instâncias primária e secundária usam o mesmo endpoint, cujo endereço de rede física faz a transição para a secundária como parte do processo de failover. Não é necessário reconfigurar seu aplicativo quando ocorre um failover.

Autenticação de Kerberos

A autenticação baseada em Kerberos para o RDS Custom para SQL Server exige que as conexões sejam feitas com um nome de entidade principal de serviço (SPN) específico. No entanto, após um evento de failover, a aplicação pode não estar ciente do novo SPN. Para resolver isso, o RDS Custom para SQL Server oferece um endpoint baseado em Kerberos.

O endpoint baseado em Kerberos segue um formato específico. Caso o endpoint do RDS seja *rds-instance-name*.*account-region-hash*.*aws-region*.rds.amazonaws.com, o endpoint correspondente baseado em Kerberos será *rds-instance-name*.*account-region-hash*.*aws-region*.awsrds.*fully qualified domain name (FQDN)*.

Por exemplo, se o endpoint do RDS for ad-test.cocv6zwtircu.us-east-1.rds.amazonaws.com e o nome do domínio for corp-ad.company.com, o endpoint baseado em Kerberos será ad-test.cocv6zwtircu.us-east-1.awsrds.corp-ad.company.com.

Esse endpoint baseado em Kerberos pode ser usado para se autenticar na instância do SQL Server usando o Kerberos, mesmo após um evento de failover, pois o endpoint é atualizado automaticamente para apontar para o novo SPN da instância primária do SQL Server.

Encontrar o CNAME

Para encontrar o CNAME, conecte-se ao controlador de domínio e abra o Gerenciador de DNS. Acesse Forward Lookup Zones e o FQDN.

Navegue por awsrds, aws-region e hash específico de conta e região.

Se você estiver conectando a instância do EC2 do RDS Custom e tentando estabelecer conexão com o banco de dados localmente usando o CNAME, a conexão usará a autenticação do NTLM em vez do Kerberos.

Se uma conexão do NTLM for retornada depois que você conectar o CNAME por meio do cliente remoto, confira se as portas necessárias estão na lista de permissões.

Para conferir se a conexão está usando o Kerberos, realize a seguinte consulta:

```
SELECT net_transport, auth_scheme
    FROM sys.dm_exec_connections
    WHERE session_id = @@SSPID;
```

Gerenciamento de uma instância de banco de dados em um domínio

É possível usar o console, a AWS CLI ou a API do Amazon RDS para gerenciar a instância de banco de dados e a respectiva relação no domínio. Por exemplo, é possível mover a instância de banco de dados para dentro, para fora, de e entre os domínios.

Por exemplo, usando a API do Amazon RDS, você pode fazer o seguinte:

- Para tentar uma união de domínio novamente cuja associação falhou, use a operação ModifyDBInstance da API e especifique o ID do diretório da associação atual.
- Para atualizar o nome da função do IAM para a associação, use a operação `ModifyDBInstance` da API e especifique o ID do diretório da associação atual e a nova função do IAM.
- Para remover uma instância de banco de dados de um domínio, use a operação `ModifyDBInstance` da API e especifique `none` como o parâmetro do domínio.
- Para mover uma instância de banco de dados de um domínio para outro, use a operação `ModifyDBInstance` da API e especifique o identificador do novo domínio como o parâmetro do domínio.
- Para listar a associação de cada instância de banco de dados, use a operação DescribeDBInstances da API.

Restaurar uma instância de banco de dados do RDS Custom para SQL Server e adicioná-la a um domínio de Active Directory

É possível restaurar um snapshot de banco de dados ou fazer uma recuperação para um ponto no tempo (PITR) de uma instância de banco de dados do SQL Server e adicioná-la a um domínio de Active Directory. Depois que a instância de banco de dados for restaurada, modifique-a utilizando o processo explicado em Etapa 5: criar ou modificar a instância de banco de dados do RDS Custom para SQL Server para adicionar a instância de banco de dados a um domínio do AD.

Compreensão da associação de domínio

Após criar ou modificar sua instância de banco de dados, a instância se tornará um membro do domínio. O console da AWS indica o status da associação de domínio para a instância de banco de dados. O status da instância de banco de dados pode ser um dos seguintes:

- joined – a instância é membro do domínio.
- joining – a instância está em processo de se tornar membro do domínio.
- pending-join – a associação da instância está pendente.
- pending-maintenance-join: a AWS tentará tornar a instância um membro do domínio durante a próxima janela de manutenção agendada.
- pending-removal – a remoção da instância do domínio está pendente.
- pending-maintenance-removal: a AWS tentará remover a instância do domínio durante a próxima janela de manutenção programada.

- failed – um problema de configuração impediu que a instância se associasse ao domínio. Verifique e corrija sua configuração antes de emitir novamente o comando de modificação da instância.
- removing – a instância está sendo removida do domínio.

Uma solicitação para se tornar um membro de um domínio pode falhar devido a um problema de conectividade de rede ou a uma função do IAM incorreta. Por exemplo, é possível criar uma instância de banco de dados ou modificar uma instância existente e não conseguir transformar a instância de banco de dados em um membro de um domínio. Nesse caso, reexecute o comando para criar ou modificar a instância de banco de dados ou modifique a instância recém-criada para ingressar no domínio.

Solucionar problemas no Active Directory

Veja a seguir alguns problemas que você pode encontrar ao configurar ou modificar um AD.

Código de erro	Descrição	Causas comuns	Sugestões de solução de problemas
Erro 2 / 0x2	O sistema não conseguiu encontrar o arquivo especificado.	O formato ou a localização da unidade organizacional (OU) especificada com o parâmetro `–domain-ou` é inválido. A conta de serviço do domínio especificada por meio do AWS Secrets Manager não tem as permissões necessárias para se associar à OU.	Revise o parâmetro `–domain-ou`. Certifique-se de que a conta de serviço do domínio tenha as permissões corretas para a OU.
Erro 5 / 0x5	Acesso negado.	Permissões configuradas incorretamente para a conta de serviço do domínio, ou a conta de computador já existe no domínio.	Revise as permissões da conta de serviço no domínio e verifique se a conta de computador do RDS não está duplicada no domínio. É possível verificar o nome da conta

Código de erro	Descrição	Causas comuns	Sugestões de solução de problemas
			de computador do RDS executando `SELECT @@SERVERNAME` na instância de banco de dados do RDS Custom para SQL Server. Se você estiver usando multi-AZ, tente reinicializar com failover, depois verifique a conta de computador do RDS novamente. Para ter mais informações, consulte Reinicializar uma instância de banco de dados.
Erro 87 / 0x57	O parâmetro está incorreto.	A conta de serviço do domínio especificada por meio do AWS Secrets Manager não tem as permissões corretas. O perfil do usuário também pode estar corrompido.	Revise os requisitos da conta de serviço do domínio.
Erro 234 / 0xEA	A unidade organizacional (OU) especificada não existe.	A UO especificada com o parâmetro `–domain-ou` não existe no AD.	Revise o parâmetro `–domain-ou` e verifique se a UO especificada existe no AD.

Código de erro	Descrição	Causas comuns	Sugestões de solução de problemas
Erro 1326 / 0x52E	O nome de usuário ou a senha estão incorretos.	As credenciais da conta de serviço do domínio fornecidas no AWS Secrets Manager contêm um nome de usuário desconhecido ou uma senha incorreta. A conta do domínio também pode estar desabilitada no AD.	Verifique se as credenciais fornecidas no AWS Secrets Manager estão corretas e se a conta de domínio está habilitada no Active Directory.
Erro 1355 / 0x54B	O domínio especificado não existe ou não foi possível estabelecer conexão.	O domínio está inativo, o conjunto especificado de IPs DNS está inacessível ou o FQDN especificado está inacessível.	Revise os parâmetros `–domain-dns-ips` e `–domain-fqdn` para garantir que estejam corretos. Revise a configuração de rede da instância de banco de dados do RDS Custom para SQL Server e garanta que o AD esteja acessível.
Erro 1722 / 0x6BA	O servidor RPC não está disponível.	Houve um problema ao acessar o serviço RPC do domínio de AD. Isso pode ser devido a um problema de serviço ou rede.	Confirme se o serviço RPC está sendo executado nos controladores de domínio e se as portas TCP `135` e `49152-65535` no domínio podem ser acessadas pela instância de banco de dados do RDS Custom para SQL Server.

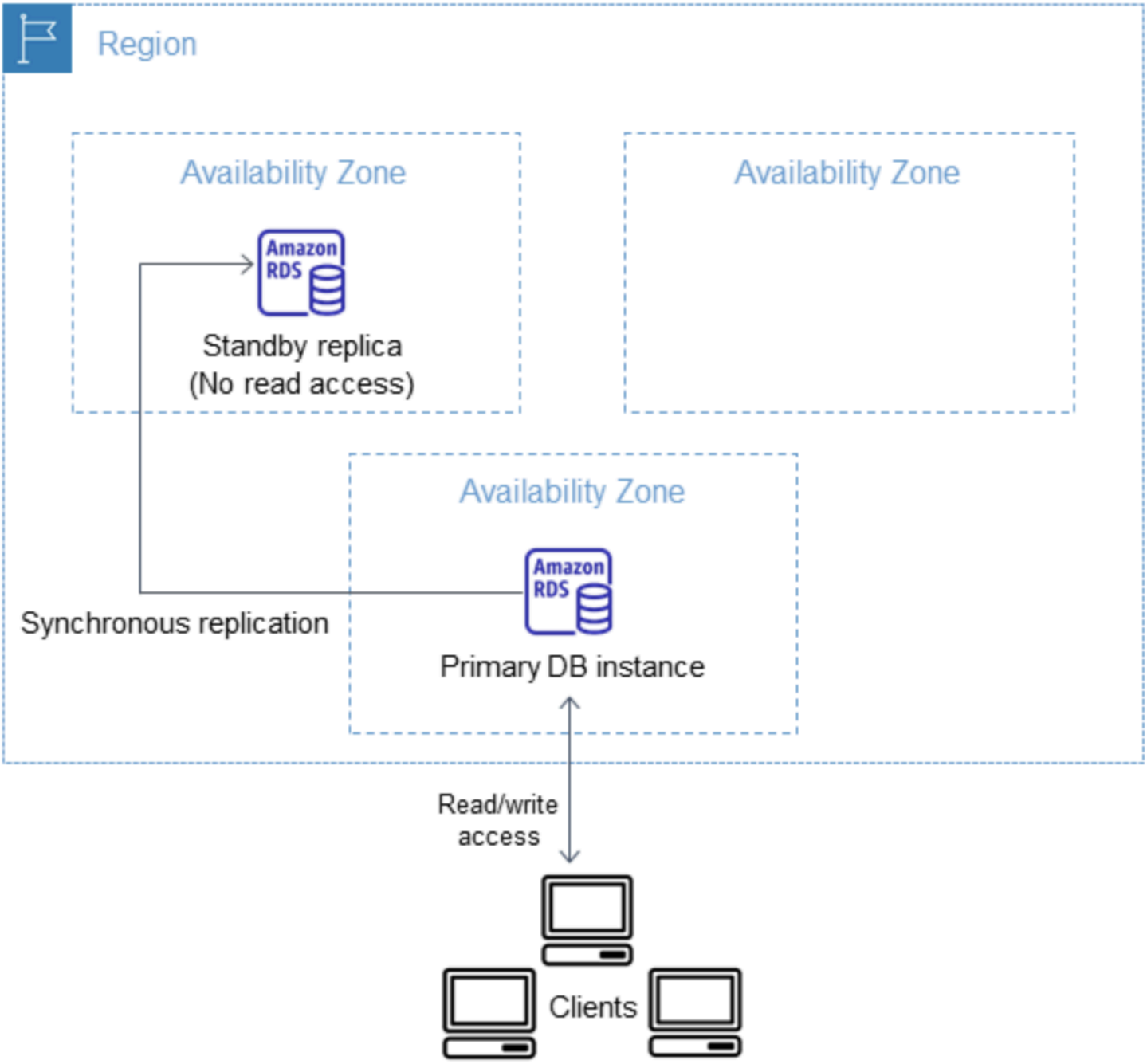
Código de erro	Descrição	Causas comuns	Sugestões de solução de problemas
Erro 2224 / 0x8B0	A conta de usuário já existe.	A conta de computador que está tentando ser adicionada ao AD já existe.	Identifique a conta de computador executando o `SELECT @@SERVERNAME` na instância de banco de dados do RDS Custom para SQL Server, depois remova-a cuidadosamente do AD.
Erro 2242 / 0x8c2	A senha deste usuário expirou.	A senha da conta de serviço do domínio especificada por meio do AWS Secrets Manager expirou.	Atualize a senha da conta de serviço do domínio usada para associar a instância de banco de dados do RDS Custom para SQL Server ao AD.

Gerenciar uma implantação multi-AZ para o RDS Custom para SQL Server

Em uma implantação de instância de banco de dados multi-AZ para RDS para SQL Server, o Amazon RDS provisiona e mantém automaticamente uma réplica em espera síncrona em outra zona de disponibilidade (AZ). A instância de banco de dados primária é sincronicamente replicada nas zonas de disponibilidade para uma réplica em espera a fim de oferecer redundância de dados.

> **Important**
>
> Uma implantação multi-AZ para RDS Custom para SQL Server é diferente de multi-AZ para RDS para SQL Server. Ao contrário do multi-AZ para RDS para SQL Server, você deve configurar pré-requisitos para o RDS Custom para SQL Server antes de criar sua instância de banco de dados multi-AZ, pois o RDS Custom é executado em sua própria conta, o que exige permissões.
>
> Se você não atender aos pré-requisitos, poderá ocorrer uma falha na execução de sua instância de banco de dados multi-AZ ou ela poderá ser revertida automaticamente para uma instância de banco de dados single-AZ. Para ter mais informações sobre pré-requisitos, consulte Pré-requisitos para uma implantação multi-AZ com RDS Custom para SQL Server.

Executar uma instância de banco de dados com alta disponibilidade pode aumentar a disponibilidade durante a manutenção planejada do sistema. No caso de uma manutenção planejada do banco de dados ou de uma interrupção não planejada do serviço, o Amazon RDS faz failover automático para a instância de banco de dados secundário atualizada. Essa funcionalidade permite que as operações do banco de dados sejam retomadas rapidamente sem intervenção manual. As instâncias primária e em espera usam o mesmo endpoint, cujo endereço de rede física faz a transição para a réplica secundária como parte do processo de failover. Não é necessário reconfigurar seu aplicativo quando ocorre um failover.



Você pode criar uma implantação multi-AZ do RDS Custom para SQL Server ao especificar multi-AZ ao criar uma instância de banco de dados do RDS Custom. Você pode usar o console para converter instâncias de banco de dados existentes do RDS Custom para SQL Server em implantações multi-AZ modificando a instância de banco de dados e especificando a opção multi-AZ. Você também pode especificar uma implantação de instância de banco de dados multi-AZ com a CLI da AWS ou a API do Amazon RDS.

O console do RDS mostra a zona de disponibilidade da réplica em espera (a AZ secundária). Você também pode usar o comando `describe-db-instances` da CLI ou a operação `DescribeDBInstances` da API para localizar a AZ secundária.

Instâncias de banco de dados do RDS Custom para SQL Server com implantação multi-AZ podem ter maior latência de gravação e confirmação em comparação com uma implantação single-AZ. Esse aumento pode acontecer devido à replicação de dados síncrona entre as instâncias de banco de dados. É possível ter uma alteração na latência se sua implantação falhar na réplica em espera, ainda que o AWS seja desenvolvido com conectividade de rede de baixa latência entre zonas de disponibilidade.

> **Note**
>
> Para workloads de produção, recomendamos usar uma classe de instância de banco de dados com IOPS provisionadas (operações de entrada/saída por segundo) para obter uma performance rápida e consistente. Para ter mais informações sobre classes de instância de banco de dados, consulte Requisitos e limitações do Amazon RDS Custom for SQL Server.

Tópicos

- Disponibilidade de região e versão
- Gerenciar uma implantação multi-AZ do RDS Custom para SQL Server
- Pré-requisitos para uma implantação multi-AZ com RDS Custom para SQL Server
- Criar uma implantação multi-AZ do RDS Custom para SQL Server
- Modificar uma implantação single-AZ do RDS Custom para SQL Server para uma implantação multi-AZ
- Modificar uma implantação do RDS Custom para SQL Server de uma implantação multi-AZ para single-AZ
- Processo de failover para uma implantação multi-AZ do RDS Custom para SQL Server

Disponibilidade de região e versão

As implantações multi-AZ do RDS Custom para SQL Server são compatíveis com as seguintes edições do SQL Server:

- SQL Server 2022 e 2019, edições Enterprise, Standard, Web e Developer

> **Note**
>
> Implantações multi-AZ para RDS Custom para SQL Server não são aceitas no SQL Server 2019 CU8 (15.00.4073.23) ou versões inferiores.

As implantações multi-AZ para RDS Custom para SQL Server estão disponíveis em todas as regiões em que o RDS Custom para SQL Server está disponível. Para ter mais informações sobre a disponibilidade de regiões de implantações multi-AZ para o RDS Custom para SQL Server, consulte Regiões e mecanismos de banco de dados compatíveis com o RDS Custom para SQL Server.

Gerenciar uma implantação multi-AZ do RDS Custom para SQL Server

As implantações multi-AZ do RDS Custom para SQL Server têm as seguintes limitações:

- Implantações multi-AZ entre regiões não são compatíveis.
- Não é possível configurar a instância de banco de dado secundária para aceitar a atividade de leitura de banco de dados.
- Quando você usa uma versão de mecanismo personalizada (CEV) com uma implantação multi-AZ, sua instância de banco de dados secundária também usa a mesma CEV. A instância de banco de dados secundária não pode usar uma CEV diferente.

Pré-requisitos para uma implantação multi-AZ com RDS Custom para SQL Server

Se você tiver uma implantação single-AZ do RDS Custom para SQL Server existente, os pré-requisitos adicionais a seguir serão necessários antes de modificá-la para uma implantação multi-AZ. Você pode optar por atender aos pré-requisitos manualmente ou com o modelo do CloudFormation fornecido. O modelo mais recente do CloudFormation contém os pré-requisitos para implantações single-AZ e multi-AZ.

> **Important**
>
> Para simplificar a configuração, recomendamos que você use o arquivo de modelo AWS CloudFormation mais recente fornecido nas instruções de configuração da rede para criar os pré-requisitos. Para ter mais informações, consulte Configurar com o AWS CloudFormation.

> **Note**
>
> Ao modificar uma implantação single-AZ do RDS Custom para SQL Server existente para uma implantação multi-AZ, você deverá cumprir esses pré-requisitos. Se você não cumprir os pré-requisitos, a configuração multi-AZ falhará. Para configurar os pré-requisitos, conclua as etapas em Modificar uma implantação single-AZ do RDS Custom para SQL Server para uma implantação multi-AZ.

- Atualize as regras de entrada e saída do grupo de segurança do RDS para permitir a porta 1120.
- Adicione uma regra à sua lista de controle de acesso (ACL) de rede privada que permita portas TCP `0-65535` para a VPC de instância de banco de dados.
- Crie endpoints de VPC do Amazon SQS que permitam que a instância de banco de dados do RDS Custom para SQL Server se comunique com o SQS.
- Atualize as permissões do SQS na função do perfil da instância.

Criar uma implantação multi-AZ do RDS Custom para SQL Server

Para criar uma implantação multi-AZ do RDS Custom para SQL Server, siga as etapas em Criar e conectar-se a uma instância de banco de dados Amazon RDS Custom for SQL Server.

> **Important**
>
> Para simplificar a configuração, recomendamos que você use o arquivo de modelo AWS CloudFormation mais recente fornecido nas instruções de configuração da rede. Para ter mais informações, consulte Configurar com o AWS CloudFormation.

A criação de uma implantação multi-AZ leva alguns minutos para ser concluída.

Modificar uma implantação single-AZ do RDS Custom para SQL Server para uma implantação multi-AZ

É possível modificar uma instância de banco de dados do RDS Custom para SQL Server de uma implantação single-AZ para multi-AZ. Quando você modifica a instância de banco de dados, o Amazon RDS realiza várias ações:

- Tira um snapshot da instância de banco de dados primária.
- Cria volumes para a réplica em espera baseados no snapshot. Esses volumes são inicializados em segundo plano e a performance máxima do volume é alcançada depois que os dados são totalmente inicializados.
- Ativa a replicação síncrona em nível de bloco entre as instâncias de banco de dados primária e secundária.

> **Important**
>
> Recomendamos que você evite modificar sua instância de banco de dados do RDS Custom para SQL Server de uma implantação single-AZ para multi-AZ em uma instância de banco de dados de produção durante períodos de pico de atividade.

A AWS usa um snapshot para criar a instância em espera para evitar tempo de inatividade ao converter da implantação single-AZ em multi-AZ. No entanto, poderá haver um impacto na performance durante e após a conversão em multi-AZ. Esse impacto pode ser significativo para workloads sensíveis à latência de gravação. Embora esse recurso permita que grandes volumes sejam restaurados rapidamente de snapshots, ele pode causar aumento na latência das operações de E/S devido à replicação síncrona. Essa latência pode afetar a performance do seu banco de dados.

> **Note**
>
> Se você criou a instância de banco de dados do RDS Custom para SQL Server antes de 29 de agosto de 2024, aplique o patch para a versão secundária mais recente antes de modificá-la.
>
> - Para instâncias do SQL Server 2019, atualize a versão do mecanismo de banco de dados para `15.00.4410.1.v1` ou posterior.
> - Para instâncias do SQL Server 2022, atualize a versão do mecanismo de banco de dados para `16.00.4150.1.v1` ou posterior.

Tópicos

- Configurar pré-requisitos para modificar uma implantação single-AZ para multi-AZ usando o CloudFormation
- Configurar pré-requisitos para modificar uma implantação single-AZ para multi-AZ manualmente
- Modifique usando o console do RDS, a CLI da AWS ou a API do RDS.

Configurar pré-requisitos para modificar uma implantação single-AZ para multi-AZ usando o CloudFormation

Para usar uma implantação multi-AZ, você deve garantir que aplicou o modelo mais recente do CloudFormation com pré-requisitos ou configurar manualmente os pré-requisitos mais recentes. Se você já tiver aplicado o modelo de pré-requisitos mais recente do CloudFormation, poderá ignorar essas etapas.

Como configurar os pré-requisitos de implantação multi-AZ do RDS Custom para SQL Server usando o CloudFormation

1. Abra o console do CloudFormation em https://console.aws.amazon.com/cloudformation.
2. Para iniciar o assistente de criação de pilha, selecione a pilha existente que você usou para criar uma implantação single-AZ e escolha Atualizar.

   A página Criar pilha é exibida.
3. Em Pré-requisito: preparar modelo, selecione O modelo está pronto.
4. Na página Specify template (Especificar modelo), faça o seguinte:

   a. Baixe o arquivo de modelo do AWS CloudFormation mais recente. Abra o menu de contexto (clique com o botão direito do mouse) do link custom-sqlserver-onboard.zip e selecione Save Link As (Salvar link como).

   b. Salve e extraia o arquivo `custom-sqlserver-onboard.json` no computador.

   c. Para Template source (Origem do template), escolha Upload a template file (Fazer upload de um arquivo de template).

   d. Em Choose file (Escolher arquivo), acesse `custom-sqlserver-onboard.json` e escolha essa opção.
5. Escolha Próximo.

   A página Specify stack details (Especificar detalhes da pilha) é exibida.
6. Para manter as opções padrão, escolha Next (Próximo).

A página Opções avançadas é exibida.

7. Para manter as opções padrão, escolha Next (Próximo).
8. Para manter as opções padrão, escolha Next (Próximo).
9. Na página Revisar alterações, faça o seguinte:

   a. Para Capabilities (Recursos), marque a caixa de seleção para confirmar que o AWS CloudFormation pode criar os recursos do IAM com nomes personalizados.

   b. Selecione Enviar.
10. Verifique se a atualização foi bem-sucedida. O status de uma operação bem-sucedida exibe `UPDATE_COMPLETE`.

Se a atualização falhar, todas as novas configurações especificadas no processo de atualização serão revertidas. O recurso existente ainda poderá ser usado. Por exemplo, se você adicionar regras de ACL de rede 18 e 19, mas existissem regras com os mesmos números, a atualização retornará o seguinte erro: `Resource handler returned message: "The network acl entry identified by 18 already exists.` Nesse cenário, você pode modificar as regras de ACL existentes para usar um número menor que 18 e, depois, tentar a atualização novamente.

Configurar pré-requisitos para modificar uma implantação single-AZ para multi-AZ manualmente

> **Important**
>
> Para simplificar a configuração, recomendamos que você use o arquivo de modelo AWS CloudFormation mais recente fornecido nas instruções de configuração da rede. Para ter mais informações, consulte Configurar pré-requisitos para modificar uma implantação single-AZ para multi-AZ usando o CloudFormation.

Se você optar por configurar recursos manualmente, realize as tarefas a seguir.

1. Abra o console da Amazon VPC, em https://console.aws.amazon.com/vpc/.
2. Selecione Endpoint. Aparecerá a página Create Endpoint (Criar endpoint).
3. Em Categoria do serviço, selecione Serviços da AWS.
4. Em Serviços, pesquise *SQS*

5. Na VPC, selecione a VPC em que sua instância de banco de dados do RDS Custom para SQL Server é implantada.
6. Em Sub-redes, selecione as sub-redes em que sua instância de banco de dados do RDS Custom para SQL Server está implantada.
7. Em Grupos de segurança, selecione o grupo *-vpc-endpoint-sg*.
8. Em Política, selecione Personalizada
9. Em sua política personalizada, substitua a *partição da AWS*, a *região*, o *accountId* e o *perfil da instância do IAM* por seus próprios valores.

   JSON

```
                 {
        "Version": "2012-10-17",
        "Statement": [
            {
                "Condition": {
                    "StringLike": {
                        "aws:ResourceTag/AWSRDSCustom": "custom-sqlserver"
                    }
                },
                "Action": [
                    "SQS:SendMessage",
                    "SQS:ReceiveMessage",
                    "SQS:DeleteMessage",
                    "SQS:GetQueueUrl"
                ],
                "Resource": "arn:${AWS::Partition}:sqs:${AWS::Region}:
${AWS::AccountId}:do-not-delete-rds-custom-*",
                "Effect": "Allow",
                "Principal": {
                    "AWS": "arn:${AWS::Partition}:iam::
${AWS::AccountId}:role/{IAM-Instance-role}"
                }
            }
        ]
    }
```

10. Atualize o Perfil da instância com permissão para acessar o Amazon SQS. Substitua a *partição da AWS*, a *região* e o *accountId* por seus próprios valores.

```
                        {
    "Sid": "SendMessageToSQSQueue",
    "Effect": "Allow",
    "Action": [
      "SQS:SendMessage",
      "SQS:ReceiveMessage",
      "SQS:DeleteMessage",
      "SQS:GetQueueUrl"

    ],
    "Resource": [
      {
        "Fn::Sub": "arn:${AWS::Partition}:sqs:${AWS::Region}:${AWS::AccountId}:do-
not-delete-rds-custom-*"
      }
    ],
    "Condition": {
      "StringLike": {
        "aws:ResourceTag/AWSRDSCustom": "custom-sqlserver"
      }
    }
  }
                        >
```

11. Atualize as regras de entrada e saída do grupo de segurança do Amazon RDS para permitir a porta 1120.

    a. Em Grupos de segurança, selecione o grupo *-rds-custom-instance-sg*.

    b. Em Regras de entrada, crie uma regra TCP personalizada para permitir a porta *1120* do grupo de origem *-rds-custom-instance-sg*.

    c. Em Regras de saída, crie uma regra TCP personalizada a fim de permitir a porta *1120* ao grupo de destino *-rds-custom-instance-sg*.

12. Adicione uma regra à sua lista de controle de acesso (ACL) de rede privada que permita portas TCP `0-65535` para a sub-rede de origem da instância de banco de dados.

> **Note**
>
> Ao criar uma regra de entrada e uma regra de saída, anote o maior número de regra existente. As novas regras que você criar devem ter um número de regra menor que 100 e não corresponder a nenhum número de regra existente.

a. Em ACLs de rede, selecione o grupo *`-private-network-acl`*.
b. Em Regras de entrada, crie uma regra Todas TCP para permitir portas TCP `0-65535` com uma fonte de *`privatesubnet1`* e *`privatesubnet2`*.
c. Em Regras de saída, crie uma regra Todas TCP para permitir portas TCP `0-65535` para *`privatesubnet1`* e *`privatesubnet2`* de destino.

Modifique usando o console do RDS, a CLI da AWS ou a API do RDS.

Depois de concluir os pré-requisitos, você pode modificar uma instância de banco de dados do RDS Custom para SQL Server de uma implantação single-AZ para multi-AZ usando o console do RDS, a CLI da AWS ou a API do RDS.

Console

Como modificar uma implantação do RDS Custom para SQL Server de single-AZ para multi-AZ

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No console do Amazon RDS, escolha Databases (Bancos de dados).

   O painel Databases (Bancos de dados) é exibido.
3. Selecione a instância de banco de dados do RDS Custom para SQL Server que você deseja modificar.
4. Em Ações, selecione Converter em implantação multi-AZ.
5. Na página Confirmação, selecione Aplicar imediatamente para aplicar as alterações imediatamente. A escolha dessa opção não causa tempo de inatividade, mas pode causar um possível impacto na performance. Você também pode optar por aplicar a atualização durante a próxima janela de manutenção. Para ter mais informações, consulte Usar a configuração de programação de modificações.

6. Na página Confirmação, selecione Converter em multi-AZ.

AWS CLI

Para converter em uma implantação de instância de banco de dados multi-AZ usando oAWS CLI, chame o comando modify-db-instance e defina a opção `--multi-az`. Especifique o identificador da instância de banco de dados e os valores para as outras opções que deseja modificar. Para ter mais informações sobre cada opção, consulte Configurações para instâncias de banco de dados.

Example

O código a seguir modifica `mycustomdbinstance` incluindo a opção `--multi-az`. As alterações serão aplicadas durante a janela de manutenção usando `--no-apply-immediately`. Use `--apply-immediately` para aplicar as alterações imediatamente. Para obter mais informações, consulte Usar a configuração de programação de modificações.

Para Linux, macOS ou Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier mycustomdbinstance \
    --multi-az \
    --no-apply-immediately
```

Para Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mycustomdbinstance ^
    --multi-az  \ ^
    --no-apply-immediately
```

API do RDS

Para converter em uma implantação de instância de banco de dados multi-AZ com a API do RDS, chame a operação ModifyDBInstance e defina o parâmetro `MultiAZ` como verdadeiro.

Modificar uma implantação do RDS Custom para SQL Server de uma implantação multi-AZ para single-AZ

É possível modificar uma instância de banco de dados do RDS Custom para SQL Server de uma implantação multi-AZ para single-AZ.

Console

Como modificar uma instância de banco de dados do RDS Custom para SQL Server de uma implantação multi-AZ para single-AZ.

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No console do Amazon RDS, escolha Databases (Bancos de dados).

   O painel Databases (Bancos de dados) é exibido.
3. Selecione a instância de banco de dados do RDS Custom para SQL Server que você deseja modificar.
4. Em Implantação multi-AZ, selecione Não.
5. Na página Confirmação, selecione Aplicar imediatamente para aplicar as alterações imediatamente. A escolha dessa opção não causa tempo de inatividade, mas pode causar um possível impacto na performance. Você também pode optar por aplicar a atualização durante a próxima janela de manutenção. Para ter mais informações, consulte Usar a configuração de programação de modificações.
6. Na página Confirmação, selecione Modificar instância de banco de dados.

AWS CLI

Para modificar uma implantação multi-AZ para single-AZ usando oAWS CLI, chame o comando modify-db-instance e inclua a opção `--no-multi-az`. Especifique o identificador da instância de banco de dados e os valores para as outras opções que deseja modificar. Para ter mais informações sobre cada opção, consulte Configurações para instâncias de banco de dados.

Example

O código a seguir modifica `mycustomdbinstance` incluindo a opção `--no-multi-az`. As alterações serão aplicadas durante a janela de manutenção usando `--no-apply-immediately`. Use `--apply-immediately` para aplicar as alterações imediatamente. Para ter mais informações, consulte Usar a configuração de programação de modificações.

Para Linux, macOS ou Unix:

```
aws rds modify-db-instance \
```

```
    --db-instance-identifier mycustomdbinstance \
    --no-multi-az  \
    --no-apply-immediately
```

Para Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mycustomdbinstance ^
    --no-multi-az \ ^
    --no-apply-immediately
```

API do RDS

Para modificar uma implantação multi-AZ para single-AZ usando a API do RDS, chame a operação ModifyDBInstance e defina o parâmetro `MultiAZ` como `false`.

Processo de failover para uma implantação multi-AZ do RDS Custom para SQL Server

Se uma interrupção planejada ou não planejada da sua instância de banco de dados for o resultado de um defeito de infraestrutura, o Amazon RDS alternará automaticamente para uma réplica em espera em outra zona de disponibilidade se você tiver ativado o multi-AZ. O tempo de conclusão do failover depende da atividade do banco de dados e de outras condições no momento em que a instância de banco de dados primária se tornou indisponível. Em geral, os tempos de failover variam de 60 a 120 segundos. No entanto, transações grandes ou um processo de recuperação longo podem aumentar o tempo de failover. Quando o failover é concluído, o console do RDS pode levar mais um tempo para mostrar a nova zona de disponibilidade.

> **Note**
>
> Você pode forçar um failover manualmente ao reinicializar uma instância de banco de dados com failover. Para ter mais informações sobre como reinicializar uma instância de banco de dados, consulte Reinicializar uma instância de banco de dados .

O Amazon RDS processa os failovers automaticamente para que você possa retomar as operações de banco de dados o mais rápido possível e sem intervenção administrativa. A instância de banco de dados principal muda automaticamente para a réplica em espera se alguma das condições descritas na tabela a seguir ocorrer. Os motivos do failover podem ser visualizados no log de eventos do RDS.

Motivo do failover	Descrição
`The operating system for the RDS Custom for SQL Server Multi-AZ DB instance is being patched in an offline operation`	Um failover foi acionado durante a janela de manutenção para um patch de SO ou uma atualização de segurança. Para ter mais informações, consulte Manutenção de uma instância de banco de dados.
`The primary host of the RDS Custom for SQL Server Multi-AZ DB instance is unhealthy.`	A implantação de instância de banco de dados multi-AZ detectou uma instância de banco de dados primária danificada e executou failover.
`The primary host of the RDS Custom for SQL Server Multi-AZ DB instance is unreachable due to loss of network connectivity.`	O monitoramento do RDS detectou uma falha de alcançabilidade de rede na instância de banco de dados principal e acionou um failover.
`The RDS Custom for SQL Server Multi-AZ DB instance was modified by the customer.`	Uma modificação da instância de banco de dados acionou um failover. Para ter mais informações, consulte Modificar uma instância de banco de dados do RDS Custom for SQL Server.
`The storage volume of the primary host of the RDS Custom for SQL Server Multi-AZ DB instance experienced a failure.`	A implantação de instância de banco de dados multi-AZ detectou um problema de armazenamento na instância de banco de dados primária e executou o failover.

Motivo do failover	Descrição
`The user requested a failover of the RDS Custom for SQL Server Multi-AZ DB instance.`	A instância de banco de dados multi-AZ do RDS Custom para SQL Server foi reinicializada com failover. Para ter mais informações, consulte Reinicializar uma instância de banco de dados .
`The RDS Custom for SQL Server Multi-AZ primary DB instance is busy or unresponsive.`	A instância de banco de dados principal não responde. Recomendamos tentar as seguintes etapas: • Examine os logs de evento e os logs do CloudWatch quanto ao uso excessivo de CPU, memória ou espaço de troca. Para ter mais informações, consulte Trabalhar com a notificação de eventos do Amazon RDS. • Crie uma regra que seja acionada em um evento do Amazon RD. Para ter mais informações, consulte Criar uma regra que é acionada em um evento do Amazon RDS. • Avalie sua workload para determinar se você está usando a classe de instância de banco de dados apropriada. Para ter mais informações, consulte Classes de instâncias de banco de dados .

Para determinar se ocorreu failover na instância de banco de dados multi-AZ, faça o seguinte:

- Configure assinaturas de eventos de banco de dados para notificar você por e-mail ou SMS de que um failover foi iniciado. Para ter mais informações sobre eventos do , consulte Trabalhar com a notificação de eventos do Amazon RDS.
- Visualize seus eventos de banco de dados usando o console do RDS ou operações de API.
- Visualize o estado atual da implantação de instância de banco de dados multi-AZ do RDS Custom para SQL Server usando o console do RDS, a CLI ou operações de API.

Configurações de tempo de vida (TTL) com aplicações usando uma implantação multi-AZ do RDS Custom para SQL Server

O mecanismo de failover modifica automaticamente o registro de Domain Name System (DNS) da instância de banco de dados para apontar para a instância de banco de dados em espera. Como

resultado, você precisará restabelecer todas as conexões existentes para sua instância de banco de dados. Garanta que qualquer valor de configuração de tempo de vida útil do cache DNS (TTL) seja baixo e confirme que sua aplicação não armazene o DNS em cache por um período prolongado. Um valor alto de TTL pode impedir que sua aplicação se reconecte rapidamente à instância de banco de dados após o failover.

Fazer backup e restauração de uma instância de banco de dados do Amazon RDS Custom

Como o Amazon RDS, o RDS Custom cria e salva backups automatizados da instância de banco de dados do RDS Custom para SQL Server quando a retenção do backup está habilitada. Também é possível fazer backup da instância de banco de dados manualmente. Os backups automatizados são compostos de backups de snapshots e backups de logs de transações. Os backups de snapshots são feitos para todo o volume de armazenamento da instância de banco de dados durante a janela de backup especificada. Os backups de logs de transações são feitos para os bancos de dados elegíveis para PITR em um intervalo regular. O RDS Custom salva os backups automatizados da instância de banco de dados de acordo com o período de retenção de backup especificado. É possível usar backups automatizados para recuperar a instância de banco de dados para um ponto no tempo dentro do período de retenção de backup.

Também é possível fazer backups de snapshots manualmente. É possível criar uma instância de banco de dados por meio desses backups de snapshots a qualquer momento. Para ter mais informações sobre como criar um snapshot de banco de dados manualmente, consulte Criar um snapshot do RDS Custom for SQL Server.

Embora os backups de snapshots sirvam operacionalmente como backups completos, você é cobrado somente pelo uso incremental do armazenamento. O primeiro snapshot de uma instância de banco de dados do RDS Custom os dados da instância de banco de dados completa. Os snapshots subsequentes do mesmo banco de dados são incrementais, o que significa que somente os dados que foram alterados depois do snapshot mais recente serão salvos.

Tópicos

- Criar um snapshot do RDS Custom for SQL Server
- Restaurar de um snapshot de banco de dados do RDS Custom for SQL Server
- Restaurar uma instância do RDS Custom for SQL Server para um ponto anterior no tempo
- Excluir um snapshot do RDS Custom for SQL Server
- Excluir backups automatizados do RDS Custom for SQL Server

Criar um snapshot do RDS Custom for SQL Server

O RDS Custom for SQL Server cria um snapshot do volume de armazenamento da instância de banco de dados, fazendo backup de toda a instância de banco de dados e não apenas dos bancos

de dados individuais. Ao criar um snapshot, especifique de qual instância de banco de dados do RDS Custom for SQL Server é necessário fazer backup. Atribua um nome ao snapshot para que ele possa ser restaurado posteriormente.

Ao criar um snapshot, o RDS Custom para SQL Server cria um snapshot do Amazon EBS para o volume `(D:)`, que é o volume de banco de dados anexado à instância de banco de dados. Para facilitar a associação de snapshots a uma instância de banco de dados específica, eles são marcados com `DBSnapshotIdentifier`, `DbiResourceId` e `VolumeType`.

A criação de um snapshot de banco de dados resulta em uma breve suspensão de E/S. Essa suspensão pode durar desde alguns segundos até alguns minutos, dependendo do tamanho e da classe da sua instância de banco de dados. O tempo de criação do snapshot varia dependendo do número total e do tamanho dos bancos de dados. Para saber mais sobre o número de bancos de dados elegíveis para uma operação de recuperação para um ponto no tempo (PITR), consulte Número de bancos de dados elegíveis para PITR por tipo de classe de instância.

Como o snapshot inclui todo o volume de armazenamento, o tamanho de arquivos, como arquivos temporários, também afeta o tempo necessário para criar esse snapshot. Para saber mais sobre como criar snapshots, consulte Criar um snapshot de banco de dados para uma instância de banco de dados single-AZ para o Amazon RDS.

Crie um snapshot do RDS Custom for SQL Server utilizando o console ou a AWS CLI.

Console

Para criar um snapshot do RDS Custom

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Bancos de dados.
3. Na lista de instâncias de banco de dados do RDS Custom, escolha a instância da qual você deseja obter um snapshot.
4. Em Actions (Ações), escolha Take snapshot (Fazer snapshot).

   A janela Take snapshot de banco de dados (Fazer snapshot de banco de dados) é exibida.
5. Para Snapshot name (Nome do snapshot), insira o nome do snapshot.
6. Selecione Take Snapshot (Fazer snapshot).

AWS CLI

Você cria um snapshot de uma instância de banco de dados do RDS Custom utilizando o comando create-db-snapshot da AWS CLI.

Especifique as seguintes opções:

- `--db-instance-identifier` – Identifica de qual instância de banco de dados do RDS Custom você fará backup
- `--db-snapshot-identifier` – Dê um nome para seu snapshot do RDS Custom para que ele possa ser restaurado mais tarde

Neste exemplo, você cria um snapshot de banco de dados chamado *`my-custom-snapshot`* para uma instância de banco de dados do RDS Custom chamada *`my-custom-instance`*.

Example

Para Linux, macOS ou Unix:

```
aws rds create-db-snapshot \
    --db-instance-identifier my-custom-instance \
    --db-snapshot-identifier my-custom-snapshot
```

Para Windows:

```
aws rds create-db-snapshot ^
    --db-instance-identifier my-custom-instance ^
    --db-snapshot-identifier my-custom-snapshot
```

Restaurar de um snapshot de banco de dados do RDS Custom for SQL Server

Ao restaurar uma instância de banco de dados do RDS Custom for SQL Server, você fornece o nome do snapshot de banco de dados e um nome para a nova instância. Não é possível restaurar de um snapshot para uma instância de banco de dados do RDS Custom existente. Uma nova instância de banco de dados do RDS Custom for SQL Server é criada quando você realiza a restauração.

A restauração a partir de um snapshot restaurará o volume de armazenamento até o momento em que o snapshot foi criado. Isso incluirá todos os bancos de dados e quaisquer outros arquivos presentes no volume (`D:`).

Console

Para restaurar uma instância de banco de dados do RDS Custom a partir de um snapshot de banco de dados

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Snapshots.
3. Escolha o DB snapshot do qual você deseja restaurar.
4. Em Actions (Ações), escolha Restore snapshot (Restaurar snapshot).
5. Na página Restore DB instance (Restaurar instância de banco de dados), para DB Instance Identifier (Identificador da instância de banco de dados), insira o nome da instância de banco de dados do RDS Custom restaurada.
6. Escolha Restore DB Instance.

AWS CLI

Você restaura um snapshot de banco de dados do RDS Custom utilizando o comando restore-db-instance-from-db-snapshot da AWS CLI.

Se o snapshot do qual você está restaurando for para uma instância de banco de dados privada, certifique-se de especificar `db-subnet-group-name` e `no-publicly-accessible`, ambos corretos. Caso contrário, a instância de banco de dados assumirá como padrão o estado de acesso público. São necessárias as seguintes opções:

- `db-snapshot-identifier` – Identifica o snapshot do qual restaurar
- `db-instance-identifier` – Especifica o nome da instância de banco de dados do RDS Custom a ser criada a partir do snapshot de banco de dados
- `custom-iam-instance-profile`: especifica o perfil da instância associado à instância subjacente do Amazon EC2 de uma instância de banco de dados do RDS Custom.

O código a seguir restaura o snapshot chamado `my-custom-snapshot` para `my-custom-instance`.

Example

Para Linux, macOS ou Unix:

```
aws rds restore-db-instance-from-db-snapshot \
  --db-snapshot-identifier my-custom-snapshot \
  --db-instance-identifier my-custom-instance \
  --custom-iam-instance-profile AWSRDSCustomInstanceProfileForRdsCustomInstance \
  --no-publicly-accessible
```

Para Windows:

```
aws rds restore-db-instance-from-db-snapshot ^
  --db-snapshot-identifier my-custom-snapshot ^
  --db-instance-identifier my-custom-instance ^
  --custom-iam-instance-profile AWSRDSCustomInstanceProfileForRdsCustomInstance ^
  --no-publicly-accessible
```

Restaurar uma instância do RDS Custom for SQL Server para um ponto anterior no tempo

É possível restaurar uma instância de banco de dados para um ponto anterior no tempo (PITR) criando uma nova instância de banco de dados. Para oferecer compatibilidade com a PITR, as instâncias de banco de dados devem ter a retenção de backup habilitada.

O tempo de restauração mais recente de uma instância de banco de dados do RDS Custom for SQL Server depende de vários fatores, mas em geral é de até cinco minutos do horário atual. Para visualizar o tempo restaurável mais recente para uma instância de banco de dado, use o comando AWS CLI describe-db-instances e confira o valor retornado no campo `LatestRestorableTime` para a instância de banco de dados. Para ver o tempo de restauração mais recente para cada instância de banco de dados no console Amazon RDS, selecione Backups automatizados.

É possível fazer a restauração para qualquer momento dentro do período de retenção de backup. Para ver o tempo de restauração mais antigo para cada instância de banco de dados, selecione Backups automatizados no console do Amazon RDS.

Para obter informações gerais sobre PITR, consulte Restaurar uma instância de banco de dados para um momento especificado no Amazon RDS.

Tópicos

- Considerações sobre a PITR para o RDS Custom for SQL Server
- Número de bancos de dados elegíveis para PITR por tipo de classe de instância

- Tornar bancos de dados não qualificados para PITR
- Logs de transações no Amazon S3
- Restauração de PITR usando o AWS Management Console, a AWS CLI ou a API do RDS.

Considerações sobre a PITR para o RDS Custom for SQL Server

No RDS Custom for SQL Server, a PITR difere das seguintes maneiras importantes da PITR no Amazon RDS:

- A PITR apenas restaura os bancos de dados na instância de banco de dados. Ele não restaura o sistema operacional ou arquivos na unidade C:.
- Para uma instância de banco de dados do RDS Custom for SQL Server, um banco de dados recebe backup automaticamente e está qualificado para PITR somente nestas condições:
  - O banco de dados está online.
  - Seu modelo de recuperação está definido como `FULL`.
  - É gravável.
  - Ele tem seus arquivos físicos na unidade D:.
  - Ele não está listado na tabela `rds_pitr_blocked_databases`. Para ter mais informações, consulte Tornar bancos de dados não qualificados para PITR.
- Os bancos de dados elegíveis para PITR são determinados por ordem de ID de banco de dados. O RDS Custom for SQL Server permite até 5.000 bancos de dados por instância de banco de dados. Contudo, o número máximo de bancos de dados restaurados por uma operação de PITR para uma instância de banco de dados do RDS Custom para SQL Server depende do tipo de classe de instância. Para ter mais informações, consulte Número de bancos de dados elegíveis para PITR por tipo de classe de instância.

  Outros bancos de dados que não fazem parte da PITR podem ser restaurados por meio de snapshots de banco de dados, incluindo os backups de snapshots automatizados utilizados para a PITR.
- Adicionar um novo banco de dados, renomear um banco de dados ou restaurar um banco de dados elegível para PITR inicia um snapshot da instância de banco de dados.
- O número máximo de bancos de dados elegíveis para PITR muda quando a instância do banco de dados passa por uma operação de computação em escala, dependendo do tipo de classe da instância de destino. Se você aumentar a escala da instância verticalmente, permitindo que mais bancos de dados na instância sejam elegíveis para PITR, um novo snapshot será criado.

- Os bancos de dados restaurados têm o mesmo nome que a instância de banco de dados de origem. Se desejar, especifique um nome diferente.
- `AWSRDSCustomSQLServerIamRolePolicy` requer acesso a outros serviços da AWS. Para ter mais informações, consulte Adicionar uma política de acesso a AWSRDSCustomSQLServerInstanceRole.
- Alterações de fuso horário não têm suporte para o RDS Custom for SQL Server. Se você alterar o fuso horário do sistema operacional ou da instância de banco de dados, a PITR (e outras automações) não funcionará.

Número de bancos de dados elegíveis para PITR por tipo de classe de instância

A tabela a seguir mostra o número máximo de bancos de dados elegíveis para PITR com base no tipo de classe de instância.

Tipo de classe de instância	Número máximo de bancos de dados elegíveis para PITR
db.*.large	100
db.*.xlarge to db.*.2xlarge	150
db.*.4xlarge to db.*.8xlarge	300
db.*.12xlarge to db.*.16xlarge	600
db.*.24xlarge, db.*32xlarge	1000

* Representa os diferentes tipos de classes da instância.

O número máximo de bancos de dados elegíveis para PITR em uma instância de banco de dados depende do tipo de classe da instância. O número varia de cem no menor a mil nos maiores tipos de classe de instância compatíveis com o RDS Custom para SQL Server. Os bancos de dados de sistemas do SQL Server (`master, model, msdb, tempdb`) não estão incluídos nesse limite. Quando você aumenta ou reduz a escala de uma instância de banco de dados verticalmente, dependendo do tipo de classe de instância de destino, o RDS Custom atualizará automaticamente o número de bancos de dados elegíveis para PITR. O RDS Custom para SQL Server enviará `RDS-EVENT-0352` quando o número máximo de bancos de dados elegíveis para PITR for alterado em uma instância de banco de dados. Para ter mais informações, consulte Eventos de versão de mecanismos personalizados.

> **Note**
>
> O suporte à PITR para mais de cem bancos de dados só está disponível em instâncias de banco de dados criadas após 26 de agosto de 2023. Para instâncias criadas antes de 26 de agosto de 2023, o número máximo de bancos de dados elegíveis para PITR é cem, independentemente da classe da instância. Para habilitar o suporte a PITR para mais de cem bancos de dados em instâncias de banco de dados criadas antes de 26 de agosto de 2023, é possível realizar a seguinte ação:
>
> - Atualizar a versão do mecanismo de banco de dados para 15.00.4322.2.v1 ou posterior

Durante uma operação de PITR, o RDS Custom vai restaurar todos os bancos de dados que faziam parte da PITR na instância de banco de dados de origem no momento da restauração. Depois que a instância de banco de dados de destino concluir as operações de restauração, se a retenção de backup estiver habilitada, a instância de banco de dados começará a fazer backup com base no número máximo de bancos de dados elegíveis para PITR na instância de banco de dados de destino.

Por exemplo, se a instância de banco de dados for executada em uma `db.*.xlarge` que tenha duzentos bancos de dados:

1. O RDS Custom para SQL Server escolherá os primeiros 150 bancos de dados, ordenados pelo ID de banco de dados, para backup da PITR.
2. Você modifica a instância para escalar até db.*.4xlarge.

3. Uma vez concluída a operação de computação em escala, o RDS Custom para SQL Server escolherá os primeiros trezentos bancos de dados, ordenados pelo ID de banco de dados, para backup da PITR. Cada um dos duzentos bancos de dados que atendem às condições exigidas pela PITR agora será elegível para PITR.
4. Agora, modifique a instância para reduzir a escala verticalmente para db.*.xlarge.
5. Uma vez concluída a operação de computação em escala, o RDS Custom para SQL Server selecionará novamente os primeiros 150 bancos de dados, ordenados pelo ID de banco de dados, para backup da PITR.

Tornar bancos de dados não qualificados para PITR

É possível optar por excluir bancos de dados individuais da PITR. Para fazer isso, coloque seus valores de `database_id` em uma tabela `rds_pitr_blocked_databases`. Utilize o seguinte script SQL para criar a tabela.

Para criar a tabela rds_pitr_blocked_databases

- Execute o seguinte script SQL.

```
create table msdb..rds_pitr_blocked_databases
(
database_id INT NOT NULL,
database_name SYSNAME NOT NULL,
db_entry_updated_date datetime NOT NULL DEFAULT GETDATE(),
db_entry_updated_by SYSNAME NOT NULL DEFAULT CURRENT_USER,
PRIMARY KEY (database_id)
);
```

Para conhecer a lista de bancos de dados qualificados e não qualificados, consulte o arquivo `RI.End` no diretório `RDSCustomForSQLServer/Instances/`*`DB_instance_resource_ID`*`/TransactionLogMetadata` do bucket do Amazon S3 `do-not-delete-rds-custom-`*`$ACCOUNT_ID`*`-`*`$REGION`*`-`*`unique_identifier`*. Para obter mais informações sobre o arquivo `RI.End`, consulte Logs de transações no Amazon S3.

Também é possível determinar a lista de bancos de dados elegíveis para PITR usando o script SQL a seguir. Defina a variável `@limit` como o número máximo de bancos de dados elegíveis para PITR para a classe de instância. Para ter mais informações, consulte Número de bancos de dados elegíveis para PITR por tipo de classe de instância.

Como determinar a lista de bancos de dados elegíveis para PITR em uma classe de instância de banco de dados

- Execute o seguinte script SQL.

```
DECLARE @Limit INT;
SET @Limit = (insert-database-instance-limit-here);

USE msdb;
IF (EXISTS (SELECT * FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_SCHEMA = 'dbo' AND
 TABLE_NAME = 'rds_pitr_blocked_databases'))
    WITH TABLE0 AS (
        SELECT hdrs.database_id as DatabaseId, sdb.name as DatabaseName,
 'ALWAYS_ON_NOT_WRITABLE_REPLICA' as Reason, NULL as DatabaseNameOnPitrTable
        FROM sys.dm_hadr_database_replica_states hdrs
        INNER JOIN sys.databases sdb ON sdb.database_id = hdrs.database_id
        WHERE (hdrs.is_local = 1 AND hdrs.is_primary_replica = 0)
        OR (sys.fn_hadr_is_primary_replica (sdb.name) = 1 AND DATABASEPROPERTYEX
 (sdb.name, 'Updateability') = 'READ_ONLY')
    ),
    TABLE1 as (
            SELECT dbs.database_id as DatabaseId, sysdbs.name as DatabaseName,
 'OPTOUT' as Reason,
            CASE WHEN dbs.database_name = sysdbs.name THEN NULL ELSE
 dbs.database_name END AS DatabaseNameOnPitrTable
            FROM msdb.dbo.rds_pitr_blocked_databases dbs
            INNER JOIN sys.databases sysdbs ON dbs.database_id = sysdbs.database_id
            WHERE sysdbs.database_id > 4
            ),
    TABLE2 as (
            SELECT
            db.name AS DatabaseName,
            db.create_date AS CreateDate,
            db.state_desc AS DatabaseState,
            db.database_id AS DatabaseId,
            rs.database_guid AS DatabaseGuid,
            rs.last_log_backup_lsn AS LastLogBackupLSN,
            rs.recovery_fork_guid RecoveryForkGuid,
            rs.first_recovery_fork_guid AS FirstRecoveryForkGuid,
            db.recovery_model_desc AS RecoveryModel,
            db.is_auto_close_on AS IsAutoClose,
            db.is_read_only as IsReadOnly,
            NEWID() as FileName,
```

```
            CASE WHEN(db.state_desc = 'ONLINE'
                    AND db.recovery_model_desc != 'SIMPLE'
                    AND((db.is_auto_close_on = 0 and db.collation_name IS NOT NULL)
 OR db.is_auto_close_on = 1))
                    AND db.is_read_only != 1
                    AND db.user_access = 0
                    AND db.source_database_id IS NULL
                    AND db.is_in_standby != 1
                    THEN 1 ELSE 0 END AS IsPartOfSnapshot,
            CASE WHEN db.source_database_id IS NULL THEN 0 ELSE 1 END AS
 IsDatabaseSnapshot
            FROM sys.databases db
            INNER JOIN sys.database_recovery_status rs
            ON db.database_id = rs.database_id
            WHERE DB_NAME(db.database_id) NOT IN('tempdb') AND
            db.database_id NOT IN (SELECT DISTINCT DatabaseId FROM TABLE1) AND
            db.database_id NOT IN (SELECT DISTINCT DatabaseId FROM TABLE0)
        ),
        TABLE3 as(
            Select @Limit+count(DatabaseName) as TotalNumberOfDatabases from TABLE2
 where TABLE2.IsPartOfSnapshot=1 and DatabaseName in ('master','model','msdb')
        )
        SELECT TOP(SELECT TotalNumberOfDatabases from TABLE3)
 DatabaseName,CreateDate,DatabaseState,DatabaseId from TABLE2 where
 TABLE2.IsPartOfSnapshot=1
        ORDER BY TABLE2.DatabaseID ASC
ELSE
    WITH TABLE0 AS (
        SELECT hdrs.database_id as DatabaseId, sdb.name as DatabaseName,
 'ALWAYS_ON_NOT_WRITABLE_REPLICA' as Reason, NULL as DatabaseNameOnPitrTable
        FROM sys.dm_hadr_database_replica_states hdrs
        INNER JOIN sys.databases sdb ON sdb.database_id = hdrs.database_id
        WHERE (hdrs.is_local = 1 AND hdrs.is_primary_replica = 0)
        OR (sys.fn_hadr_is_primary_replica (sdb.name) = 1 AND DATABASEPROPERTYEX
 (sdb.name, 'Updateability') = 'READ_ONLY')
    ),
    TABLE1 as (
            SELECT
            db.name AS DatabaseName,
            db.create_date AS CreateDate,
            db.state_desc AS DatabaseState,
            db.database_id AS DatabaseId,
            rs.database_guid AS DatabaseGuid,
            rs.last_log_backup_lsn AS LastLogBackupLSN,
```

```
            rs.recovery_fork_guid RecoveryForkGuid,
            rs.first_recovery_fork_guid AS FirstRecoveryForkGuid,
            db.recovery_model_desc AS RecoveryModel,
            db.is_auto_close_on AS IsAutoClose,
            db.is_read_only as IsReadOnly,
            NEWID() as FileName,
            CASE WHEN(db.state_desc = 'ONLINE'
                    AND db.recovery_model_desc != 'SIMPLE'
                    AND((db.is_auto_close_on = 0 and db.collation_name IS NOT NULL)
OR db.is_auto_close_on = 1))
                    AND db.is_read_only != 1
                    AND db.user_access = 0
                    AND db.source_database_id IS NULL
                    AND db.is_in_standby != 1
                    THEN 1 ELSE 0 END AS IsPartOfSnapshot,
            CASE WHEN db.source_database_id IS NULL THEN 0 ELSE 1 END AS
IsDatabaseSnapshot
            FROM sys.databases db
            INNER JOIN sys.database_recovery_status rs
            ON db.database_id = rs.database_id
            WHERE DB_NAME(db.database_id) NOT IN('tempdb') AND
            db.database_id NOT IN (SELECT DISTINCT DatabaseId FROM TABLE0)
        ),
        TABLE2 as(
            SELECT @Limit+count(DatabaseName) as TotalNumberOfDatabases from TABLE1
where TABLE1.IsPartOfSnapshot=1 and DatabaseName in ('master','model','msdb')
        )
        select top(select TotalNumberOfDatabases from TABLE2)
DatabaseName,CreateDate,DatabaseState,DatabaseId from TABLE1 where
TABLE1.IsPartOfSnapshot=1
        ORDER BY TABLE1.DatabaseID ASC
```

> **Note**
>
> Os bancos de dados que são apenas links simbólicos também são excluídos dos bancos de dados elegíveis para operações de PITR. A consulta acima não é filtrada com base nesses critérios.

Logs de transações no Amazon S3

O período de retenção de backup determina se os logs de transações para instâncias de banco de dados RDS Custom for SQL Server são automaticamente extraídos e carregados no Amazon S3. Um valor diferente de zero significa que backups automáticos são criados e que o agente do RDS Custom carrega os logs de transações no S3 a cada 5 minutos.

Os arquivos de log de transações no S3 são criptografados em repouso utilizando a AWS KMS key que você forneceu quando criou sua instância de banco de dados. Para obter mais informações, consulte Como proteger dados usando criptografia do lado do servidor no Guia do usuário do Amazon Simple Storage Service.

Os logs de transações de cada banco de dados são carregados em um bucket do S3 denominado `do-not-delete-rds-custom-`*`$ACCOUNT_ID`*`-`*`$REGION`*`-`*`unique_identifier`*. O diretório `RDSCustomForSQLServer/Instances/`*`DB_instance_resource_ID`* no bucket do S3 contém dois subdiretórios:

- `TransactionLogs` – contém os logs de transações para cada banco de dados e seus respectivos metadados.

  O nome do arquivo de log de transações segue o padrão *`yyyyMMddHHmm`*`.`*`database_id`*`.`*`timestamp`*, por exemplo:

  ```
  202110202230.11.1634769287
  ```

  O mesmo nome de arquivo com o sufixo `_metadata` contém informações sobre o log de transações, como números de sequência de log, nome do banco de dados e `RdsChunkCount`. `RdsChunkCount` determina quantos arquivos físicos representam um único arquivo de log de transações. Você pode ver arquivos com os sufixos `_0001`, `_0002` e assim por diante, o que significa os blocos físicos de um arquivo de log de transações. Se quiser utilizar um arquivo de log de transações em blocos, certifique-se de mesclar os blocos depois de baixá-los.

  Considere um cenário com os seguintes arquivos:

  - `202110202230.11.1634769287`
  - `202110202230.11.1634769287_0001`
  - `202110202230.11.1634769287_0002`
  - `202110202230.11.1634769287_metadata`

O `RdsChunkCount` é `3`. A ordem de mesclagem dos arquivos é a seguinte: `202110202230.11.1634769287`, `202110202230.11.1634769287_0001`, `202110202230.11.1634769287_0002`.

- `TransactionLogMetadata` – contém informações de metadados sobre cada iteração da extração do log de transações.

  O arquivo `RI.End` contém informações para todos os bancos de dados que tiveram seus logs de transações extraídos e para todos os bancos de dados existentes, mas que não tiveram seus logs de transações extraídos. O nome do arquivo `RI.End` segue o padrão *`yyyyMMddHHmm`*`.RI.End.`*`timestamp`*, por exemplo:

  ```
  202110202230.RI.End.1634769281
  ```

Restauração de PITR usando o AWS Management Console, a AWS CLI ou a API do RDS.

É possível restaurar uma instância de banco de dados do RDS Custom for SQL Server em um ponto anterior no tempo utilizando o AWS Management Console, a AWS CLI ou a API do RDS.

Console

Para restaurar uma instância de banco de dados do RDS Custom para um ponto anterior especificado

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Automated backups (Backups automatizados).
3. Escolha a instância de banco de dados do RDS Custom que você deseja restaurar.
4. Em Actions (Ações), escolha Restore to point in time (Restaurar para point-in-time).

   A janela Restore to point in time (Restaurar para point-in-time) é exibida.
5. Escolha Latest restorable time (Hora da última restauração) para restaurar no último horário possível ou escolha Custom (Personalizar) para escolher um horário.

   Se você escolher Custom (Personalizado), insira a data e a hora para a qual deseja restaurar a instância.

Os horários são mostrados no fuso horário local, que é indicado por um deslocamento do Tempo Universal Coordenado (UTC). Por exemplo, UTC-5 é a Hora Padrão do Leste dos EUA/Horário de Verão Central.

6. Para DB instance identifier (Identificador de instância de banco de dados), insira o nome da instância de banco de dados do RDS Custom restaurada de destino. O nome deve ser exclusivo.
7. Escolha outras opções conforme necessário, como a classe da instância de banco de dados.
8. Escolha Restore to point in time (Restaurar para point-in-time).

AWS CLI

Você restaura uma instância de banco de dados em um horário especificado utilizando o comando restore-db-instance-to-point-in-time da AWS CLI para criar uma nova instância de banco de dados do RDS Custom.

Utilize uma das seguintes opções para especificar o backup a ser restaurado:

- `--source-db-instance-identifier` *`mysourcedbinstance`*
- `--source-dbi-resource-id` *`dbinstanceresourceID`*
- `--source-db-instance-automated-backups-arn` *`backupARN`*

A opção `custom-iam-instance-profile` é obrigatória.

O exemplo a seguir restaura `my-custom-db-instance` para uma nova instância de banco de dados denominada `my-restored-custom-db-instance`, a partir do ponto anterior especificado.

Example

Para Linux, macOS ou Unix:

```
aws rds restore-db-instance-to-point-in-time \
    --source-db-instance-identifier my-custom-db-instance\
    --target-db-instance-identifier my-restored-custom-db-instance \
    --custom-iam-instance-profile AWSRDSCustomInstanceProfileForRdsCustomInstance \
    --restore-time 2022-10-14T23:45:00.000Z
```

Para Windows:

```
aws rds restore-db-instance-to-point-in-time ^
    --source-db-instance-identifier my-custom-db-instance ^
    --target-db-instance-identifier my-restored-custom-db-instance ^
    --custom-iam-instance-profile AWSRDSCustomInstanceProfileForRdsCustomInstance ^
    --restore-time 2022-10-14T23:45:00.000Z
```

Excluir um snapshot do RDS Custom for SQL Server

É possível excluir snapshots de banco de dados gerenciados pelo RDS Custom for SQL Server quando eles não são mais necessários. O procedimento de exclusão é o mesmo para instâncias de banco de dados Amazon RDS e RDS Custom.

Os snapshots do Amazon EBS para os volumes binário e raiz permanecem na sua conta por mais tempo, pois podem estar vinculados a algumas instâncias em execução na sua conta ou a outros snapshots do RDS Custom for SQL Server. Esses snapshots do EBS serão excluídos automaticamente quando não estiverem mais relacionados a nenhum recurso existente do RDS Custom for SQL Server (instâncias de banco de dados ou backups).

Console

Para excluir um snapshot de uma instância de banco de dados do RDS Custom

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Snapshots.
3. Escolha o snapshot de banco de dados que você deseja excluir.
4. Em Actions (Ações), selecione Delete Snapshot (Excluir snapshot).
5. Escolha Delete (Excluir) na página de confirmação.

AWS CLI

Para excluir um snapshot do RDS Custom, utilize o comando da AWS CLI delete-db-snapshot.

A seguinte opção é necessária:

- `--db-snapshot-identifier` # – o snapshot a ser excluído

O exemplo a seguir exclui o snapshot de banco de dados `my-custom-snapshot`.

Example

Para Linux, macOS ou Unix:

```
aws rds delete-db-snapshot \
  --db-snapshot-identifier my-custom-snapshot
```

Para Windows:

```
aws rds delete-db-snapshot ^
  --db-snapshot-identifier my-custom-snapshot
```

Excluir backups automatizados do RDS Custom for SQL Server

Você pode excluir backups automatizados retidos para o RDS Custom for SQL Server quando eles não são mais necessários. O procedimento é idêntico ao de exclusão de backups do Amazon RDS.

Console

Como excluir um backup automatizado retido

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Automated backups (Backups automatizados).
3. Escolha Retained (Retido).
4. Escolha o backup automatizado retido que você deseja excluir.
5. Em Actions, selecione Delete.
6. Na página de confirmação, insira **`delete me`** e escolha Delete (Excluir).

AWS CLI

Você pode excluir um backup automatizado retido usando o comando da AWS CLI delete-db-instance-automated-backup.

A seguinte opção é usada para excluir um backup automatizado retido:

- `--dbi-resource-id` – o identificador de recurso da instância de banco de dados do RDS Custom de origem.

Você pode encontrar o identificador de recurso da instância de banco de dados de origem de um backup automatizado retido usando o comando da AWS CLI describe-db-instance-automated-backups.

O exemplo a seguir exclui o backup automatizado retido com o identificador de recurso da instância de banco de dados de origem `custom-db-123ABCEXAMPLE`.

Example

Para Linux, macOS ou Unix:

```
aws rds delete-db-instance-automated-backup \
    --dbi-resource-id custom-db-123ABCEXAMPLE
```

Para Windows:

```
aws rds delete-db-instance-automated-backup ^
    --dbi-resource-id custom-db-123ABCEXAMPLE
```

Copiar um snapshot de banco de dados do Amazon RDS Custom para SQL Server

Com o RDS Custom para SQL Server, é possível copiar snapshots de banco de dados manuais ou backups automatizados. Depois de copiar um snapshot, a cópia é um snapshot manual. É possível fazer várias cópias de um backup automatizado ou um snapshot manual, mas cada cópia deve ter um identificador exclusivo.

Só é possível copiar um snapshot na mesma conta AWS em diferentes Regiões da AWS em que o RDS Custom para SQL Server está disponível. No momento, os seguintes operadores não são aceitos:

- Copiar snapshots de banco de dados na mesma Região da AWS.
- Copiar snapshots de banco de dados entre contas AWS.

O RDS Custom para SQL Server é compatível com cópia de snapshot incremental. Para obter mais informações, consulte Considerações sobre a cópia de snapshots incrementais.

Tópicos

- Limitações
- Lidar com a criptografia
- Cópia entre regiões
- Snapshots de instâncias de banco de dados criadas com versões personalizadas do mecanismo (CEV)
- Conceder as permissões necessárias à sua entidade principal do IAM
- Copiar um snapshot de banco de dados

Limitações

As seguintes limitações se aplicam à cópia de um snapshot de banco de dados para o RDS Custom para SQL Server:

- Se você excluir um snapshot de origem antes que o snapshot de destino fique disponível, a cópia do snapshot poderá falhar. Verifique se o snapshot de destino tem o status `AVAILABLE` antes de excluir um snapshot de origem.

- Não é possível especificar um nome de grupo de opções nem copiar um grupo de opções em sua solicitação de cópia do snapshot de banco de dados.
- Se você excluir qualquer recurso dependente da AWS do DB snapshot de banco de dados de origem antes ou durante o processo de cópia, sua solicitação de cópia do snapshot poderá falhar de forma assíncrona.
  - Se você excluir o arquivo de backup Service Master Key (SMK) da sua instância de banco de dados de origem armazenado no bucket do S3 gerenciado pelo RDS Custom em sua conta, a cópia do snapshot de banco de dados será bem-sucedida de forma assíncrona. No entanto, os recursos do SQL Server dependentes do SMK, como bancos de dados habilitados para TDE, apresentam problemas. Para obter mais informações, consulte Solucionar problemas do estado PENDING_RECOVERY para bancos de dados habilitados para TDE no RDS Custom para SQL Server.
- Atualmente, não é possível copiar snapshots de banco de dados na mesma Região da AWS.
- Atualmente, não é possível copiar snapshots de banco de dados entre contas AWS.

As limitações à cópia de um snapshot de banco de dados para Amazon RDS também se aplicam ao RDS Custom para SQL Server. Para obter mais informações, consulte Limitações.

Lidar com a criptografia

Todas as instâncias e snapshots de banco de dados do RDS Custom para SQL Server são criptografados com chaves do KMS. Só é possível copiar um snapshot criptografado em um snapshot criptografado, portanto, é necessário especificar uma chave do KMS válida na Região da AWS de destino para a solicitação de cópia do snapshot de banco de dados.

O snapshot de origem permanece criptografado ao longo do processo de cópia. O Amazon RDS usa criptografia envelopada para proteger os dados durante a operação de cópia com a chave do KMS da Região da AWS de destino especificada. Para obter mais informações, consulte Criptografia envelopada no Guia do desenvolvedor do AWS Key Management Service.

Cópia entre regiões

É possível copiar snapshots de banco de dados entre Regiões da AWS. No entanto, existem certas restrições e considerações para a cópia de snapshot entre regiões.

Autorizar o RDS a se comunicar entre Regiões da AWS para cópia de snapshots

Depois que uma solicitação de cópia de snapshot de banco de dados entre regiões é processada com êxito, o RDS inicia a cópia. Uma solicitação de autorização para o RDS acessar o snapshot de origem é criada. Essa solicitação de autorização vincula o snapshot de banco de dados de origem ao snapshot de banco de dados de destino. Isso permite que o RDS copie apenas no snapshot de destino especificado.

O RDS verifica a autorização usando a permissão `rds:CrossRegionCommunication` no perfil do IAM vinculado ao serviço. Se a cópia for autorizada, o RDS poderá se comunicar com a região de origem e concluirá a operação de cópia.

O RDS não tem acesso a snapshots de banco de dados que não foram autorizados anteriormente por uma solicitação de CopyDBSnapshot. A autorização é revogada quando a cópia é concluída.

O RDS usa a função vinculada ao serviço para verificar a autorização na região de origem. A cópia falhará se você excluir o perfil vinculado ao serviço durante o processo de cópia.

Para ter mais informações, consulte Usar perfis vinculados ao serviço no Guia do usuário do AWS Identity and Access Management.

Uso de credenciais do AWS Security Token Service

Os tokens de sessão do endpoint global do AWS Security Token Service (AWS STS) são válidos apenas nas Regiões da AWS habilitadas por padrão (regiões comerciais). Se você usar credenciais da operação da API `assumeRole` no AWS STS, use o endpoint regional se a região de origem for uma região de aceitação. Caso contrário, haverá falha na solicitação. Suas credenciais devem ser válidas nas duas regiões, o que é verdadeiro para regiões opcionais somente quando o endpoint regional do AWS STS é usado.

Para usar o endpoint global, certifique-se de que ele esteja ativado para ambas as regiões nas operações. Defina o endpoint global como `Valid` em todas as Regiões da AWS nas configurações da conta AWS STS.

Para ter mais informações, consulte Gerenciar o AWS STS em uma Região da AWS no Guia do usuário do AWS Identity and Access Management.

Snapshots de instâncias de banco de dados criadas com versões personalizadas do mecanismo (CEV)

Para um snapshot de banco de dados de uma instância de banco de dados que use uma versão personalizada do mecanismo (CEV), o RDS associa a CEV ao snapshot de banco de dados. Para copiar um snapshot de banco de dados de origem associado a uma CEV entre Regiões da AWS, o RDS copia a CEV junto com o snapshot de banco de dados de origem na região de destino.

Se você estiver copiando vários snapshots de banco de dados associados à mesma CEV na mesma região de destino, a primeira solicitação de cópia copia a CEV associada. O processo de cópia das solicitações a seguir localiza a CEV inicialmente copiada e a associa às cópias do snapshot de banco de dados a seguir. A cópia de CEV existente deve estar no estado `AVAILABLE` para ser associada às cópias do snapshot de banco de dados.

Para copiar um snapshot de banco de dados associado a uma CEV, a política do IAM do solicitante deve ter as permissões para autorizar tanto a cópia do snapshot de banco de dados quanto a cópia da CEV associada. As seguintes permissões são necessárias na política do IAM do solicitante para permitir a cópia associada da CEV:

- `rds:CopyCustomDBEngineVersion`: a entidade principal do IAM do solicitante precisa ter a permissão para copiar a CEV de origem na região de destino junto com o snapshot de banco de dados de origem. A solicitação de cópia do snapshot falhará devido a erros de autorização se a entidade principal do IAM do solicitante não estiver autorizada a copiar a CEV de origem.
- `ec2:CreateTags`: a AMI do EC2 subjacente da CEV de origem é copiada na região de destino como parte da cópia da CEV. O RDS Custom tenta marcar a AMI com a tag `AWSRDSCustom` antes de copiar a AMI. A entidade principal do IAM do solicitante deve ter permissão para criar a tag na AMI subjacente à CEV de origem na região de origem.

Para ter mais informações sobre permissões de cópia da CEV, consulte Conceder as permissões necessárias à sua entidade principal do IAM.

Conceder as permissões necessárias à sua entidade principal do IAM

Verifique se você tem acesso suficiente para copiar um snapshot de banco de dados do RDS Custom para SQL Server. O perfil ou o usuário do IAM (conhecido como a entidade principal do IAM) para copiar um snapshot de banco de dados usando o console ou a CLI deve ter uma das seguintes políticas para a criação bem-sucedida de uma instância de banco de dados:

- A política `AdministratorAccess` ou,
- A política do `AmazonRDSFullAccess` mostra as seguintes permissões adicionais:

```
s3:CreateBucket
s3:GetBucketPolicy
s3:PutBucketPolicy
kms:CreateGrant
kms:DescribeKey
ec2:CreateTags
```

O RDS Custom usa essas permissões durante a cópia do snapshot entre Regiões da AWS. Essas permissões configuram recursos em sua conta que são necessários para operações do RDS Custom. Para ter mais informações sobre a permissão `kms:CreateGrant`, consulte Gerenciamento de AWS KMS key.

O exemplo de política JSON a seguir concede as permissões necessárias além da política `AmazonRDSFullAccess`.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "CreateS3BucketAndReadWriteBucketPolicy",
            "Effect": "Allow",
            "Action": [
                "s3:CreateBucket",
                "s3:PutBucketPolicy",
                "s3:GetBucketPolicy"
            ],
            "Resource": "arn:aws:s3:::do-not-delete-rds-custom-*"
        },
        {
            "Sid": "CreateKmsGrant",
            "Effect": "Allow",
            "Action": [
                "kms:CreateGrant",
                "kms:DescribeKey"
            ],
```

```
            "Resource": "*"
        },
        {
            "Sid": "CreateEc2Tags",
            "Effect": "Allow",
            "Action": [
                "ec2:CreateTags"
            ],
            "Resource": "*"
        }
    ]
}
```

> **Note**
>
> As permissões listadas não devem ser restritas por políticas de controle de serviço (SCPs), limites de permissão nem políticas de sessão associadas à entidade principal do IAM.

Se você usar condições com chaves de contexto na política do IAM do solicitante, determinadas condições poderão fazer com que a solicitação falhe. Para ter mais informações sobre dificuldades comuns causadas pelas condições da política do IAM, consulte Solicitações de cópia de snapshot de banco de dados entre regiões.

Copiar um snapshot de banco de dados

Use o procedimento a seguir para copiar um snapshot de banco de dados. Para cada conta da AWS, é possível copiar até 20 snapshots de banco de dados ao mesmo tempo de uma Região da AWS em outra. Se copiar um snapshot de banco de dados para outra Região da AWS, você criará um snapshot de banco de dados manual que será retido nessa Região da AWS. Copiar um snapshot de banco de dados fora da Região da AWS de origem resultará em cobranças de transferência de dados do Amazon RDS. Para obter mais informações sobre a definição de preço da transferência de dados, consulte Definição de preço do Amazon RDS.

Depois que a cópia do snapshot de banco de dados tiver sido criada na nova Região da AWS, ela se comportará da mesma forma como todos os outros snapshots de banco de dados nessa Região da AWS.

É possível copiar um snapshot de banco de dados usando o AWS Management Console, a AWS CLI ou a API do Amazon RDS.

Console

O procedimento a seguir copia um snapshot de banco de dados do RDS Custom para SQL Server usando o AWS Management Console.

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Snapshots.
3. Selecione o snapshot de banco de dados do RDS Custom para SQL Server que você deseja copiar.
4. No menu suspenso Ações, escolha Copiar snapshot.



RDS > Snapshots > Copy snapshot



Copy snapshot

Settings

Source DB snapshot
DB snapshot identifier for the snapshot being copied.
db1-snapshot

New DB snapshot identifier
DB snapshot identifier for the new snapshot.
copied-db1-snapshot
Must start with a letter and only contain letters, digits, or hyphens.

Destination Region Info
Choose a Region

Resource tags Info
☐ Copy tags from the source DB snapshot

Depending on the amount of data to be copied and the Region you choose, this operation can take several hours to complete. The display on the progress bar might be delayed until setup is complete.

Encryption

Encryption Info
☑ Enable Encryption
Choose to encrypt the copy of the source DB snapshot. Master key IDs and aliases appear in the list after they have been created using KMS. You cannot remove encryption from an encrypted DB snapshot.

AWS KMS key Info
Enter a key ARN

Amazon Resource Name (ARN)
[redacted]
Example: arn:aws:kms:<region>:<accountID>:key/<key-id>

Account
[redacted]

KMS key ID
[redacted]

5. Para copiar o snapshot de banco de dados em uma Região da AWS diferente, defina Região de destino como o valor exigido.

   > **Note**
   >
   > A Região da AWS de destino deve ter a mesma versão de mecanismo de banco de dados disponível que a Região da AWS de origem.

6. Em Novo identificador de snapshot de banco de dados, digite um nome exclusivo para o snapshot de banco de dados. É possível fazer várias cópias de um backup automatizado ou snapshot manual, mas cada cópia deve ter um identificador exclusivo.
7. (Opcional) Selecione Copy Tags (Copiar tags) para copiar tags e valores do snapshot para a cópia do snapshot.
8. Em Criptografia, especifique o identificador da chave do KMS a ser usado para criptografar a cópia do snapshot de banco de dados.

   > **Note**
   >
   > O RDS Custom para SQL Server criptografa todos os snapshots de banco de dados. Não é possível criar um snapshot de banco de dados não criptografado.

9. Escolha Copy snapshot (Copiar snapshot).

O RDS Custom para SQL Server cria uma cópia do snapshot de banco de dados da instância de banco de dados na Região da AWS selecionada.

AWS CLI

É possível copiar um snapshot de banco de dados do RDS Custom para SQL Server usando o comando copy-db-snapshot da AWS CLI. Se você estiver copiando o snapshot para uma nova Região da AWS, execute o comando na nova Região da AWS. As seguintes opções são usadas para copiar um snapshot de banco de dados. Nem todas as opções são necessárias para todos os cenários.

- `--source-db-snapshot-identifier`: o identificador do snapshot de banco de dados de origem.

  - Se o snapshot de origem estiver em uma Região da AWS diferente da cópia, especifique um ARN válido do snapshot de banco de dados. Por exemplo, .`arn:aws:rds:us-west-2:123456789012:snapshot:instance1-snapshot-12345678`
- `--target-db-snapshot-identifier`: o identificador da nova cópia do snapshot de banco de dados.
- `--kms-key-id`: o identificador da chave do KMS para um snapshot do banco de dados criptografado. O identificador da chave do KMS é o nome do recurso da Amazon (ARN), o identificador de chave ou o alias da chave da chave do KMS.
  - Ao copiar um snapshot criptografado para outra Região da AWS, você deve especificar uma chave do KMS para a Região da AWS de destino. As chaves do KMS são específicas da Região da AWS em que são criadas. Não é possível usar chaves de criptografia de uma Região da AWS em outra Região da AWS, a menos que uma chave de várias regiões seja usada. Para obter mais informações sobre chaves do KMS de várias regiões, consulte Usar chaves de várias regiões no AWS KMS.
- `--copy-tags`: Inclua as tags e os valores do snapshot de origem na cópia do snapshot.

As seguintes opções não aceitam a cópia de um snapshot de banco de dados do RDS Custom para SQL Server:

- `--copy-option-group`
- `--option-group-name`
- `--pre-signed-url`
- `--target-custom-availability-zone`

O exemplo de código a seguir copia um snapshot de banco de dados criptografado da região Oeste dos EUA (Oregon) para a região Leste dos EUA (N. da Virgínia). Execute o comando na região de destino (us-east-1).

Para Linux, macOS ou Unix:

```
aws rds copy-db-snapshot \
     --region us-east-1 \
    --source-db-snapshot-identifier arn:aws:rds:us-
west-2:123456789012:snapshot:instance1-snapshot-12345678 \
    --target-db-snapshot-identifier mydbsnapshotcopy \
    --kms-key-id a1b2c3d4-1234-5678-wxyz-a1b2c3d4d5e6
```

Para Windows:

```
aws rds copy-db-snapshot ^
     --region us-east-1 ^
    --source-db-snapshot-identifier arn:aws:rds:us-
west-2:123456789012:snapshot:instance1-snapshot-12345678 ^
    --target-db-snapshot-identifier mydbsnapshotcopy ^
    --kms-key-id a1b2c3d4-1234-5678-wxyz-a1b2c3d4d5e6
```

RDS API

É possível copiar um snapshot de banco de dados do RDS Custom para SQL Server usando a operação CopyDBSnapshot da API do Amazon RDS. Se estiver copiando o snapshot para uma nova Região da AWS, execute a ação na nova Região da AWS. As seguintes opções são usadas para copiar um snapshot de banco de dados. Nem todos os parâmetros são obrigatórios:

- `SourceDBSnapshotIdentifier`: o identificador do snapshot de banco de dados de origem.
  - Se o snapshot de origem estiver em uma Região da AWS diferente da cópia, especifique um ARN válido do snapshot de banco de dados. Por exemplo, .`arn:aws:rds:us-west-2:123456789012:snapshot:instance1-snapshot-12345678`
- `TargetDBSnapshotIdentifier`: o identificador da nova cópia do snapshot de banco de dados.
- `KmsKeyId`: o identificador da chave do KMS de um snapshot do banco de dados criptografado. O identificador da chave do KMS é o nome do recurso da Amazon (ARN), o identificador de chave ou o alias da chave da chave do KMS.
  - Ao copiar um snapshot criptografado para outra Região da AWS, você deve especificar uma chave do KMS para a Região da AWS de destino. As chaves do KMS são específicas da Região da AWS em que são criadas. Não é possível usar chaves de criptografia de uma Região da AWS em outra Região da AWS, a menos que uma chave de várias regiões seja usada. Para obter mais informações sobre chaves do KMS de várias regiões, consulte Usar chaves de várias regiões no AWS KMS.
- `CopyTags`: defina esse parâmetro como `true` para copiar tags e valores do snapshot de origem na cópia do snapshot. O padrão é `false`.

As seguintes opções não aceitam a cópia de um snapshot de banco de dados do RDS Custom para SQL Server:

- `CopyOptionGroup`
- `OptionGroupName`
- `PreSignedUrl`
- `TargetCustomAvailabilityZone`

O código a seguir cria uma cópia de um snapshot, com o novo nome `mydbsnapshotcopy`, na região Leste dos EUA (N. da Virgínia).

```
https://rds.us-east-1.amazonaws.com/
    ?Action=CopyDBSnapshot
    &KmsKeyId=a1b2c3d4-1234-5678-wxyz-a1b2c3d4d5e6
    &SourceDBSnapshotIdentifier=arn%3Aaws%3Ards%3Aus-
west-2%3A123456789012%3Asnapshot%3Ainstance1-snapshot-12345678
    &TargetDBSnapshotIdentifier=mydbsnapshotcopy
    &Version=2014-10-31
    &X-Amz-Algorithm=AWS4-HMAC-SHA256
    &X-Amz-Credential=AKIADQKE4SARGYLE/20161117/us-east-1/rds/aws4_request
    &X-Amz-Date=20161117T221704Z
    &X-Amz-SignedHeaders=content-type;host;user-agent;x-amz-content-sha256;x-amz-
date
    &X-Amz-
Signature=da4f2da66739d2e722c85fcfd225dc27bba7e2b8dbea8d8612434378e52adccf
```

Migrar um banco de dados on-premises para o Amazon RDS Custom for SQL Server

Você pode utilizar o seguinte processo para migrar um banco de dados Microsoft SQL Server on-premises para o Amazon RDS Custom for SQL Server utilizando backup e restauração nativos:

1. Faça backup completo do banco de dados na instância de banco de dados on-premises.
2. Carregue o arquivo de backup no Amazon S3.
3. Baixe o arquivo de backup do S3 para a instância de banco de dados do RDS Custom for SQL Server.
4. Restaure um banco de dados utilizando o arquivo de backup baixado na instância de banco de dados do RDS Custom for SQL Server.

Esse processo explica a migração de um banco de dados on-premises para o RDS Custom for SQL Server, utilizando backup completo e restauração nativos. Para reduzir o tempo de substituição durante o processo de migração, considere também o uso de backups diferenciais ou de log.

Para obter informações gerais sobre backup e restauração nativos do RDS para SQL Server, consulte Importar e exportar bancos de dados do SQL Server usando backup e restauração nativos.

Tópicos

- Pré-requisitos
- Fazer backup do banco de dados on-premises
- Carregar o arquivo de backup no Amazon S3
- Baixar o arquivo de backup do Amazon S3
- Restaurar o arquivo de backup para a instância de banco de dados do RDS Custom for SQL Server

Pré-requisitos

Realize as seguintes tarefas antes de migrar o banco de dados:

1. Configure o Remote Desktop Connection (RDP) para a sua instância de banco de dados do RDS Custom for SQL Server. Para obter mais informações, consulte Conectar-se à sua instância de banco de dados do RDS Custom via RDP.

2. Configure o acesso ao Amazon S3 para que você possa carregar e baixar o arquivo de backup do banco de dados.

Fazer backup do banco de dados on-premises

Você utiliza o backup nativo do SQL Server para fazer backup completo do banco de dados na instância de banco de dados on-premises.

O exemplo a seguir mostra o backup de um banco de dados chamado `mydatabase`, com a opção `COMPRESSION` especificada para reduzir o tamanho do arquivo de backup.

Para fazer backup do banco de dados on-premises

1. Usando o SQL Server Management Studio (SSMS), conecte-se à instância do SQL Server on-premises.
2. Execute o seguinte comando T-SQL.

```
backup database mydatabase to
disk ='C:\Program Files\Microsoft SQL Server\MSSQL13.MSSQLSERVER\MSSQL\Backup\mydb-full-compressed.bak'
with compression;
```

Carregar o arquivo de backup no Amazon S3

Você usa o AWS Management Console para carregar o arquivo de backup `mydb-full-compressed.bak` no Amazon S3.

Para carregar o arquivo de backup no S3

1. Faça login no AWS Management Console e abra o console do Amazon S3 em https://console.aws.amazon.com/s3/.
2. Na lista Buckets, escolha o nome do bucket no qual você deseja carregar seu arquivo de backup.
3. Escolha Upload (Fazer upload).
4. Na janela Upload (Fazer upload), siga um destes procedimentos:
   - Arraste e solte `mydb-full-compressed.bak` na janela Upload (Carregar).

- Selecione Add file (Adicionar arquivo), escolha `mydb-full-compressed.bak` e depois Open (Abrir).

O Amazon S3 carrega o arquivo de backup como um objeto do S3. Quando o carregamento for concluído, você verá uma mensagem de sucesso na página Upload: status.

Baixar o arquivo de backup do Amazon S3

Você usa o console para baixar o arquivo de backup do S3 para a instância de banco de dados do RDS Custom for SQL Server.

Para baixar o arquivo de backup do S3

1. Usando o RDP, conecte-se à sua instância de banco de dados do RDS Custom for SQL Server.
2. Faça login no AWS Management Console e abra o console do Amazon S3 em https://console.aws.amazon.com/s3/.
3. Na lista Buckets, escolha o nome do bucket que contém seu arquivo de backup.
4. Escolha o arquivo de backup `mydb-full-compressed.bak`.
5. Em Actions (Ações), selecione Download as (Baixar como).
6. Abra o menu de contexto (clique com o botão direito do mouse) para o link fornecido e escolha Save As (Salvar como).
7. Salve `mydb-full-compressed.bak` no diretório `D:\rdsdbdata\BACKUP`.

Restaurar o arquivo de backup para a instância de banco de dados do RDS Custom for SQL Server

Utilize a restauração nativa do SQL Server para restaurar o arquivo de backup na sua instância de banco de dados do RDS Custom for SQL Server.

Neste exemplo, a opção `MOVE` é especificada porque os diretórios de dados e arquivos de log são diferentes da instância de banco de dados on-premises.

Para restaurar o arquivo de backup

1. Usando o SSMS, conecte-se à sua instância de banco de dados do RDS Custom for SQL Server.

2. Execute o seguinte comando T-SQL.

```
restore database mydatabase from disk='D:\rdsdbdata\BACKUP\mydb-full-
compressed.bak'
with move 'mydatabase' to 'D:\rdsdbdata\DATA\mydatabase.mdf',
move 'mydatabase_log' to 'D:\rdsdbdata\DATA\mydatabase_log.ldf';
```

Fazer atualização de uma instância de banco de dados do Amazon RDS Custom para SQL Server

Você pode atualizar uma instância de banco de dados do Amazon RDS Custom para SQL Server modificando-a para utilizar uma nova versão do mecanismo de banco de dados. Para obter informações gerais sobre como atualizar instâncias de banco de dados, consulte Atualizar a versão de mecanismo de uma instância de banco de dados.

Tópicos

- Visão geral das atualizações no RDS Custom para SQL Server
- Atualizar a versão principal e secundária do mecanismo
- Nível de compatibilidade do banco de dados

Visão geral das atualizações no RDS Custom para SQL Server

O Amazon RDS Custom para SQL Server aceita atualizações de versões principais e secundárias. As atualizações de versão secundária podem incluir patches de segurança, correções de bugs e melhorias no mecanismo. A Microsoft lança essas atualizações como atualizações cumulativas (CUs). As atualizações de versão principal introduzem novos recursos e mudanças de mecanismo entre as versões, como a atualização do SQL Server 2019 para 2022. Você pode aplicar ambas as atualizações imediatamente ou durante as janelas de manutenção programada. Para evitar possíveis problemas de compatibilidade com versões anteriores, recomendamos que antes da atualização você teste suas aplicações em um ambiente de não produção.

O RDS Custom para SQL Server permite que você atualize uma versão de mecanismo fornecida pelo RDS (RPEV) ou uma versão de mecanismo personalizada (CEV).

- As versões do mecanismo fornecidas pelo RDS (RPEVs) contêm patches atualizados do sistema operacional (SO) e atualizações cumulativas (CUs) do SQL Server.
- Para uma versão de mecanismo personalizada (CEV), você deve seguir um processo de duas etapas. Primeiro, crie uma CEV com sua versão de destino do SQL Server. Consulte Preparar-se para criar uma CEV para RDS Custom para SQL Server. Essa versão de destino deve ser igual ou mais recente que sua versão atual. Depois que a nova CEV for criada, modifique a instância de banco de dados para usar essa nova versão. Para ter mais informações, consulte Performing a minor version upgrade for Amazon RDS Custom for SQL Server CEV with Multi-AZ.

Não aplique atualizações cumulativas do SQL Server no local à sua instância personalizada do RDS em execução. Depois de criar uma CEV com uma versão específica do SQL Server (por exemplo, SQL Server 2022 CU16), a aplicação de uma atualização cumulativa mais recente diretamente na instância a remove do perímetro de suporte e rela o erro `SP-S3006`. Para corrigir uma instância existente do SQL Server usando uma CEV, crie outra CEV que inclua a atualização cumulativa desejada e modifique a instância existente para mudar para a nova CEV.

Se você atualizar uma instância de banco de dados do RDS Custom para SQL Server em uma implantação multi-AZ, o RDS Custom para SQL Server executará atualizações progressivas para a instância. Essa abordagem minimiza o tempo de inatividade ao atualizar uma instância por vez. O RDS executa as seguintes ações para realizar atualizações progressivas:

1. Atualiza a instância de banco de dados em espera.
2. Faz failover para a instância de banco de dados em espera atualizada, tornando-a a nova instância de banco de dados primária.
3. Atualiza a nova instância de banco de dados em espera.

O tempo de inatividade da instância de banco de dados para implantações multi-AZ é o tempo necessário para o failover.

As seguintes limitações aplicam-se à atualização de uma instância de banco de dados do RDS Custom para SQL Server:

- Não há suporte para a opção de banco de dados personalizado e grupos de parâmetros.
- Quaisquer volumes de armazenamento adicionais que você anexar à instância de banco de dados do RDS Custom para SQL Server não serão anexados após a atualização.
- Para CEVs, a aplicação de atualizações cumulativas do SQL Server no local não é permitida e resulta na remoção da instância do perímetro de suporte.

Atualizar a versão principal e secundária do mecanismo

As atualizações de versões principais e secundárias do mecanismo são irreversíveis e sempre devem ser feitas para uma versão mais recente. Para identificar as versões de destino disponíveis, use o AWS Management Console e escolha entre as versões disponíveis ao modificar a instância de

banco de dados. Também é possível usar o comando `describe-db-engine-versions` da CLI ou o comando DescribeDBEngineVersions da API do RDS.

Para Linux, macOS ou Unix:

```
aws rds describe-db-engine-versions \
    --engine custom-sqlserver-se \
    --engine-version 15.00.4322.2.v1 \
    --query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" \
    --output table
```

Para Windows:

```
aws rds describe-db-engine-versions ^
    --engine custom-sqlserver-se ^
    --engine-version 15.00.4322.2.v1 ^
    --query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" ^
    --output table
```

A saída mostra as versões disponíveis do mecanismo de destino:

```
--------------------------
|DescribeDBEngineVersions|
+------------------------+
|      EngineVersion     |
+------------------------+
|  15.00.4410.1.v1       |
|  15.00.4415.2.v1       |
|  15.00.4430.1.v1       |
|  16.00.4165.4.v1       |
|  16.00.4175.1.v1       |
|  16.00.4185.3.v1       |
+------------------------+
```

Depois de identificar sua versão de destino, use o AWS Management Console e siga as instruções em Modificar uma instância de banco de dados do RDS Custom for SQL Server. Também é possível usar o comando `modify-db-instance` da CLI ou o comando ModifyDBInstance da API do RDS.

Para Linux, macOS ou Unix:

```
aws rds modify-db-instance \
```

```
    --db-instance-identifier DB_INSTANCE_IDENTIFIER \
    --engine-version ENGINE_VERSION \
    --allow-major-version-upgrade \
    --region Region \
    --no-apply-immediately
```

Para Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier DB_INSTANCE_IDENTIFIER ^
    --engine-version ENGINE_VERSION ^
    --allow-major-version-upgrade ^
    --region Region ^
    --no-apply-immediately
```

> **Note**
>
> Você deve incluir o parâmetro `--allow-major-version-upgrade` para realizar atualizações de versão principal.

Nível de compatibilidade do banco de dados

Você pode usar os níveis de compatibilidade do banco de dados Microsoft SQL Server para ajustar alguns comportamentos de banco de dados de forma a imitar as versões anteriores do SQL Server. Para ter mais informações, consulte Nível de compatibilidade na documentação da Microsoft.

Quando você atualiza sua instância de banco de dados, todos os bancos de dados existentes permanecem em seus níveis de compatibilidade originais. Por exemplo, se você fizer a atualização do SQL Server 2019 para o SQL Server 2022, todos os bancos de dados existentes terão um nível de compatibilidade de 150. Qualquer novo banco de dados criado após a atualização terá um nível de compatibilidade de 160.

Você pode alterar o nível de compatibilidade de um banco de dados usando o comando ALTER DATABASE. Por exemplo, para alterar um banco de dados denominado `customeracct` para que ele seja compatível com o SQL Server 2022, emita o seguinte comando:

```
ALTER DATABASE customeracct SET COMPATIBILITY_LEVEL = 160
```

Solucionar problemas de banco de dados do Amazon RDS Custom para SQL Server

O modelo de responsabilidade compartilhada do RDS Custom fornece acesso ao nível de shell do SO e acesso pelo administrador do banco de dados. O RDS Custom executa recursos na sua conta, ao contrário do Amazon RDS, que executa recursos em uma conta do sistema. A extensão do acesso aumenta a responsabilidade. Nas seções a seguir, você vai aprender a solucionar problemas em instâncias de banco de dados do Amazon RDS Custom para SQL Server.

> **Note**
>
> Esta seção explica como solucionar problemas do RDS Custom para SQL Server. Para solucionar problemas do RDS Custom para Oracle, consulte Solucionar problemas de banco de dados do Amazon RDS Custom para Oracle.

Tópicos

- Visualizar eventos personalizados do RDS Custom
- Assinar eventos do RDS Custom
- Solucionar erros de CEV para o RDS Custom para SQL Server
- Corrigir configurações não compatíveis no RDS Custom para SQL Server
- Solucionar problemas de Storage-Full no RDS Custom para SQL Server
- Solucionar problemas do estado PENDING_RECOVERY para bancos de dados habilitados para TDE no RDS Custom para SQL Server

Visualizar eventos personalizados do RDS Custom

O procedimento para visualizar eventos é o mesmo para instâncias de banco de dados do RDS Custom e do Amazon RDS. Para ter mais informações, consulte Visualizar eventos do Amazon RDS.

Para visualizar a notificação de eventos do RDS Custom utilizando a AWS CLI, execute o comando `describe-events`. O RDS Custom introduz vários novos eventos. As categorias dos eventos são as mesmas que as do Amazon RDS. Para visualizar a lista de eventos, consulte Categorias de eventos e mensagens de eventos do Amazon RDS.

O exemplo a seguir recupera detalhes dos eventos que ocorreram para a instância de banco de dados do RDS Custom especificada.

```
aws rds describe-events \
    --source-identifier my-custom-instance \
    --source-type db-instance
```

Assinar eventos do RDS Custom

O procedimento para assinar eventos é o mesmo para instâncias de banco de dados do RDS Custom e do Amazon RDS. Para ter mais informações, consulte Inscrever-se em notificações de eventos do Amazon RDS.

Para assinar a notificação de eventos do RDS Custom usando a CLI, execute o comando `create-event-subscription`. Inclua os seguintes parâmetros necessários:

- `--subscription-name`
- `--sns-topic-arn`

O exemplo a seguir cria uma assinatura para eventos de backup e recuperação para uma instância de banco de dados do RDS Custom na conta atual da AWS. As notificações são enviadas para um tópico do Amazon Simple Notification Service (Amazon SNS), especificado por `--sns-topic-arn`.

```
aws rds create-event-subscription \
    --subscription-name my-instance-events \
    --source-type db-instance \
    --event-categories '["backup","recovery"]' \
    --sns-topic-arn arn:aws:sns:us-east-1:123456789012:interesting-events
```

Solucionar erros de CEV para o RDS Custom para SQL Server

Ao tentar criar uma CEV, ela pode falhar. Nesse caso, o RDS Custom emite a mensagem de evento `RDS-EVENT-0198`. Para ter mais informações sobre a visualização de eventos do RDS, consulte Categorias de eventos e mensagens de eventos do Amazon RDS.

Use as informações a seguir para ajudar você a resolver as possíveis causas.

Mensagem	Sugestões de solução de problemas		
`Custom Engine Version creation expected a`	Execute o Sysprep na instância do EC2 que você criou a partir		

Mensagem	Sugestões de solução de problemas
`Sysprep'd AMI. Retry creation using a Sysprep'd AMI.`	da AMI. Para ter mais informaçõ es sobre como preparar uma AMI utilizando o Sysprep, consulte Criar uma imagem de máquina da Amazon (AMI) padronizada usando o Sysprep.
`EC2 Image permissio ns for image (AMI_ID) weren't found for customer (Customer_ID). Verify customer (Customer_ID) has valid permissions on the EC2 Image.`	Verifique se sua conta e perfil usados para criação têm as permissões necessárias na `create EC2 Instance` e em `Describe Images` na AMI selecionada.
`Failed to rebuild databases with server collation (collation name) due to missing setup.exe file for SQL Server.`	Verifique se o arquivo `setup` está disponível em `C:\Progra m Files\Microsoft SQL Server\nnn\Setup Bootstrap \SQLnnnn\setup.exe` .
`Image (AMI_ID) doesn't exist in your account (ACCOUNT_ID). Verify (ACCOUNT_ID) is the owner of the EC2 image.`	A AMI deve existir na mesma conta de cliente.
`Image id (AMI_ID) isn't valid. Specify a valid image id, and try again.`	O nome da AMI está incorreto. O ID do AMI correto deve ser fornecido.

Mensagem	Sugestões de solução de problemas
`Image (AMI_ID) operating system platform isn't supported. Specify a valid image, and try again.`	Selecione uma AMI compatível que tenha o Windows Server com a edição SQL Server Enterprise, Standard ou Web. Selecione uma AMI com um dos seguintes códigos de operação de uso do EC2 Marketplace: • RunInstances:0102: Windows com SQL Server Enterprise • RunInstances:0006: Windows com SQL Server Standard • RunInstances:0202: Windows com SQL Server Web
`SQL Server Web Edition isn't supported for creating a Custom Engine Version using Bring Your Own Media. Specify a valid image, and try again.`	Use uma AMI que contenha uma edição compatível do SQL Server. Para obter mais informações, consulte Versões compatíveis com CEVs do RDS Custom para SQL Server.
`The custom engine version can't be the same as the OEV engine version. Specify a valid CEV, and try again.`	Não há suporte para as versões do mecanismo Classic RDS Custom para SQL Server. Por exemplo, versão 15.00.4073.23.v1. Use um número de versão compatível.
`The custom engine version isn't in an active state. Specify a valid CEV, and try again.`	A CEV deve estar em um estado `AVAILABLE` para concluir a operação. Modifique a CEV de `INACTIVE` para `AVAILABLE` .

Mensagem	Sugestões de solução de problemas
`The custom engine version isn't valid for an upgrade. Specify a valid CEV with an engine version greater or equal to (X), and try again.`	A CEV de destino não é válida. Verifique os requisitos para um caminho de atualização válido.
`The custom engine version isn't valid. Names can include only lowercase letters (a-z), dashes (-), underscores (_), and periods (.). Specify a valid CEV, and try again.`	Siga a convenção de nomenclatura da CEV necessária. Para ter mais informações, consulte Requisitos para CEVs do RDS Custom para SQL Server.
`The custom engine version isn't valid. Specify valid database engine version, and try again. Example: 15.00.4073.23-cev123.`	Foi fornecida uma versão do mecanismo de banco de dados não compatível. Use uma versão de mecanismo de banco de dados compatível.
`The expected architecture is (X) for image (AMI_ID), but architecture (Y) was found.`	Use uma AMI baseada na arquitetura x86_64.
`The expected owner of image (AMI_ID) is customer account ID (ACCOUNT_ID), but owner (ACCOUNT_ID) was found.`	Crie a instância do EC2 a partir da AMI para a qual você tem permissão. Execute o Sysprep na instância do EC2 para criar e salvar uma imagem base.
`The expected platform is (X) for image (AMI_ID), but platform (Y) was found.`	Use uma AMI criada com a plataforma Windows.

Mensagem	Sugestões de solução de problemas
`The expected root device type is (X) for image %s, but root device type (Y) was found.`	Crie a AMI com o tipo de dispositivo EBS.
`The expected SQL Server edition is (X), but (Y) was found.`	Selecione uma AMI compatível que tenha o Windows Server com a edição SQL Server Enterpris e, Standard ou Web. Selecione uma AMI com um dos seguintes códigos de operação de uso do EC2 Marketplace: • RunInstances:0102: Windows com SQL Server Enterprise • RunInstances:0006: Windows com SQL Server Standard • RunInstances:0202: Windows com SQL Server Web
`The expected state is (X) for image (AMI_ID), but the following state was found: (Y).`	A AMI deve estar em um estado de `AVAILABLE` .
`The provided Windows OS name (X) isn’t valid. Make sure the OS is one of the following: (Y).`	Use um sistema operacional Windows compatível.
`Unable to find bootstrap log file in path.`	Verifique se o arquivo de log está disponível em `C:\Program Files\Microsoft SQL Server \nnn\Setup Bootstrap\Log \Summary.txt` .

Mensagem	Sugestões de solução de problemas		
`RDS expected a Windows build version greater than or equal to (X), but found version (Y)..`	Use uma AMI com uma versão mínima de compilação do sistema operacional de 14393.		
`RDS expected a Windows major version greater than or equal to (X), but found version (Y)..`	Use uma AMI com uma versão principal mínima do sistema operacional 10.0 ou superior.		

Corrigir configurações não compatíveis no RDS Custom para SQL Server

Devido ao modelo de responsabilidade compartilhada, é sua responsabilidade corrigir problemas de configuração que colocam a instância de banco de dados do RDS Custom para SQL Server no estado `unsupported-configuration`. Se o problema for com a infraestrutura da AWS, será possível utilizar o console ou a AWS CLI para corrigi-lo. Se o problema for com o sistema operacional ou a configuração do banco de dados, será possível fazer login no host para corrigi-lo.

> **Note**
>
> Esta seção explica como corrigir configurações não compatíveis no RDS Custom para SQL Server. Para ter informações sobre o RDS Custom para Oracle, consulte Corrigir configurações não compatíveis no RDS Custom para Oracle.

As tabelas a seguir apresentam a descrição de notificações e eventos que o perímetro de suporte envia e como corrigi-los. Essas notificações e o perímetro de suporte estão sujeitos a alterações. Para obter informações básicas sobre o perímetro de suporte, consultePerímetro de suporte do RDS Custom. Para obter informações sobre descrições de eventos, consulte Categorias de eventos e mensagens de eventos do Amazon RDS.

Código do evento	Área de configuração	Mensagem de evento do RDS	Processo de validação
SP-S0000	Configuração manual incompatível	O status da instância de banco de dados do RDS Custom está definido como [Configuração incompatível] pelo seguinte motivo: X.	Para resolver esse problema, crie um caso de suporte.

Recurso da AWS (infraestrutura)

Código do evento	Área de configuração	Mensagem de evento do RDS	Processo de validação
SP-S1001	Estado da instância do EC2	O status da instância de banco de dados do RDS Custom está definido como [Configuração incompatível] pelo seguinte motivo: a instância do EC2 subjacente %s foi interrompida sem interromper a instância do RDS. É possível resolver isso iniciando a instância do EC2 subjacente e garantindo que os volumes	Para conferir o status de uma instância de banco de dados, use o console ou execute o seguinte comando AWS CLI: `aws rds describe-db-instances \ --db-instance-identifier db-instance-name |grep DBInstanceStatus`

Código do evento	Área de configuração	Mensagem de evento do RDS	Processo de validação
		binários e de dados estejam conectados. Se sua intenção for interromper a instância do RDS, certifique-se primeiro de que a instância do EC2 subjacente esteja no estado DISPONÍVEL e, depois, use o console do RDS ou a CLI para interromper a instância do RDS.	

Código do evento	Área de configuração	Mensagem de evento do RDS	Processo de validação

SP-S1002	Estado da instância do EC2	O status da instância de banco de dados do RDS Custom está definido como [Configuração incompatível] porque: o status da instância de banco de dados do RDS está definido como STOPPED, mas a instância do EC2 subjacente %s foi iniciada. É possível resolver isso interrompendo a instância do EC2 subjacente. Se sua intenção for iniciar a instância do RDS, use o console ou a CLI.	Use o seguinte comando AWS CLI para conferir o status de uma instância de banco de dados:
			`aws rds describe-db-instances \ --db-instance-identifier db-instance-name | grep DBInstanceStatus`
			Também é possível conferir o status da instância do EC2 usando o console do EC2.
			Para iniciar uma instância de banco de dados, use o console ou execute o seguinte comando AWS CLI:
			`aws rds start-db-instance \ --db-instance-identifier db-instance-name`

Código do evento	Área de configuração	Mensagem de evento do RDS	Processo de validação
SP-S1003	Classe de instância do EC2	O status da instância de banco de dados do RDS Custom está definido como [Configuração incompatível] pelo seguinte motivo: há uma incompatibilidade entre a classe de instância de banco de dados esperada e a configurada do host do EC2. É possível resolver isso modificando a classe da instância de banco de dados de acordo com o tipo de classe original.	Use o seguinte comando da CLI para conferir a classe de instância de banco de dados esperada: `aws rds describe-db-instances \ --db-instance-identifier db-instance-name | grep DBInstanceClass`

Código do evento	Área de configuração	Mensagem de evento do RDS	Processo de validação
SP-S1004	Volume de armazenamento do EBS não acessível	O status da instância de banco de dados do RDS Custom está definido como [Configuração incompatível] pelo seguinte motivo: o volume de armazenamento original do EBS %s que estava associado à instância do EC2 não está acessível no momento.	

Código do evento	Área de configuração	Mensagem de evento do RDS	Processo de validação
SP-S1005	Volume de armazenamento do EBS separado	O status da instância de banco de dados do RDS Custom está definido como [Configuração incompatível] pelo seguinte motivo: o volume de armazenamento original do EBS “volume-id” não está anexado. É possível resolver isso anexando o volume do EBS associado à instância do EC2.	Depois de reconectar o volume do EBS, use os seguintes comandos da CLI para conferir se o volume “volume-id” do EBS está conectado corretamente à instância do RDS: `aws ec2 describe-volumes \ --volume-ids volume-id | grep InstanceId`

Código do evento	Área de configuração	Mensagem de evento do RDS	Processo de validação

SP-S1006	Tamanho do volume de armazenamento do EBS	O status da instância de banco de dados do RDS Custom está definido como [Configuração incompatível] pelo seguinte motivo: há uma incompatibilidade entre as configurações esperadas e definidas do "volume-id" do volume de armazenamento do EBS. O tamanho do volume foi alterado manualmente no nível do EC2 a partir dos valores originais de [%s]. Para resolver esse problema, crie um caso de suporte.	Use o seguinte comando da CLI para comparar o tamanho do volume dos detalhes do "volume-id" do volume do EBS e os detalhes da instância do RDS:
			`aws rds describe-db-instances \ --db-instance-identifier db-instance-name | grep AllocatedStorage`
			Use o seguinte comando da CLI para ver o tamanho real do volume alocado:
			`aws ec2 describe-volumes \ --volume-ids |grep Size`

Código do evento	Área de configuração	Mensagem de evento do RDS	Processo de validação

SP-S1007	Configuração do volume de armazenamento do EBS	O status da instância de banco de dados do RDS Custom está definido como [Configuração incompatível] pelo seguinte motivo: há uma incompatibilidade entre as configurações esperadas e definidas do "volume-id" do volume de armazenamento do EBS. É possível resolver isso modificando a configuração do volume de armazenamento do EBS [IOPS, Throughput, Tipo de volume] para os valores originais de [IOPS: %s, Throughput: %s, Tipo de volume: %s] no nível do EC2. Para futuras modificações de armazenamento,	Use o comando da CLI a seguir para comparar o tipo de volume dos detalhes do "volume-id" do volume do EBS e os detalhes da instância do RDS. Garanta que os valores no nível do EBS correspondam aos valores no nível do RDS:
			`aws rds describe-db-instances \ --db-instance-identifier db-instance-name | grep StorageType`
			Para saber o valor esperado do throughput de armazenamento no nível do RDS:
			`aws rds describe-db-instances \ --db-instance-identifier db-instance-name | grep StorageThroughput`
			Para saber o valor esperado de IOPS de volume no nível do RDS:
			`aws rds describe-db-instances \ --db-instance-identifier db-instance-name | grep Iops`
			Para saber o tipo de armazenamento atual no nível do EC2:

Código do evento	Área de configuração	Mensagem de evento do RDS	Processo de validação

		use o console do RDS ou a CLI. O tamanho do volume também foi alterado manualmente no nível do EC2 dos valores originais de [%s]. Para resolver esse problema, crie um caso de suporte.	`aws ec2 describe-volumes \ --volume-ids |grep VolumeType`
			
			Para saber o valor atual do throughput de armazenamento no nível do EC2:
			
			`aws ec2 describe-volumes \ --volume-ids |grep Throughput`
			
			Para saber o valor esperado de IOPS de volume no nível do EC2:
			
			`aws ec2 describe-volumes \ --volume-ids |grep Iops`

Código do evento	Área de configuração	Mensagem de evento do RDS	Processo de validação

SP-S1008	Configuração e tamanho do volume de armazenamento do EBS	O status da instância de banco de dados do RDS Custom está definido como [Configuração incompatível] pelo seguinte motivo: há uma incompatibilidade entre as configurações esperadas e definidas do "volume-id" do volume de armazenamento do EBS. É possível resolver isso modificando a configuração do volume de armazenamento do EBS [IOPS, Throughput, Tipo de volume] para os valores originais de [IOPS: %s, Throughput: %s, Tipo de volume: %s] no nível do EC2. Para futuras modificações de armazenamento,	Use o comando da CLI a seguir para comparar o tipo de volume dos detalhes do "volume-id" do volume do EBS e os detalhes da instância do RDS. Garanta que os valores no nível do EBS correspondam aos valores no nível do RDS:
			`aws rds describe-db-instances \ --db-instance-identifier db-instance-name | grep StorageType`
			Para saber o valor esperado do throughput de armazenamento no nível do RDS:
			`aws rds describe-db-instances \ --db-instance-identifier db-instance-name | grep StorageThroughput`
			Para saber o valor esperado de IOPS de volume no nível do RDS:
			`aws rds describe-db-instances \ --db-instance-identifier db-instance-name | grep Iops`
			Para saber o tipo de armazenamento atual no nível do EC2:

Código do evento	Área de configuração	Mensagem de evento do RDS	Processo de validação

		use o console do RDS ou a CLI. O tamanho do volume também foi alterado manualmente no nível do EC2 dos valores originais de [%s]. Para resolver esse problema, crie um caso de suporte.	`aws ec2 describe-volumes \ --volume-ids |grep VolumeType`
			
			Para saber o valor atual do throughput de armazenamento no nível do EC2:
			
			`aws ec2 describe-volumes \ --volume-ids |grep Throughput`
			
			Para saber o valor esperado de IOPS de volume no nível do EC2:
			
			`aws ec2 describe-volumes \ --volume-ids |grep Iops`
			
			Para saber o tamanho esperado do volume alocado:
			
			`aws rds describe-db-instances \ --db-instance-identifier db-instance-name | grep AllocatedStorage`
			
			Para saber o tamanho real do volume alocado:
			
			`aws ec2 describe-volumes \ --volume-ids |grep Size`

Código do evento	Área de configuração	Mensagem de evento do RDS	Processo de validação
SP-S1009	Permissões do SQS	O status da instância de banco de dados do RDS Custom está definido como [Configuração incompatível] pelo seguinte motivo: faltam permissões do Amazon Simple Queue Service (SQS) para o perfil da instância do IAM. É possível resolver isso garantindo que o perfil do IAM associado ao host tenha as seguintes permissões: [“SQS:SendMessage”,”SQS:ReceiveMessage”,”SQS:DeleteMessage”,”SQS:GetQueueUrl”].	

Código do evento	Área de configuração	Mensagem de evento do RDS	Processo de validação
SP-S1010	Endpoint da VPC do SQS	O status da instância de banco de dados do RDS Custom está definido como [Configuração incompatível] pelo seguinte motivo: uma política de endpoint da VPC está bloqueando as operações do Amazon Simple Queue Service (SQS). É possível resolver isso modificando a política de endpoint da VPC para permitir as ações necessárias do SQS.	

Código do evento	Área de configuração	Mensagem de evento do RDS	Processo de validação
SP-S1011	Política de barramento de eventos	O status da instância de banco de dados do RDS Custom está definido como [Configuração incompatível] pelo seguinte motivo: a política baseada em recursos para seu barramento de eventos arn:aws:events:region-1:123456789012:event-bus/default nega ações events:PutEvents do Amazon CloudWatch. Resolva isso modificando sua política baseada em recursos para permitir ações de events:PutEvents para EventBus %s.	

Código do evento	Área de configuração	Mensagem de evento do RDS	Processo de validação
SP-S1012	Permissões do CloudWatch VPC	O status da instância de banco de dados do RDS Custom está definido como [Configuração incompatível] pelo seguinte motivo: uma política de endpoint da VPC está sem permissões para acessar os eventos do Amazon CloudWatch. Resolva isso modificando sua política de endpoint da VPC para permitir events:PutEvents no EventBus arn:aws:events:region-1:123456789012:event-bus/default.	

Código do evento	Área de configuração	Mensagem de evento do RDS	Processo de validação
SP-S1013	Política de controle de serviço	O status da instância de banco de dados do RDS Custom está definido como [Configuração incompatível] pelo seguinte motivo: uma política de controle de serviço em seu AWS Organizations está sem permissões para acessar os eventos do Amazon CloudWatch. Resolva isso modificando sua política de controle de serviço para permitir events:PutEvents no EventBus arn:aws:events:region-1:123456789012:event-bus/default.	

Código do evento	Área de configuração	Mensagem de evento do RDS	Processo de validação
SP-S1014	Perfil de instância do IAM	O status da instância de banco de dados do RDS Custom está definido como [Configuração incompatível] pelo seguinte motivo: as permissões do seu perfil %s da instância do IAM negam os eventos do Amazon CloudWatch. Resolva isso definindo ["events:PutEvents"] como 'Permitir' e permitindo events:PutEvents no EventBus arn:aws:events:region-1:123456789012:event-bus/default no seu perfil do IAM associado à instância.	

Código do evento	Área de configuração	Mensagem de evento do RDS	Processo de validação
SP-S1015	Perfil de instância do IAM	O status da instância de banco de dados do RDS Custom está definido como [Configuração incompatível] pelo seguinte motivo: o perfil %s da instância do IAM está sem permissões do evento do Amazon CloudWatch. Resolva isso incluindo as permissões ["events:PutEvents"] e permitindo events:PutEvents no EventBus arn:aws:events:region-1:123456789012:event-bus/default no seu perfil do IAM associado à instância.	

Código do evento	Área de configuração	Mensagem de evento do RDS	Processo de validação
SP-S1016	Limite de permissões do IAM	O status da instância de banco de dados do RDS Custom está definido como [Configuração incompatível] pelo seguinte motivo: o seu perfil %s da instância do IAM tem limite de permissões que negam os eventos do Amazon CloudWatch. Resolva isso definindo ["events:PutEvents"] como 'Permitir' para o EventBus arn:aws:events:region-1:123456789012:event-bus/default no limite de permissões do perfil da sua instância do IAM.	

Sistema operacional

Código do evento	Área de configuração	Mensagem de evento do RDS	Processo de validação
SP-S2001	Status do serviço SQL	O status da instância de banco de dados do RDS Custom está definido como [Configuração incompatível] pelo seguinte motivo: o serviço SQL Server não foi iniciado. É possível resolver isso reiniciando o serviço SQL Server no host. Se essa instância de banco de dados for uma instância de banco de dados multi-AZ e a reinicialização falhar, pare e inicialize o host para iniciar um failover.	

SP-S2002	Status do Agente do RDS Custom	O status da instância de banco de dados do RDS Custom está definido como [Configuração incompatível] pelo seguinte motivo: o serviço Agente do RDS Custom	Faça login no host e certifique-se de que o agente do RDS Custom esteja em execução.
			
			É possível utilizar os comandos a seguir para visualizar o status do agente.
			
			`$name = "RDSCustomAgent"`
			`$service = Get-Service $name`
			`Write-Host $service.Status`

Código do evento	Área de configuração	Mensagem de evento do RDS	Processo de validação
		não está instalado ou não pôde ser iniciado. É possível resolver isso examinando o log de eventos do Windows para determinar por que o serviço não foi iniciado e tomar as medidas apropriadas para corrigir o problema. Para receber assistência adicional, crie um caso de suporte.	Se o status não for `Running`, é possível iniciar o serviço com o seguinte comando: `Start-Service $name` Se o agente não for iniciado, confira os eventos do Windows para ver por que ele não pode ser iniciado. O agente exige que um usuário do Windows inicie o serviço. Verifique se existe um usuário do Windows e ele tem privilégios para executar o serviço.

Código do evento	Área de configuração	Mensagem de evento do RDS	Processo de validação
`SP-S2003`	Status do Agente do SSM	O status da instância de banco de dados do RDS Custom está definido como [Configuração incompatível] pelo seguinte motivo: o serviço Agente do Amazon SSM não pode ser acessado. É possível solucionar esse problema conferindo o status do serviço com o comando `Get-Service AmazonSSMAgent` do PowerShell ou iniciando o serviço com `Start-Service AmazonSSMAgent` . Garanta que o tráfego de saída HTTPS (porta 443) para os endpoints regionais ssm, ssmmessages e ec2messages seja permitido.	Para ter mais informações, consulte Solucionar problemas com o SSM Agent. Para solucionar problemas de endpoints do SSM, consulte Unable to connect to SSM endpoints e Use ssm-cli to troubleshoot managed node availability.

Código do evento	Área de configuração	Mensagem de evento do RDS	Processo de validação
SP-S2004	Login do Agente do RDS Custom	O status da instância de banco de dados do RDS Custom está definido como [Configuração incompatível] pelo seguinte motivo: ocorreu um problema inesperado com o login do SQL "$HOSTNAME/RDSAgent" . Para resolver esse problema, crie um caso de suporte.	

Código do evento	Área de configuração	Mensagem de evento do RDS	Processo de validação
SP-S2005	Fuso horário	O status da instância de banco de dados do RDS Custom está definido como [Configuração incompatível] pelo seguinte motivo: o fuso horário na instância do Amazon EC2 [%s] foi alterado. É possível resolver esse problema modificando o fuso horário de volta para a configuração especificada durante a criação da instância. Se você quiser criar uma instância com um fuso horário específico, consulte a documentação do RDS Custom.	Execute o comando `Get-Timezone` do PowerShell para confirmar o fuso horário. Para obter mais informações, consulte Fuso horário local para instâncias de banco de dados do RDS Custom para SQL Server.

Código do evento	Área de configuração	Mensagem de evento do RDS	Processo de validação
SP-S2006	Versão da solução de software de alta disponibilidade	O status da instância de banco de dados do RDS Custom está definido como [Configuração incompatível] pelo seguinte motivo: a solução de software de alta disponibilidade da instância atual é diferente da versão esperada. Para resolver esse problema, crie um caso de suporte.	

Código do evento	Área de configuração	Mensagem de evento do RDS	Processo de validação
SP-S2007	Configuração da solução de software de alta disponibilidade	O status da instância de banco de dados do RDS Custom está definido como [Configuração incompatível] pelo seguinte motivo: as configurações da solução de software de alta disponibilidade foi modificada para valores inesperados na instância %s. Para corrigir esse problema, reinicialize a instância do EC2. Ao reinicializar a instância do EC2, ela atualiza automaticamente as configurações para a configuração necessária para a solução de software de alta disponibilidade.	

Código do evento	Área de configuração	Mensagem de evento do RDS	Processo de validação

SP-S2008	SQL Server Service	A instância de banco de dados do RDS Custom está definida como [Configuração não suportada]: o serviço SQL Server (MSSQLServer) não existe no host. Para resolver isso, crie um caso de suporte.	É possível utilizar os comandos a seguir para visualizar o status do agente.
			`$name = "MSSQLServer"`
			`$service = Get-Service $name`
			`Write-Host $service.Status`

SP-2009	Certificado SSL	A instância de banco de dados do RDS Custom está definida como [Configuração incompatível] devido a certificados SSL não autoassinados que causam interrupção no RDS. Para resolver esse problema, remova os certificados não autoassinados do armazenamento de certificados raiz confiáveis.	Execute o comando do PowerShell a seguir para examinar os certificados não autoassinados. `Get-ChildItem cert:\LocalMachine\root -Recurse | Where-Object {$_.Issuer -ne $_.Subject -and $_.Issuer -notlike "*RDSCustomAgentCA*"}` Para ter mais informações, consulte HTTP Error 403.16 when you try to access a website that's hosted on IIS.

Código do evento	Área de configuração	Mensagem de evento do RDS	Processo de validação
SP-2010	Status do armazenamento do volume raiz	The RDS Custom DB instance is set to [Unsupported configuration] because of: Root volume storage is full. To resolve this issue, free up at least 500 MiB of storage space in the root EBS volume "volume-id" or increase the volume size and resize the C drive on the EC2 instance "instance-id". The root volume size changes do not persist when you replace the EC2 instance.	Use o comando a seguir para visualizar o armazenamento disponível no volume raiz (C:). `(Get-PSDrive -Name C).Free / 1MB` Para ter mais informações sobre como modificar o volume raiz do EBS, consulte Como.

Banco de dados

Código do evento	Área de configuração	Mensagem de evento do RDS	Processo de validação
SP-S3001	Protocolo de memória compartilhada do SQL Server	O status da instância de banco de dados do RDS Custom está definido como [Configuração	É possível validar isso conferindo: SQL Server Configuration Manager > Configuração de rede do SQL Server > Protocolos para MSSQLSERVER> Memória compartilhada como Habilitado.

Código do evento	Área de configuração	Mensagem de evento do RDS	Processo de validação
		incompatível] pelo seguinte motivo: o protocolo de memória compartilhada do SQL Server está desabilitado. É possível resolver esse problema habilitando o protocolo de memória compartilhada no SQL Server Configuration Manager.	Depois de habilitar o protocolo, reinicie o processo do SQL Server.
SP-S3002	Chave mestra de serviço	O status da instância de banco de dados do RDS Custom está definido como [Configuração incompatível] pelo seguinte motivo: a automação do RDS não consegue fazer backup da chave mestra de serviço (SMK) como parte da nova geração da SMK. Para resolver esse problema, crie um caso de suporte.	

Código do evento	Área de configuração	Mensagem de evento do RDS	Processo de validação
SP-S3003	Chave mestra de serviço	O status da instância de banco de dados do RDS Custom está definido como [Configuração incompatível] pelo seguinte motivo: os metadados relacionados à chave mestra de serviço (SMK) estão ausentes ou incompletos. Para resolver esse problema, crie um caso de suporte.	

Código do evento	Área de configuração	Mensagem de evento do RDS	Processo de validação

SP-S3004	Versão e edição de mecanismos de banco de dados	O status da instância de banco de dados do RDS Custom está definido como [Configuração não suportada] pelo seguinte motivo: há uma incompatibilidade entre a versão e a edição esperadas e instaladas do SQL Server: a modificação da edição do SQL Server não é suportada no RDS Custom para SQL Server. Além disso, não é possível alterar manualmente a versão do SQL Server na instância do EC2 no RDS Custom. Para resolver esse problema, crie um caso de suporte.	Execute a seguinte consulta para saber a versão do SQL:
			`select @@version`
			Execute o seguinte comando da AWS CLI para saber a versão do mecanismo SQL no RDS:
			`aws rds describe-db-instances \ --db-instance-identifier db-instance-name |grep EngineVersion`
			`aws rds describe-db-instances \ --db-instance-identifier db-instance-name |grep Engine`
			Para obter mais informações, consulte Modificar uma instância de banco de dados do RDS Custom for SQL Server e Atualizar a versão de mecanismo de uma instância de banco de dados.

Código do evento	Área de configuração	Mensagem de evento do RDS	Processo de validação

SP-S3005	Edição do mecanismo de banco de dados	O status da instância de banco de dados do RDS Custom está definido como [Configuração incompatível] pelo seguinte motivo: a edição atual do SQL Server não coincide com a edição esperada do SQL Server [%s]: a modificação do SQL Server não tem suporte no RDS Custom para SQL Server. Para resolver esse problema, crie um caso de suporte.	Execute a seguinte consulta para saber a edição do SQL:
			Example
			`select @@version`
			Execute o seguinte comando AWS CLI para saber a edição do mecanismo SQL do RDS:
			`aws rds describe-db-instances \ --db-instance-identifier db-instance-name |grep Engine`

Código do evento	Área de configuração	Mensagem de evento do RDS	Processo de validação

SP-S3006	Versão do mecanismo de banco de dados	O status da instância de banco de dados do RDS Custom está definido como [Configuração incompatível] pelo seguinte motivo: a versão atual do SQL Server não coincide com a versão esperada do SQL Server [%s]: não é possível alterar manualmente a versão do SQL Server na instância do RDS Custom EC2. Para resolver esse problema, crie um caso de suporte. No caso de quaisquer modificações futuras na versão do SQL Server, você pode modificar a instância no console do AWS RDS ou por meio do comando modify-db-instance da CLI.	Execute a seguinte consulta para saber a versão do SQL:
			
			Example
			
			`select @@version`
			
			Execute o seguinte comando AWS CLI para saber a versão do mecanismo SQL do RDS:
			
			`aws rds describe-db-instances \ --db-instance-identifier db-instance-name |grep EngineVersion`
			
			Para obter mais informações, consulte Modificar uma instância de banco de dados do RDS Custom for SQL Server e Atualizar a versão de mecanismo de uma instância de banco de dados.

Código do evento	Área de configuração	Mensagem de evento do RDS	Processo de validação

SP-S3007	Local dos arquivos de banco de dados	O status da instância de banco de dados do RDS Custom está definido como [Configuração incompatível] pelo seguinte motivo: os arquivos do banco de dados estão configurados fora da unidade D:\. É possível resolver esse problema garantindo que todos os arquivos do banco de dados, incluindo ROW, LOG, FILESTREAM, etc... estejam armazenados na unidade D:\.	Execute a consulta a seguir para listar a localização dos arquivos do banco de dados que não estão no caminho padrão:
			`USE master;`
			`SELECT physical_name as files_not_in_default_path`
			`FROM sys.master_files`
			`WHERE SUBSTRING(physical_name,1,3)!='D:\';`

Código do evento	Área de configuração	Mensagem de evento do RDS	Processo de validação
SP-S3008	Limite excedido de contagem de banco de dados	O status da instância de banco de dados do RDS Custom está definido como [Configuração não suportada] pelo seguinte motivo: o número total de bancos de dados na instância de banco de dados excede o limite máximo de 5000. Para resolver isso, reduza o número de bancos de dados para abaixo do limite máximo suportado.	Use o seguinte comando para visualizar a contagem total de bancos de dados: `SELECT COUNT(name) as databaseCount FROM sys.databases WHERE name not in ('tempdb','master','model','msdb','DWDiagnostics','DWConfiguration','DWQueue');`

Solucionar problemas de **`Storage-Full`** no RDS Custom para SQL Server

O RDS Custom também monitora o volume raiz (C:). O RDS Custom para SQL Server move a instância de banco de dados para o estado `unsupported-configuration` quando o volume raiz tem menos de 500 MiB de espaço em disco disponível. Consulte `Event SP-S2010` em Corrigir configurações não compatíveis no RDS Custom para SQL Server.

Solucionar problemas do estado PENDING_RECOVERY para bancos de dados habilitados para TDE no RDS Custom para SQL Server

Os bancos de dados do SQL Server com criptografia de dados transparente (TDE) ativada poderão permanecer no estado `PENDING_RECOVERY` se a descriptografia automática apresentar problemas. Isso normalmente ocorre após a restauração de uma instância de banco de dados, se o arquivo de backup do Service Master Key (SMK) da instância de banco de dados de origem armazenado no

bucket do S3 gerenciado pelo RDS Custom em sua conta tiver sido excluído antes da conclusão da restauração.

Para ativar a descriptografia automática e os bancos de dados habilitados para TDE, você precisa abrir a Database Master Key (DMK) com a senha e criptografar a DMK usando o SMK.

Use os seguintes comandos do SQL Server como referência:

```
-- Identify PENDING_RECOVERY TDE databases
USE MASTER;
GO
SELECT name, is_encrypted, state_desc FROM sys.databases;
GO

-- Open DMK using password
OPEN MASTER KEY DECRYPTION BY PASSWORD = '<password>';
GO

-- Encrypt DMK using SMK
ALTER MASTER KEY ADD ENCRYPTION BY SERVICE MASTER KEY;
GO

-- Close SMK
CLOSE MASTER KEY;
GO

-- Bring the TDE databases online
ALTER DATABASE <database_name> SET ONLINE;
GO

-- Verify TDE databases are now in ONLINE state
SELECT name, is_encrypted, state_desc FROM sys.databases;
GO
```

Amazon RDS em AWS Outposts

O Amazon RDS no AWS Outposts amplia os bancos de dados do RDS para SQL Server, RDS para MySQL e RDS para PostgreSQL para ambientes de rack de primeira e segunda geração do AWS Outposts. O AWS Outposts usa o mesmo hardware das Regiões da AWS públicas para levar serviços, infraestrutura e modelos de operação da AWS para o ambiente on-premises. Com o RDS no Outposts, é possível provisionar instâncias de banco de dados gerenciados próximo aos aplicativos de negócios que devem ser executados no local. Para obter mais informações sobre AWS Outposts, consulte a documentação do AWS Outposts e a página do produto AWS Outposts.

Você usa o mesmo AWS Management Console, a AWS CLI e a API do RDS para provisionar e gerenciar instâncias de banco de dados do RDS on Outposts on-premises que usa para instâncias de banco de dados do RDS em execução na Nuvem AWS. O RDS on Outposts automatiza tarefas, como provisionamento de banco de dados, aplicação de patches no sistema operacional e no banco de dados, backup e arquivamento em longo prazo no Amazon S3.

O RDS no Outposts oferece suporte a backups automatizados de instâncias de banco de dados. A conectividade de rede entre o Outpost e sua Região da AWS é necessária para fazer backup e restaurar instâncias de banco de dados. Todos os snapshots do banco de dados e os logs de transação de um Outpost são armazenados em sua Região da AWS. Na região da AWS, é possível restaurar uma instância de banco de dados de um snapshot do banco de dados para um Outpost diferente. Para obter mais informações, consulte Introdução aos backups.

RDS no Outposts oferece suporte à manutenção automatizada e às atualizações de instâncias de banco de dados. Para obter mais informações, consulte Manutenção de uma instância de banco de dados.

O RDS on Outposts usa criptografia em repouso para instâncias de banco de dados e snapshots de banco de dados usando a sua AWS KMS key. Para obter mais informações sobre criptografia em repouso, consulte Criptografar recursos do Amazon RDS.

Por padrão, as instâncias do EC2 em sub-redes Outposts podem usar o Serviço Amazon Route 53 DNS para resolver nomes de domínio para endereços IP. Você pode encontrar tempos mais longos de resolução de DNS com Route 53, dependendo da latência do caminho entre o Outpost e a Região da AWS. Nesses casos, você pode usar os servidores DNS instalados localmente em seu ambiente local. Para obter mais informações, consulte DNS no Guia do usuário do AWS Outposts.

Quando a conectividade de rede com a Região da AWS não estiver disponível, a instância de banco de dados continuará a ser executada localmente. Você pode continuar acessando instâncias

de banco de dados usando a resolução de nomes DNS configurando um servidor DNS local como um servidor secundário. No entanto, não é possível criar instâncias de banco de dados nem modificar instâncias de banco de dados existentes. Os backups automáticos não ocorrem quando não há conectividade. Se houver uma falha na instância de banco de dados, ela não será substituída automaticamente até que a conectividade seja restaurada. Recomendamos restaurar a conectividade de rede o mais rápido possível.

Tópicos

- Pré-requisitos para o Amazon RDS on AWS Outposts
- Suporte do Amazon RDS on AWS Outposts para recursos do Amazon RDS
- Classes de instância de banco de dados compatíveis com o Amazon RDS on AWS Outposts
- Endereços IP de propriedade do cliente para o Amazon RDS no AWS Outposts.
- Trabalhar com implantações multi-AZ para o Amazon RDS no AWS Outposts
- Criar instâncias de banco de dados para o Amazon RDS on AWS Outposts
- Criar réplicas de leitura do Amazon RDS no AWS Outposts
- Considerações para restaurar instâncias de banco de dados no Amazon RDS on AWS Outposts

Pré-requisitos para o Amazon RDS on AWS Outposts

Veja a seguir os pré-requisitos para usar o Amazon RDS on AWS Outposts:

- Instale o AWS Outposts em seu datacenter no local. Para obter mais informações, consulte Instalação de um servidor AWS Outposts no Guia de instalação do servidor AWS Outposts.
- É necessário ter pelo menos uma sub-rede disponível para o RDS no Outposts. Você pode usar a mesma sub-rede para outras workloads.
- Você deve ter uma conexão de rede confiável entre o Outpost e uma região da AWS.

Suporte do Amazon RDS on AWS Outposts para recursos do Amazon RDS

A tabela a seguir descreve os recursos do Amazon RDS compatíveis com o Amazon RDS no AWS Outposts.

Atributo	Compatível	Observações	Mais informações

Provisionamento de instância de banco de dados	Sim	Você só pode criar instâncias de banco de dados para os mecanismos de banco de dados do RDS for SQL Server, RDS for MySQL e RDS for PostgreSQL. As seguintes versões são compatíveis:
		• Microsoft SQL Server:
		  • 15.00.4043.16.v1 e versões 2019 posteriores
		  • 14.00.3294.2.v1 e versões 2017 posteriores
		  • 13.00.5820.21.v1 e versões 2016 posteriores
		• MySQL versões 8.0 e 8.4
		• Todas as versões do PostgreSQL 16, 15, 14 e 13, PostgreSQL versão 12.5 e versões posteriores do PostgreSQL 12
			Criar instâncias de banco de dados para o Amazon RDS on AWS Outposts

Atributo	Compatível	Observações	Mais informações
Conecte-se a uma instância de banco de dados do Microsoft SQL Server com o Microsoft SQL Server Management Studio	Sim	Algumas versões TLS e cifras de criptografia podem não ser seguras. Para desativá-las, siga as instruções em ???.	Conectar-se à instância de banco de dados do Microsoft SQL Server
Modificar a senha do usuário mestre	Sim	Nenhum	Modificar uma instância de banco de dados do Amazon RDS
Renomear uma instância de banco de dados	Sim	Nenhum	Modificar uma instância de banco de dados do Amazon RDS
Reinicializar uma instância de banco de dados	Sim	Nenhum	Reinicializar uma instância de banco de dados
Interromper uma instância de banco de dados	Sim	Nenhum	Parar uma instância de banco de dados do Amazon RDS temporariamente
Iniciar uma instância de banco de dados	Sim	Nenhum	Iniciar uma instância de banco de dados do Amazon RDS que foi anteriormente interrompida

Atributo	Compatível	Observações	Mais informações
Implantações multi-AZ	Sim	Implantações multi-AZ são compatíveis com instâncias de banco de dados do MySQL e do PostgreSQL. As implantações multi-AZ não oferecem suporte ao roteamento direto de VPC (DVR).	Criar instâncias de banco de dados para o Amazon RDS on AWS Outposts Configurar e gerenciar uma implantação multi-AZ para o Amazon RDS
Grupos de parâmetros do banco de dados	Sim	Nenhum	Grupos de parâmetros para Amazon RDS
Réplicas de leitura	Sim	As réplicas de leitura são compatíveis com instâncias de banco de dados do MySQL e do PostgreSQL. As réplicas de leitura não oferecem suporte ao roteamento direto de VPC (DVR).	Criar réplicas de leitura do Amazon RDS no AWS Outposts
Criptografia em repouso	Sim	O RDS no Outposts não oferece suporte a instâncias de banco de dados não criptografadas.	Criptografar recursos do Amazon RDS
AWS Identity and Access Managemen tAutenticação do banco de dados do (IAM)	Não	Nenhum	Autenticação do banco de dados do IAM para MariaDB, MySQL e PostgreSQL

Atributo	Compatível	Observações	Mais informações
Associar uma função do IAM a uma instância de banco de dados	Não	Nenhum	Comando add-role-to-db-instance AWS CLI Operação AddRoleToDBInstance da API do RDS
Autenticação de Kerberos	Não	Nenhum	Autenticação de Kerberos
Marcar recursos do Amazon RDS	Sim	Nenhum	Marcar recursos do Amazon RDS
Grupos de opções	Sim	Nenhum	Trabalhar com grupos de opções
Modificar a janela de manutenção	Sim	Nenhum	Manutenção de uma instância de banco de dados
Atualização automática da versão secundária	Sim	Nenhum	Atualizar automaticamente a versão do mecanismo espelho
Modificar a janela de backup	Sim	Nenhum	Introdução aos backups Modificar uma instância de banco de dados do Amazon RDS
Alterar a classe da instância de banco de dados	Sim	Nenhum	Modificar uma instância de banco de dados do Amazon RDS

Atributo	Compatível	Observações	Mais informações
Alterar o armazenamento alocado	Sim	Nenhum	Modificar uma instância de banco de dados do Amazon RDS
Storage autoscaling (Autoscaling do armazenamento)	Sim	Nenhum	Gerenciar a capacidade automaticamente com a escalabilidade automática de armazenamento do Amazon RDS
Snapshots manuais e automáticos da instância de banco de dados	Sim	É possível armazenar backups automatizados e snapshots manuais em sua Região da AWS. Ou você pode armazená-los localmente no Outpost. Os backups locais são compatíveis com instâncias de banco de dados do MySQL e do PostgreSQL. Para armazenar backups no seu Outpost, certifique-se de ter o Amazon S3 on Outposts configurado. As implantações de instâncias multi-AZ não oferecem suporte a backups locais.	Criar instâncias de banco de dados para o Amazon RDS on AWS Outposts Amazon S3 on Outposts Criar um snapshot de banco de dados para uma instância de banco de dados single-AZ para o Amazon RDS

Atributo	Compatível	Observações	Mais informações
Restaurar de um snapshot de banco de dados	Sim	É possível armazenar backups automatizados e snapshots manuais para a instância de banco de dados restaurada na Região da AWS principal ou localmente no seu Outpost.	Considerações para restaurar instâncias de banco de dados no Amazon RDS on AWS Outposts Restaurar uma instância de banco de dados
Restaurar uma instância de banco de dados do Amazon S3	Não	Nenhum	Restauração de um backup em uma instância de banco de dados MySQL
Exportar dados de snapshot para o Amazon S3	Não	Nenhum	Exportação de dados de snapshot de banco de dados para o Amazon S3 para Amazon RDS
Recuperação para um ponto no tempo	Sim	Você pode armazenar backups automatizados e snapshots manuais para a instância de banco de dados restaurad a na Região da AWS principal ou localmente no seu Outpost, com uma exceção.	Considerações para restaurar instâncias de banco de dados no Amazon RDS on AWS Outposts Restaurar uma instância de banco de dados para um momento especificado no Amazon RDS
Monitoramento avançado	Não	Nenhum	Monitorar métricas do SO com o monitoramento avançado

Atributo	Compatível	Observações	Mais informações
Monitoramento do Amazon CloudWatch	Sim	É possível visualizar o mesmo conjunto de métricas disponíveis para seus bancos de dados na Região da AWS.	Monitorar métricas do Amazon RDS com o Amazon CloudWatch
Publicar logs de mecanismos de banco de dados no CloudWatch Logs	Sim	Nenhum	Publicação de logs de banco de dados no Amazon CloudWatch Logs
Notificação de eventos	Sim	Nenhum	Trabalhar com a notificação de eventos do Amazon RDS
Amazon RDS Performance Insights	Não	Nenhum	Monitorar a carga de banco de dados com o Performance Insights no Amazon RDS

Atributo	Compatível	Observações	Mais informações
Visualizar ou baixar logs de banco de dados	Não	O RDS no Outposts não é compatível com a visualização de logs de banco de dados utilizando o console nem com a descrição de logs de banco de dados utilizando a AWS CLI ou a API do RDS. O RDS no Outposts não é compatível com o download de logs de banco de dados utilizando o console nem ao download de logs de banco de dados utilizando a AWS CLI ou a API do RDS.	Monitorar arquivos de log do Amazon RDS
Proxy do Amazon RDS	Não	Nenhum	Amazon RDS Proxy
Procedimentos armazenados para Amazon RDS for MySQL	Sim	Nenhum	Referência de procedimentos armazenados do RDS para MySQL
Replicação com bancos de dados externos para o RDS for MySQL	Não	Nenhum	Configurar a replicação da posição do arquivo de log binário com uma instância de origem externa

Atributo	Compatível	Observações	Mais informações
Opção de backup e restauração nativos do Amazon RDS for Microsoft SQL Server	Sim	Nenhum	Importar e exportar bancos de dados do SQL Server usando backup e restauração nativos

Classes de instância de banco de dados compatíveis com o Amazon RDS on AWS Outposts

O Amazon RDS on AWS Outposts é compatível com as classes de instância de banco de dados a seguir:

- Classes de instâncias de banco de dados de uso geral
  - db.m7i.48xlarge
  - db.m7i.24xlarge
  - db.m7i.16xlarge
  - db.m7i.12xlarge
  - db.m7i.8xlarge
  - db.m7i.4xlarge
  - db.m7i.2xlarge
  - db.m7i.xlarge
  - db.m7i.large
  - db.m5.24xlarge
  - db.m5.16xlarge
  - db.m5.12xlarge
  - db.m5.8xlarge
  - db.m5.4xlarge
  - db.m5.2xlarge

  - db.m5.xlarge
  - db.m5.large
- Classes de instâncias de banco de dados otimizadas para memória
  - db.r7i.48xlarge
  - db.r7i.24xlarge
  - db.r7i.16xlarge
  - db.r7i.12xlarge
  - db.r7i.8xlarge
  - db.r7i.4xlarge
  - db.r7i.2xlarge
  - db.r7i.xlarge
  - db.r7i.large
  - db.r5.24xlarge
  - db.r5.16xlarge
  - db.r5.12xlarge
  - db.r5.8xlarge
  - db.r5.4xlarge
  - db.r5.2xlarge
  - db.r5.xlarge
  - db.r5.large

Dependendo de como você configurou seu Outpost, talvez você não tenha todas essas classes disponíveis. Por exemplo, se você não comprou as classes db.r5 para o Outpost, não poderá usá-las com o RDS on Outposts.

Somente o armazenamento SSD de uso geral é compatível com as instâncias de banco de dados do RDS no Outposts. Para mais informações sobre classes de instância de banco de dados, consulte Classes de instâncias de banco de dados .

Amazon RDS gerencia a manutenção e a recuperação de suas instâncias de banco de dados e requer capacidade ativa no Outpost para fazer isso. Recomendamos configurar instâncias N+1 EC2 para cada classe de instância de banco de dados em seus ambientes de produção. O RDS on Outposts pode usar a capacidade extra dessas instâncias do EC2 para operações de manutenção.

e reparo. Por exemplo, se seus ambientes de produção tiverem 3 classes de instância de banco de dados db.m5.large e 5 db.r5.xlarge, recomendamos que eles tenham pelo menos 4 instâncias do EC2 m5.large e 6 instâncias do EC2 r5.xlarge. Para obter mais informações, consulte o tópico sobre Resiliência no AWS Outposts, no Guia do usuário do AWS Outposts.

Endereços IP de propriedade do cliente para o Amazon RDS no AWS Outposts.

O Amazon RDS on AWS Outposts usa as informações fornecidas sobre sua rede on-premises para criar um grupo de endereços. Esse grupo é conhecido como grupo de endereços IP de propriedade do cliente (grupo de CoIPs). Os endereços IP de propriedade do cliente (CoIPs) fornecem conectividade local ou externa aos recursos nas sub-redes do Outpost por meio de sua rede on-premises. Para obter mais informações sobre CoIPs, consulte Customer-owned IP addresses (Endereço IP de propriedade do cliente) no Guia do usuário do AWS Outposts.

Cada instância de banco de dados do RDS no Outposts tem um endereço IP privado para tráfego dentro de sua Virtual Private Cloud (VPC). Esse endereço IP privado não é acessível publicamente. É possível usar a opção Public (Público) para definir se a instância de banco de dados também deve ter um endereço IP público além do endereço IP privado. O uso do endereço IP público para conexões os roteia pela Internet e pode resultar em altas latências em alguns casos.

Em vez de usar esses endereços IP públicos e privados, o RDS on Outposts é compatível com o uso de CoIPs para instâncias de banco de dados por meio de suas sub-redes. Ao usar um CoIP para uma instância de banco de dados do RDS on Outposts, você se conecta à instância de banco de dados com o endpoint da instância de banco de dados. O RDS on Outposts usa automaticamente o CoIP para todas as conexões de dentro e fora da VPC.

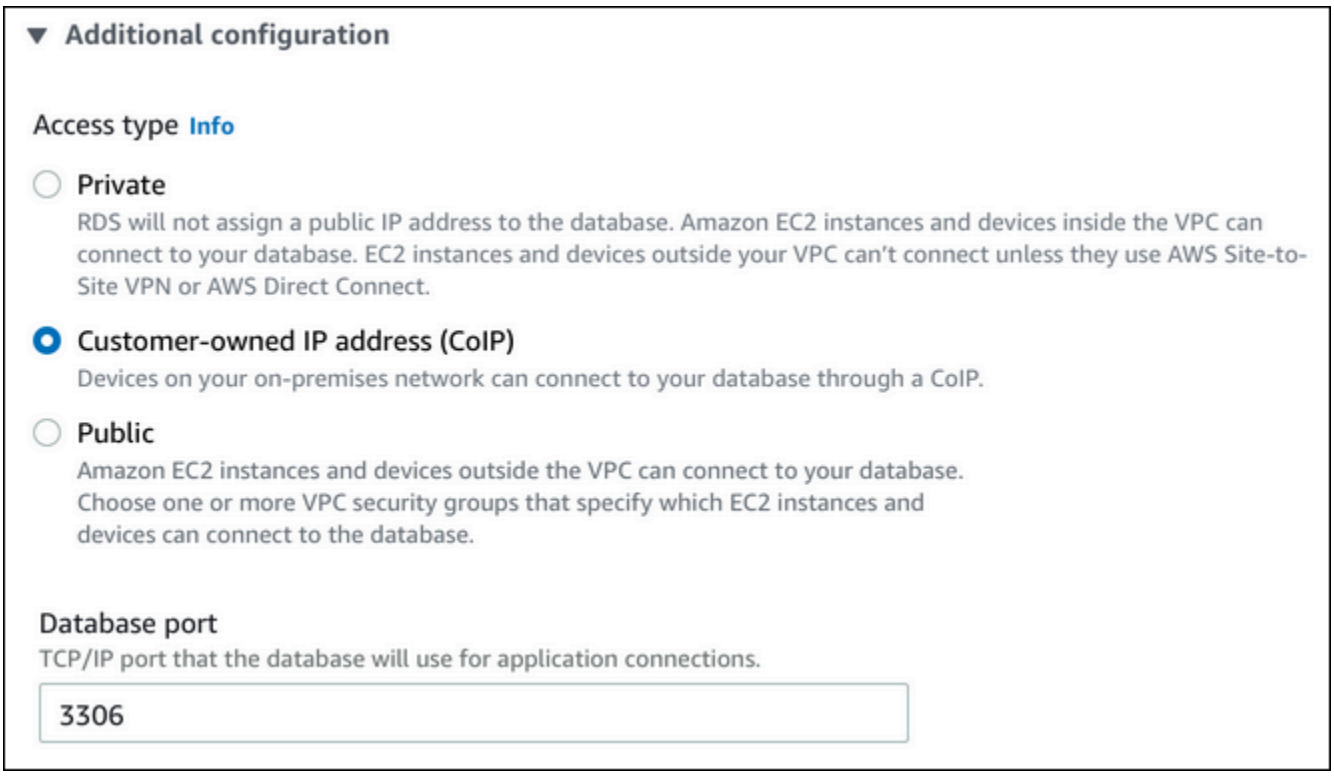
Os CoIPs podem fornecer os seguintes benefícios para instâncias de banco de dados do RDS no Outposts:

- Menor latência de conexão
- Segurança aprimorada

Usar CoIPs

Você pode ativar ou desativar um CoIP para uma instância de banco de dados do RDS on Outposts usando o AWS Management Console, a AWS CLI ou a API do RDS:

- Com o AWS Management Console, escolha a configuração Customer-owned IP address (CoIP) [Endereço IP de propriedade do cliente (CoIP)] em Access type (Tipo de acesso) para usar CoIPs. Escolha uma das outras configurações para desativá-las.



- Com a AWS CLI, use a opção `--enable-customer-owned-ip | --no-enable-customer-owned-ip`.
- Com a API do RDS, use o parâmetro `EnableCustomerOwnedIp`.

É possível ativar ou desativar CoIPs ao realizar qualquer uma das seguintes ações:

- Criar uma instância de banco de dados

  Para obter mais informações, consulte Criar instâncias de banco de dados para o Amazon RDS on AWS Outposts.
- Modificar uma instância de banco de dados

  Para obter mais informações, consulte Modificar uma instância de banco de dados do Amazon RDS.
- Criar uma réplica de leitura

  Para obter mais informações, consulte Criar réplicas de leitura do Amazon RDS no AWS Outposts.
- Restaurar uma instância de banco de dados usando um snapshot

  Para obter mais informações, consulte Restaurar uma instância de banco de dados.

- Para restaurar uma instância de banco de dados para um horário especificado

  Para obter mais informações, consulte Restaurar uma instância de banco de dados para um momento especificado no Amazon RDS.

> **Note**
>
> Em alguns casos, você pode ativar CoIPs para uma instância de banco de dados, mas o Amazon RDS não consegue alocar um CoIP para a instância de banco de dados. Nesses casos, o status da instância de banco de dados é alterado para rede incompatível. Para obter mais informações sobre o status da instância de banco de dados, consulte Visualizar o status de uma instância de banco de dados Amazon RDS em um .

Limitações

As seguintes limitações se aplicam ao suporte a CoIP para instâncias de banco de dados doRDS no Outposts:

- Ao usar um CoIP para uma instância de banco de dados, verifique se a acessibilidade pública está desativada para essa instância de banco de dados.
- As regras de entrada para os grupos de segurança de VPC devem incluir o intervalo de endereços CoIP (bloco CIDR). Para obter mais informações sobre como configurar um grupo de segurança, consulte Fornecer acesso à instância de banco de dados na VPC criando um grupo de segurança.
- Você não pode atribuir um CoIP de um grupo de CoIPs a uma instância de banco de dados. Ao usar um CoIP para uma instância de banco de dados, o Amazon RDS atribui automaticamente um CoIP de um grupo de CoIPs à instância de banco de dados.
- Você deve utilizar a Conta da AWS com os recursos do Outpost (proprietário) ou compartilhar os recursos a seguir com outras Contas da AWS (consumidores) na mesma organização.
  - O Outpost
  - A tabela de rotas do gateway local (LGW) para a VPC da instância de banco de dados
  - O grupo de CoIPs ou os grupos da tabela de rotas do LGW

  Para obter mais informações, consulte Working with shared AWS Outposts resources (Trabalhar com recursos compartilhados AWS Outposts) no AWS Outposts User Guide (Guia do usuário do AWS Outposts).

Trabalhar com implantações multi-AZ para o Amazon RDS no AWS Outposts

No caso de implantações multi-AZ, o Amazon RDS cria uma instância de banco de dados primário em um AWS Outpost. O RDS replica de forma síncrona os dados em uma instância de banco de dados em espera em outro Outpost.

Implantações multi-AZ no AWS Outposts funcionam como implantações multi-AZ nas Regiões da AWS, mas com as seguintes diferenças:

- Elas exigem uma conexão local entre dois ou mais Outposts.
- Elas exigem grupos de IPs de propriedade do cliente (CoIP). Para ter mais informações, consulte Endereços IP de propriedade do cliente para o Amazon RDS no AWS Outposts..
- A replicação é realizada na sua rede local.

O recurso multi-AZ no AWS Outposts está disponível para todas as versões compatíveis do MySQL e do PostgreSQL no RDS on Outposts. Não há compatibilidade para backups locais em implantações multi-AZ. Para ter mais informações, consulte Criar instâncias de banco de dados para o Amazon RDS on AWS Outposts.

Trabalhar com o modelo de responsabilidade compartilhada

Embora a AWS utilize esforços comercialmente razoáveis para fornecer instâncias de banco de dados configuradas para alta disponibilidade, a disponibilidade usa um modelo de responsabilidade compartilhada. A capacidade do RDS on Outposts de fazer failover e reparar instâncias de banco de dados exige que todos os seus Outposts estejam conectados à sua Região da AWS.

O RDS on Outposts também exige conectividade entre o Outpost que está hospedando a instância de banco de dados primário e o Outpost que está hospedando a instância de banco de dados em espera para replicação síncrona. Qualquer impacto nessa conexão pode impedir que o RDS on Outposts realize um failover.

Talvez você perceba latências elevadas para uma implantação de instância de banco de dados padrão como resultado da replicação de dados síncrona. A largura de banda e a latência da conexão entre o Outpost que hospeda a instância de banco de dados primário e o Outpost que hospeda a instância de banco de dados em espera afetam diretamente as latências. Para ter mais informações, consulte Pré-requisitos.

Melhorar a disponibilidade

Recomendamos as seguintes ações para melhorar a disponibilidade:

- Aloque capacidade adicional suficiente para suas aplicações essenciais à missão para permitir recuperação e o failover se houver um problema de host subjacente. Isso se aplica a todos os Outposts que contêm sub-redes em seu grupo de sub-redes de banco de dados. Para ter mais informações, consulte Resiliência no AWS Outposts.
- Forneça conectividade de rede redundante para o Outposts.
- Use mais de dois Outposts. Ter mais de dois Outposts permite que o Amazon RDS recupere uma instância de banco de dados. O RDS faz essa recuperação movendo a instância de banco de dados para outro Outpost se o Outpost atual tiver uma falha.
- Forneça fontes de alimentação duplas e conectividade de rede redundante para o Outpost.

Recomendamos o seguinte para suas redes locais:

- A latência do tempo de ida e volta (RTT) entre o Outpost que hospeda sua instância de banco de dados primário e o Outpost que hospeda sua instância de banco de dados em espera afeta diretamente a latência de gravação. Mantenha a latência RTT entre o AWS Outposts em milissegundos de um dígito baixo. Recomendamos até cinco milissegundos, mas seus requisitos podem variar.

  Você pode encontrar o impacto líquido na latência da rede nas métricas do Amazon CloudWatch para `WriteLatency`. Para ter mais informações, consulte Métricas do Amazon CloudWatch para o Amazon RDS.
- A disponibilidade da conexão entre os Outposts afeta a disponibilidade geral de suas instâncias de banco de dados. Tenha conectividade de rede redundante entre os Outposts.

Pré-requisitos

As implantações multi-AZ no RDS on Outposts têm os seguintes pré-requisitos:

- Tenha pelo menos dois Outposts, conectados por conexões locais e anexados a diferentes zonas de disponibilidade em uma Região da AWS.
- Seus grupos de sub-redes de banco de dados devem conter o seguinte:

  - No mínimo duas sub-redes em pelo menos duas zonas de disponibilidade em determinada Região da AWS.
  - Sub-redes somente nos Outposts.
  - Pelo menos duas sub-redes em pelo menos dois Outposts dentro da mesma nuvem privada virtual (VPC).
- Associe a VPC de sua instância de banco de dados a todas as tabelas de rotas de gateway local. Essa associação é necessária porque a replicação é realizada por sua rede local usando os gateways locais de seus Outposts.

  Por exemplo, suponha que sua VPC contenha a sub-rede A no Outpost-A e a sub-rede-B no Outpost-B. O Outpost-A usa LocalGateway-A (LGW-A) e o Outpost-B usa LocalGateway-B (LGW-B). O LGW-A tem RouteTable-A e o LGW-B tem RouteTable-B. Você precisa usar RouteTable-A e RouteTable-B para o tráfego de replicação. Para fazer isso, associe sua VPC a RouteTable-A e RouteTable-B.

  Para ter mais informações sobre como criar uma associação, consulte o comando create-local-gateway-route-table-vpc-association AWS CLI do Amazon EC2.
- Verifique se seus Outposts usam o roteamento IP de propriedade do cliente (CoIP). Cada tabela de rotas também deve ter pelo menos um grupo de endereços. O Amazon RDS aloca um endereço IP adicional para as instâncias de banco de dados primário e em espera para sincronização de dados.
- Verifique se a Conta da AWS que possui as instâncias de banco de dados do RDS tem as tabelas de rotas de gateway local e os grupos de CoIPs. Ou verifique se faz parte de um compartilhamento do Resource Access Manager com acesso às tabelas de rotas de gateway local e aos grupos de CoIPs.
- Verifique se os endereços IP em seus grupos de CoIPs podem ser roteados de um gateway local do Outpost para os outros.
- Verifique se os blocos CIDR da VPC (por exemplo, 10.0.0.0/4) e seus blocos CIDR do grupo de CoIPs não contêm endereços IP da Classe E (240.0.0.0/4). O RDS usa esses endereços IP internamente.
- Não se esqueça de configurar corretamente o tráfego de entrada de saída e relacionado.

  O RDS on Outposts estabelece conexão de rede privada virtual (VPN) entre as instâncias de banco de dados primário e em espera. Para que isso funcione corretamente, sua rede local deve permitir tráfego de entrada de saída e relacionado para o Internet Security Association and Key Management Protocol (ISAKMP). Ele faz isso usando a porta 500 do User Datagram Protocol

(UDP) e a porta 4500 do IP Security (IPsec) Network Address Translation Traversal (NAT-T) usando UDP.

Para ter mais informações sobre CoIPs, consulte Endereços IP de propriedade do cliente para o Amazon RDS no AWS Outposts. neste guia e Endereços IP de propriedade do cliente no Guia do usuário do AWS Outposts.

Trabalhar com operações de API para permissões do Amazon EC2

Independentemente de você usar COIPs para sua instância de banco de dados no AWS Outposts, o RDS precisa de acesso aos recursos do grupo de CoIPs. O RDS pode chamar as seguintes operações de API de permissões do EC2 para CoIPs em seu nome para implantações multi-AZ:

- `CreateCoipPoolPermission`: quando você cria uma instância de banco de dados multi-AZ no RDS on Outposts
- `DeleteCoipPoolPermission`: quando você exclui uma instância de banco de dados multi-AZ no RDS on Outposts

Essas operações de API concedem ou removem de contas do RDS internas a permissão para alocar endereços de IP elásticos do grupo de CoIPs especificado pela permissão. Você pode visualizar esses endereços IP usando a operação de API do `DescribeCoipPoolUsage`. Para ter mais informações sobre CoIPs, consulte Endereços IP de propriedade do cliente para o Amazon RDS no AWS Outposts. e Endereços IP de propriedade do cliente no Guia do usuário do AWS Outposts.

O RDS também pode chamar as seguintes operações de API de permissões do EC2 para tabelas de rotas de gateway local em seu nome para implantações multi-AZ:

- `CreateLocalGatewayRouteTablePermission`: quando você cria uma instância de banco de dados multi-AZ no RDS on Outposts
- `DeleteLocalGatewayRouteTablePermission`: quando você exclui uma instância de banco de dados multi-AZ no RDS on Outposts

Essas operações de API concedem ou removem de contas do RDS internas a permissão para associar internas VPCs do RDS às tabelas de rotas de gateway local. Você pode visualizar essas associações de tabelas de rotas e VPC usando as operações de API do `DescribeLocalGatewayRouteTableVpcAssociations`.

Criar instâncias de banco de dados para o Amazon RDS on AWS Outposts

Criar uma instância de banco de dados do Amazon RDS on AWS Outposts é semelhante a criar uma instância de banco de dados Amazon RDS na Nuvem AWS. No entanto, certifique-se de especificar um grupo de sub-redes de banco de dados associado ao seu Outpost.

Uma nuvem privada virtual (VPC) com base no serviço Amazon VPC pode abranger todas as zonas de disponibilidade em uma Região da AWS. É possível estender qualquer VPC na Região da AWS ao Outpost adicionando uma sub-rede do Outpost. Para adicionar uma sub-rede do Outpost a uma VPC, especifique o nome do recurso da Amazon (ARN) do Outpost ao criar a sub-rede.

Antes de criar uma instância de banco de dados do RDS no Outposts, você pode criar um grupo de sub-redes de banco de dados que inclua uma sub-rede associada ao Outpost. Ao criar uma instância de banco de dados do RDS no Outposts, especifique esse grupo de sub-redes de banco de dados. Você também pode optar por criar um grupo de sub-redes de banco de dados ao criar a instância de banco de dados.

Para obter informações sobre como configurar o AWS Outposts, consulte o Guia do usuário do AWS Outposts.

Console

Criar um grupo de sub-redes de banco de dados

Crie um grupo de sub-redes de banco de dados com uma sub-rede associada ao Outpost.

Também é possível criar um grupo de sub-redes de banco de dados para o Outpost ao criar a instância de banco de dados. Se quiser fazer isso, pule este procedimento.

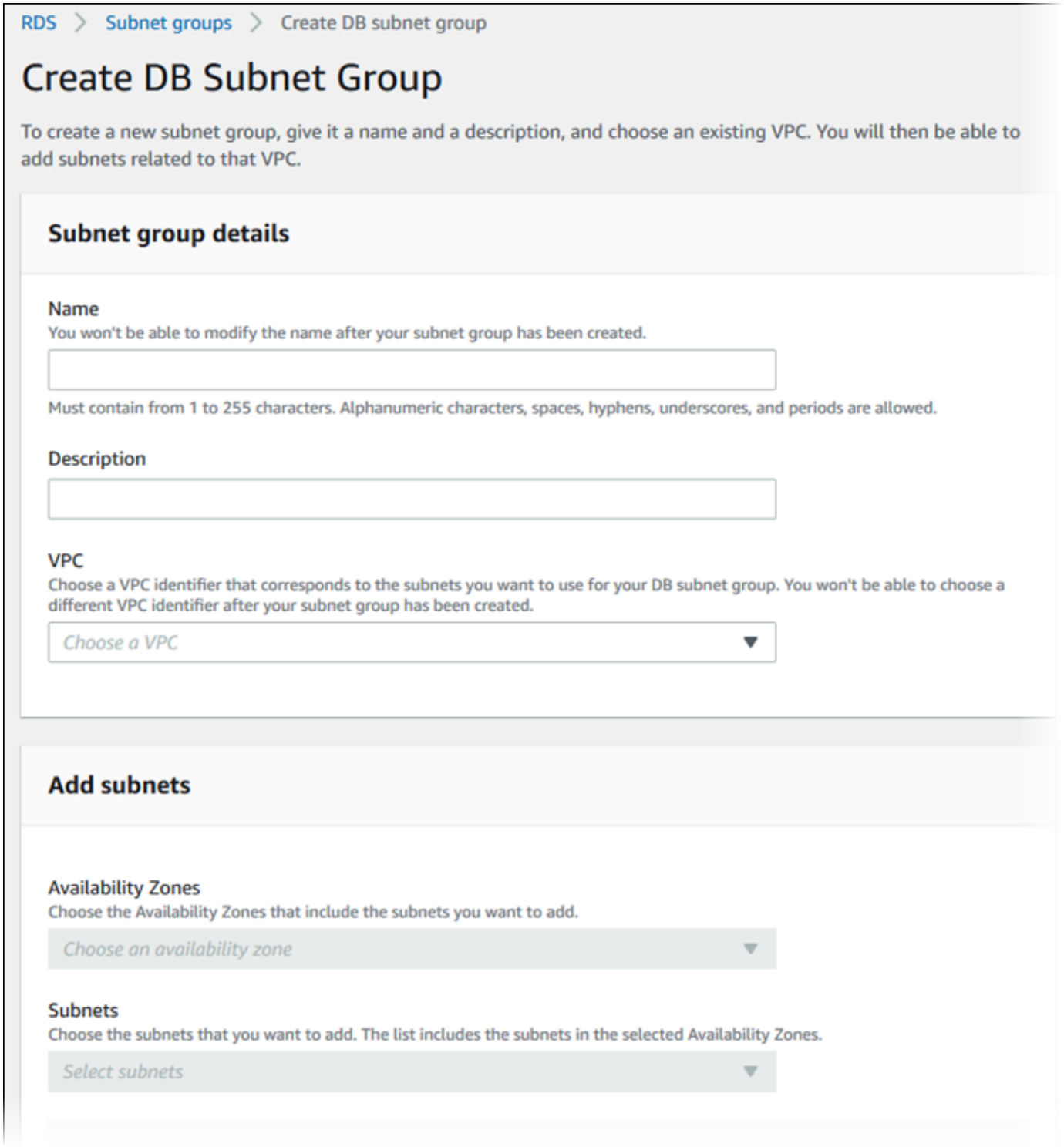
> **Note**
>
> Para criar um grupo de sub-redes de banco de dados para a Nuvem AWS, especifique pelo menos duas sub-redes.

Para criar um grupo de sub-redes de banco de dados para o seu Outpost

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.

2. No canto superior direito do console do Amazon RDS, escolha a Região da AWS na qual você quer criar o grupo de sub-redes de banco de dados.
3. Escolha Subnet groups (Grupos de sub-redes) e Create DB Subnet Group (Criar grupo de sub-redes de banco de dados).

   A página Create DB subnet group (Criar grupo de sub-redes de banco de dados) é exibida.



4. Para Name (Nome), escolha nome do grupo de sub-redes do banco de dados.
5. Para Description (Descrição), escolha uma descrição para o grupo de sub-redes do banco de dados.

6. Para VPC, escolha a VPC para a qual você está criando o grupo de sub-redes do banco de dados.
7. Em Zonas de disponibilidade, selecione a zona de disponibilidade para o Outpost.
8. Em Sub-redes, selecione a sub-rede que será usada pelo RDS no Outposts.
9. Selecione Create (Criar) para criar o grupo de sub-redes de banco de dados.

Criar a instância de banco de dados do RDS on Outposts

Crie a instância de banco de dados e selecione o Outpost para a instância de banco de dados.

Para criar uma instância de banco de dados do RDS no Outposts usando o console

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No canto superior direito do console do Amazon RDS, escolha a Região da AWS onde o Outpost no qual você quer criar a instância de banco de dados está anexado.
3. No painel de navegação, escolha Databases (Bancos de dados).
4. Escolha Create database (Criar banco de dados).

   O AWS Management Console detecta Outposts disponíveis que você configurou e apresenta a opção No local na seção Local do banco de dados.

   > **Note**
   >
   > Se você não configurou nenhum Outpost, a seção Local do banco de dados não será exibida ou a opção RDS no Outposts não estará disponível na seção Escolher um método de criação local.

5. Para Database location (Localização do banco de dados), escolha On-premises.
6. Para On-premises creation method, (Método de criação no local), escolha RDS on Outposts.
7. Especifique as configurações para Outposts Connectivity (Conectividade com o Outposts). Essas configurações são para o Outpost que usa a VPC com o grupo de sub-redes de banco de dados da sua instância de banco de dados. Essa VPC deve se basear no serviço Amazon VPC.

   a. Para Virtual Private Cloud (VPC), escolha a VPC que contém o grupo de sub-redes de banco de dados para a sua instância de banco de dados.

b. Para VPC grupo de segurança (Grupo de segurança da VPC), escolha o grupo de segurança da Amazon VPC para a sua instância de banco de dados.

c. Para DB subnet group (Grupo de sub-redes do banco de dados), escolha o grupo de sub-redes de banco de dados para a sua instância de banco de dados.

É possível escolher um grupo de sub-redes de banco de dados existente associado ao Outpost, por exemplo, se tiver realizando o procedimento em Criar um grupo de sub-redes de banco de dados.

Também é possível criar um grupo de sub-redes de banco de dados para o Outpost.

8. Em Multi-AZ deployment (Implantação multi-AZ), escolha Create a standby instance (recommended for production usage) (Criar uma instância em espera (recomendada para uso em produção) para criar uma instância de banco de dados em espera em outro Outpost.

> **Note**
>
> Essa opção não está disponível para o Microsoft SQL Server.
> Se você optar por criar uma implantação multi-AZ, não poderá armazenar backups no Outpost.

9. Em Backup, faça o seguinte:

a. Para Backup target (Destino de backup), escolha uma das seguintes opções:

- Nuvem AWS para armazenar backups automatizados e snapshots manuais na Região da AWS principal.
- Outposts (on-premises) para criar backups locais.

> **Note**
>
> Para armazenar backups no seu Outpost, este deve ter o recurso Amazon S3. Para obter mais informações, consulte Amazon S3 on Outposts.
> Não há compatibilidade para backups locais em implantações multi-AZ nem réplicas de leitura.

b. Escolha Enable automated backups (Habilitar backups automatizados) para criar snapshots pontuais da sua instância de banco de dados.

Se você ativar backups automatizados, poderá escolher valores para Backup retention period (Período de retenção de backup) e Backup window (Janela de backup) ou deixar os valores padrão.

10. Especifique outras configurações de instância de banco de dados, conforme necessário.

    Para obter mais informações sobre cada configuração ao criar uma instância de banco de dados, consulte Configurações para instâncias de banco de dados.

11. Escolha Create database (Criar banco de dados).

    O painel Databases (Bancos de dados) é exibido. Um banner informa que a sua instância de banco de dados está sendo criada e mostra o botão View credential details (Visualizar detalhes de credenciais).

Visualizar detalhes da instância de banco de dados

Depois de criar sua instância de banco de dados, você pode visualizar credenciais e outros detalhes dela.

Para visualizar detalhes da instância de banco de dados

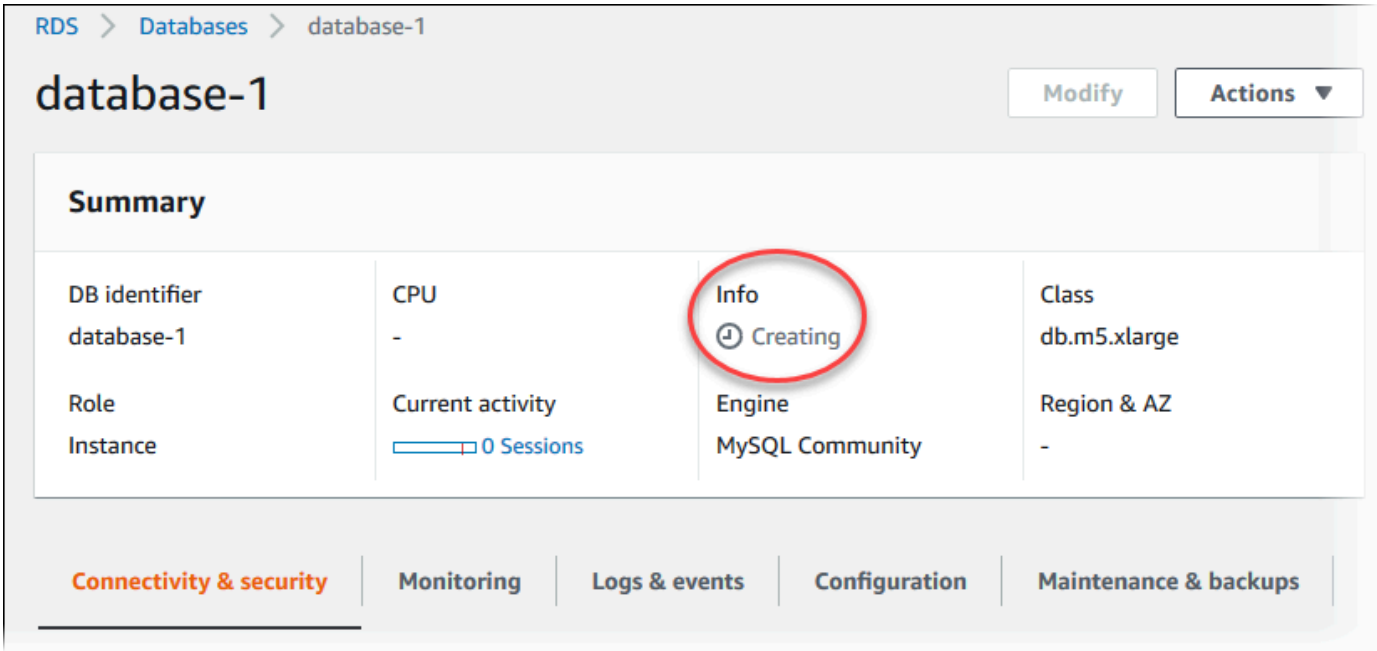
1. Para visualizar o nome de usuário primário e a senha da instância de banco de dados, escolha View credential details (Visualizar detalhes de credenciais), na página Databases (Bancos de dados).

   Você pode se conectar à instância de banco de dados como o usuário primário usando essas credenciais.

   > **Important**
   >
   > Não é possível visualizar a senha do usuário mestre novamente. Caso você não a registre, talvez seja necessário alterá-la. Para alterar a senha do usuário mestre depois que a instância do banco de dados estiver disponível, modifique a instância de banco de dados. Para obter mais informações sobre a modificação de uma instância de banco de dados , consulte Modificar uma instância de banco de dados do Amazon RDS.

2. Escolha o nome da nova instância de banco de dados na página Databases (Bancos de dados).

No console do RDS, os detalhes para a nova instância de banco de dados são exibidos. A instância de banco de dados fica com o status Creating (Criando) até que esteja criada e pronta para uso. Quando o status muda para Available (Disponível), você pode se conectar à instância de banco de dados. Dependendo da classe da instância de banco de dados e do armazenamento alocado, pode levar alguns minutos até que a nova instância de banco de dados fique disponível.



Quando a instância de banco de dados estiver disponível, será possível gerenciá-la da mesma maneira que você gerencia as instâncias de banco de dados do RDS na Nuvem AWS.

AWS CLI

Antes de criar uma nova instância de banco de dados em um Outpost com a AWS CLI, crie um grupo de sub-redes de banco de dados para uso pelo RDS on Outposts.

Para criar um grupo de sub-redes de banco de dados para o seu Outpost

- Use o comando create-db-subnet-group. Em `--subnet-ids`, especifique o grupo de sub-redes no Outpost que será usado pelo RDS no Outposts.

  Para Linux, macOS ou Unix:

```
aws rds create-db-subnet-group \
```

```
    --db-subnet-group-name myoutpostdbsubnetgr \
    --db-subnet-group-description "DB subnet group for RDS on Outposts" \
    --subnet-ids subnet-abc123
```

Para Windows:

```
aws rds create-db-subnet-group ^
    --db-subnet-group-name myoutpostdbsubnetgr ^
    --db-subnet-group-description "DB subnet group for RDS on Outposts" ^
    --subnet-ids subnet-abc123
```

Para criar uma instância de banco de dados do RDS on Outposts utilizando a AWS CLI

- Use o comando create-db-instance. Especifique uma zona de disponibilidade para o Outpost, um grupo de segurança da Amazon VPC associado ao Outpost e o grupo de sub-redes de banco de dados criado para o Outpost. É possível incluir as seguintes opções:
  - `--db-instance-identifier`
  - `--db-instance-class`
  - `--engine` – o mecanismo de banco de dados. Use um dos seguintes valores:
    - MySQL – especifique `mysql`.
    - PostgreSQL – especifique `postgres`.
    - Microsoft SQL Server: especifique `sqlserver-ee`, `sqlserver-se` ou `sqlserver-web`.
  - `--availability-zone`
  - `--vpc-security-group-ids`
  - `--db-subnet-group-name`
  - `--allocated-storage`
  - `--max-allocated-storage`
  - `--master-username`
  - `--master-user-password`
  - `--multi-az | --no-multi-az`: (opcional) se deverá ou não ser criada uma instância de banco de dados em espera em uma zona de disponibilidade diferente. O padrão é `--no-multi-az`.

    A opção `--multi-az` não está disponível para o SQL Server.

- `--backup-retention-period`
- `--backup-target` – (opcional) onde armazenar backups automatizados e snapshots manuais. Use um dos seguintes valores:
  - `outposts` – armazene-os localmente no seu Outpost.
  - `region`: armazene-os na Região da AWS principal. Este é o valor padrão.

  Se você usar a opção `--multi-az`, não poderá usar o `outposts` para `--backup-target`. Além disso, a instância de banco de dados não poderá ter réplicas de leitura se você usar `outposts` para `--backup-target`.
- `--storage-encrypted`
- `--kms-key-id`

Example

O seguinte exemplo cria uma instância de banco de dados MySQL denominada `myoutpostdbinstance` com backups armazenados no seu Outpost.

Para Linux, macOS ou Unix:

```
aws rds create-db-instance \
    --db-instance-identifier myoutpostdbinstance \
    --engine-version 8.0.17 \
    --db-instance-class db.m5.large \
    --engine mysql \
    --availability-zone us-east-1d \
    --vpc-security-group-ids outpost-sg \
    --db-subnet-group-name myoutpostdbsubnetgr \
    --allocated-storage 100 \
    --max-allocated-storage 1000 \
    --master-username masterawsuser \
    --manage-master-user-password \
    --backup-retention-period 3 \
    --backup-target outposts \
    --storage-encrypted \
    --kms-key-id mykey
```

Para Windows:

```
aws rds create-db-instance ^
```

```
    --db-instance-identifier myoutpostdbinstance ^
    --engine-version 8.0.17 ^
    --db-instance-class db.m5.large ^
    --engine mysql ^
    --availability-zone us-east-1d ^
    --vpc-security-group-ids outpost-sg ^
    --db-subnet-group-name myoutpostdbsubnetgr ^
    --allocated-storage 100 ^
    --max-allocated-storage 1000 ^
    --master-username masterawsuser ^
    --manage-master-user-password ^
    --backup-retention-period 3 ^
    --backup-target outposts ^
    --storage-encrypted ^
    --kms-key-id mykey
```

Para obter mais informações sobre cada configuração ao criar uma instância de banco de dados, consulte Configurações para instâncias de banco de dados.

API do RDS

Para criar uma instância de banco de dados em um Outpost com a API do RDS, primeiro crie um grupo de sub-redes de banco de dados que será usado pelo RDS no Outposts chamando a operação CreateDBSubnetGroup. Em `SubnetIds`, especifique o grupo de sub-redes no Outpost que será usado pelo RDS no Outposts.

Depois, chame a operação CreateDBInstance com os seguintes parâmetros. Especifique uma zona de disponibilidade para o Outpost, um grupo de segurança da Amazon VPC associado ao Outpost e o grupo de sub-redes de banco de dados criado para o Outpost.

- `AllocatedStorage`
- `AvailabilityZone`
- `BackupRetentionPeriod`
- `BackupTarget`

  Se você estiver criando uma implantação de instância de banco de dados multi-AZ, não poderá usar `outposts` para `BackupTarget`. Além disso, a instância de banco de dados não poderá ter réplicas de leitura se você usar `outposts` para `BackupTarget`.
- `DBInstanceClass`
- `DBInstanceIdentifier`

- `VpcSecurityGroupIds`
- `DBSubnetGroupName`
- `Engine`
- `EngineVersion`
- `MasterUsername`
- `MasterUserPassword`
- `MaxAllocatedStorage` (opcional)
- `MultiAZ` (opcional)
- `StorageEncrypted`
- `KmsKeyID`

Para obter mais informações sobre cada configuração ao criar uma instância de banco de dados, consulte Configurações para instâncias de banco de dados.

Criar réplicas de leitura do Amazon RDS no AWS Outposts

O Amazon RDS no AWS Outposts utiliza a funcionalidade de replicação integrada nos mecanismos de banco de dados MySQL e PostgreSQL para criar uma réplica de leitura de uma instância de banco de dados de origem. A instância de banco de dados de origem se torna a instância de banco de dados primária. As atualizações feitas à instância de banco de dados primária são copiadas de forma assíncrona na réplica de leitura. É possível reduzir a carga na instância de banco de dados primária roteando as consultas de leitura dos aplicativos para a réplica de leitura. Com o uso de réplicas de leitura, é possível expandir de maneira elástica para além das limitações de capacidade de uma única instância de banco de dados para workloads de banco de dados com muita leitura.

Quando você cria uma réplica de leitura de uma instância de banco de dados do RDS no Outposts, a réplica de leitura pode usar um endereço IP (CoIP) de propriedade do cliente. Para ter mais informações, consulte Endereços IP de propriedade do cliente para o Amazon RDS no AWS Outposts..

As réplicas de leitura no RDS no Outposts têm as seguintes limitações:

- Não é possível criar réplicas de leitura no RDS para SQL Server em instâncias de banco de dados do RDS no Outposts.
- As réplicas de leitura entre regiões não são compatíveis com o RDS no Outposts.
- As réplicas de leitura em cascata não são compatíveis com o RDS no Outposts.
- O RDS de origem na instância de banco de dados do Outposts não pode ter backups locais. O destino de backup da instância de banco de dados de origem deve ser sua Região da AWS.
- Réplicas de leitura exigem grupos de IPs de propriedade do cliente (CoIP). Para ter mais informações, consulte Endereços IP de propriedade do cliente para o Amazon RDS no AWS Outposts..
- As réplicas de leitura no RDS no Outposts só podem ser criadas na mesma nuvem privada virtual (VPC) que a instância de banco de dados de origem.
- As réplicas de leitura no RDS no Outposts podem estar localizadas no mesmo Outpost ou em outro Outpost na mesma VPC da instância de banco de dados de origem.
- Não é possível criar réplicas de leitura para instâncias de banco de dados criptografadas com AWS KMS External Key Store (XKS).

Você pode criar uma réplica de leitura a partir de uma instância de banco de dados do RDS no Outposts usando o AWS Management Console, a AWS CLI ou a API do RDS. Para ter mais

informações sobre réplicas de leitura, consulte Trabalhar com réplicas de leitura de instância de banco de dados.

Console

Para criar uma réplica de leitura a partir de uma instância de banco de dados de origem

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Databases (Bancos de dados).
3. Escolha a instância de banco de dados que você deseja usar como a origem da réplica de leitura.
4. Em Actions (Ações), escolha Create read replica (Criar réplica de leitura).
5. Em DB instance identifier (Identificador da instância de banco de dados), insira um nome para a réplica de leitura.
6. Especifique as configurações para Outposts Connectivity (Conectividade com o Outposts). Essas configurações são para o Outpost que usa a nuvem privada virtual (VPC) com o grupo de sub-redes de banco de dados de sua instância de banco de dados. Essa VPC deve se basear no serviço Amazon VPC.
7. Selecione sua DB instance class (classe de instância de banco de dados). Recomendamos que você use a classe de instância de banco de dados e o tipo de armazenamento iguais ou superiores aos da instância de banco de dados de origem para a réplica de leitura.
8. Em Multi-AZ deployment (Implantação multi-AZ), selecione Create a standby instance (recommended for production usage) (Criar uma instância em espera (recomendada para uso em produção) para criar uma instância de banco de dados em espera em outra zona de disponibilidade.

   Você pode criar a réplica de leitura como uma instância de banco de dados multi-AZ independentemente de o banco de dados de origem ser ou não uma instância de banco de dados multi-AZ.
9. (Opcional) Em Connectivity (Conectividade), defina valores para Subnet Group (Grupo de sub-redes) e Availability Zone (Zona de disponibilidade).

   Se você especificar valores para Subnet Group (Grupo de sub-redes) e Availability Zone (Zona de disponibilidade), a réplica de leitura será criada em um Outpost associado à zona de disponibilidade no grupo de sub-redes do banco de dados.

Se você especificar um valor para Subnet Group (Grupo de sub-redes) e No preference (Nenhuma preferência) para Availability Zone (Zona de disponibilidade), a réplica de leitura será criada em um Outpost aleatório no grupo de sub-redes do banco de dados.

10. Em AWS KMS key, escolha o identificador da AWS KMS key da chave do KMS.

    A réplica de leitura deve ser criptografada.
11. Selecione outras opções conforme necessário.
12. Escolha Create read replica (Criar réplica de leitura).

Depois que a réplica de leitura for criada, você poderá vê-la na página Databases (Bancos de dados) no console do RDS. Ele mostra Replica (Réplica) na coluna Role (Função).

AWS CLI

Para criar uma réplica de leitura de uma instância de banco de dados de origem do MySQL ou do PostgreSQL, use o comando create-db-instance-read-replica da AWS CLI.

Você pode controlar onde a réplica de leitura é criada especificando as opções `--db-subnet-group-name` e `--availability-zone`:

- Se você especificar as opções `--db-subnet-group-name` e `--availability-zone`, a réplica de leitura será criada em um Outpost associado à zona de disponibilidade no grupo de sub-redes do banco de dados.
- Se você especificar a opção `--db-subnet-group-name` e não especificar a opção `--availability-zone`, a réplica de leitura será criada em um Outpost aleatório no grupo de sub-redes do banco de dados.
- Se você não especificar nenhuma das opções, a réplica de leitura será criada no mesmo Outpost que a instância de banco de dados de origem do RDS no Outposts.

O exemplo a seguir cria uma réplica e especifica a localização da réplica de leitura por meio da inclusão das opções `--db-subnet-group-name` e `--availability-zone`.

Example

Para Linux, macOS ou Unix:

```
aws rds create-db-instance-read-replica \
```

```
    --db-instance-identifier myreadreplica \
    --source-db-instance-identifier mydbinstance \
    --availability-zone us-west-2a
```

Para Windows:

```
aws rds create-db-instance-read-replica ^
    --db-instance-identifier myreadreplica ^
    --source-db-instance-identifier mydbinstance ^
    --availability-zone us-west-2a
```

API do RDS

Para criar uma réplica de leitura a partir de uma instância de banco de dados MySQL ou PostgreSQL, chame a operação CreateDBInstanceReadReplica da API do Amazon RDS com os seguintes parâmetros obrigatórios:

- `DBInstanceIdentifier`
- `SourceDBInstanceIdentifier`

Você pode controlar onde a réplica de leitura é criada especificando os parâmetros `DBSubnetGroupName` e `AvailabilityZone`:

- Se você especificar os parâmetros `DBSubnetGroupName` e `AvailabilityZone`, a réplica de leitura será criada em um Outpost associado à zona de disponibilidade no grupo de sub-redes do banco de dados.
- Se você especificar o parâmetro `DBSubnetGroupName` e não especificar o parâmetro `AvailabilityZone`, a réplica de leitura será criada em um Outpost aleatório no grupo de sub-redes do banco de dados.
- Se você não especificar nenhum dos parâmetros, a réplica de leitura será criada no mesmo Outpost que a instância de banco de dados de origem do RDS no Outposts.

Considerações para restaurar instâncias de banco de dados no Amazon RDS on AWS Outposts

Ao restaurar uma instância de banco de dados no Amazon RDS on AWS Outposts, você pode escolher o local de armazenamento para backups automatizados e snapshots manuais da instância de banco de dados restaurada.

- Ao restaurar a partir de um snapshot do banco de dados manual, é possível armazenar backups na Região da AWS principal ou localmente no seu Outpost.
- Ao restaurar a partir de um backup automatizado (recuperação em um ponto anterior no tempo), você tem menos opções:
  - Se estiver restaurando da Região da AWS principal, você poderá armazenar backups na Região da AWS ou no seu Outpost.
  - Se estiver restaurando do seu Outpost, você poderá armazenar backups somente nesse Outpost.

Amazon RDS Proxy

Com o proxy do Amazon RDS, você pode permitir que suas aplicações agrupem e compartilhem conexões de banco de dados para melhorar sua capacidade de escala. O proxy do RDS torna as aplicações mais resilientes a falhas de banco de dados conectando-se automaticamente a uma instância de banco de dados em espera e preservando as conexões de aplicações. Ao usar o RDS Proxy, você também pode impor a autenticação do AWS Identity and Access Management (IAM) para bancos de dados e armazenar credenciais com segurança no AWS Secrets Manager.

Com o RDS Proxy, você pode lidar com picos imprevisíveis no tráfego de banco de dados. Caso contrário, esses picos podem causar problemas devido a conexões com excesso de assinaturas ou à criação rápida de conexões. O RDS Proxy estabelece um grupo de conexões de banco de dados e reutiliza conexões nesse grupo. Essa abordagem evita sobrecarregar a memória e a CPU de abrir uma nova conexão de banco de dados todas as vezes. Para proteger um banco de dados contra o excesso de assinaturas, é possível controlar o número de conexões do banco de dados criadas.

O RDS Proxy coloca na fila ou limita as conexões das aplicações que não podem ser atendidas imediatamente do grupo de conexões. Embora as latências possam aumentar, sua aplicação pode continuar a escalar sem falhar abruptamente ou sobrecarregar o banco de dados. Se as solicitações de conexão excederem os limites especificados, o proxy do RDS rejeitará as conexões de aplicações (ou seja, grandes quantidades de carga). Ao mesmo tempo, ele mantém uma performance previsível para a carga que pode ser atendida pelo RDS com a capacidade disponível.

Você pode reduzir a sobrecarga para processar credenciais e estabelecer uma conexão segura para cada nova conexão. O proxy do RDS pode lidar com parte desse trabalho em nome do banco de dados.

O proxy do RDS é totalmente compatível com as versões dos mecanismos com os quais é compatível. É possível habilitar o proxy do RDS para a maioria das aplicações sem alterações de código.

Tópicos

- Disponibilidade de região e versão
- Cotas e limitações do RDS Proxy
- Planejar onde usar o RDS Proxy
- Conceitos e terminologia do RDS Proxy

- Conceitos básicos do RDS Proxy
- Gerenciar um RDS Proxy
- Como trabalhar com endpoints do proxy do Amazon RDS
- Monitorar métricas do proxy do RDS com o Amazon CloudWatch
- Trabalhar com eventos do RDS Proxy
- Solução de problemas do RDS Proxy
- Usar o proxy do RDS com o AWS CloudFormation

Disponibilidade de região e versão

A disponibilidade e a compatibilidade de recursos variam entre versões específicas de cada mecanismo de banco de dados e entre Regiões da AWS. Para ter mais informações sobre a disponibilidade de versões e regiões do Amazon RDS com o RDS Proxy, consulte Regiões e mecanismos de banco de dados compatíveis com o Amazon RDS Proxy.

Cotas e limitações do RDS Proxy

As seguintes limitações aplicam-se ao RDS Proxy:

- Cada ID de Conta da AWS é limitado a 20 proxies. Se a aplicação exigir mais proxies, solicite um aumento na página Service Quotas no AWS Management Console. Na página Service Quotas, selecione Amazon Relational Database Service (Amazon RDS) e localize Proxies para solicitar um aumento de cota. A AWS pode aumentar automaticamente a cota ou aguardar a análise da solicitação pelo Suporte.
- Cada proxy pode ter até 200 segredos associados do Secrets Manager. Assim, cada proxy pode se conectar com até 200 contas de usuário diferentes em qualquer momento.
- Cada proxy tem um endpoint padrão. Também é possível adicionar até vinte endpoints a cada proxy. É possível criar, visualizar, modificar e excluir esses endpoints.
- Para instâncias de banco de dados do RDS em configurações de replicação, você pode associar um proxy somente à instância de banco de dados do gravador, não a uma réplica de leitura.
- O proxy do RDS deve estar na mesma nuvem privada virtual (VPC) que o banco de dados. O proxy não pode ser acessível publicamente, embora o banco de dados sim. Por exemplo, se você estiver fazendo protótipos do banco de dados em um host local, não poderá se conectar ao proxy,

a menos que você configure os requisitos de rede necessários para permitir a conexão com o proxy. O motivo disso é que o host local está fora da VPC do proxy.

- Não é possível usar o proxy do RDS com uma VPC cuja locação está definida como `dedicated`.
- Se você usar o RDS Proxy com uma instância de banco de dados do RDS com a autenticação do IAM habilitada, confira a autenticação do usuário. Os usuários que se conectam por meio de um proxy devem ser autenticados por credenciais de login. Para obter detalhes sobre o suporte do Secrets Manager e do IAM no RDS Proxy, consulte Configurar credenciais de banco de dados no AWS Secrets Manager para RDS Proxy e Configurar políticas do AWS Identity and Access Management (IAM) para o RDS Proxy.
- Você não pode usar o RDS Proxy com DNS personalizado ao usar a validação do nome de host SSL.
- Cada proxy pode ser associado a uma única instância de banco de dados de destino. No entanto, é possível associar vários proxies à mesma instância de banco de dados.
- Qualquer instrução com um tamanho de texto maior do que 16 KB faz com que o proxy fixe a sessão à conexão atual.
- Certas regiões têm restrições de zona de disponibilidade (AZ) a serem consideradas ao criar seu proxy. A região Leste dos EUA (Norte da Virgínia) não é compatível com o RDS Proxy na zona de `use1-az3` disponibilidade. A região Oeste dos EUA (Norte da Califórnia) não é compatível com o RDS Proxy na zona de `usw1-az2` disponibilidade. Ao selecionar sub-redes ao criar seu proxy, certifique-se de não selecionar sub-redes nas zonas de disponibilidade mencionadas acima.
- No momento, o RDS Proxy não comporta nenhuma chave de contexto de condição global.

  Para obter mais informações sobre chaves de contexto de condição global, consulte Chaves de contexto de condição global AWS no Guia do usuário do IAM.
- Não é possível usar o RDS Proxy com o RDS Custom para SQL Server.
- Para refletir qualquer modificação do grupo de parâmetros do banco de dados no proxy, é necessária uma reinicialização da instância, mesmo que você opte por aplicar as alterações imediatamente. Com relação a parâmetros no nível do cluster, é necessária uma reinicialização em todo o cluster.
- Seu proxy cria automaticamente o usuário do banco de dados `rdsproxyadmin` quando você registra um destino de proxy. Este é um usuário protegido que é essencial para a funcionalidade de proxy. Você deve evitar alterar o usuário `rdsproxyadmin` em qualquer função. A exclusão ou modificação do usuário `rdsproxyadmin` ou das respectivas permissões pode afetar a disponibilidade do proxy para sua aplicação.

Para saber as limitações adicionais para cada mecanismo de banco de dados, consulte as seguintes seções:

- Limitações adicionais do RDS para MariaDB
- Limitações adicionais do RDS para Microsoft SQL Server
- Limitações adicionais do RDS para MySQL
- Limitações adicionais do RDS para PostgreSQL

Limitações adicionais do RDS para MariaDB

As seguintes limitações adicionais se aplicam ao RDS Proxy com bancos de dados do RDS para MariaDB:

- No momento, todos os proxies escutam na porta 3306 para o MariaDB. Os proxies ainda se conectam ao banco de dados usando a porta especificada nas configurações do banco de dados.
- Não é possível usar o RDS Proxy com bancos de dados do MariaDB autogerenciados em instâncias do Amazon EC2.
- Você não pode usar o proxy do RDS com uma instância de banco de dados do RDS para MariaDB que tenha o parâmetro `read_only` em seu grupo de parâmetros de banco de dados definido como 1.
- O RDS Proxy não é compatível com o modo compactado do MariaDB. Por exemplo, ele não é compatível com a compactação usada pelas opções `--compress` ou `-C` do comando `mysql`.
- Algumas funções e instruções SQL podem alterar o estado da conexão sem causar fixação. Para obter o comportamento de fixação mais atual, consulte Evitar a fixação de um RDS Proxy.
- O RDS Proxy não é compatível com o plug-in `auth_ed25519` do Memcached.
- O RDS Proxy não é compatível com Transport Layer Security (TLS) versão 1.3 para bancos de dados MariaDB.
- Conexões de banco de dados que processam um comando `GET DIAGNOSTIC` podem retornar informações imprecisas quando o RDS Proxy reutiliza a mesma conexão de banco de dados para executar outra consulta. Isso pode acontecer quando o RDS Proxy realiza multiplexação de conexões de banco de dados. Para obter mais informações, consulte Visão geral dos conceitos do RDS Proxy.
- Atualmente, o RDS Proxy não é compatível com a opção `caching_sha2_password` para `ClientPasswordAuthType` para MariaDB.

> **Important**
>
> Para proxies associados a bancos de dados MariaDB, não defina o parâmetro de configuração `sql_auto_is_null` como `true` ou um valor diferente de zero na consulta de inicialização. Isso pode causar um comportamento incorreto da aplicação.

Limitações adicionais do RDS para Microsoft SQL Server

As seguintes limitações adicionais se aplicam ao RDS Proxy com RDS para bancos de dados do Microsoft SQL Server:

- o número de segredos do Secrets Manager que você precisa criar para um proxy depende do agrupamento que sua instância de banco de dados usa. Por exemplo, suponha que sua instância de banco de dados use agrupamento que diferencie maiúsculas de minúsculas. Se sua aplicação aceitar “Admin” e “admin”, seu proxy precisará de dois segredos separados. Para ter mais informações sobre agrupamento no SQL Server, consulte a documentação do Microsoft SQL Server.
- O RDS Proxy não é compatível com conexões que usam o Active Directory.
- Você não pode usar a autenticação do IAM com clientes que não sejam compatíveis com as propriedades do token. Para ter mais informações, consulte Considerações sobre como se conectar a um proxy com o Microsoft SQL Server.
- Os resultados de `@@IDENTITY`, `@@ROWCOUNT` e `SCOPE_IDENTITY` nem sempre são precisos. Como solução alternativa, recupere seus valores na mesma declaração de sessão para garantir que eles retornem as informações corretas.
- Se a conexão usar vários conjuntos de resultados ativos (MARS), o RDS Proxy não realizará as consultas de inicialização. Para obter informações sobre MARS, consulte a documentação do Microsoft SQL Server.
- No momento, o RDS Proxy não é compatível com instâncias de banco de dados do RDS para SQL Server executadas na versão principal do SQL Server 2022.
- O proxy do RDS não é compatível com instâncias de banco de dados do RDS para SQL Server que são executadas na versão principal do SQL Server 2014.
- O RDS Proxy não comporta aplicações clientes que não conseguem lidar com várias mensagens de resposta em um registro do TLS.

Limitações adicionais do RDS para MySQL

As seguintes limitações adicionais se aplicam ao RDS Proxy com bancos de dados do RDS para MySQL:

- O suporte do RDS Proxy para autenticação `caching_sha2_password` requer uma conexão segura (TLS).
- Sabe-se que o suporte do RDS Proxy para `caching_sha2_password` tem problemas de compatibilidade com determinadas versões do driver go-sql.
- Ao usar o driver MySQL 8.4 C, a API `mysql_stmt_bind_named_param` poderá formar pacotes malformados se a contagem de parâmetros exceder a contagem de espaços reservados em instruções preparadas. Isso resulta em respostas incorretas. Para obter mais informações, consulte Relatório de bugs do MySQL.
- No momento, todos os proxies escutam na porta 3306 para o MySQL. Os proxies ainda se conectam ao banco de dados usando a porta especificada nas configurações do banco de dados.
- Não é possível usar o proxy do RDS com bancos de dados MySQL autogerenciados em instâncias do EC2.
- Você não pode usar o proxy do RDS com uma instância de banco de dados do RDS para MySQL que tenha o parâmetro `read_only` em seu grupo de parâmetros de banco de dados definido como `1`.
- O proxy do RDS não é compatível com o modo compactado do MySQL. Por exemplo, ele não é compatível com a compactação usada pelas opções `--compress` ou `-C` do comando `mysql`.
- Conexões de banco de dados que processam um comando `GET DIAGNOSTIC` podem retornar informações imprecisas quando o RDS Proxy reutiliza a mesma conexão de banco de dados para executar outra consulta. Isso pode acontecer quando o RDS Proxy realiza multiplexação de conexões de banco de dados.
- Algumas funções e declarações SQL, como `SET LOCAL`, podem alterar o estado da conexão sem causar fixação. Para obter o comportamento de fixação mais atual, consulte Evitar a fixação de um RDS Proxy.
- Usar a função `ROW_COUNT()` em uma consulta com várias declarações não é aceito.
- O RDS Proxy não comporta aplicações clientes que não conseguem lidar com várias mensagens de resposta em um registro do TLS.
- O RDS Proxy não aceita senhas duplas do MySQL.

> **Important**
>
> Para proxies associados a bancos de dados MySQL, não defina o parâmetro de configuração `sql_auto_is_null` como `true` ou um valor diferente de zero na consulta de inicialização. Isso pode causar um comportamento incorreto da aplicação.

Limitações adicionais do RDS para PostgreSQL

As seguintes limitações adicionais se aplicam ao RDS Proxy com bancos de dados do RDS para PostgreSQL:

- O proxy do RDS não é compatível com os filtros de fixação de sessão do PostgreSQL.
- Atualmente, todos os proxies escutam na porta 5432 para o PostgreSQL.
- No PostgreSQL, o proxy do RDS atualmente não é compatível com o cancelamento de uma consulta de um cliente emitindo um `CancelRequest`. Esse é o caso, por exemplo, quando você cancela uma consulta de longa duração em uma sessão psql interativa usando Ctrl+C.
- Os resultados da função do PostgreSQL lastval nem sempre são precisos. Como uma solução alternativa, use a instrução INSERT com a cláusula `RETURNING`.
- Atualmente, o RDS Proxy não é compatível com o modo de replicação de streaming.
- Com o RDS para PostgreSQL 16, as modificações no valor `scram_iterations` afetam exclusivamente o processo de autenticação entre o proxy e o banco de dados. Especificamente, se você configurar `ClientPasswordAuthType` como `scram-sha-256`, nenhuma personalização feita no valor `scram_iterations` influenciará a autenticação de senha do cliente para o proxy. Em vez disso, o valor de iteração para autenticação de senha do cliente para proxy é fixado em 4096.
- O banco de dados `default` deve existir.
- Se você usar `ALTER ROLE` para alterar o perfil do usuário com `SET ROLE`, as conexões subsequentes desse usuário com o proxy talvez não usem essa configuração de perfil, se essas conexões encontrarem fixação. Para evitar isso, ao usar o proxy, use `SET ROLE` na consulta de inicialização do proxy. Para obter mais informações, consulte Consulta de inicialização no Criar um proxy para o Amazon RDS.

> **Important**
>
> Para proxies existentes com bancos de dados PostgreSQL, se você modificar a autenticação do banco de dados para usar apenas `SCRAM`, o proxy ficará indisponível por até 60 segundos. Para evitar o problema, faça o seguinte:
>
> - O banco de dados deve permitir tanto a autenticação `SCRAM` quanto a `MD5`.
> - Para usar somente a autenticação `SCRAM`, crie um proxy, migre o tráfego da aplicação para o novo proxy e exclua o proxy anteriormente associado ao banco de dados.

Planejar onde usar o RDS Proxy

Você pode determinar quais de suas instâncias de banco de dados, clusters e aplicações podem se beneficiar mais com o uso do RDS Proxy. Para fazer isso, considere estes fatores:

- Qualquer instância de banco de dados que encontrar erros de “conexões em excesso” é uma boa candidata para associação a um proxy. Isso geralmente é caracterizado por um alto valor da métrica `ConnectionAttempts` do CloudWatch. O proxy permite que as aplicações abram muitas conexões de cliente enquanto gerencia um número menor de conexões de longa duração para a instância de banco de dados.
- Para instâncias de bancos de dados que usam classes de instância menores da AWS, como T2 ou T3, o uso de um proxy pode ajudar a evitar condições de falta de memória. Isso também pode ajudar a reduzir a sobrecarga da CPU para estabelecer conexões. Essas condições podem ocorrer ao lidar com um grande número de conexões.
- É possível monitorar certas métricas do Amazon CloudWatch para determinar se uma instância de banco de dados está se aproximando de certos tipos de limite. Esses limites são para o número de conexões e para a memória associada ao gerenciamento de conexão. Você também pode monitorar certas métricas do CloudWatch para determinar se uma instância de banco de dados está lidando com muitas conexões de curta duração. A abertura e o fechamento dessas conexões podem aplicar um sobrecarga de performance a seu banco de dados. Para obter informações sobre as métricas a serem monitoradas, consulte Monitorar métricas do proxy do RDS com o Amazon CloudWatch.
- As funções do AWS Lambda também podem ser boas candidatas para o uso de um proxy. Essas funções fazem conexões curtas frequentes com o banco de dados que se beneficiam do grupo de conexões oferecido pelo RDS Proxy. É possível aproveitar qualquer autenticação do IAM que você

já tenha para funções do Lambda, em vez de gerenciar credenciais de banco de dados em seu código de aplicação do Lambda.

- Essas aplicações que geralmente abrem e encerram um grande número de conexões de banco de dados e não têm mecanismos internos de agrupamento de conexões são ideais para usar um proxy.
- As aplicações que mantêm um grande número de conexões abertas por longos períodos geralmente são boas candidatas para o uso de um proxy. As aplicações em setores como software como serviço (SaaS) ou comércio eletrônico geralmente minimizam a latência de solicitações de banco de dados deixando as conexões abertas. Com o RDS Proxy, uma aplicação pode manter mais conexões abertas do que quando se conecta diretamente à instância de banco de dados.
- Talvez você não tenha adotado a autenticação do IAM e do Secrets Manager devido à complexidade da configuração dessa autenticação para todas as instâncias de banco de dados. Nesse caso, você pode manter os métodos de autenticação existentes implantados e delegar a autenticação para um proxy. O proxy pode aplicar as políticas de autenticação para conexões de cliente para aplicações específicas. É possível aproveitar qualquer autenticação do IAM que você já tenha para funções do Lambda, em vez de gerenciar credenciais de banco de dados em seu código de aplicação do Lambda.
- O RDS Proxy pode ajudar a tornar as aplicações mais resilientes e transparentes às falhas do banco de dados. Ele ignora os caches do Sistema de Nomes de Domínio (DNS) para reduzir os tempos de failover em até 66% para bancos de dados multi-AZ do Amazon RDS. O RDS Proxy também direciona automaticamente o tráfego para uma nova instância de banco de dados, preservando as conexões da aplicação. Isso torna os failovers mais transparentes para as aplicações.

Conceitos e terminologia do RDS Proxy

É possível simplificar o gerenciamento das conexões das instâncias de banco de dados do Amazon RDS e de clusters de bancos de dados Amazon Aurora usando o RDS Proxy.

O proxy do RDS lida com o tráfego de rede entre a aplicação cliente e o banco de dados. Ele faz isso de uma maneira ativa primeiro, entendendo o protocolo de banco de dados. Depois, ele ajusta seu comportamento com base nas operações SQL da aplicação e nos conjuntos de resultados do banco de dados.

O proxy do RDS reduz a sobrecarga de memória e de CPU para gerenciamento de conexão no banco de dados. O banco de dados precisa de menos memória e recursos de CPU quando as

aplicações abrem muitas conexões simultâneas. Ele também não requer lógica em suas aplicações para fechar e reabrir conexões que permanecem ociosas por um longo período. Da mesma forma, ele requer menos lógica nas aplicações para restabelecer conexões em caso de um problema de banco de dados.

A infraestrutura do proxy do RDS é altamente disponível e implantada em várias zonas de disponibilidade (AZs). A computação, a memória e o armazenamento do RDS Proxy são independentes da instância de banco de dados do RDS. Essa separação ajuda a reduzir a sobrecarga em seus servidores de banco de dados, para que eles possam dedicar seus recursos para atender às workloads do banco de dados. Os recursos de computação do proxy do RDS não têm servidor, realizando a escalabilidade automática com base na workload do banco de dados.

Tópicos

- Visão geral dos conceitos do RDS Proxy
- Agrupamento de conexões
- Segurança do RDS Proxy
- Failover
- Transações

Visão geral dos conceitos do RDS Proxy

O proxy do RDS lida com a infraestrutura para executar o agrupamento de conexões e os outros recursos descritos nas seções a seguir. Os proxies estão representados no console do RDS na página Proxies.

Cada proxy lida com conexões com uma única instância de banco de dados do RDS. O proxy determina automaticamente a instância de gravador atual para as instâncias de banco de dados multi-AZ do RDS e para os clusters provisionados do Aurora.

As conexões que um proxy mantém abertas e disponíveis para a aplicação de banco de dados usar compõem o grupo de conexões.

Por padrão, o proxy do RDS pode reutilizar uma conexão após cada transação em sua sessão. Essa reutilização em nível de transação é chamada de multiplexação. Quando o proxy do RDS remove temporariamente uma conexão do grupo de conexões para reutilizá-la, essa operação é chamada de empréstimo da conexão. Quando é seguro fazer isso, o proxy do RDS retorna essa conexão ao grupo de conexões.

Em alguns casos, o proxy do RDS não consegue ter certeza de que é seguro reutilizar uma conexão de banco de dados fora da sessão atual. Nesses casos, ele mantém a sessão na mesma conexão até que a sessão termine. Esse comportamento de fallback é chamado de fixação.

Um proxy tem um endpoint padrão. Você se conecta a esse endpoint ao trabalhar com uma instância de banco de dados do RDS ou com um cluster de banco de dados do Aurora. Você faz isso em vez de se conectar ao endpoint de leitura/gravação que se conecta diretamente à instância ou ao cluster. Os endpoints de finalidade especial de um cluster do Aurora permanecem disponíveis para uso. Para clusters de banco de dados do RDS, também é possível criar endpoints adicionais de leitura/gravação e somente leitura. Para ter mais informações, consulte Visão geral dos endpoints de proxy.

Por exemplo, você ainda pode se conectar ao endpoint do cluster para conexões de leitura/gravação sem agrupamento de conexões. Você ainda pode se conectar ao endpoint de leitor para conexões somente leitura com balanceamento de carga. Você ainda pode se conectar aos endpoints da instância para diagnóstico e solução de problemas de instâncias de banco de dados específicas em um cluster do . Se você usa outros serviços da AWS, como o AWS Lambda para se conectar a bancos de dados do RDS, altere as configurações de conexão para usar o endpoint do proxy. Por exemplo, você especifica o endpoint do proxy para permitir que as funções do Lambda acessem seu banco de dados enquanto aproveita a funcionalidade do RDS Proxy.

Cada proxy contém um grupo de destino. Esse grupo de destino incorpora a instância de banco de dados do RDS à qual o proxy pode se conectar. Para um cluster do Aurora, por padrão, o grupo de destino é associado a todas as instâncias de banco de dados nesse cluster. Dessa forma, o proxy pode se conectar a qualquer instância de bancos de dados Aurora promovida para ser a instância de gravação no cluster. A instância de banco de dados do RDS associada a um proxy é chamada de destino desse proxy. Por conveniência, quando você cria um proxy pelo console, o proxy do RDS também cria o grupo de destino correspondente e registra os destinos associados automaticamente.

Uma família de mecanismos é um conjunto relacionado de mecanismos de banco de dados que usam o mesmo protocolo de banco de dados. Escolha a família de mecanismos para cada proxy criado.

Agrupamento de conexões

Cada proxy executa o agrupamento de conexões separadamente para a instância de gravação e leitura de seu banco de dados do RDS ou do associado. O agrupamento de conexões é uma otimização que reduz a sobrecarga associada à abertura e ao fechamento de conexões e à manutenção de muitas conexões abertas simultaneamente. Essa sobrecarga inclui a memória necessária para lidar com cada nova conexão. Também envolve a sobrecarga da CPU para fechar

cada conexão e abrir uma nova. Os exemplos incluem handshaking TLS/SSL (Transport Layer Security/Secure Sockets Layer), autenticação, recursos de negociação etc. O agrupamento de conexões simplifica a lógica da aplicação. Você não precisa escrever código da aplicação para minimizar o número de conexões abertas simultâneas.

Cada proxy também executa a multiplexação de conexões, também conhecida como reutilização de conexões. Com a multiplexação, o RDS Proxy executa todas as operações para uma transação usando uma conexão de banco de dados subjacente. Depois, o RDS pode usar uma conexão diferente para a próxima transação. Você pode abrir muitas conexões simultâneas com o proxy, e o proxy mantém um número menor de conexões abertas com a instância ou o cluster de banco de dados. Isso minimiza ainda mais a sobrecarga na memória de conexões no servidor de banco de dados. Essa técnica também reduz a possibilidade de erros de “excesso de conexões”.

Segurança do RDS Proxy

O proxy do RDS usa os mecanismos de segurança do RDS existentes, como TLS/SSL e o AWS Identity and Access Management (IAM). Para obter informações gerais sobre esses recursos de segurança, consulte Segurança no Amazon RDS. Além disso, familiarize-se com a forma como o RDS e o Aurora trabalham com autenticação, autorização e outras áreas de segurança.

O proxy do RDS pode atuar como uma camada adicional de segurança entre aplicações cliente e o banco de dados subjacente. Por exemplo, você pode se conectar ao proxy usando o TLS 1.3, mesmo que a instância de banco de dados subjacente ofereça suporte a uma versão mais antiga do TLS. Você pode se conectar ao proxy com um perfil do IAM. Isso ocorre mesmo que o proxy se conecte ao banco de dados usando o método de autenticação nativa de usuário e senha. Usando essa técnica, você pode aplicar requisitos de autenticação fortes para aplicações de banco de dados sem um esforço de migração dispendioso para as próprias instâncias de banco de dados.

Armazene as credenciais de banco de dados usadas pelo proxy do RDS no AWS Secrets Manager. Cada usuário de banco de dados da instância de banco de dados do RDS acessada por um proxy deve ter um segredo correspondente no Secrets Manager. Também é possível configurar a autenticação do IAM para os usuários do RDS Proxy. Ao fazer isso, é possível aplicar a autenticação do IAM para acesso ao banco de dados mesmo que os bancos de dados usem autenticação nativa de senha. Recomendamos usar esses recursos de segurança em vez de incorporar credenciais de banco de dados no código da aplicação.

Usar TLS/SSL com o RDS Proxy

Você pode se conectar ao proxy do RDS usando o protocolo TLS/SSL.

> **Note**
>
> O proxy do Amazon RDS usa certificados do AWS Certificate Manager (ACM). Se você estiver usando o RDS Proxy, não será necessário baixar os certificados do Amazon RDS ou atualizar as aplicações que usam conexões RDS Proxy.

Para aplicar o TLS a todas as conexões entre o proxy e o banco de dados, é possível especificar uma configuração Exigir Transport Layer Security ao criar ou modificar um proxy no AWS Management Console.

O proxy do RDS também pode garantir que a sessão use TLS/SSL entre o cliente e o endpoint do RDS Proxy. Para que o proxy do RDS faça isso, especifique o requisito no lado do cliente. As variáveis de sessão SSL não são definidas para conexões SSL a um banco de dados usando o RDS Proxy.

- Para o RDS para MySQL e Aurora MySQL, especifique o requisito no lado do cliente com o parâmetro ao executar o comando .
- Para o PostgreSQL do Amazon RDS e o Aurora PostgreSQL, especifique como parte da string ao executar o comando .

O RDS Proxy é compatível com o protocolo TLS versões 1.0, 1.1, 1.2 e 1.3. É possível se conectar ao proxy usando uma versão mais recente do TLS do que a usada no banco de dados subjacente.

Por padrão, os programas cliente estabelecem uma conexão criptografada com o RDS Proxy, com controle adicional disponível por meio da opção `--ssl-mode`. No lado do cliente, o proxy do RDS oferece suporte a todos os modos SSL.

Para o cliente, os modos SSL são os seguintes:

PREFERRED

O SSL é a primeira opção, mas não é obrigatória.

DISABLED

Nenhum SSL é permitido.

REQUIRED

Impor SSL.

VERIFY_CA

Impor SSL e verificar a autoridade de certificação (CA - certificate authority).

VERIFY_IDENTITY

Impor SSL e verificar a CA, além do nome de host da CA.

Ao usar um cliente com `--ssl-mode VERIFY_CA` ou `VERIFY_IDENTITY`, especifique a opção `--ssl-ca` apontando para uma CA em formato `.pem`. Para o arquivo `.pem` a ser usado, baixe todos os PEMs de CA raiz de Amazon Trust Services e coloque-os em um único arquivo `.pem`.

O RDS Proxy usa certificados curinga, que se aplicam a um domínio e aos seus subdomínios. No momento, se você usa o cliente `mysql` para se conectar ao modo SSL `VERIFY_IDENTITY`, é necessário usar o comando `mysql` compatível com MySQL 8.0.

Failover

O failover é um recurso de alta disponibilidade que substitui uma instância de banco de dados por outra quando a instância original fica indisponível. Um failover pode ocorrer devido a um problema com uma instância de banco de dados. Ele também pode fazer parte de procedimentos normais de manutenção, como durante uma atualização de banco de dados. O failover se aplica a instâncias de banco de dados do RDS em uma configuração multi-AZ e a clusters de bancos de dados Aurora com uma ou mais instâncias de leitor além da instância de gravador.

A conexão por meio de um proxy torna a aplicação mais resiliente a failovers de banco de dados. Quando a instância de banco de dados original se torna indisponível, o proxy do RDS conecta-se ao banco de dados em espera sem eliminar as conexões de aplicações ociosas. Isso ajuda a acelerar e simplificar o processo de failover. Isso causa menos interrupções na aplicação do que uma reinicialização típica ou um problema de banco de dados.

Sem o RDS Proxy, um failover envolve uma breve interrupção. Durante a interrupção, não é possível executar operações de gravação no banco de dados em failover. Todas as conexões de banco de dados existentes são interrompidas e sua aplicação precisa reabri-las. O banco de dados fica disponível para novas conexões e operações de gravação quando uma instância de banco de dados somente leitura é promovida para substituir a que não está disponível.

Durante os failovers de banco de dados, o proxy do RDS continua aceitando conexões no mesmo endereço IP e direciona conexões automaticamente para a nova instância do banco de dados primário. Os clientes que se conectam pelo proxy do RDS não são suscetíveis ao seguinte:

- Atrasos de propagação do Domain Name System (DNS) no failover.
- Cache DNS local.
- Tempos limite de conexão.
- Incerteza sobre qual instância de banco de dados é o gravador atual.
- Espera pela resposta de uma consulta de um gravador anterior que ficou indisponível sem fechar conexões.

Para aplicações que mantêm seu próprio grupo de conexões, passar pelo proxy do RDS significa que a maioria das conexões permanecem ativas durante failovers ou outras interrupções. Somente conexões que estão no meio de uma transação ou de uma instrução SQL são canceladas. O proxy do RDS aceita novas conexões imediatamente. Quando o gravador do banco de dados não estiver disponível, o proxy do RDS coloca as solicitações de entrada em fila.

Para aplicações que não mantêm seus próprios grupos de conexão, o proxy do RDS oferece taxas de conexão mais rápidas e conexões mais abertas. Ele descarrega a sobrecarga cara de reconexões frequentes do banco de dados. E faz isso reutilizando as conexões de banco de dados mantidas no grupo de conexões do RDS Proxy. Essa abordagem é particularmente importante para conexões TLS, em que os custos da configuração são significativos.

Transações

Todas as instruções dentro de uma única transação sempre usam a mesma conexão ao banco de dados subjacente. A conexão se torna disponível para uso de outra sessão quando a transação termina. O uso da transação como a unidade de granularidade tem as seguintes consequências:

- A reutilização da conexão pode ocorrer após cada declaração individual, quando a configuração do RDS para MySQL ou do Aurora MySQL estiver ativada.
- Por outro lado, quando a configuração `autocommit` está desativada, a primeira declaração emitida em uma sessão inicia uma nova transação. Por exemplo, suponhamos que você insira uma sequência de `SELECT`, `INSERT`, `UPDATE` e outras declarações em linguagem de manipulação de dados (DML). Nesse caso, a reutilização da conexão não ocorrerá até que você emita um `COMMIT`, `ROLLBACK` ou encerre a transação.
- A inserção de uma instrução DDL (Data Definition Language, linguagem de definição de dados) faz com que a transação termine depois que essa instrução é concluída.

O proxy do RDS detecta quando uma transação termina por meio do protocolo de rede usado pela aplicação cliente de banco de dados. A detecção da transação não depende de palavras-chave, como `COMMIT` ou `ROLLBACK`, que aparecem no texto da instrução SQL.

Em alguns casos, o proxy do RDS pode detectar uma solicitação de banco de dados que torna impraticável mover a sessão para uma conexão diferente. Nesses casos, ele desativa a multiplexação para essa conexão o restante da sessão. A mesma regra se aplicará se o proxy do RDS não puder ter certeza de que a multiplexação é prática para a sessão. Essa operação é chamada de fixação. Para obter formas de detectar e minimizar a fixação, consulte Evitar a fixação de um RDS Proxy.

Conceitos básicos do RDS Proxy

Use as informações nas páginas a seguir para configurar, gerenciar Amazon RDS Proxy e definir opções de segurança relacionadas. As opções de segurança controlam quem pode acessar cada proxy e como cada proxy se conecta a instâncias de banco de dados.

Se você é novo no RDS Proxy, recomendamos seguir as páginas na ordem em que as apresentamos.

Tópicos

- Configurar os pré-requisitos de rede para o RDS Proxy
- Configurar credenciais de banco de dados no AWS Secrets Manager para RDS Proxy
- Configurar políticas do AWS Identity and Access Management (IAM) para o RDS Proxy
- Criar um proxy para o Amazon RDS
- Visualizar um proxy
- Conectar-se a um banco de dados pelo RDS Proxy

Configurar os pré-requisitos de rede para o RDS Proxy

O uso do RDS Proxy requer que você tenha uma nuvem privada virtual (VPC) comum entre a instância de banco de dados do RDS e o RDS Proxy. Essa VPC deve ter um mínimo de duas sub-redes em zonas de disponibilidade diferentes. Sua conta pode possuir essas sub-redes ou compartilhá-las com outras contas. Para obter informações sobre compartilhamento de VPC, consulte Trabalhar com VPCs compartilhadas.

Seus recursos de aplicações cliente, como Amazon EC2, Lambda ou Amazon ECS, podem estar na mesma VPC que o proxy. Ou podem estar em uma VPC separada do proxy. Se você se conectou com êxito a quaisquer instâncias de banco de dados do RDS ou clusters de banco de dados do Aurora, você já terá os recursos de rede necessários.

Tópicos

- Obter informações sobre suas sub-redes
- Planejar a capacidade de endereços IP

Obter informações sobre suas sub-redes

Para criar um proxy, é necessário fornecer as sub-redes e a VPC em que o proxy opera. O exemplo do Linux a seguir mostra comandos da AWS CLI que examinam as VPCs e as sub-redes pertencentes à sua Conta da AWS. Em particular, você passa IDs de sub-rede como parâmetros quando cria um proxy usando o CLI.

```
aws ec2 describe-vpcs
aws ec2 describe-internet-gateways
aws ec2 describe-subnets --query '*[].[VpcId,SubnetId]' --output text | sort
```

O exemplo do Linux a seguir mostra comandos da AWS CLI para determinar os IDs de sub-rede correspondentes a uma instância específica de banco de dados do RDS. Encontre o ID da VPC para a instância de banco de dados. Examine a VPC para encontrar as sub-redes. O exemplo do Linux a seguir mostra como fazer isso.

```
$ #From the DB instance, trace through the DBSubnetGroup and Subnets to find the subnet
 IDs.
$ aws rds describe-db-instances --db-instance-identifier my_instance_id --query '*[].
[DBSubnetGroup]|[0]|[0]|[Subnets]|[0]|[*].SubnetIdentifier' --output text
```

```
subnet_id_1
subnet_id_2
subnet_id_3
...
```

```
$ #From the DB instance, find the VPC.
$ aws rds describe-db-instances --db-instance-identifier my_instance_id --query '*[].
[DBSubnetGroup]|[0]|[0].VpcId' --output text
```

```
my_vpc_id
```

```
$ aws ec2 describe-subnets --filters Name=vpc-id,Values=my_vpc_id --query '*[].
[SubnetId]' --output text
```

```
subnet_id_1
subnet_id_2
subnet_id_3
subnet_id_4
subnet_id_5
subnet_id_6
```

Planejar a capacidade de endereços IP

Um RDS Proxy ajusta automaticamente sua capacidade conforme necessário com base no tamanho e no número de instâncias de banco de dados registradas nele. Algumas operações também podem precisar de capacidade adicional de proxy, como aumentar o tamanho de um banco de dados registrado ou operações internas de manutenção do RDS Proxy. Durante essas operações, seu proxy pode precisar de mais endereços IP para provisionar a capacidade extra. Esses endereços adicionais possibilitam que seu proxy se expanda sem afetar a workload. A falta de endereços IP gratuitos em suas sub-redes impede que um proxy aumente a escala verticalmente. Isso pode ocasionar maiores latências de consulta ou falhas na conexão do cliente. O RDS notifica você por meio de um evento `RDS-EVENT-0243` quando não há endereços IP livres suficientes em suas sub-redes. Para obter informações sobre esse evento, consulte Trabalhar com eventos do RDS Proxy.

Reserve os números mínimos a seguir de endereços IP livres nas sub-redes para o proxy, com base no tamanho das classes de instâncias de banco de dados.

Classe de instância de banco de dados	Mínimo de endereços IP livres
db.*.xlarge ou menor	10
db.*.2xlarge	15
db.*.4xlarge	25
db.*.8xlarge	45
db.*.12xlarge	60

Classe de instância de banco de dados	Mínimo de endereços IP livres
db.*.16xlarge	75
db.*.24xlarge	110

Esses números de endereços IP recomendados são estimativas para um proxy com apenas o endpoint padrão. Um proxy com endpoints adicionais ou réplicas de leitura pode precisar de mais endereços IP livres. Para cada endpoint adicional, recomendamos que você reserve mais três endereços IP. Para cada réplica de leitura, recomendamos que você reserve endereços IP adicionais conforme especificado na tabela com base no tamanho dessa réplica de leitura.

> **Note**
>
> O RDS Proxy não consome mais de 215 endereços IP em uma VPC.

Configurar credenciais de banco de dados no AWS Secrets Manager para RDS Proxy

O RDS Proxy no Amazon RDS usa o AWS Secrets Manager para armazenar e gerenciar credenciais de banco de dados com segurança. Em vez de incorporar credenciais na aplicação, você associa um proxy a um segredo do Secrets Manager que contenha os detalhes de autenticação necessários. Crie um segredo separado do Secrets Manager para cada conta de usuário de banco de dados ao qual o proxy se conecta na instância de banco de dados do RDS.

Tópicos

- Criar segredos para usá-los com o RDS Proxy

Criar segredos para usá-los com o RDS Proxy

Antes de criar um proxy, você deve primeiro criar pelo menos um segredo que armazene suas credenciais de banco de dados.

Console

Como criar um segredo

1. Abra o console Secrets Manager em https://console.aws.amazon.com/secretsmanager/.
2. Escolha Armazenar Novo Segredo.
3. Escolha Credenciais para um banco de dados do Amazon RDS.
4. Insira um nome de usuário e senha. As credenciais que você inserir devem corresponder às credenciais de um usuário do banco de dados que exista no banco de dados do RDS correlacionado. O RDS Proxy usa essas credenciais para autenticar e estabelecer conexões com o banco de dados em nome das aplicações.

   Se houver alguma incompatibilidade, você poderá atualizar o segredo para que corresponda à senha do banco de dados. Enquanto você não atualizar o segredo, as tentativas de conexão por meio do proxy usando esse segredo falharão, mas as conexões que usam outros segredos válidos ainda funcionarão.

   > **Note**
   >
   > No caso do RDS para SQL Server, o RDS Proxy exige um segredo com distinção entre maiúsculas e minúsculas no Secrets Manager, independentemente das configurações de agrupamento da instância de banco de dados. Se a aplicação permitir nomes de usuário com diferentes usos de maiúsculas e minúsculas, como “Admin” e “admin”, você deverá criar segredos separados para cada um. O RDS Proxy não aceita autenticação de nome de usuário sem distinção entre maiúsculas e minúsculas entre o cliente e o proxy.
   > Para ter mais informações sobre agrupamento no SQL Server, consulte a documentação do Microsoft SQL Server.

5. Em Banco de dados, selecione o banco de dados do Amazon RDS que o segredo acessará.
6. Preencha outras configurações referentes ao segredo e escolha Armazenar. Para obter instruções mais abrangentes, consulte Creating an AWS Secrets Manager secret no Guia do usuário do AWS Secrets Manager.

AWS CLI

Ao criar um proxy por meio da AWS CLI, especifique os nomes de recurso da Amazon (ARNs) dos segredos correspondentes. Você faz isso para todas as contas de usuário do banco de dados que o proxy pode acessar. No AWS Management Console, escolha os segredos por seus nomes descritivos.

- Para criar um segredo do Secrets Manager e utilizá-lo com o RDS Proxy, use o comando create-secret:

```
aws secretsmanager create-secret \
  --name "secret_name" \
  --description "secret_description" \
  --region region_name \
  --secret-string '{"username":"db_user","password":"db_user_password"}'
```

- Também é possível criar uma chave personalizada para criptografar o segredo do Secrets Manager. O comando a seguir cria uma chave de exemplo.

```
aws kms create-key --description "test-key" --policy '{
  "Id":"kms-policy",
  "Version":"2012-10-17",
  "Statement":
    [
      {
        "Sid":"Enable IAM User Permissions",
        "Effect":"Allow",
        "Principal":{"AWS":"arn:aws:iam::account_id:root"},
        "Action":"kms:*","Resource":"*"
      },
      {
        "Sid":"Allow access for Key Administrators",
        "Effect":"Allow",
        "Principal":
          {
            "AWS":
              ["$USER_ARN","arn:aws:iam:account_id::role/Admin"]
          },
        "Action":
          [
            "kms:Create*",
            "kms:Describe*",
```

```
            "kms:Enable*",
            "kms:List*",
            "kms:Put*",
            "kms:Update*",
            "kms:Revoke*",
            "kms:Disable*",
            "kms:Get*",
            "kms:Delete*",
            "kms:TagResource",
            "kms:UntagResource",
            "kms:ScheduleKeyDeletion",
            "kms:CancelKeyDeletion"
          ],
        "Resource":"*"
      },
      {
        "Sid":"Allow use of the key",
        "Effect":"Allow",
        "Principal":{"AWS":"$ROLE_ARN"},
        "Action":["kms:Decrypt","kms:DescribeKey"],
        "Resource":"*"
      }
    ]
}'
```

Por exemplo, os seguintes comandos criam segredos do Secrets Manager para dois usuários de banco de dados:

```
aws secretsmanager create-secret \
  --name secret_name_1 --description "db admin user" \
  --secret-string '{"username":"admin","password":"choose_your_own_password"}'

aws secretsmanager create-secret \
  --name secret_name_2 --description "application user" \
  --secret-string '{"username":"app-user","password":"choose_your_own_password"}'
```

Para criar esses segredos criptografados com a chave do AWS KMS personalizada, use os seguintes comandos:

```
aws secretsmanager create-secret \
  --name secret_name_1 --description "db admin user" \
  --secret-string '{"username":"admin","password":"choose_your_own_password"}' \
```

```
  --kms-key-id arn:aws:kms:us-east-2:account_id:key/key_id

aws secretsmanager create-secret \
  --name secret_name_2 --description "application user" \
  --secret-string '{"username":"app-user","password":"choose_your_own_password"}' \
  --kms-key-id arn:aws:kms:us-east-2:account_id:key/key_id
```

Para ver os segredos que pertencem à sua conta da AWS, use o comando list-secrets:

```
aws secretsmanager list-secrets
```

Ao criar um proxy utilizando a CLI, você transmite os nomes de recursos da Amazon (ARNs) de um ou mais segredos ao parâmetro `--auth`. O exemplo a seguir mostra como preparar um relatório com apenas o nome e o ARN de cada segredo pertencente à sua conta da AWS. Este exemplo usa o parâmetro `--output table` que está disponível na AWS CLI versão 2. Se você estiver usando a AWS CLI versão 1, use `--output text` em vez disso.

```
aws secretsmanager list-secrets --query '*[].[Name,ARN]' --output table
```

Para confirmar se o segredo contém as credenciais corretas no formato adequado, use o comando get-secret-value. Substitua *your_secret_name* pelo nome abreviado ou ARN do segredo.

```
aws secretsmanager get-secret-value --secret-id your_secret_name
```

A saída contém uma linha com um valor codificado em JSON semelhante ao seguinte:

```
...
"SecretString": "{\"username\":\"your_username\",\"password\":\"your_password\"}",
...
```

Configurar políticas do AWS Identity and Access Management (IAM) para o RDS Proxy

Depois de criar os segredos no Secrets Manager, crie uma política do IAM que possa acessar esses segredos. Para obter mais informações sobre como usar a IAM, consulte Gerenciamento de identidade e acesso no Amazon RDS.

> **Tip**
>
> O procedimento a seguir se aplicará se você usar o console do IAM. Se você usar o AWS Management Console para RDS, o RDS poderá criar a política do IAM para você automaticamente. Nesse caso, você pode ignorar o procedimento a seguir.

Como criar uma política do IAM que acessa os segredos do Secrets Manager para uso com o proxy

1. Faça login no console do IAM. Siga o processo Criar perfil, conforme descrito em Criar perfis do IAM, selecionando Criar um perfil para delegar permissões a um serviço da AWS.

   Selecione Serviço da AWS para o Tipo de entidade confiável. Em Caso de uso, selecione RDS no menu suspenso Casos de uso para outros serviços da AWS. Selecione RDS: adicionar perfil ao banco de dados.

2. Para a nova função, execute a etapa Add inline policy (Adicionar política em linha). Use os mesmos procedimentos gerais que em Editar políticas do IAM. Cole o seguinte JSON na caixa de texto JSON. Substitua o ID da sua própria conta. Substitua sua região da AWS por `us-east-2`. Substitua os nomes de recurso da Amazon (ARNs) pelos segredos criados. Consulte Especificar chaves do KMS em instruções de política do IAM. Para a ação `kms:Decrypt`, substitua o ARN da chave padrão AWS KMS key ou sua própria chave do KMS. O que você usa depende de qual deles você usou para criptografar os segredos do Secrets Manager.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": "secretsmanager:GetSecretValue",
            "Resource": [
                "arn:aws:secretsmanager:us-east-2:account_id:secret:secret_name_1",
                "arn:aws:secretsmanager:us-east-2:account_id:secret:secret_name_2"
            ]
        },
        {
            "Sid": "VisualEditor1",
            "Effect": "Allow",
            "Action": "kms:Decrypt",
            "Resource": "arn:aws:kms:us-east-2:account_id:key/key_id",
```

```
            "Condition": {
                "StringEquals": {
                    "kms:ViaService": "secretsmanager.us-east-2.amazonaws.com"
                }
            }
        }
    ]
}
```

3. Edite a política de confiança desse perfil do IAM. Cole o seguinte JSON na caixa de texto JSON.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "rds.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

O comando a seguir executa a mesma operação na AWS CLI.

```
PREFIX=my_identifier
USER_ARN=$(aws sts get-caller-identity --query "Arn" --output text)

aws iam create-role --role-name my_role_name \
  --assume-role-policy-document '{"Version":"2012-10-17","Statement":
[{"Effect":"Allow","Principal":{"Service":
["rds.amazonaws.com"]},"Action":"sts:AssumeRole"}]}'

ROLE_ARN=arn:aws:iam::account_id:role/my_role_name

aws iam put-role-policy --role-name my_role_name \
  --policy-name $PREFIX-secret-reader-policy --policy-document '{
    "Version": "2012-10-17",
    "Statement": [
        {
```

```
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": "secretsmanager:GetSecretValue",
            "Resource": [
                "arn:aws:secretsmanager:us-east-2:account_id:secret:secret_name_1",
                "arn:aws:secretsmanager:us-east-2:account_id:secret:secret_name_2"
            ]
        },
        {
            "Sid": "VisualEditor1",
            "Effect": "Allow",
            "Action": "kms:Decrypt",
            "Resource": "arn:aws:kms:us-east-2:account_id:key/key_id",
            "Condition": {
                "StringEquals": {
                    "kms:ViaService": "secretsmanager.us-east-2.amazonaws.com"
                }
            }
        }
    ]
}
```

Criar um proxy para o Amazon RDS

É possível associar um proxy a uma instância de banco de dados do RDS para MariaDB, do RDS para Microsoft SQL Server, do RDS para MySQL ou do RDS para PostgreSQL.

Console

Como criar um proxy

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Proxies.
3. Escolha Create proxy (Criar proxy).
4. Defina as configurações de proxy a seguir.

Configuração	Descrição
Família de mecanismos	O protocolo de rede de banco de dados que o proxy reconhece quando interpreta o tráfego de rede de entrada e saída do banco de dados. **Note** Para usar o RDS para PostgreSQL, mantenha o banco de dados `postgres` na sua instância. Consulte Solução de problemas de exclusão do banco de dados `postgres`.
Identificador do proxy	Um nome exclusivo no ID da conta da AWS e na região da AWS atual.
Tempo-limite de inatividade da conexão do cliente	O proxy fechará a conexão do cliente se ela permanecer inativa por determinado período. Por padrão, é 1.800 segundos (30 minutos). Uma conexão fica ociosa quando a aplicação não envia uma nova solicitação dentro do período especificado após a conclusão da solicitação anterior. O proxy mantém a conexão do banco de dados subjacente aberta e a devolve ao grupo de conexões, disponibilizando-a para novas conexões de clientes. Para que o proxy remova proativamente as conexões ociosas, reduza o tempo-limite de inatividade da conexão do cliente. Para minimizar os custos de conexão durante picos de workload, aumente o tempo-limite.

Configuração	Descrição
Banco de dados	A instância de banco de dados do RDS para acesso por meio desse proxy. A lista inclui somente instâncias e clusters de banco de dados com mecanismos de banco de dados compatíveis, versões de mecanismo e outras configurações. Se a lista estiver vazia, crie uma instância ou cluster de banco de dados compatível com o RDS Proxy. Para fazer isso, siga o procedimento em Criar uma instância de banco de dados do Amazon RDS. Depois, tente criar o proxy novamente.
Conexões máximas do grupo de conexões	Um valor entre 1 e 100 para definir a porcentagem do limite de `max_connections` que o RDS Proxy pode usar. Se pretender usar apenas um proxy com esse cluster ou instância de banco de dados, defina esse valor como 100. Para ter mais informaçõ es sobre como o RDS Proxy usa essa configuração, consulte the section called “MaxConnectionsPercent”.

Configuração	Descrição
Filtros de fixação de sessão	Impede que o RDS Proxy fixe determinados estados de sessão detectados, o que ignora as medidas de segurança padrão para conexões de multiplexação. No momento, o PostgreSQL não aceita essa configuração, e a única opção disponível é `EXCLUDE_VARIABLE_SETS`. Habilitá-la pode fazer com que as variáveis de sessão de uma conexão afetem outras, gerando erros ou problemas de correção caso as consultas dependam de variáveis de sessão definidas fora da transação atual. Use essa opção somente depois de confirmar que as aplicações podem compartilhar conexões de banco de dados com segurança. Os seguintes padrões são considerados seguros: • Declarações `SET` nas quais não haja alteração no valor efetivo da variável de sessão. Em outras palavras, não há alteração na variável de sessão. • Você altera o valor da variável de sessão e executa uma instrução na mesma transação. Para obter mais informações, consulte Evitar a fixação de um RDS Proxy.
Tempo-limite de empréstimo da conexão	Se você espera que o proxy use todas as conexões de banco de dados disponíveis, defina o tempo de espera antes que ele gere um erro de tempo-limite. É possível especificar até cinco minutos. Essa configuração só se aplica quando o proxy atingiu o número máximo de conexões e todas as conexões já estão em uso.

Configuração	Descrição
Consulta de inicialização	(Opcional) Adicione uma consulta de inicialização ou modifique a atual. Você pode especificar uma ou mais instruções SQL para o proxy executar ao abrir cada nova conexão de banco de dados. A configuração normalmente é usada com instruções SET para garantir que cada conexão tenha configurações idênticas. Verifique se a consulta adicionada é válida. Para incluir diversas variáveis em uma única instrução SET, use vírgulas como separadores. Por exemplo: `SET variable1 =value1, variable2 =value2` Para várias instruções, use um ponto e vírgula como separador. **Important** Como você pode acessar a consulta de inicialização como parte da configuração do grupo de destino, ela não é protegida por autenticação ou métodos criptográficos. Qualquer pessoa com acesso para visualizar ou gerenciar a configuração do seu grupo de destino de proxy pode visualizar a consulta de inicialização. Você não deve adicionar dados sensíveis, como senhas ou chaves de criptografia de longa duração, a essa opção.
Perfil do AWS Identity and Access Management (IAM)	Um perfil do IAM com permissão para acessar os segredos do Secrets Manager, que representam as credenciais das contas de usuário do banco de dados que o proxy pode usar. Também é possível criar um perfil do IAM no AWS Management Console.
Segredos do Secrets Manager.	Selecione pelo menos um segredo do Secrets Manager que contenha credenciais de usuário de banco de dados que permita ao proxy acessar a instância de banco de dados do RDS.

Configuração	Descrição
Tipo de autenticação de cliente	O tipo de autenticação usada pelo proxy para conexões de clientes. Sua escolha se aplica a todos os segredos do Secrets Manager que você associa a esse proxy. Se você precisar especificar um tipo de autenticação de cliente diferente para cada segredo, crie seu proxy usando a AWS CLI ou a API.
Autenticação do IAM	Se deseja ou não exigir, permitir ou negar a autenticação do IAM para conexões com o proxy. A opção de permissão só é válida para proxies do RDS para SQL Server. Sua escolha se aplica a todos os segredos do Secrets Manager que você associa a esse proxy. Se você precisar especificar um tipo de autenticação do IAM diferente para cada segredo, crie seu proxy usando a AWS CLI ou a API.
Exigir Transport Layer Security	Impõe TLS/SSL para todas as conexões de cliente. O proxy usa a mesma configuração de criptografia para sua conexão com o banco de dados subjacente, quer a conexão do cliente seja criptografada ou não criptografada.
Sub-redes	Esse campo é pré-preenchido com todas as sub-redes associadas à sua VPC. É possível remover quaisquer sub-redes que não sejam necessárias para o proxy, mas pelo menos duas sub-redes devem ser mantidas.

Configuração	Descrição
Grupo de segurança de VPC	Escolha um grupo de segurança da VPC existente ou crie um no AWS Management Console. Configure as regras de entrada para possibilitar que suas aplicações acessem o proxy e as regras de saída para permitir o tráfego de seus destinos de banco de dados. **Note** O grupo de segurança deve permitir conexões do proxy com o banco de dados. Ele serve para entrada das aplicações para o proxy e saída do proxy para o banco de dados. Por exemplo, se você usar o mesmo grupo de segurança para o banco de dados e o proxy, verifique se os recursos nesse grupo de segurança podem se comunicar entre si. Ao usar uma VPC compartilhada, evite usar o grupo de segurança padrão para a VPC ou um associado a outra conta. Em vez disso, escolha um grupo de segurança que pertença à sua conta. Se não houver nenhum, crie um. Para ter mais informações, consulte Trabalhar com VPCs compartilhadas. O RDS implanta um proxy em várias zonas de disponibilidade para garantir a alta disponibilidade. Para habilitar a comunicação entre AZs, a lista de controle de acesso (ACL) de rede da sub-rede de proxy deve permitir saída na porta do mecanismo e entrada em todas as portas. Para obter mais informações sobre ACLs de rede, consulte Controlar o tráfego para sub-redes com ACLs de rede. Se a ACL de rede do proxy e do destino for idêntica, você deverá adicionar uma regra de entrada do protocolo TCP em que a Fonte esteja definida como o CIDR da VPC. Você também deve adicionar uma regra de saída de

Configuração	Descrição
	protocolo TCP específica da porta do mecanismo em que o Destino esteja definido como o CIDR da VPC.
Ativar o registro em log aprimorado	Habilite essa configuração para solucionar problemas de compatibilidade ou de performance do proxy. Quando habilitado, o RDS Proxy registra em log informações detalhadas de performance para ajudar você a depurar o comportamento do SQL ou a performance e a escalabilidade da conexão do proxy. Habilite essa configuração apenas para depuração e garanta que as medidas de segurança adequadas estejam em vigor para proteger as informações confidenciais nos logs. Para minimizar as despesas operacionais, o RDS Proxy desabilita automaticamente essa configuração 24 horas após a ativação. Habilite-a temporariamente para solucionar problemas específicos.

5. Escolha Create proxy (Criar proxy).

AWS CLI

Para criar um proxy utilizando a AWS CLI, chame o comando create-db-proxy com os seguintes parâmetros obrigatórios:

- `--db-proxy-name`
- `--engine-family`
- `--role-arn`
- `--auth`
- `--vpc-subnet-ids`

O valor `--engine-family` diferencia letras maiúsculas de minúsculas.

Example

Para Linux, macOS ou Unix:

```
aws rds create-db-proxy \
    --db-proxy-name proxy_name \
    --engine-family { MYSQL | POSTGRESQL | SQLSERVER } \
    --auth ProxyAuthenticationConfig_JSON_string \
    --role-arn iam_role \
    --vpc-subnet-ids space_separated_list \
    [--vpc-security-group-ids space_separated_list] \
    [--require-tls | --no-require-tls] \
    [--idle-client-timeout value] \
    [--debug-logging | --no-debug-logging] \
    [--tags comma_separated_list]
```

Para Windows:

```
aws rds create-db-proxy ^
    --db-proxy-name proxy_name ^
    --engine-family { MYSQL | POSTGRESQL | SQLSERVER } ^
    --auth ProxyAuthenticationConfig_JSON_string ^
    --role-arn iam_role ^
    --vpc-subnet-ids space_separated_list ^
    [--vpc-security-group-ids space_separated_list] ^
    [--require-tls | --no-require-tls] ^
    [--idle-client-timeout value] ^
    [--debug-logging | --no-debug-logging] ^
    [--tags comma_separated_list]
```

Veja a seguir um exemplo do valor JSON da opção `--auth`. Este exemplo aplica um tipo de autenticação de cliente diferente a cada segredo.

```
[
  {
    "Description": "proxy description 1",
    "AuthScheme": "SECRETS",
    "SecretArn": "arn:aws:secretsmanager:us-
west-2:123456789123:secret/1234abcd-12ab-34cd-56ef-1234567890ab",
    "IAMAuth": "DISABLED",
    "ClientPasswordAuthType": "POSTGRES_SCRAM_SHA_256"
  },

  {
    "Description": "proxy description 2",
    "AuthScheme": "SECRETS",
```

```
    "SecretArn": "arn:aws:secretsmanager:us-
west-2:111122223333:secret/1234abcd-12ab-34cd-56ef-1234567890cd",
    "IAMAuth": "DISABLED",
    "ClientPasswordAuthType": "POSTGRES_MD5"

  },

  {
    "Description": "proxy description 3",
    "AuthScheme": "SECRETS",
    "SecretArn": "arn:aws:secretsmanager:us-
west-2:111122221111:secret/1234abcd-12ab-34cd-56ef-1234567890ef",
    "IAMAuth": "REQUIRED"
  }

]
```

Tip

Se você ainda não souber os IDs de sub-rede que serão usados para o parâmetro `--vpc-subnet-ids`, consulte Configurar os pré-requisitos de rede para o RDS Proxy para obter exemplos de como encontrá-los.

Note

O grupo de segurança deve permitir acesso ao banco de dados ao qual o proxy se conecta. O mesmo grupo de segurança é usado para entrada das aplicações para o proxy e para saída do proxy para o banco de dados. Por exemplo, suponha que você use o mesmo grupo de segurança para o banco de dados e o proxy. Nesse caso, especifique que os recursos nesse grupo de segurança podem se comunicar com outros recursos no mesmo grupo de segurança.

Ao usar uma VPC compartilhada, você não pode usar o grupo de segurança padrão para a VPC ou um grupo de segurança pertencente a outra conta. Escolha um grupo de segurança que pertença à sua conta. Se não houver, crie um. Para obter mais informações sobre essa limitação, consulte Trabalhar com VPCs compartilhadas.

Para criar as associações corretas para o proxy, também é possível usar o comando register-db-proxy-targets. Especificar o nome do grupo de destino do `default` O proxy do RDS cria automaticamente um grupo de destino com este nome ao criar cada proxy.

```
aws rds register-db-proxy-targets
    --db-proxy-name value
    [--target-group-name target_group_name]
    [--db-instance-identifiers space_separated_list]  # rds db instances, or
    [--db-cluster-identifiers cluster_id]        # rds db cluster (all instances)
```

API do RDS

Para criar um proxy do RDS , chame a operação CreateDBProxy da API do Amazon RDS. Passe um parâmetro com a estrutura de dados AuthConfig.

O proxy do RDS cria automaticamente um grupo de destino chamado `default` ao criar cada proxy. Associe uma instância de banco de dados do RDS ao grupo de destino chamando a função RegisterDBProxyTargets.

Visualizar um proxy

Depois de criar um ou mais proxies do RDS, você pode visualizá-los e gerenciá-los no AWS Management Console, na AWS CLI ou na API do RDS. É possível analisar seus detalhes de configuração, monitorar a performance e determinar quais proxies devem ser modificados ou excluídos, conforme necessário.

Para possibilitar que as aplicações de banco de dados direcionem o tráfego por meio de um proxy, é necessário especificar o endpoint do proxy na string de conexão.

Console

Como visualizar um proxy no console

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Proxies.
3. Selecione o nome do proxy para visualizar os detalhes.
4. Na página de detalhes, a seção Grupos de destino mostra como o proxy está vinculado a uma instância de banco de dados específica do RDS. É possível acessar a página do grupo de

destino padrão para ter uma visão mais detalhada dessa associação, incluindo as configurações definidas durante a criação do proxy. Essas configurações incluem a porcentagem máxima de conexão, o tempo-limite de empréstimo da conexão, a família de mecanismos e os filtros de fixação de sessão.

CLI

Para visualizar o proxy usando a CLI, use o comando describe-db-proxies. Por padrão, a solicitação exibe todos os proxies pertencentes à sua conta da AWS. Para ver detalhes de um único proxy, especifique o nome dele com o parâmetro `--db-proxy-name`.

```
aws rds describe-db-proxies [--db-proxy-name proxy_name]
```

Para visualizar as outras informações associadas ao proxy, use os comandos a seguir.

```
aws rds describe-db-proxy-target-groups  --db-proxy-name proxy_name

aws rds describe-db-proxy-targets --db-proxy-name proxy_name
```

Use a seguinte sequência de comandos para ver mais detalhes sobre as coisas associadas ao proxy:

1. Para obter uma lista de proxies, execute describe-db-proxies.
2. Para mostrar parâmetros de conexão, como a porcentagem máxima de conexões que o proxy pode usar, execute describe-db-proxy-target-groups `--db-proxy-name`. Use o nome do proxy como o valor do parâmetro.
3. Para ver os detalhes da instância de banco de dados do RDS associada ao grupo de destino exibido, execute describe-db-proxy-targets.

API do RDS

Para visualizar seus proxies usando a API do RDS, use a operação DescribeDBProxies. Ela retorna valores do tipo de dados DBProxy.

Para ver detalhes das configurações de conexão do proxy, use os identificadores de proxy desse valor de retorno com a operação DescribeDBProxyTargetGroups. Ela retorna valores do tipo de dados DBProxyTargetGroup.

Para ver a instância do RDS ou o cluster de bancos de dados Aurora associado ao proxy, use a operação DescribeDBProxyTargets. Ela retorna valores do tipo de dados DBProxyTarget.

Conectar-se a um banco de dados pelo RDS Proxy

A maneira de se conectar a uma instância de banco de dados do RDS por meio de um proxy ou se conectar ao banco de dados, em geral, é a mesma. Para ter mais informações, consulte Visão geral dos endpoints de proxy.

Tópicos

- Conectar-se a um proxy usando autenticação nativa
- Conectar-se a um proxy usando autenticação do IAM
- Considerações sobre como se conectar a um proxy com o Microsoft SQL Server
- Considerações sobre como e conectar a um proxy com o PostgreSQL

Conectar-se a um proxy usando autenticação nativa

Use as seguintes etapas para se conectar a um proxy usando autenticação nativa:

1. Localize o endpoint do proxy. No AWS Management Console, você pode encontrar o endpoint na página de detalhes do proxy correspondente. Com a AWS CLI, é possível usar o comando describe-db-proxies. O exemplo a seguir mostra como.

```
# Add --output text to get output as a simple tab-separated list.
$ aws rds describe-db-proxies --query '*[*].
{DBProxyName:DBProxyName,Endpoint:Endpoint}'
[
    [
        {
            "Endpoint": "the-proxy.proxy-demo.us-east-1.rds.amazonaws.com",
            "DBProxyName": "the-proxy"
        },
        {
            "Endpoint": "the-proxy-other-secret.proxy-demo.us-
east-1.rds.amazonaws.com",
            "DBProxyName": "the-proxy-other-secret"
        },
        {
            "Endpoint": "the-proxy-rds-secret.proxy-demo.us-
east-1.rds.amazonaws.com",
```

```
            "DBProxyName": "the-proxy-rds-secret"
        },
        {
            "Endpoint": "the-proxy-t3.proxy-demo.us-east-1.rds.amazonaws.com",
            "DBProxyName": "the-proxy-t3"
        }
    ]
]
```

2. Especifique o endpoint como o parâmetro host na string de conexão da aplicação cliente. Por exemplo, especifique o endpoint do proxy como o valor da opção `mysql -h` ou da opção `psql -h`.
3. Forneça o mesmo nome de usuário e senha do banco de dados como o faz normalmente.

Conectar-se a um proxy usando autenticação do IAM

Ao usar a autenticação do IAM com o RDS Proxy, configure os usuários do banco de dados para fazer a autenticação com nomes de usuário e senhas regulares. A autenticação do IAM se aplica ao proxy do RDS recuperando as credenciais de nome de usuário e senha do Secrets Manager. A conexão do proxy do RDS com o banco de dados subjacente não passa pelo IAM.

Para conectar-se ao RDS Proxy usando a autenticação do IAM, siga o mesmo procedimento de conexão geral utilizado na autenticação do IAM com uma instância de banco de dados do RDS. Para obter mais informações sobre como usar a IAM, consulte Segurança no Amazon RDS.

As principais diferenças no uso do IAM para o proxy do RDS incluem o seguinte:

- Não configure cada usuário de banco de dados individual com um plug-in de autorização. Os usuários do banco de dados ainda têm nomes de usuário e senhas regulares dentro do banco de dados. Você configura segredos do Secrets Manager contendo esses nomes de usuário e senhas e autoriza o proxy do RDS a recuperar as credenciais do Secrets Manager.

  A autenticação do IAM aplica-se à conexão entre o programa cliente e o proxy. Depois, o proxy faz a autenticação no banco de dados usando as credenciais de nome de usuário e senha recuperadas pelo Secrets Manager.
- Em vez da instância, do cluster ou do endpoint de leitor, especifique o endpoint do proxy. Para obter detalhes sobre o endpoint do proxy, consulte Conectar-se à instância de banco de dados usando a autenticação do IAM.

- No caso da autenticação do IAM direta do banco de dados, você escolhe os usuários do banco de dados seletivamente e os configura para que sejam identificados com um plug-in de autenticação especial. Em seguida, pode se conectar a esses usuários com a autenticação do IAM.

  No caso de uso do proxy, é necessário fornecer a esse proxy segredos que contenham o nome de usuário e a senha de algum usuário (autenticação nativa). Em seguida, você se conecta ao proxy com a autenticação do IAM. Isso é feito gerando um token de autenticação com o endpoint do proxy, e não com o endpoint do banco de dados. Você também utiliza um nome de usuário que corresponde a um dos nomes de usuários dos segredos fornecidos.
- Certifique-se de usar os protocolos Transport Layer Security (TLS)/Secure Sockets Layer (SSL) ao se conectar a um proxy usando a autenticação do IAM.

É possível conceder a um usuário específico acesso ao proxy modificando a política do IAM. Veja a seguir um exemplo.

```
"Resource": "arn:aws:rds-db:us-east-2:1234567890:dbuser:prx-ABCDEFGHIJKL01234/db_user"
```

Considerações sobre como se conectar a um proxy com o Microsoft SQL Server

Para se conectar a um proxy usando a autenticação do IAM, você não usa o campo de senha. Em vez disso, você fornece a propriedade de token apropriada para cada tipo de driver de banco de dados no campo token. Por exemplo, use a propriedade `accessToken` para JDBC ou a propriedade `sql_copt_ss_access_token` para ODBC. Ou use a propriedade `AccessToken` do driver .NET SqlClient. Você não pode usar a autenticação do IAM com clientes que não sejam compatíveis com as propriedades do token.

Em algumas condições, um proxy não pode compartilhar uma conexão de banco de dados e, em vez disso, fixa a conexão de sua aplicação cliente ao proxy em uma conexão de banco de dados dedicada. Para ter mais informações sobre essas condições, consulte Evitar a fixação de um RDS Proxy.

Considerações sobre como e conectar a um proxy com o PostgreSQL

Se você criar um usuário de banco de dados PostgreSQL para se conectar ao RDS Proxy, conceda a ele o privilégio `CONNECT` no banco de dados. Sem isso, o usuário não poderá estabelecer uma conexão. Para ter mais informações, consulte the section called “Adicionar um novo usuário de banco de dados a um banco de dados do PostgreSQL ao usar o RDS Proxy”.

Quando um cliente inicia uma conexão com um banco de dados PostgreSQL, ele envia uma mensagem de inicialização. Essa mensagem inclui pares de strings de caracteres de nome e valor do parâmetro. Para obter mais detalhes, consulte o `StartupMessage` em PostgreSQL Message Formats na documentação do PostgreSQL.

Ao se conectar por meio de um proxy do RDS, a mensagem de inicialização pode incluir os seguintes parâmetros reconhecidos atualmente:

- `user`
- `database`

A mensagem de inicialização também pode incluir os seguintes parâmetros de tempo de execução adicionais:

- `application_name`
- `client_encoding`
- `DateStyle`
- `TimeZone`
- `extra_float_digits`
- `search_path`

Para ter mais informações sobre sistemas de mensagens PostgreSQL, consulte Frontend/Backend Protocol na documentação do PostgreSQL.

Para o PostgreSQL, se você usar JDBC, recomendamos o seguinte para evitar a fixação:

- Defina o parâmetro de conexão JDBC `assumeMinServerVersion` como pelo menos `9.0` para evitar a fixação. Isso impede que o driver JDBC execute uma viagem de ida e volta adicional durante a inicialização da conexão quando ele executa `SET extra_float_digits = 3`.
- Defina o parâmetro de conexão JDBC `ApplicationName` como *`any/your-application-name`* para evitar a fixação. Isso impede que o driver JDBC execute uma viagem de ida e volta adicional durante a inicialização da conexão quando ele executa `SET application_name = "PostgreSQL JDBC Driver"`. Observe que o parâmetro JDBC é `ApplicationName`, mas o parâmetro `StartupMessage` do PostgreSQL é `application_name`.

Para ter mais informações, consulte Evitar a fixação de um RDS Proxy. Para ter mais informações sobre como se conectar usando JDBC, consulte Connecting to the Database na documentação do PostgreSQL.

Gerenciar um RDS Proxy

Esta seção fornece informações sobre como gerenciar a operação e a configuração do RDS Proxy. Esses procedimentos ajudam a aplicação a fazer o uso mais eficiente de conexões de banco de dados e alcançar o máximo de reutilização de conexão. Quanto mais você puder aproveitar a reutilização de conexões, poderá economizar mais sobrecarga de CPU e de memória. Isso, por sua vez, reduz a latência da aplicação e permite que o banco de dados dedique mais de seus recursos ao processamento de solicitações da aplicação.

Tópicos

- Modificar um RDS Proxy
- Adicionar um novo usuário de banco de dados ao usar o RDS Proxy
- Considerações sobre a conexão do RDS Proxy
- Evitar a fixação de um RDS Proxy
- Excluir um RDS Proxy

Modificar um RDS Proxy

Você pode alterar determinadas configurações associadas a um proxy depois de criar o proxy. Faça isso modificando o próprio proxy, seu grupo de destino associado ou ambos. Cada proxy tem um grupo de destino associado.

AWS Management Console

> **Important**
>
> Os valores nos campos Client authentication type (Tipo de autenticação do cliente) e IAM authentication (Autenticação do IAM) se aplicam a todos os segredos do Secrets Manager associados a esse proxy. Para especificar valores diferentes para cada segredo, modifique seu proxy usando a AWS CLI ou a API.

Como modificar as configurações de um proxy

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Proxies.
3. Na lista de proxies, escolha o proxy cujas configurações você deseja modificar ou vá para sua página de detalhes.
4. Para Actions (Ações), escolha Modify (Modificar).
5. Insira ou escolha as propriedades a serem modificadas. Você pode modificar o seguinte:
   - Proxy identifier (Identificador de proxy): renomeie o proxy inserindo um novo identificador.
   - Idle client connection timeout (Tempo limite da conexão do cliente ociosa): insira um período para o tempo limite da conexão do cliente ociosa.
   - IAM role (Perfil do IAM): altere o perfil do IAM usada para recuperar os segredos do Secrets Manager.
   - Secrets Manager secrets (Segredos do Secrets Manager): adicione ou remova segredos do Secrets Manager. Esses segredos correspondem a nomes de usuário e senhas de banco de dados.
   - Client authentication type (Tipo de autenticação do cliente): (somente PostgreSQL) Altere o tipo de autenticação das conexões do cliente com o proxy.
   - IAM Authentication (Autenticação do IAM): exija ou desabilite a autenticação do IAM para conexões com o proxy.
   - Require Transport Layer Security (Exija Transport Layer Security): ative ou desative o requisito de segurança do Transport layer Security (TLS).
   - VPC grupo de segurança (Grupos de segurança da VPC): adicione ou remova grupos de segurança da VPC para uso do proxy.
   - Enable enhanced logging (Habilite o registro em log aprimorado): habilite ou desabilite o registro em log aprimorado.
6. Selecione Modify.

Se você não encontrou as configurações listadas que deseja alterar, use o procedimento a seguir para atualizar o grupo de destino para o proxy. O grupo de destino associado a um proxy controla as configurações relacionadas às conexões físicas do banco de dados. Cada proxy tem um grupo de

destino associado, chamado `default`, que é criado automaticamente com o proxy. Não é possível renomear o grupo de destino padrão.

Você só pode modificar o grupo de destino na página de detalhes do proxy, não na lista da página Proxies.

Como modificar as configurações de um grupo de destino de proxy

1. Na página Proxies, acesse a página de detalhes de um proxy.
2. Em Target groups (Grupos de destino), escolha o link `default`. Atualmente, todos os proxies têm um único grupo de destino chamado `default`.
3. Na página de detalhes do grupo de destino default (padrão) escolha Modify (Modificar).
4. Escolha novas configurações para as propriedades que você pode modificar:

   - Database (Banco de dados): escolha outra instância de banco de dados do RDS ou outro cluster do Aurora.
   - Connection pool maximum connections (Conexões máximas do grupo de conexões): ajuste a porcentagem das conexões máximas disponíveis que o proxy pode usar.
   - Session pinning filters (Filtros de fixação de sessões): (opcional) escolha um filtro de fixação de sessões. Isso contorna as medidas de segurança padrão para multiplexar conexões de banco de dados entre conexões de clientes. No momento, a configuração não é compatível com o PostgreSQL. A única opção é `EXCLUDE_VARIABLE_SETS`.

     Habilitar essa configuração pode fazer com que as variáveis da sessão de uma conexão afetem outras conexões. Isso pode causar erros ou problemas de correção se suas consultas dependerem dos valores das variáveis de sessão definidos fora da transação atual. Considere a possibilidade de usar essa opção depois de verificar se é seguro as aplicações compartilharem conexões de banco de dados entre conexões de clientes.

     Os seguintes padrões podem ser considerados seguros:

     - Instruções `SET` em que não há alteração no valor efetivo da variável de sessão; isto é, não há alteração na variável de sessão.
     - Você altera o valor da variável de sessão e executa uma instrução na mesma transação.

     Para ter mais informações, consulte Evitar a fixação de um RDS Proxy.
   - Connection borrow timeout (Tempo limite do empréstimo de conexões): ajuste o intervalo do tempo limite de empréstimo de conexões. Essa configuração se aplica quando o número

máximo de conexões já está sendo usado para o proxy. A configuração determina quanto tempo o proxy espera que uma conexão fique disponível antes de retornar um erro de tempo limite.

- Consulta de inicialização. (Opcional) Adicione uma consulta de inicialização ou modifique a atual. Você pode especificar uma ou mais instruções SQL para o proxy executar ao abrir cada nova conexão de banco de dados. A configuração normalmente é usada com instruções `SET` para garantir que cada conexão tenha configurações idênticas. Verifique se a consulta adicionada é válida. Para incluir diversas variáveis em uma única instrução `SET`, use vírgulas como separadores. Por exemplo:

  ```
  SET variable1=value1, variable2=value2
  ```

  Para várias instruções, use um ponto e vírgula como separador.

Não é possível alterar determinadas propriedades, como o identificador do grupo de destino e o mecanismo de banco de dados.

5. Escolha Modify target group (Modificar grupo de destino).

AWS CLI

Para modificar um proxy usando a AWS CLI, use os comandos modify-db-proxy, modify-db-proxy-target-group, deregister-db-proxy-targets e register-db-proxy-targets.

Com o comando `modify-db-proxy`, é possível alterar propriedades como as seguintes:

- O conjunto de segredos do Secrets Manager usados pelo proxy.
- Se o TLS é necessário.
- O tempo limite do cliente ocioso.
- Se deseja registrar em log informações adicionais de instruções SQL para depuração.
- O perfil do IAM usada para recuperar segredos do Secrets Manager.
- Os grupos de segurança usados pelo proxy.

O exemplo a seguir mostra como renomear um proxy existente.

```
aws rds modify-db-proxy --db-proxy-name the-proxy --new-db-proxy-name the_new_name
```

Para modificar configurações relacionadas à conexão ou renomear o grupo de destino, use o comando `modify-db-proxy-target-group`. Atualmente, todos os proxies têm um único grupo de destino chamado `default`. Ao trabalhar com esse grupo de destino, especifique o nome do proxy e `default` para o nome do grupo de destino. Não é possível renomear o grupo de destino padrão.

O exemplo a seguir mostra como verificar primeiro a configuração de `MaxIdleConnectionsPercent` de um proxy e alterá-la usando o grupo de destino.

```
aws rds describe-db-proxy-target-groups --db-proxy-name the-proxy

{
    "TargetGroups": [
        {
            "Status": "available",
            "UpdatedDate": "2019-11-30T16:49:30.342Z",
            "ConnectionPoolConfig": {
                "MaxIdleConnectionsPercent": 50,
                "ConnectionBorrowTimeout": 120,
                "MaxConnectionsPercent": 100,
                "SessionPinningFilters": []
            },
            "TargetGroupName": "default",
            "CreatedDate": "2019-11-30T16:49:27.940Z",
            "DBProxyName": "the-proxy",
            "IsDefault": true
        }
    ]
}

aws rds modify-db-proxy-target-group --db-proxy-name the-proxy --target-group-name
 default --connection-pool-config '
{ "MaxIdleConnectionsPercent": 75 }'

{
    "DBProxyTargetGroup": {
        "Status": "available",
        "UpdatedDate": "2019-12-02T04:09:50.420Z",
        "ConnectionPoolConfig": {
            "MaxIdleConnectionsPercent": 75,
            "ConnectionBorrowTimeout": 120,
            "MaxConnectionsPercent": 100,
            "SessionPinningFilters": []
```

```
        },
        "TargetGroupName": "default",
        "CreatedDate": "2019-11-30T16:49:27.940Z",
        "DBProxyName": "the-proxy",
        "IsDefault": true
    }
}
```

Com os comandos `deregister-db-proxy-targets` e , você altera a qual instância de banco de dados do RDS ou cluster de bancos de dados Aurora o proxy está associado por meio de seu grupo de destino. No momento, cada proxy pode se conectar a uma instância de banco de dados do RDS. O grupo de destino controla os detalhes da conexão de todas as instâncias de banco de dados do RDS em uma configuração multi-AZ.

O exemplo a seguir começa com um proxy associado a um cluster do Aurora MySQL chamado `cluster-56-2020-02-25-1399`. O exemplo mostra como alterar o proxy para que ele possa se conectar a outro cluster chamado `provisioned-cluster`.

Ao trabalhar com uma instância de banco de dados do RDS, você especifica a opção `--db-instance-identifier`.

O exemplo a seguir modifica um proxy Aurora MySQL. Um proxy Aurora PostgreSQL tem a porta 5432.

```
aws rds describe-db-proxy-targets --db-proxy-name the-proxy

{
    "Targets": [
        {
            "Endpoint": "instance-9814.demo.us-east-1.rds.amazonaws.com",
            "Type": "RDS_INSTANCE",
            "Port": 3306,
            "RdsResourceId": "instance-9814"
        },
        {
            "Endpoint": "instance-8898.demo.us-east-1.rds.amazonaws.com",
            "Type": "RDS_INSTANCE",
            "Port": 3306,
            "RdsResourceId": "instance-8898"
        },
        {
            "Endpoint": "instance-1018.demo.us-east-1.rds.amazonaws.com",
```

```
            "Type": "RDS_INSTANCE",
            "Port": 3306,
            "RdsResourceId": "instance-1018"
        },
        {
            "Type": "TRACKED_CLUSTER",
            "Port": 0,
            "RdsResourceId": "cluster-56-2020-02-25-1399"
        },
        {
            "Endpoint": "instance-4330.demo.us-east-1.rds.amazonaws.com",
            "Type": "RDS_INSTANCE",
            "Port": 3306,
            "RdsResourceId": "instance-4330"
        }
    ]
}

aws rds deregister-db-proxy-targets --db-proxy-name the-proxy --db-cluster-identifier
 cluster-56-2020-02-25-1399

aws rds describe-db-proxy-targets --db-proxy-name the-proxy

{
    "Targets": []
}

aws rds register-db-proxy-targets --db-proxy-name the-proxy --db-cluster-identifier
 provisioned-cluster

{
    "DBProxyTargets": [
        {
            "Type": "TRACKED_CLUSTER",
            "Port": 0,
            "RdsResourceId": "provisioned-cluster"
        },
        {
            "Endpoint": "gkldje.demo.us-east-1.rds.amazonaws.com",
            "Type": "RDS_INSTANCE",
            "Port": 3306,
            "RdsResourceId": "gkldje"
        },
        {
```

```
            "Endpoint": "provisioned-1.demo.us-east-1.rds.amazonaws.com",
            "Type": "RDS_INSTANCE",
            "Port": 3306,
            "RdsResourceId": "provisioned-1"
        }
    ]
}
```

API do RDS

Para modificar um proxy utilizando a API do RDS, utilize as operações ModifyDBProxy, ModifyDBProxyTargetGroup, DeregisterDBProxyTargets e RegisterDBProxyTargets.

Com `ModifyDBProxy`, é possível alterar propriedades como as seguintes:

- O conjunto de segredos do Secrets Manager usados pelo proxy.
- Se o TLS é necessário.
- O tempo limite do cliente ocioso.
- Se deseja registrar em log informações adicionais de instruções SQL para depuração.
- O perfil do IAM usada para recuperar segredos do Secrets Manager.
- Os grupos de segurança usados pelo proxy.

Com `ModifyDBProxyTargetGroup`, é possível modificar as configurações relacionadas à conexão. Atualmente, todos os proxies têm um único grupo de destino chamado `default`. Ao trabalhar com esse grupo de destino, especifique o nome do proxy e `default` para o nome do grupo de destino. Não é possível renomear o grupo de destino padrão.

Com `DeregisterDBProxyTargets` e `RegisterDBProxyTargets`, altere a qual instância de banco de dados do RDS o proxy está associado por meio do grupo de destino. Atualmente, cada proxy pode se conectar a uma instância de banco de dados do RDS ou cluster de bancos de dados Aurora. O grupo de destino controla os detalhes da conexão de todas as instâncias de banco de dados do RDS em uma configuração multi-AZ ou todas as instâncias de banco de dados em um cluster do Aurora.

Adicionar um novo usuário de banco de dados ao usar o RDS Proxy

Em alguns casos, é possível adicionar um novo usuário de banco de dados a uma instância de banco de dados do RDS ou um cluster do Aurora associado a um proxy. Nesse caso, adicione ou

reformule um segredo do Secrets Manager para armazenar as credenciais desse usuário. Para isso, siga estas etapas:

1. Crie um segredo do Secrets Manager usando o procedimento descrito em Configurar credenciais de banco de dados no AWS Secrets Manager para RDS Proxy.
2. Atualize o perfil do IAM para conceder ao proxy do RDS acesso ao novo segredo do Secrets Manager. Para fazer isso, atualize a seção de recursos da política de perfil do IAM.
3. Modifique o RDS Proxy para adicionar o novo segredo do Secrets Manager em Segredos do Secrets Manager.
4. Se o novo usuário tomar o lugar de um existente, atualize as credenciais armazenadas no segredo do Secrets Manager do proxy do usuário existente.

Adicionar um novo usuário de banco de dados a um banco de dados do PostgreSQL ao usar o RDS Proxy

Ao adicionar um novo usuário ao banco de dados do PostgreSQL, se você tiver executado o seguinte comando:

```
REVOKE CONNECT ON DATABASE postgres FROM PUBLIC;
```

Conceda ao usuário `rdsproxyadmin` o privilégio `CONNECT` para que ele possa monitorar as conexões no banco de dados de destino.

```
GRANT CONNECT ON DATABASE postgres TO rdsproxyadmin;
```

Também é possível permitir que outros usuários do banco de dados de destino realizem verificações de integridade mudando `rdsproxyadmin` para o usuário do banco de dados no comando acima.

Alterar a senha de um usuário de banco de dados ao usar o RDS Proxy

Em alguns casos, você pode alterar a senha de um usuário de banco de dados em uma instância de banco de dados do RDS ou cluster do Aurora associado a um proxy. Nesse caso, atualize o segredo do Secrets Manager correspondente com a nova senha.

Considerações sobre a conexão do RDS Proxy

Configurar configurações de conexões

Para ajustar o grupo de conexões do RDS Proxy, você pode modificar as seguintes configurações:

- IdleClientTimeout
- MaxConnectionsPercent
- MaxIdleConnectionsPercent
- ConnectionBorrowTimeout

IdleClientTimeout

É possível especificar por quanto tempo uma conexão de cliente pode ficar antes que o proxy a feche. O padrão é de 1.800 segundos (30 minutos).

Uma conexão de cliente é considerada ociosa quando a aplicação não envia uma nova solicitação dentro do tempo especificado após a conclusão da solicitação anterior. A conexão de banco de dados subjacente permanece aberta e é retornada ao grupo de conexões. Portanto, ela está disponível para ser reutilizada para novas conexões de cliente. Reduza o tempo limite de conexão do cliente ocioso se desejar que o proxy remova proativamente as conexões obsoletas. Se a workload estabelecer conexões frequentes com o proxy, aumente o tempo limite de conexão do cliente ocioso para reduzir o custo de estabelecimento de conexões.

Essa configuração é representada pelo Tempo limite de conexão do cliente ocioso no console do RDS e na configuração `IdleClientTimeout` na AWS CLI e a API. Para saber como alterar o valor do Tempo limite de conexão do cliente ocioso no console do RDS, consulte AWS Management Console. Para saber como alterar o valor da configuração `IdleClientTimeout`, use o comando da CLI modify-db-proxy ou a operação da API ModifyDBProxy.

MaxConnectionsPercent

Você pode limitar o número de conexões que um Proxy RDS pode estabelecer com o banco de dados de destino. Você especifica o limite como uma porcentagem do máximo de conexões disponíveis para o banco de dados. Essa configuração é representada pelo campo Connection pool maximum connections (Número máximo de conexões para o grupo de conexões) no console do proxy do RDS ou pelo parâmetro `MaxConnectionsPercent` na AWS CLI e na API.

O valor `MaxConnectionsPercent` é expresso como uma porcentagem da configuração `max_connections` da instância de banco de dados do RDS usada pelo grupo de destino. O proxy não cria todas essas conexões com antecedência. Essa configuração permite ao proxy estabelecer essas conexões conforme a necessidade da workload.

Por exemplo, para um destino de banco de dados registrado com `max_connections` definido como 1.000 e `MaxConnectionsPercent` definido como 95, o RDS Proxy define 950 conexões como o limite superior para conexões simultâneas com esse destino de banco de dados.

Um efeito colateral comum de sua workload atingir o número máximo de conexões de banco de dados permitidas é um aumento na latência geral da consulta, bem como um aumento na métrica `DatabaseConnectionsBorrowLatency`. Você pode monitorar as conexões de banco de dados usadas atualmente e o total permitido comparando as métricas `DatabaseConnections` e `MaxDatabaseConnectionsAllowed`.

Ao definir esse parâmetro, observe as seguintes práticas recomendadas:

- Permita espaço de conexão suficiente para mudanças no padrão da workload. É recomendável definir o parâmetro pelo menos 30% acima do seu uso máximo monitorado recentemente. Como o RDS Proxy redistribui as cotas de conexão do banco de dados em vários nós, as mudanças na capacidade interna podem exigir pelo menos 30% de espaço livre para conexões adicionais, a fim de evitar o aumento das latências de empréstimo.
- O RDS Proxy reserva um certo número de conexões para monitoramento ativo para comportar falhas rápidas, roteamento de tráfego e operações internas. A métrica `MaxDatabaseConnectionsAllowed` não inclui essas conexões reservadas. Ela representa o número de conexões disponíveis para atender à workload e pode ser menor do que o valor derivado da configuração `MaxConnectionsPercent`.

  Valores `MaxConnectionsPercent` mínimos recomendados

  - db.t3.small: 30
  - db.t3.medium ou superior: 20

Para saber como alterar o valor Connection pool maximum connections (Número máximo de conexões para o grupo de conexões) no console do RDS, consulte AWS Management Console. Para saber como alterar o valor de `MaxConnectionsPercent`, consulte o comando da CLI modify-db-proxy-target-group ou a operação da API ModifyDBProxyTargetGroup.

Para obter informações sobre os limites de conexões de banco de dados, consulte Número máximo de conexões de banco de dados.

MaxIdleConnectionsPercent

Você pode controlar o número de conexões de banco de dados ociosas que o proxy do RDS pode manter no grupo de conexões. Por padrão, o RDS Proxy considera uma conexão de banco de dados no grupo como ociosa quando não há atividade na conexão por cinco minutos.

O valor `MaxIdleConnectionsPercent` é expresso como uma porcentagem da configuração `max_connections` do grupo de destino da instância de banco de dados do RDS. O valor padrão é 50% de `MaxConnectionsPercent` e o limite superior é o valor de `MaxConnectionsPercent`. Por exemplo, se `MaxConnectionsPercent` for 80, o valor padrão de `MaxIdleConnectionsPercent` será 40. Se o valor de `MaxConnectionsPercent` não for especificado, para o SQL Server, `MaxIdleConnectionsPercent` será 5 e, para todos os outros mecanismos, o padrão será 50.

Com um valor alto, o proxy deixa uma alta porcentagem de conexões de banco de dados ociosas abertas. Com um valor baixo, o proxy fecha uma alta porcentagem de conexões de banco de dados ociosas. Se suas workloads forem imprevisíveis, considere definir um valor alto para `MaxIdleConnectionsPercent`. Isso significa que o RDS Proxy pode acomodar picos de atividade sem abrir muitas novas conexões de banco de dados.

Essa configuração é representada pela configuração `MaxIdleConnectionsPercent` do `DBProxyTargetGroup` na AWS CLI e na API. Para saber como alterar o valor de `MaxIdleConnectionsPercent`, consulte o comando da CLI modify-db-proxy-target-group ou a operação da API ModifyDBProxyTargetGroup.

Para obter informações sobre os limites de conexões de banco de dados, consulte Número máximo de conexões de banco de dados.

ConnectionBorrowTimeout

É possível especificar quanto tempo o proxy do RDS espera que uma conexão de banco de dados no grupo de conexão fique disponível antes de retornar um erro de tempo limite. O padrão é 120 segundos. Esta configuração se aplica quando o número de conexões está no máximo e, portanto, nenhuma conexão está disponível no grupo de conexões. Isso também se aplica caso nenhuma instância de banco de dados apropriada esteja disponível para lidar com a solicitação, por exemplo, quando uma operação de failover está em andamento. Usando essa configuração, é possível definir o melhor período de espera para a aplicação sem precisar alterar o tempo limite da consulta no código.

Essa configuração é representada pelo campo Connection borrow timeout (Tempo limite de empréstimo de conexões) no console do RDS ou pela configuração `ConnectionBorrowTimeout` de `DBProxyTargetGroup` na AWS CLI ou API. Para saber como alterar o valor do campo Connection borrow timeout (Tempo limite de empréstimo de conexões) no console do RDS, consulte AWS Management Console. Para saber como alterar o valor de `ConnectionBorrowTimeout`, consulte o comando da CLI modify-db-proxy-target-group ou a operação da API ModifyDBProxyTargetGroup.

Conexões de cliente e banco de dados

As conexões do seu aplicativo com o RDS Proxy são conhecidas como conexões de cliente. As conexões de um proxy com o banco de dados são conexões de banco de dados. Ao usar o RDS Proxy, as conexões do cliente terminam no proxy, enquanto as conexões do banco de dados são gerenciadas no RDS Proxy.

O grupo de conexões do lado da aplicação pode oferecer o benefício de reduzir o estabelecimento de conexão recorrente entre a aplicação e o RDS Proxy.

Pense nos seguintes aspectos de configuração antes de implementar um grupo de conexões do lado da aplicação:

- Vida útil máxima da conexão do cliente: o RDS Proxy impõe uma vida útil máxima das conexões do cliente de 24 horas. Esse valor não é configurável. Configure o grupo com uma vida útil máxima de conexão inferior a 24 horas para evitar quedas inesperadas na conexão do cliente.
- Tempo limite de inatividade da conexão do cliente: o RDS Proxy impõe um tempo máximo de inatividade para as conexões do cliente. Configure seu pool com um tempo limite de conexão ociosa de um valor menor do que a configuração de tempo limite de inatividade da conexão do cliente para o RDS Proxy para evitar quedas inesperadas de conexão.

O número máximo de conexões de cliente configuradas no grupo de conexões do lado da aplicação não precisa ser limitado à configuração max_connections do RDS Proxy.

O grupo de conexões do cliente resulta em maior vida útil da conexão do cliente. Se suas conexões experimentarem fixação, o agrupamento de conexões de clientes pode reduzir a eficiência da multiplexação. As conexões de cliente fixas, mas ociosas, no grupo de conexões do lado da aplicação continuam mantendo uma conexão com o banco de dados e impedem que a conexão seja reutilizada por outras conexões do cliente. Analise os logs do proxy para conferir se as conexões estão sendo fixadas.

> **Note**
>
> O proxy do RDS encerra as conexões com o banco de dados depois de 24 horas, quando elas não estiverem mais em uso. O proxy executa essa ação independentemente do valor da configuração máxima de conexões ociosas.

Evitar a fixação de um RDS Proxy

A multiplexação é mais eficiente quando as solicitações de banco de dados não dependem de informações de estado de solicitações anteriores. Nesse caso, o proxy do RDS pode reutilizar uma conexão na conclusão de cada transação. Exemplos de tais informações de estado incluem a maioria das variáveis e parâmetros de configuração que você pode alterar por meio de instruções `SET` ou `SELECT`. Por padrão, as transações SQL em uma conexão de cliente podem multiplexar entre conexões de banco de dados subjacentes.

Suas conexões com o proxy podem entrar em um estado conhecido como fixação. Quando uma conexão é fixada, cada transação posterior usa a mesma conexão de banco de dados subjacente até que a sessão termine. Outras conexões de cliente também não podem reutilizar essa conexão de banco de dados até que a sessão termine. A sessão termina quando a conexão do cliente é descartada.

O proxy do RDS automaticamente fixa uma conexão de cliente a uma conexão de banco de dados específica quando detecta uma alteração no estado de sessão que não é apropriada para outras sessões. A fixação reduz a eficácia da reutilização de conexões. Se todas ou quase todas as suas conexões forem fixadas, considere modificar o código da aplicação ou a workload para reduzir as condições que provocam a fixação.

Por exemplo, a aplicação altera uma variável de sessão ou um parâmetro de configuração. Nesse caso, as declarações posteriores poderão contar que a nova variável ou parâmetro estará em vigor. Assim, quando o proxy do RDS processa solicitações para alterar variáveis ou definições da configuração da sessão, ele fixa essa sessão para a conexão com o banco de dados. Dessa forma, o estado da sessão permanece em vigor para todas as transações posteriores na mesma sessão.

Em alguns mecanismos de banco de dados, essa regra não se aplica a todos os parâmetros que você pode definir. O proxy do RDS monitora determinadas instruções e variáveis. Portanto, o RDS Proxy não fixa a sessão quando você as modifica. Nesse caso, o proxy do RDS apenas reutiliza a conexão para outras sessões que tenham os mesmos valores para essas configurações. Para obter

detalhes sobre o que o RDS Proxy monitora para um mecanismo de banco de dados, consulte o seguinte:

- O que o RDS Proxy monitora para bancos de dados do RDS para SQL Server
- O que o RDS Proxy monitora para bancos de dados do RDS para MariaDB e o RDS para MySQL

O que o RDS Proxy monitora para bancos de dados do RDS para SQL Server

Veja a seguir as declarações do SQL Server que o RDS Proxy monitora:

- `USE`
- `SET ANSI_NULLS`
- `SET ANSI_PADDING`
- `SET ANSI_WARNINGS`
- `SET ARITHABORT`
- `SET CONCAT_NULL_YIELDS_NULL`
- `SET CURSOR_CLOSE_ON_COMMIT`
- `SET DATEFIRST`
- `SET DATEFORMAT`
- `SET LANGUAGE`
- `SET LOCK_TIMEOUT`
- `SET NUMERIC_ROUNDABORT`
- `SET QUOTED_IDENTIFIER`
- `SET TEXTSIZE`
- `SET TRANSACTION ISOLATION LEVEL`

O que o RDS Proxy monitora para bancos de dados do RDS para MariaDB e o RDS para MySQL

Veja a seguir as declarações do MySQL e do MariaDB que o RDS Proxy monitora:

- DROP DATABASE
- DROP SCHEMA

- USE

Veja a seguir as variáveis do MySQL e MariaDB que o RDS Proxy monitora:

- AUTOCOMMIT
- AUTO_INCREMENT_INCREMENT
- CHARACTER SET (or CHAR SET)
- CHARACTER_SET_CLIENT
- CHARACTER_SET_DATABASE
- CHARACTER_SET_FILESYSTEM
- CHARACTER_SET_CONNECTION
- CHARACTER_SET_RESULTS
- CHARACTER_SET_SERVER
- COLLATION_CONNECTION
- COLLATION_DATABASE
- COLLATION_SERVER
- INTERACTIVE_TIMEOUT
- NAMES
- NET_WRITE_TIMEOUT
- QUERY_CACHE_TYPE
- SESSION_TRACK_SCHEMA
- SQL_MODE
- TIME_ZONE
- TRANSACTION_ISOLATION (or TX_ISOLATION)
- TRANSACTION_READ_ONLY (or TX_READ_ONLY)
- WAIT_TIMEOUT

Reduzir a fixação

O ajuste de performance do proxy do RDS envolve a tentativa de maximizar a reutilização de conexões em nível de transação (multiplexação) minimizando a fixação.

É possível reduzir a fixação da seguinte maneira:

- Evite solicitações de banco de dados desnecessárias que podem causar fixação.
- Defina variáveis e definições de configuração de forma consistente em todas as conexões. Dessa forma, as sessões posteriores têm mais probabilidade de reutilizar conexões que têm essas configurações específicas.

  No entanto, para a configuração do PostgreSQL, uma variável leva à fixação da sessão.
- Para um banco de dados da família de mecanismos do MySQL, aplique um filtro de fixação de sessão ao proxy. Você pode isentar certos tipos de operações de fixar a sessão se souber que isso não afeta o funcionamento correto da aplicação.
- Veja com que frequência a fixação ocorre monitorando a métrica `DatabaseConnectionsCurrentlySessionPinned` do Amazon CloudWatch. Para obter informações sobre essa e outras métricas do CloudWatch, consulte Monitorar métricas do proxy do RDS com o Amazon CloudWatch.
- Se você usar instruções `SET` para executar uma inicialização idêntica para cada conexão de cliente, poderá fazê-lo enquanto preserva a multiplexação no nível de transação. Nesse caso, você move as instruções que configuram o estado da sessão inicial para a consulta de inicialização usada por um proxy. Esta propriedade é uma string que contém uma ou mais instruções SQL, separadas por ponto e vírgula.

  Por exemplo, você pode definir uma consulta de inicialização para um proxy que define determinados parâmetros de configuração. O proxy do RDS aplica essas configurações sempre que configura uma nova conexão para esse proxy. Você pode remover as instruções `SET` correspondentes do código de sua aplicação, para que elas não interfiram na multiplexação em nível de transação.

  Para ver métricas sobre a frequência com que a fixação ocorre para um proxy, consulte Monitorar métricas do proxy do RDS com o Amazon CloudWatch.

Condições que causam fixação para todas as famílias de mecanismos

O proxy fixa a sessão à conexão atual nas seguintes situações em que a multiplexação pode causar um comportamento inesperado:

- Qualquer instrução com um tamanho de texto maior do que 16 KB faz com que o proxy fixe a sessão.

Condições que podem causar fixação no RDS para Microsoft SQL Server

Para o RDS para SQL Server, as seguintes interações também geram fixação:

- Utilizando vários conjuntos de resultados ativos (MARS). Para obter informações sobre MARS, consulte a documentação do SQL Server.
- Utilizando a comunicação do coordenador de transações distribuídas (DTC).
- Criando tabelas temporárias, transações, cursores ou declarações preparadas.
- Usando as seguintes declarações `SET`:
  - `SET ANSI_DEFAULTS`
  - `SET ANSI_NULL_DFLT`
  - `SET ARITHIGNORE`
  - `SET DEADLOCK_PRIORITY`
  - `SET FIPS_FLAGGER`
  - `SET FMTONLY`
  - `SET FORCEPLAN`
  - `SET IDENTITY_INSERT`
  - `SET NOCOUNT`
  - `SET NOEXEC`
  - `SET OFFSETS`
  - `SET PARSEONLY`
  - `SET QUERY_GOVERNOR_COST_LIMIT`
  - `SET REMOTE_PROC_TRANSACTIONS`
  - `SET ROWCOUNT`
  - `SET SHOWPLAN_ALL`, `SHOWPLAN_TEXT`, e `SHOWPLAN_XML`
  - `SET STATISTICS`
  - `SET XACT_ABORT`

Condições que causam fixação no RDS para MariaDB e RDS para MySQL

Para o MySQL e o MariaDB, as seguintes interações também podem causar fixação:

- As instruções explícitas de bloqueio de tabela `LOCK TABLE`, `LOCK TABLES` ou `FLUSH TABLES WITH READ LOCK` fazem com que o proxy fixe a sessão.
- A criação de bloqueios nomeados usando `GET_LOCK` faz com que o proxy fixe a sessão.
- A definição de uma variável de usuário ou definir uma variável de sistema (com algumas exceções) faz com que o proxy fixe a sessão. Se essa situação reduzir muito a reutilização de conexões, determine que as operações `SET` não causem fixação. Para obter informações sobre como fazer isso definindo a propriedade de fixação de filtros, consulte Criar um proxy para o Amazon RDS e Modificar um RDS Proxy.
- A criação de uma tabela temporária faz com que o proxy fixe a sessão. Dessa forma, o conteúdo da tabela temporária é preservado durante toda a sessão, independentemente dos limites de transação.
- A chamada das funções `ROW_COUNT`, `FOUND_ROWS` e `LAST_INSERT_ID` às vezes causa fixação.
- As instruções preparadas fazem com que o proxy fixe a sessão. Essa regra se aplicará se a instrução preparada usar texto SQL ou o protocolo binário.
- O RDS Proxy não fixa conexões quando você usa SET LOCAL.
- A chamada de procedimentos armazenados e de funções armazenadas não causa fixação. O proxy do RDS não detecta nenhuma alteração de estado de sessão resultante dessas chamadas. Verifique se a aplicação não altera o estado da sessão dentro de rotinas armazenadas se você confia nesse estado de sessão para persistir entre transações. Por exemplo, no momento, o proxy do RDS não é compatível com um procedimento armazenado que cria uma tabela temporária que persiste em todas as transações.

Se você tiver conhecimento especializado sobre o comportamento da aplicação, poderá ignorar o comportamento de fixação de determinadas instruções da aplicação. Para fazer isso, escolha a opção Filtros de fixação de sessão ao criar o proxy. Atualmente, é possível cancelar a fixação de sessão para definir variáveis de sessão e configurações.

Condições que causam fixação no RDS para PostgreSQL

Para o PostgreSQL, as seguintes interações também geram a fixação:

- Usar comandos `SET`.
- Usar comandos `PREPARE`, `DISCARD`, `DEALLOCATE` ou `EXECUTE` para gerenciar instruções preparadas.
- Criar sequências, tabelas ou visualizações temporárias.

- Declarar cursores.
- Descartar o estado da sessão.
- Escutar em um canal de notificação.
- Carregar um módulo de biblioteca, como `auto_explain`.
- Manipular sequências usando funções como `nextval` e `setval`.
- Interagir com bloqueios usando funções como `pg_advisory_lock` e `pg_try_advisory_lock`.

> **Note**
>
> O RDS Proxy não fixa bloqueios consultivos em nível de transação, especificamente `pg_advisory_xact_lock`, `pg_advisory_xact_lock_shared`, `pg_try_advisory_xact_lock` e `pg_try_advisory_xact_lock_shared`.

- Definir ou redefinir um parâmetro como o padrão. Especificamente, usar comandos `SET` e `set_config` para atribuir valores padrão às variáveis da sessão.
- A chamada de procedimentos armazenados e de funções armazenadas não causa fixação. O proxy do RDS não detecta nenhuma alteração de estado de sessão resultante dessas chamadas. Verifique se a aplicação não altera o estado da sessão dentro de rotinas armazenadas se você confia nesse estado de sessão para persistir entre transações. Por exemplo, no momento, o proxy do RDS não é compatível com um procedimento armazenado que cria uma tabela temporária que persiste em todas as transações.

Excluir um RDS Proxy

Será possível excluir um proxy quando não precisar mais dele. Ou poderá excluir um proxy se você tirar de serviço a instância ou o cluster de banco de dados associado.

AWS Management Console

Como excluir um proxy

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Proxies.
3. Escolha o proxy a ser excluído da lista.

4. Escolha Delete Proxy (Excluir proxy).

AWS CLI

Para excluir um proxy de banco de dados, use o comando delete-db-proxy da AWS CLI. Para remover associações relacionadas, use também o comando deregister-db-proxy-targets.

```
aws rds delete-db-proxy --name proxy_name
```

```
aws rds deregister-db-proxy-targets
    --db-proxy-name proxy_name
    [--target-group-name target_group_name]
    [--target-ids comma_separated_list]       # or
    [--db-instance-identifiers instance_id]       # or
    [--db-cluster-identifiers cluster_id]
```

API do RDS

Para excluir um proxy de banco de dados, chame a função DeleteDBProxy da API do Amazon RDS. Para excluir itens e associações relacionados, você também chama as funções DeleteDBProxyTargetGroup e DeregisterDBProxyTargets.

Como trabalhar com endpoints do proxy do Amazon RDS

Os endpoints do RDS Proxy oferecem maneiras flexíveis e eficientes de gerenciar conexões de banco de dados, o que melhora a escalabilidade, disponibilidade e segurança. Com endpoints de proxy, é possível:

- Simplificar o monitoramento e a solução de problemas: use vários endpoints para monitorar e gerenciar conexões de diferentes aplicações de maneira independente.

Tópicos

- Visão geral dos endpoints de proxy
- Limitações de endpoints de proxy
- Endpoints de proxy para clusters de banco de dados multi-AZ
- Acesso aos bancos de dados do Aurora e do RDS entre VPCs

- Criação de um endpoint de proxy
- Visualização dos endpoints de proxy
- Modificação de um endpoint de proxy
- Exclusão de um endpoint de proxy

Visão geral dos endpoints de proxy

Trabalhar com endpoints de proxy do RDS envolve os mesmos tipos de procedimentos usados com cluster de bancos de dados Aurora, endpoints de leitor e endpoints de instâncias RDS. Se você não tem familiaridade com os endpoints do RDS, encontre mais informações em Conexão a uma instância de banco de dados executando o mecanismo de banco de dados do MySQL e Conectar a uma instância de banco de dados executando o mecanismo de banco de dados do PostgreSQL.

Ao criar um endpoint de proxy, você pode associá-lo a uma nuvem privada virtual (VPC) diferente da utilizada pelo proxy. Isso possibilita que você se conecte ao proxy por meio de outra VPC, como uma usada por uma aplicação diferente em sua organização.

Para obter informações sobre limites associados aos endpoints de proxy, consulte Limitações de endpoints de proxy.

Os logs do RDS Proxy incluem como prefixo em cada entrada o nome do endpoint de proxy associado. Pode ser o nome especificado para um endpoint definido pelo usuário ou o nome especial `default` para o endpoint padrão de leitura/gravação do proxy.

Cada endpoint de proxy tem seu próprio conjunto de métricas do CloudWatch. Monitore métricas de todos os endpoints específicos, de um endpoint específico ou de todos os endpoints de leitura/gravação ou somente leitura de um proxy. Para obter mais informações, consulte Monitorar métricas do proxy do RDS com o Amazon CloudWatch.

Um endpoint de proxy usa o mesmo mecanismo de autenticação que o proxy associado. O proxy do RDS configura automaticamente permissões e autorizações para o endpoint definido pelo usuário, consistentes com as propriedades do proxy associado.

Limitações de endpoints de proxy

Os endpoints do RDS Proxy têm as seguintes limitações:

- O endpoint padrão do proxy do RDS não pode ser modificado.

- O número máximo de endpoints definidos pelo usuário para um proxy é 20. Assim, um proxy pode ter até 21 endpoints: o endpoint padrão e mais 20 que você cria.
- Quando você associa endpoints adicionais a um proxy, o proxy do RDS determina automaticamente quais instâncias de banco de dados em seu cluster usar para cada endpoint. Você não pode escolher instâncias específicas da mesma maneira que escolhe endpoints personalizados do Aurora.

Quando você cria um proxy, o RDS cria automaticamente um endpoint da VPC para comunicação segura entre as aplicações e o banco de dados. O endpoint da VPC é visível e pode ser acessado no console da Amazon VPC.

A adição de um novo endpoint de proxy provisiona um endpoint de interface do AWS PrivateLink. Se você adicionar um ou mais endpoints ao proxy, incorrerá em cobranças adicionais. Para ter mais informações, consulte Preços do Amazon Relational Database Service Proxy.

Endpoints de proxy para clusters de banco de dados multi-AZ

Por padrão, o endpoint ao qual você se conecta quando usa o proxy do RDS com um cluster do Aurora tem capacidade de leitura/gravação. Como resultado, esse endpoint envia todas as solicitações à instância do gravador do cluster. Todas essas conexões são consideradas no valor `max_connections` da instância do gravador. Se o seu proxy estiver associado a um cluster de banco de dados do , você pode criar endpoints adicionais de leitura/gravação ou somente leitura para esse proxy.

Você pode usar um endpoint somente leitura com seu proxy para consultas somente leitura. Você faz isso da mesma forma que usa o endpoint do leitor para um cluster provisionado do Aurora. Fazer isso ajuda você a aproveitar o dimensionamento de leitura de um cluster do com uma ou mais instâncias de banco de dados do leitor. Você pode executar mais consultas simultâneas e fazer mais conexões simultâneas usando um endpoint somente leitura e adicionando mais instâncias de banco de dados do leitor ao cluster do , conforme necessário. Esses endpoints de leitor ajudam a melhorar a escalabilidade de leitura de suas aplicações com uso intensivo de consultas. Os endpoints de leitor também ajudam a melhorar a disponibilidade de suas conexões, se uma instância de banco de dados do leitor no cluster ficar indisponível.

Endpoints de leitor para clusters de banco de dados Multi-AZ

Com o RDS Proxy, você pode criar e usar endpoints de leitor. No entanto, esses endpoints funcionam apenas para proxies associados a clusters de banco de dados do . Se você usa a CLI ou

a API do RDS, pode ver o atributo `TargetRole` com um valor `READ_ONLY`. É possível aproveitar esses proxies alterando o destino de um proxy de uma instância de banco de dados do RDS para um cluster de banco de dados multi-AZ.

Você pode criar e se conectar a endpoints somente leitura chamados endpoints de leitor quando usa o proxy do RDS com clusters do Aurora.

Como os endpoints de leitor ajudam a disponibilidade das aplicações

Em alguns casos, uma ou mais instâncias de leitor no cluster podem ficar indisponíveis. Nesses casos, as conexões que usam um endpoint leitor de um proxy de banco de dados podem se recuperar mais rapidamente do que aquelas que usam o endpoint leitor do Aurora. O proxy do RDS roteia conexões somente para as instâncias de leitor disponíveis no cluster. Não há um atraso causado pelo cache de DNS quando uma instância fica indisponível.

Se a conexão for multiplexada, o proxy do RDS direciona as consultas subsequentes para uma instância de banco de dados de leitor diferente, sem qualquer interrupção na aplicação. Se uma instância do leitor estiver em um estado indisponível, todas as conexões do cliente com esse endpoint da instância serão fechadas.

Se a conexão estiver fixada, a próxima consulta na conexão retornará um erro. Porém, a aplicação pode se reconectar imediatamente ao mesmo endpoint de proxy. O proxy do RDS roteia a conexão para uma instância de banco de dados de leitor diferente que esteja no estado `available`. Quando você se reconecta manualmente, o proxy do RDS não verifica o atraso de replicação entre as instâncias antigas e novas do leitor.

Se o seu cluster de banco de dados Multi-AZ não tiver nenhuma instância de leitor disponível, o RDS Proxy tentará se conectar a um endpoint de leitor quando ele estiver disponível. Se nenhuma instância de leitor ficar disponível dentro do tempo limite de empréstimo da conexão, a tentativa de conexão falhará. Se uma instância de leitor ficar disponível, a tentativa de conexão será bem-sucedida.

Como os endpoints de leitor contribuem para a escalabilidade da consulta

Os endpoints de leitor para um proxy ajudam no dimensionamento da consulta do das seguintes maneiras:

- Onde for prático, o proxy do RDS usa a mesma instância de banco de dados de leitor para todos os problemas de consultas usando uma conexão de endpoint de leitor específica. Dessa forma, um

conjunto de consultas relacionadas nas mesmas tabelas pode aproveitar o cache, a otimização do plano e assim por diante em uma instância de banco de dados específica.

- Se uma instância de banco de dados de leitor ficar indisponível, o efeito sobre a aplicação dependerá se a sessão for multiplexada ou fixada. Se a sessão for multiplexada, o proxy do RDS roteia todas as consultas subsequentes para uma instância de banco de dados de leitor diferente sem qualquer ação de sua parte. Se a sessão estiver fixada, a aplicação receberá um erro e deverá se reconectar. Você pode se reconectar ao endpoint de leitor imediatamente e o proxy do RDS roteará a conexão para uma instância de banco de dados de leitor disponível. Para obter mais informações sobre multiplexação e fixação para sessões de proxy, consulte Visão geral dos conceitos do RDS Proxy.

Acesso aos bancos de dados do Aurora e do RDS entre VPCs

Por padrão, os componentes da pilha de tecnologia do RDS e do Aurora estão todos na mesma Amazon VPC. Por exemplo, suponha que uma aplicação em execução em uma instância do Amazon EC2 se conecta a uma instância de banco de dados do Amazon RDS ou cluster de banco de dados doAurora. Nesse caso, o servidor da aplicação e o banco de dados devem estar dentro da mesma VPC.

Com o RDS Proxy, é possível configurar o acesso a uma instância de banco de dados do Amazon RDS em uma VPC a partir de recursos em outra VPC, como instâncias do EC2. Por exemplo, sua organização pode ter várias aplicações que acessam os mesmos recursos de banco de dados. Cada aplicação pode estar em sua própria VPC.

Para habilitar o acesso entre VPCs, crie um novo endpoint para o proxy. O proxy em si reside na mesma VPC que o cluster de bancos de dados Aurora ou instância do RDS. No entanto, o endpoint entre VPCs reside na outra VPC, juntamente com os outros recursos, como as instâncias do EC2. O endpoint entre VPC está associado a sub-redes e grupos de segurança da mesma VPC que o EC2 e outros recursos. Essas associações permitem que você se conecte ao endpoint a partir das aplicações que, de outra forma, não podem acessar o banco de dados devido às restrições da VPC.

As etapas a seguir explicam como criar e acessar um endpoint entre VPCs com o RDS Proxy:

1. Crie duas VPCs ou escolha duas VPCs para as quais você já usa para trabalhar com o Aurora e o RDS. Cada VPC deve ter os próprios recursos de rede associados, como um gateway da Internet, tabelas de roteamento, sub-redes e grupos de segurança. Se você tiver apenas uma VPC, consulte Conceitos básicos do Amazon RDS para ver as etapas de configuração de outra

VPC, a fim de usar o RDS com êxito. Também é possível examinar a VPC existente no console do Amazon EC2 para ver quais tipos de recursos podem ser conectados.

2. Crie um proxy de banco de dados associado ao cluster de bancos de dados Aurora ou instância do RDS aos quais você deseja se conectar. Siga o procedimento em Criar um proxy para o Amazon RDS.
3. Na página Details (Detalhes) do seu proxy no console do RDS, na seção Proxy endpoints (Endpoints de proxy), escolha Create endpoint (Criar endpoint). Siga o procedimento em Criação de um endpoint de proxy.
4. Escolha se deseja criar o endpoint entre VPCs como leitura/gravação ou somente leitura.
5. Em vez de aceitar o padrão da mesma VPC que o cluster de bancos de dados Aurora ou instância do RDS, escolha uma VPC diferente. Essa VPC deve estar na mesma região da AWS que a VPC, onde o proxy reside.
6. Agora, em vez de aceitar os padrões para sub-redes e grupos de segurança da mesma VPC que o cluster de bancos de dados Aurora ou instância do RDS, faça novas seleções. Faça-as com base nas sub-redes e grupos de segurança da VPC que você escolheu.
7. Você não precisa alterar nenhuma das configurações para os segredos do Secrets Manager. As mesmas credenciais funcionam para todos os endpoints de proxy, independentemente da VPC em que cada endpoint esteja.
8. Aguarde até que o novo endpoint alcance o estado Available (Disponível).
9. Anote o nome completo do endpoint. Esse é o valor que termina em *`Region_name`*`.rds.amazonaws.com`, que você fornece como parte da string de conexão para sua aplicação de banco de dados.
10. Acesse o novo endpoint por meio de um recurso na mesma VPC que o endpoint. Uma maneira simples de testar esse processo é criar uma nova instância do EC2 nessa VPC. Depois, faça login na instância do EC2 e execute os comandos `mysql` ou `psql` para se conectar usando o valor do endpoint na string de conexão.

Criação de um endpoint de proxy

Para criar um endpoint de proxy, siga estas instruções:

Console

Para criar um endpoint de proxy

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Proxies.
3. Clique no nome do proxy para o qual você deseja criar um novo endpoint.

   A página de detalhes desse proxy é exibida.
4. Na seção Proxy endpoints (Endpoints do proxy), escolha Create proxy endpoint (Criar endpoint proxy).

   A janela Create proxy endpoint (Criar endpoint proxy) aparece.
5. em Proxy endpoint name (Nome do endpoint de proxy), insira um nome descritivo de sua escolha.
6. Em Target role (Função de destino), escolha se deseja criar o endpoint como leitura/gravação ou somente leitura.

   As conexões que usam endpoints de leitura/gravação podem executar qualquer tipo de operação: declarações DDL (linguagem de definição de dados), declarações DML (linguagem de manipulação de dados) e consultas. Esses endpoints sempre se conectam à instância principal do cluster do Aurora. Você pode usar endpoints de leitura/gravação para operações gerais de banco de dados quando você usa apenas um único endpoint em sua aplicação. Você também pode usar endpoints de leitura/gravação para operações administrativas, aplicações de processamento de transações on-line (OLTP) e trabalhos de extração de transformação de carga (ETL).

   As conexões que usam um endpoint somente leitura só podem executar consultas. Quando há várias instâncias de leitor no cluster do Aurora, o proxy do RDS pode usar uma instância de leitor diferente para cada conexão com o endpoint. Dessa forma, uma aplicação com uso intensivo de consultas pode tirar proveito da capacidade de clustering do Aurora. Você pode acrescentar mais capacidade de consulta ao cluster adicionando mais instâncias de banco de dados de leitor. Essas conexões somente leitura não impõem nenhuma sobrecarga na instância primária do cluster. Dessa forma, suas consultas de relatórios e análises não retardam as operações de gravação de suas aplicações OLTP. O RDS Proxy pode usar uma das instâncias de leitor para cada conexão com o endpoint. Dessa forma, uma aplicação com uso intensivo de

consultas pode tirar proveito da capacidade de clustering do Aurora. Essas conexões somente leitura não impõem nenhuma sobrecarga na instância primária do cluster. Dessa forma, suas consultas de relatórios e análises não retardam as operações de gravação de suas aplicações OLTP.

7. Em Nuvem privada virtual (VPC), selecione o padrão para acessar o endpoint das mesmas instâncias do EC2 ou outros recursos que normalmente são usados para acessar o proxy ou o banco de dados associado. Para configurar o acesso entre VPC para esse proxy, selecione uma VPC diferente da padrão. Para obter mais informações sobre acesso entre VPCs, consulte Acesso aos bancos de dados do Aurora e do RDS entre VPCs.
8. Em Subnets (Sub-redes), o proxy do RDS preenche as mesmas sub-redes que o proxy associado por padrão. Para restringir o acesso ao endpoint para que apenas uma parte do intervalo de endereços da VPC possa se conectar a ele, remova uma ou mais sub-redes.
9. Em VPC grupo de segurança (Grupo de segurança da VPC), você pode selecionar um grupo de segurança existente ou criar outro. O proxy do RDS preenche os mesmos grupos de segurança que o proxy associado por padrão. Se as regras de entrada e saída para o proxy forem apropriadas para esse endpoint, deixe a escolha padrão.

   Se você optar por criar um novo grupo de segurança, especifique um nome para o grupo de segurança nessa página. Depois, edite as configurações do grupo de segurança no console do EC2.
10. Escolha Create proxy endpoint (Criar endpoint de proxy).

AWS CLI

Para criar um endpoint de proxy, use o comando da AWS CLI create-db-proxy-endpoint.

Inclua os seguintes parâmetros necessários:

- `--db-proxy-name` *`value`*
- `--db-proxy-endpoint-name` *`value`*
- `--vpc-subnet-ids` *`list_of_ids`*. Separe os IDs de sub-rede com espaços. Você não especifica o ID da própria VPC.

Você também pode incluir os seguintes parâmetros opcionais:

- `--target-role { READ_WRITE | READ_ONLY }`. Por padrão, esse parâmetro é `READ_WRITE`. O valor só tem um efeito sobre os clusters do Aurora provisionados que contêm uma ou mais instâncias de banco de dados do leitor. Quando o proxy está associado a um cluster de banco de dados multi-AZ que contém apenas uma instância de banco de dados de gravador, não é possível especificar `READ_ONLY`. Para ter mais informações sobre o uso pretendido de endpoints somente leitura com cluster de banco de dados multi-AZ, consulte Endpoints de leitor para clusters de banco de dados Multi-AZ.
- `--vpc-security-group-ids` *`value`*. Separe os IDs do grupo de segurança com espaços. Se você omitir esse parâmetro, o proxy do RDS usará o grupo de segurança padrão para a VPC. O proxy do RDS determina a VPC com base nos IDs de sub-rede que você especificar para o parâmetro `--vpc-subnet-ids`.

Example

O exemplo a seguir cria um endpoint de proxy chamado `my-endpoint`.

Para Linux, macOS ou Unix:

```
aws rds create-db-proxy-endpoint \
  --db-proxy-name my-proxy \
  --db-proxy-endpoint-name my-endpoint \
  --vpc-subnet-ids subnet_id subnet_id subnet_id ... \
  --target-role READ_ONLY \
  --vpc-security-group-ids security_group_id ]
```

Para Windows:

```
aws rds create-db-proxy-endpoint ^
  --db-proxy-name my-proxy ^
  --db-proxy-endpoint-name my-endpoint ^
  --vpc-subnet-ids subnet_id_1 subnet_id_2 subnet_id_3 ... ^
  --target-role READ_ONLY ^
  --vpc-security-group-ids security_group_id
```

API do RDS

Para criar um endpoint de proxy, use a ação da API do RDS CreateDBProxyEndpoint.

Visualização dos endpoints de proxy

Para visualizar os endpoints de proxy existentes, siga estas instruções:

Console

Para visualizar os detalhes de um endpoint de proxy

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Proxies.
3. Na lista, escolha o proxy cujo endpoint você deseja visualizar. Clique no nome do proxy para visualizar sua página de detalhes.
4. Na seção Proxy endpoints (Endpoints de proxy), escolha o endpoint que você deseja visualizar. Clique no nome dele para ver a página de detalhes.
5. Examine os parâmetros cujos valores lhe interessam. Você pode verificar propriedades como as seguintes:
   - Se o endpoint é leitura/gravação ou somente leitura.
   - O endereço do endpoint que você usa em uma string de conexão de banco de dados.
   - A VPC, as sub-redes e os grupos de segurança associados a um endpoint.

AWS CLI

Para visualizar um ou mais endpoints de proxy, use o comando da AWS CLI describe-db-proxy-endpoints.

Você pode incluir os seguintes parâmetros opcionais:

- `--db-proxy-endpoint-name`
- `--db-proxy-name`

O exemplo a seguir descreve o endpoint de proxy `my-endpoint`.

Example

Para Linux, macOS ou Unix:

```
aws rds describe-db-proxy-endpoints \
  --db-proxy-endpoint-name my-endpoint
```

Para Windows:

```
aws rds describe-db-proxy-endpoints ^
  --db-proxy-endpoint-name my-endpoint
```

API do RDS

Para descrever um ou mais endpoints de proxy, use a operação da API do RDS DescribeDBProxyEndpoints.

Modificação de um endpoint de proxy

Para modificar os endpoints de proxy, siga estas instruções:

Console

Para modificar um ou mais endpoints de proxy

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Proxies.
3. Na lista, escolha o proxy cujo endpoint você deseja modificar. Clique no nome do proxy para visualizar sua página de detalhes.
4. Em Proxy endpoints (Endpoints de proxy), escolha o endpoint que você deseja modificar. Você pode selecioná-lo na lista ou clicar no nome dele para visualizar a página de detalhes.
5. Na página de detalhes do proxy, na seção Proxy endpoints (Endpoints de proxy), escolha Edit (Editar). Ou na página de detalhes do endpoint de proxy, em Ações, selecione Editar.
6. Altere os valores dos parâmetros que você deseja modificar.
7. Escolha Save changes (Salvar alterações).

AWS CLI

Para modificar um endpoint de proxy, use o comando da AWS CLI modify-db-proxy-endpoint com os seguintes parâmetros obrigatórios:

- `--db-proxy-endpoint-name`

Especifique alterações nas propriedades do endpoint usando um ou mais dos seguintes parâmetros:

- `--new-db-proxy-endpoint-name`
- `--vpc-security-group-ids`. Separe os IDs do grupo de segurança com espaços.

O exemplo a seguir renomeia o endpoint de proxy `my-endpoint` para `new-endpoint-name`.

Example

Para Linux, macOS ou Unix:

```
aws rds modify-db-proxy-endpoint \
  --db-proxy-endpoint-name my-endpoint \
  --new-db-proxy-endpoint-name new-endpoint-name
```

Para Windows:

```
aws rds modify-db-proxy-endpoint ^
  --db-proxy-endpoint-name my-endpoint ^
  --new-db-proxy-endpoint-name new-endpoint-name
```

API do RDS

Para modificar um endpoint de proxy, use a operação da API do RDS ModifyDBProxyEndpoint.

Exclusão de um endpoint de proxy

Para excluir um endpoint para o proxy, siga estas instruções:

> **Note**
>
> Não é possível excluir o endpoint de proxy padrão que o RDS Proxy cria automaticamente para cada proxy.
> Quando você exclui um proxy, o proxy do RDS exclui automaticamente todos os endpoints associados.

Console

Para excluir um endpoint de proxy usando o AWS Management Console

1. No painel de navegação, escolha Proxies.
2. Na lista, escolha o proxy cujo endpoint você deseja endpoint. Clique no nome do proxy para visualizar sua página de detalhes.
3. Em Proxy endpoints (Endpoints de proxy), escolha o endpoint que você deseja excluir. Você pode selecionar um ou mais endpoints na lista ou clicar no nome de um único endpoint para visualizar a página de detalhes.
4. Na página de detalhes do proxy, na seção Proxy endpoints (Endpoints de proxy), escolha Delete (Excluir). Ou, na página de detalhes do endpoint de proxy, em Ações, selecione Excluir.

AWS CLI

Para excluir um endpoint de proxy, execute o comando delete-db-proxy-endpoint com os seguintes parâmetros obrigatórios:

- `--db-proxy-endpoint-name`

O comando a seguir exclui o endpoint de proxy chamado `my-endpoint`.

Para Linux, macOS ou Unix:

```
aws rds delete-db-proxy-endpoint \
  --db-proxy-endpoint-name my-endpoint
```

Para Windows:

```
aws rds delete-db-proxy-endpoint ^
  --db-proxy-endpoint-name my-endpoint
```

API do RDS

Para excluir um endpoint deproxy com a API do RDS, execute a operação DeleteDBProxyEndpoint. Especifique o nome do endpoint de proxy para o parâmetro `DBProxyEndpointName`.

Monitorar métricas do proxy do RDS com o Amazon CloudWatch

É possível monitorar o proxy do RDS usando o Amazon CloudWatch O CloudWatch coleta e processa dados brutos dos proxies e os transforma em métricas legíveis quase em tempo real. Para localizar essas métricas no console do CloudWatch, escolha Metrics (Métricas) e escolha RDS e Per Proxy Metrics (Métricas por Proxy). Para obter mais informações, consulte Usando métricas do Amazon CloudWatch no Guia do usuário do Amazon CloudWatch.

> **Note**
>
> O RDS publica essas métricas para cada instância subjacente do Amazon EC2 associada a um proxy. Um único proxy pode ser atendido por mais de uma instância do EC2. Use as estatísticas do CloudWatch para agregar os valores de um proxy em todas as instâncias associadas.
>
> Algumas dessas métricas podem não ser visíveis até depois da primeira conexão bem-sucedida por um proxy.

Nos logs do RDS Proxy, cada entrada é prefixada com o nome do endpoint de proxy associado. Esse nome pode ser o nome especificado para um endpoint definido pelo usuário ou o nome especial `default` para o endpoint padrão de um proxy que realiza solicitações de leitura/gravação.

Todas as métricas do proxy do RDS estão no grupo `proxy`.

Cada endpoint de proxy tem suas próprias métricas do CloudWatch. Você pode monitorar o uso de cada endpoint de proxy de forma independente. Para obter mais informações sobre os endpoints do proxy, consulte Como trabalhar com endpoints do proxy do Amazon RDS.

Você pode agregar os valores de cada métrica usando um dos seguintes conjuntos de dimensões. Por exemplo, usando o conjunto de dimensões `ProxyName` , você pode analisar todo o tráfego para um determinado proxy. Ao usar os outros conjuntos de dimensões, você pode dividir as métricas de maneiras diferentes. Você pode dividir as métricas com base nos diferentes endpoints ou bancos de dados de destino de cada proxy ou no tráfego de leitura/gravação e somente leitura para cada banco de dados.

- Conjunto de dimensões 1 : `ProxyName`
- Conjunto de dimensões 2 : `ProxyName`, `EndpointName`
- Conjunto de dimensões 3 :`ProxyName`,`TargetGroup`,`Target`

- Conjunto de dimensões 4 :`ProxyName`,`TargetGroup`,`TargetRole`

Métrica	Descrição	Período válido	Conjunto de dimensões do CloudWatch
`AvailabilityPercentage`	O percentual de tempo para o qual o grupo de destino estava disponível na função indicada pela dimensão. Essa métrica é relatada a cada minuto. A estatística mais útil para essa métrica é `Sum`.	1 minuto	Dimension set 4
`ClientConnections`	O número atual de conexões de cliente. Essa métrica é relatada a cada minuto. A estatística mais útil para essa métrica é `Sum`.	1 minuto	Dimension set 1, Dimension set 2
`ClientConnectionsClosed`	O número de conexões de cliente fechadas. A estatística mais útil para essa métrica é `Sum`.	1 minuto e acima	Dimension set 1, Dimension set 2
`ClientConnectionsInSetup`	O número atual de conexões de cliente abertas, mas que não concluíram a configuração. Essa	1 minuto	Dimension set 1, Dimension set 2

Métrica	Descrição	Período válido	Conjunto de dimensões do CloudWatch
	métrica é relatada a cada minuto. A estatística mais útil para esta métrica é Soma.		
`ClientConnectionsNoTLS`	O número atual de conexões de cliente sem Transport Layer Security (TLS). Essa métrica é relatada a cada minuto. A estatística mais útil para essa métrica é `Sum`.	1 minuto	Dimension set 1, Dimension set 2
`ClientConnectionsReceived`	O número de solicitações de conexão de cliente recebidas. A estatística mais útil para essa métrica é `Sum`.	1 minuto e acima	Dimension set 1, Dimension set 2
`ClientConnectionsSetupFailedAuth`	O número de tentativas de conexão do cliente que falharam devido à configuração incorreta da autenticação ou do TLS. A estatística mais útil para essa métrica é `Sum`.	1 minuto e acima	Dimension set 1, Dimension set 2

Métrica	Descrição	Período válido	Conjunto de dimensões do CloudWatch
`ClientConnectionsSetupSucceeded`	O número de conexões de cliente estabelecido com êxito com qualquer mecanismo de autenticação com ou sem TLS. A estatística mais útil para essa métrica é `Sum`.	1 minuto e acima	Dimension set 1, Dimension set 2
`ClientConnectionsTLS`	O número atual de conexões de cliente com TLS. Essa métrica é relatada a cada minuto. A estatística mais útil para essa métrica é `Sum`.	1 minuto	Dimension set 1, Dimension set 2
`DatabaseConnectionRequests`	O número de solicitações para criar uma conexão de banco de dados. A estatística mais útil para essa métrica é `Sum`.	1 minuto e acima	Dimension set 1, Dimension set 3, Dimension set 4
`DatabaseConnectionRequestsWithTLS`	O número de solicitações para criar uma conexão de banco de dados com TLS. A estatística mais útil para essa métrica é `Sum`.	1 minuto e acima	Dimension set 1, Dimension set 3, Dimension set 4

Métrica	Descrição	Período válido	Conjunto de dimensões do CloudWatch
`DatabaseConnections`	O número atual de conexões de banco de dados. Essa métrica é relatada a cada minuto. A estatística mais útil para essa métrica é `Sum`.	1 minuto	Dimension set 1, Dimension set 3, Dimension set 4
`DatabaseConnectionsBorrowLatency`	O tempo, em microssegundos, que leva para o proxy que está sendo monitorado obter uma conexão de banco de dados. A estatística mais útil para essa métrica é `Sum`.	1 minuto e acima	Dimension set 1, Dimension set 2
`DatabaseConnectionsCurrentlyBorrowed`	O número atual de conexões de banco de dados no estado de empréstimo. Essa métrica é relatada a cada minuto. A estatística mais útil para essa métrica é `Sum`.	1 minuto	Dimension set 1, Dimension set 3, Dimension set 4

Métrica	Descrição	Período válido	Conjunto de dimensões do CloudWatch
`DatabaseConnectionsCurrentlyInTransaction`	O número atual de conexões de banco de dados em uma transação. Essa métrica é relatada a cada minuto. A estatística mais útil para essa métrica é `Sum`.	1 minuto	Dimension set 1, Dimension set 3, Dimension set 4
`DatabaseConnectionsCurrentlySessionPinned`	O número atual de conexões de banco de dados atualmente fixadas devido a operações em solicitações de cliente que alteram o estado da sessão. Essa métrica é relatada a cada minuto. A estatística mais útil para essa métrica é `Sum`.	1 minuto	Dimension set 1, Dimension set 3, Dimension set 4
`DatabaseConnectionsSetupFailed`	O número de solicitações de conexão do banco de dados que falharam. A estatística mais útil para essa métrica é `Sum`.	1 minuto e acima	Dimension set 1, Dimension set 3, Dimension set 4

Métrica	Descrição	Período válido	Conjunto de dimensões do CloudWatch
`DatabaseConnectionsSetupSucceeded`	O número de conexões de banco de dados estabelecidas com êxito com ou sem TLS. A estatística mais útil para essa métrica é `Sum`.	1 minuto e acima	Dimension set 1, Dimension set 3, Dimension set 4
`DatabaseConnectionsWithTLS`	O número atual de conexões de banco de dados com TLS. Essa métrica é relatada a cada minuto. A estatística mais útil para essa métrica é `Sum`.	1 minuto	Dimension set 1, Dimension set 3, Dimension set 4
`MaxDatabaseConnectionsAllowed`	O número máximo de conexões de banco de dados permitidas. Essa métrica é relatada a cada minuto. A estatística mais útil para essa métrica é `Sum`.	1 minuto	Dimension set 1, Dimension set 3, Dimension set 4
`QueryDatabaseResponseLatency`	O tempo, em microssegundos, que o banco de dados levou para responder à consulta. A estatística mais útil para essa métrica é `Average`.	1 minuto e acima	Dimension set 1, Dimension set 2, Dimension set 3, Dimension set 4

Métrica	Descrição	Período válido	Conjunto de dimensões do CloudWatch
`QueryRequests`	O número de consultas recebidas . Uma consulta incluindo várias instruções é contada como uma consulta. A estatística mais útil para essa métrica é `Sum`.	1 minuto e acima	Dimension set 1, Dimension set 2
`QueryRequestsNoTLS`	O número de consultas recebidas de conexões não TLS. Uma consulta incluindo várias instruções é contada como uma consulta. A estatística mais útil para essa métrica é `Sum`.	1 minuto e acima	Dimension set 1, Dimension set 2
`QueryRequestsTLS`	O número de consultas recebidas de conexões TLS. Uma consulta incluindo várias instruções é contada como uma consulta. A estatística mais útil para essa métrica é `Sum`.	1 minuto e acima	Dimension set 1, Dimension set 2

Métrica	Descrição	Período válido	Conjunto de dimensões do CloudWatch
`QueryResponseLatency`	O tempo, em microssegundos, entre a obtenção de uma solicitação de consulta e a resposta do proxy a ela. A estatística mais útil para essa métrica é `Average`.	1 minuto e acima	Dimension set 1, Dimension set 2

Você pode encontrar os logs de atividades do proxy do RDS sob CloudWatch no AWS Management Console. Cada proxy tem uma entrada na página Log groups (Grupos de logs).

> **Important**
>
> Esses logs são destinados ao consumo humano para fins de solução de problemas e não para acesso programático. O formato e o conteúdo dos logs estão sujeitos a alterações. Em particular, os logs mais antigos não contêm nenhum prefixo indicando o endpoint para cada solicitação. Nos logs mais recentes, cada entrada é prefixada com o nome do endpoint do proxy associado. Esse nome pode ser o nome que você especificou para um endpoint definido pelo usuário ou o nome especial `default` para solicitações que usam o endpoint padrão de um proxy.

Trabalhar com eventos do RDS Proxy

Um evento indica uma alteração em um ambiente, como um ambiente da AWS, um serviço ou uma aplicação de um parceiro de software como serviço (SaaS). Ou pode ser uma de suas próprias aplicações ou serviços personalizados. Por exemplo, o Amazon RDS gera um evento quando você cria ou modifica um RDS Proxy. O Amazon RDS entrega eventos ao Amazon EventBridge quase em tempo real. A seguir, você pode encontrar uma lista de eventos do proxy do RDS que você pode assinar e um exemplo de um evento do RDS Proxy.

Para obter mais informações sobre como trabalhar com eventos, consulte o seguinte no:

- Para obter instruções sobre como visualizar eventos usando o AWS Management Console, a AWS CLI ou a API do RDS, consulte Visualizar eventos do Amazon RDS.
- Para saber como configurar o Amazon RDS para enviar eventos ao EventBridge, consulte Criar uma regra que é acionada em um evento do Amazon RDS.

Eventos do RDS Proxy

As tabelas a seguir mostram a categoria de evento e uma lista de eventos quando um proxy do RDS é o tipo de fonte.

Categoria	ID do evento do RDS	Message	Observações
alteração de configuração	RDS-EVENT-0204	Proxy de banco de dados *nome* modificado pelo RDS.	Nenhum
alteração de configuração	RDS-EVENT-0207	O RDS modificou o endpoint do proxy de banco de dados *nome*.	Nenhum
alteração de configuração	RDS-EVENT-0213	O RDS detectou a adição da instância de banco de dados e a adicionou automaticamente ao grupo de destino do *nome* do proxy de banco de dados.	Nenhum
alteração de configuração	RDS-EVENT-0214	O RDS detectou a exclusão da instância de banco de dados *nome* e a removeu automaticamente do grupo de destino *nome* do proxy de banco de dados *nome*.	Nenhum

Categoria	ID do evento do RDS	Message	Observações
alteração de configuração	RDS-EVENT-0215	O RDS detectou a exclusão do cluster de banco de dados *nome* e o removeu automaticamente do grupo de destino *nome* do proxy de banco de dados *nome*.	Nenhum
criação	RDS-EVENT-0203	O RDS criou o proxy de banco de dados *nome*.	Nenhum
criação	RDS-EVENT-0206	O RDS criou o endpoint para *nome* para o proxy do banco de dados *nome*.	Nenhum
exclusão	RDS-EVENT-0205	O RDS excluiu o proxy do banco de dados *nome*.	Nenhum
exclusão	RDS-EVENT-0208	O RDS excluiu o endpoint *nome* para o proxy do banco de dados *nome*.	Nenhum
falha	RDS-EVENT-0243	O RDS não conseguiu provisionar capacidad e para o *nome* do proxy porque não há endereços IP suficientes disponíveis em suas sub-redes: *nome*. Para resolver o problema, as sub-redes devem ter o número mínimo de endereços IP não usados, conforme recomendado na documentação do proxy do RDS.	Para determinar o número recomendado para sua classe de instância , consulte Planejar a capacidade de endereços IP.

Categoria	ID do evento do RDS	Message	Observações
falha	RDS-EVENT-0275	O RDS limitou algumas conexões com o proxy de banco de dados *nome*. O número de solicitações de conexão simultâneas do cliente para o proxy excedeu o limite.	Nenhum

Veja a seguir um exemplo de um evento proxy do RDS no formato JSON. O evento mostra que o RDS modificou o endpoint chamado `my-endpoint` do proxy do RDS chamado `my-rds-proxy`. O ID do evento é RDS-EVENT-0207.

```
{
  "version": "0",
  "id": "68f6e973-1a0c-d37b-f2f2-94a7f62ffd4e",
  "detail-type": "RDS DB Proxy Event",
  "source": "aws.rds",
  "account": "123456789012",
  "time": "2018-09-27T22:36:43Z",
  "region": "us-east-1",
  "resources": [
     "arn:aws:rds:us-east-1:123456789012:db-proxy:my-rds-proxy"
  ],
  "detail": {
    "EventCategories": [
      "configuration change"
    ],
    "SourceType": "DB_PROXY",
    "SourceArn": "arn:aws:rds:us-east-1:123456789012:db-proxy:my-rds-proxy",
    "Date": "2018-09-27T22:36:43.292Z",
    "Message": "RDS modified endpoint my-endpoint of DB Proxy my-rds-proxy.",
    "SourceIdentifier": "my-endpoint",
    "EventID": "RDS-EVENT-0207"
  }
}
```

Solução de problemas do RDS Proxy

A seguir, é possível encontrar ideias de solução de problemas para alguns problemas comuns do proxy do RDS e informações sobre logs do CloudWatch para o RDS Proxy.

Nos logs do RDS Proxy, cada entrada é prefixada com o nome do endpoint de proxy associado. Esse nome pode ser o nome que você especificou para um endpoint definido pelo usuário. Ou pode ser o nome especial `default` para o endpoint padrão de um proxy que realiza solicitações de leitura/gravação. Para obter mais informações sobre os endpoints do proxy, consulte Como trabalhar com endpoints do proxy do Amazon RDS.

Tópicos

- Verificar a conectividade para um proxy
- Problemas e soluções comuns de
- Solução de problemas do RDS Proxy com o RDS para MySQL
- Solução de problemas do RDS Proxy com o RDS para PostgreSQL

Verificar a conectividade para um proxy

É possível usar os comandos a seguir para verificar se todos os componentes, como proxy, banco de dados e instâncias de computação na conexão, podem se comunicar entre si.

Examine o próprio proxy usando o comando describe-db-proxies. Examine também o grupo de destino associado usando o comando describe-db-proxy-target-groups. Verifique se os detalhes dos destinos correspondem à instância de banco de dados do RDS ou ao cluster de banco de dados do que você pretende associar ao proxy. Use comandos como os seguintes.

```
aws rds describe-db-proxies --db-proxy-name $DB_PROXY_NAME
aws rds describe-db-proxy-target-groups --db-proxy-name $DB_PROXY_NAME
```

Para confirmar se o proxy pode se conectar ao banco de dados subjacente, examine os destinos especificados nos grupos de destino usando o comando describe-db-proxy-targets. Use um comando como o seguinte.

```
aws rds describe-db-proxy-targets --db-proxy-name $DB_PROXY_NAME
```

A saída do comando describe-db-proxy-targets inclui um campo `TargetHealth`. É possível examinar os campos `State`, `Reason` e `Description` dentro de `TargetHealth` para verificar se o proxy pode se comunicar com a instância de banco de dados subjacente.

- Um valor `State` de `AVAILABLE` indica que o proxy pode se conectar à instância de banco de dados.
- Um valor `State` de `UNAVAILABLE` indica um problema de conexão temporário ou permanente. Nesse caso, examine os campos `Reason` e `Description`. Por exemplo, se `Reason` tiver um valor de `PENDING_PROXY_CAPACITY`, tente se conectar novamente depois que o proxy terminar sua operação de escalabilidade. Se `Reason` tiver um valor de `UNREACHABLE`, `CONNECTION_FAILED` ou `AUTH_FAILURE`, use a explicação do campo `Description` para ajudá-lo a diagnosticar o problema.
- O campo `State` pode ter um valor de `REGISTERING` por um breve período antes de alterar para `AVAILABLE` ou `UNAVAILABLE`.

Se o comando do Netcat a seguir (`nc`) for bem-sucedido, você poderá acessar o endpoint do proxy na instância do EC2 ou em outro sistema em que esteja conectado. Esse comando relata uma falha se você não estiver na mesma VPC que o proxy e o banco de dados associado. Você pode fazer login diretamente no banco de dados sem estar na mesma VPC. No entanto, não é possível fazer login no proxy a menos que você esteja na mesma VPC.

```
nc -zx MySQL_proxy_endpoint 3306

nc -zx PostgreSQL_proxy_endpoint 5432
```

Você pode usar os comandos a seguir para garantir que sua instância do EC2 tenha as propriedades necessárias. Em particular, a VPC da instância do EC2 deve ser a mesma que a VPC da instância de banco de dados do RDS ou do cluster de bancos de dados Aurora ao qual o proxy se conecta.

```
aws ec2 describe-instances --instance-ids your_ec2_instance_id
```

Examine os segredos do Secrets Manager usados para o proxy.

```
aws secretsmanager list-secrets
aws secretsmanager get-secret-value --secret-id your_secret_id
```

O campo `SecretString` exibido por `get-secret-value` deve estar codificado como uma string JSON que inclua os campos `username` e `password`. O exemplo a seguir mostra o formato do campo `SecretString`.

```
{
  "ARN": "some_arn",
  "Name": "some_name",
  "VersionId": "some_version_id",
  "SecretString": '{"username":"some_username","password":"some_password"}',
  "VersionStages": [ "some_stage" ],
  "CreatedDate": some_timestamp
}
```

Problemas e soluções comuns de

Esta seção descreve alguns problemas comuns e possíveis soluções ao usar o Proxy RDS.

Depois de executar o comando `aws rds describe-db-proxy-targets` da CLI, se a descrição de `TargetHealth` indicar `Proxy does not have any registered credentials`, verifique o seguinte:

- Há credenciais registradas para que o usuário acesse o proxy.
- O perfil do IAM para acessar o segredo do Secrets Manager usado pelo proxy é válido.

Você pode encontrar os eventos de RDS a seguir ao criar ou se conectar a um proxy de banco de dados.

Categoria	ID do evento do RDS	Descrição
falha	RDS-EVENT-0243	O RDS não conseguiu provisionar capacidade para o proxy porque não há endereços IP suficient es disponíveis em suas sub-redes. Para resolver o problema, suas sub-redes devem ter o número mínimo de endereços IP não usados.

Categoria	ID do evento do RDS	Descrição
		Para determinar o número recomendado para sua classe de instância, consulte Planejar a capacidade de endereços IP.
falha	RDS-EVENT-0275	O RDS limitou algumas conexões com o proxy de banco de dados *nome*. O número de solicitações de conexão simultâneas do cliente para o proxy excedeu o limite.

Você pode encontrar os seguintes problemas ao criar um proxy ou ao se conectar a um proxy.

Erro	Causas ou soluções alternativas
`403: The security token included in the request is invalid`	Selecione uma função existente do IAM em vez de optar por criar uma.

Solução de problemas do RDS Proxy com o RDS para MySQL

Você pode encontrar os problemas a seguir ao se conectar a um proxy do MySQL.

Erro	Causas ou soluções alternativas
`ERROR 1040 (HY000): Connections rate limit`	A taxa de solicitações de conexão do cliente para o proxy excedeu o limite.

Erro	Causas ou soluções alternativas
`exceeded (limit_value )`	
`ERROR 1040 (HY000): IAM authentication rate limit exceeded`	O número de solicitações simultâneas com autenticação do IAM do cliente para o proxy excedeu o limite.
`ERROR 1040 (HY000): Number simultaneous connections exceeded (limit_value )`	O número de solicitações de conexão simultâneas do cliente para o proxy excedeu o limite.
`ERROR 1045 (28000): Access denied for user 'DB_USER'@'%' (using password: YES)`	O segredo do Secrets Manager usado pelo proxy não corresponde ao nome de usuário e à senha de um usuário de banco de dados existente. Atualize as credenciais no segredo do Secrets Manager ou verifique se o usuário do banco de dados existe e tem a mesma senha do segredo.
`ERROR 1105 (HY000): Unknown error`	Ocorreu um erro desconhecido.
`ERROR 1231 (42000): Variable ''character_set_client'' can't be set to the value of value`	O valor definido para o parâmetro `character_set_client` não é válido. Por exemplo, o valor `ucs2` não é válido porque ele pode travar o servidor do MySQL.

Erro	Causas ou soluções alternativas
`ERROR 3159 (HY000): This RDS Proxy requires TLS connections.`	Você habilitou a configuração Exigir Transport Layer Security no proxy, mas a conexão incluiu o parâmetro `ssl-mode=DISABLED` no cliente MySQL. Realize um dos procedimentos a seguir: • Desabilite a configuração Exigir Transport Layer Security para o proxy. • Conectar-se ao banco de dados usando a configuração mínima de `ssl-mode=REQUIRED` no cliente do MySQL.
`ERROR 2026 (HY000): SSL connection error: Internal Server` *`Error`*	Falha no handshake TLS para o proxy. Algumas razões possíveis incluem: • O SSL é necessário, mas o servidor não oferece suporte a ele. • Ocorreu um erro interno do servidor. • Ocorreu um handshake ruim.
`ERROR 9501 (HY000): Timed-out waiting to acquire database connection`	O tempo limite do proxy foi atingido enquanto aguardava para adquirir uma conexão de banco de dados. Algumas razões possíveis incluem: • O proxy não consegue estabelecer uma conexão de banco de dados porque o máximo de conexões máximas foi atingido • O proxy não consegue estabelecer uma conexão de banco de dados porque o banco de dados não está disponível.

Solução de problemas do RDS Proxy com o RDS para PostgreSQL

Você pode encontrar os problemas a seguir ao se conectar a um proxy do PostgreSQL.

Erro	Causa	Solução
`ERROR 28000: IAM authentication is allowed only with SSL connections.`	O usuário tentou se conectar ao banco de dados usando a autenticação do IAM com a configuração `sslmode=disable` no cliente do PostgreSQL.	O usuário precisa se conectar ao banco de dados usando a configuração mínima de `sslmode=require` no cliente do PostgreSQL. Para obter mais informações,

Erro	Causa	Solução
		consulte a documentação PostgreSQL SSL Support.
`ERROR 28000: This RDS proxy has no credentials for the role` *`role_name`* `. Check the credentials for this role and try again.`	Não há segredo do Secrets Manager para essa função.	Adicione um segredo do Secrets Manager para essa função. Para obter mais informações, consulte Configurar políticas do AWS Identity and Access Management (IAM) para o RDS Proxy.
`ERROR 28000: RDS supports only IAM, MD5, or SCRAM authentication.`	O cliente de banco de dados que está sendo usado para se conectar ao proxy está usando um mecanismo de autenticação que não é compatível proxy no momento.	Se você não estiver usando a autenticação do IAM, use a autenticação de senha MD5 ou SCRAM.
`ERROR 28000: A user name is missing from the connection startup packet. Provide a user name for this connection.`	O cliente de banco de dados que está sendo usado para se conectar ao proxy não está enviando um nome de usuário ao tentar estabelecer uma conexão.	Defina um nome de usuário ao configurar uma conexão com o proxy usando o cliente PostgreSQL de sua escolha.
`ERROR 28000: IAM is allowed only with SSL connections.`	Um cliente tentou se conectar usando a autenticação do IAM, mas o SSL não estava habilitado.	Habilite o SSL no cliente PostgreSQL.

Erro	Causa	Solução
`ERROR 28000: This RDS Proxy requires TLS connections.`	O usuário habilitou a opção Exigir Transport Layer Security, mas tentou se conectar com `sslmode=disable` no cliente do PostgreSQL.	Para corrigir esse erro, execute um dos seguintes procedimentos: • Desabilite a opção Exigir Transport Layer Security do proxy. • Conectar-se ao banco de dados usando a configuração mínima de `sslmode=allow` no cliente do PostgreSQL.
`ERROR 28P01: IAM authentication failed for user` *`user_name`* `. Check the IAM token for this user and try again.`	Esse erro pode ocorrer devido aos seguintes motivos: • O cliente forneceu o nome de usuário do IAM incorreto. • O cliente forneceu um token de autorização do IAM incorreto para o usuário. • O cliente está usando uma política do IAM que não tem as permissões necessárias. • O cliente forneceu um token de autorização do IAM expirado para o usuário.	Para corrigir esse erro, faça o seguinte: 1. Confirme se o usuário do IAM fornecido existe. 2. Confirme se o token de autorização do IAM pertence ao usuário do IAM fornecido. 3. Confirme se a política do IAM tem as permissões adequadas para o RDS. 4. Verifique a validade do token de autorização do IAM usado.
`ERROR 28P01: The password that was provided for the role` *`role_name`* `is wrong.`	A senha para essa função não corresponde ao segredo do Secrets Manager.	Verifique o segredo dessa função no Secrets Manager para ver se a senha é a mesma que está sendo usada no cliente PostgreSQL.

Erro	Causa	Solução
`ERROR 28P01: The IAM authentication failed for the role` *`role_name`* `. Check the IAM token for this role and try again.`	Há um problema com o token do IAM usado para a autenticação do IAM.	Gere outro token de autenticação e use-o em uma nova conexão.
`ERROR 0A000: Feature not supported: RDS Proxy supports only version 3.0 of the PostgreSQL messaging protocol.`	O cliente PostgreSQL usado para se conectar ao proxy usa um protocolo mais antigo que 3.0.	Use um cliente PostgreSQL mais recente que ofereça suporte ao protocolo de sistema de mensagens 3.0. Se você estiver usando a CLI `psql` do PostgreSQL, use uma versão posterior ou igual a 7.4.
`ERROR 0A000: Feature not supported: RDS Proxy currently doesn't support streaming replication mode.`	O cliente PostgreSQL usado para se conectar ao proxy está tentando usar o modo de replicação de streaming, que não é compatível com o proxy do RDS no momento.	Desative o modo de replicação de streaming no cliente PostgreSQL que está sendo usado para estabelecer a conexão.
`ERROR 0A000: Feature not supported: RDS Proxy currently doesn't support the option` *`option_name`* `.`	Pela mensagem de inicialização, o cliente PostgreSQL usado para se conectar ao proxy está solicitando uma opção que não é compatível com o proxy do RDS no momento.	Desative a opção que está sendo mostrada como não compatível na mensagem acima no cliente PostgreSQL que está sendo usado para se conectar.

Erro	Causa	Solução
`ERROR 53300: The IAM authentication failed because of too many competing requests.`	O número de solicitações simultâneas com autenticação do IAM do cliente para o proxy excedeu o limite.	Reduza a taxa na qual as conexões que usam a autenticação do IAM de um cliente PostgreSQL são estabelecidas.
`ERROR 53300: The maximum number of client connections to the proxy exceeded` *`number_value`* `.`	O número de solicitações de conexão simultâneas do cliente para o proxy excedeu o limite.	Reduza o número de conexões ativas de clientes PostgreSQL para esse proxy do RDS .
`ERROR 53300: Rate of connection to proxy exceeded` *`number_value`* `.`	A taxa de solicitações de conexão do cliente para o proxy excedeu o limite.	Reduza a taxa na qual as conexões de um cliente PostgreSQL são estabelecidas.
`ERROR XX000: Unknown error.`	Ocorreu um erro desconhecido.	Entre em contato com o AWS Support para investigar o problema.

Erro	Causa	Solução
`ERROR 08000: Timed-out waiting to acquire database connection.`	O proxy atingiu o tempo limite enquanto aguardava para obter uma conexão de banco de dados dentro do período especificado pela configuração `ConnectionBorrowTimeout`. Algumas razões possíveis incluem: • O proxy não consegue estabelecer uma conexão de banco de dados porque o máximo de conexões foi atingido. • O proxy não consegue estabelecer uma conexão de banco de dados porque o banco de dados não está disponível ou quando o estabelecimento da conexão com o banco de dados demora mais do que o `ConnectionBorrowTimeout` configurado.	As possíveis soluções são as seguintes: • Evite fixar conexões de proxy. Consulte Evitar a fixação de um RDS Proxy. • Examine as configurações `ConnectionBorrowTimeout` e `MaxConnectionsPercent`. Consulte Considerações sobre a conexão do RDS Proxy. • Examine a disponibilidade do destino. Consulte `AvailabilityPercentage` em Monitorar métricas do proxy do RDS com o Amazon CloudWatch.

Erro	Causa	Solução
`ERROR XX000: Request returned an error: database_error .`	A conexão de banco de dados estabelecida pelo proxy retornou um erro.	A solução depende do erro específico do banco de dados. Um exemplo é: `Request returned an error: database "your-database-name" does not exist`. Isso significa que o nome do banco de dados especificado não existe no servidor do banco de dados. Ou isso significa que o nome de usuário usado como um nome de banco de dados (se não for especificado um nome de banco de dados) não existe no servidor.

Solução de problemas de exclusão do banco de dados **`postgres`**

Se você excluir acidentalmente o banco de dados `postgres` da sua instância, precisará restaurá-lo para restabelecer a conectividade com sua instância. Execute os seguintes comandos em sua instância de banco de dados:

```
CREATE DATABASE postgres;
GRANT CONNECT ON DATABASE postgres TO rdsproxyadmin;
```

Usar o proxy do RDS com o AWS CloudFormation

É possível usar o proxy do RDS com o AWS CloudFormation. Isso ajuda a criar grupos de recursos relacionados. Esse grupo pode incluir um proxy que pode se conectar a uma instância de banco de dados do Amazon RDS ou a um cluster de bancos de dados do Aurora recém-criado. A compatibilidade do proxy do RDS no AWS CloudFormation envolve dois novos tipos de registro: `DBProxy` e `DBProxyTargetGroup`.

A listagem a seguir mostra um modelo de exemplo do AWS CloudFormation para o RDS Proxy.

```
Resources:
 DBProxy:
   Type: AWS::RDS::DBProxy
   Properties:
     DBProxyName: CanaryProxy
     EngineFamily: MYSQL
     RoleArn:
      Fn::ImportValue: SecretReaderRoleArn
     Auth:
       - {AuthScheme: SECRETS, SecretArn: !ImportValue ProxySecret, IAMAuth: DISABLED}
     VpcSubnetIds:
       Fn::Split: [",", "Fn::ImportValue": SubnetIds]

 ProxyTargetGroup:
   Type: AWS::RDS::DBProxyTargetGroup
   Properties:
     DBProxyName: CanaryProxy
     TargetGroupName: default
     DBInstanceIdentifiers:
       - Fn::ImportValue: DBInstanceName
   DependsOn: DBProxy
```

Para obter mais informações sobre os recursos neste exemplo, consulte DBProxy e DBProxyTargetGroup.

Para obter mais informações sobre os recursos do e do que você pode criar usando o , consulte Referência de tipo de recurso do RDS.

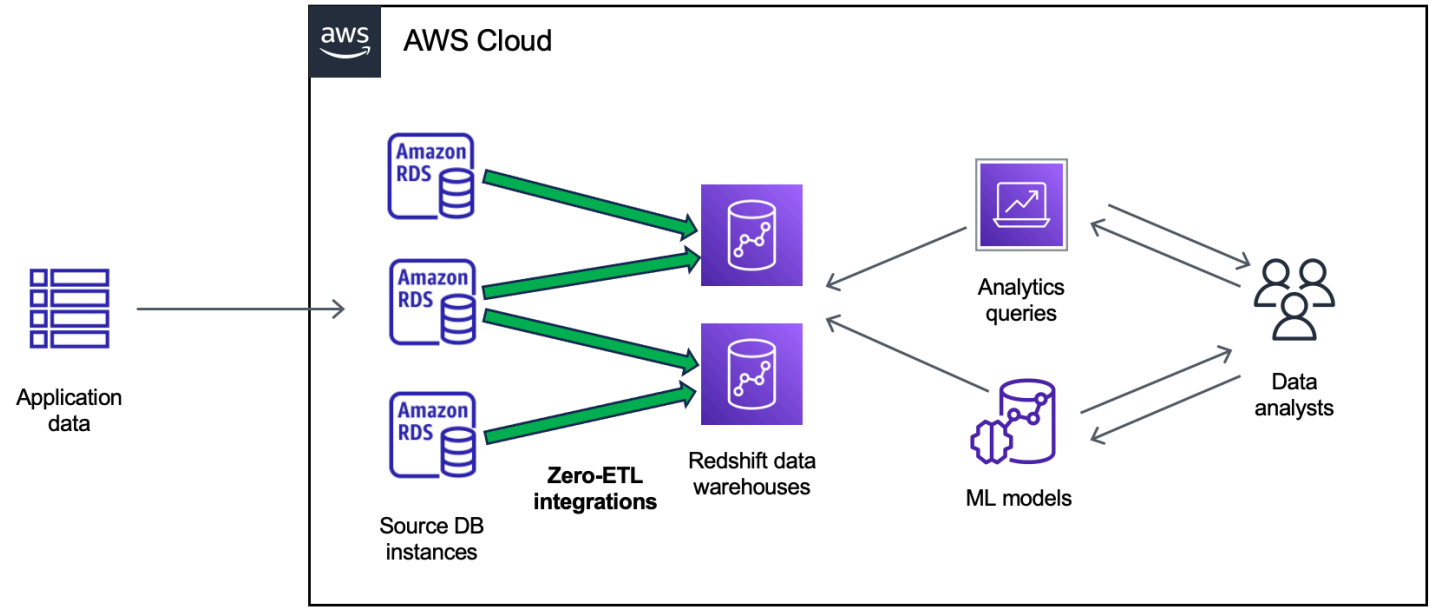
Integrações ETL zero do Amazon RDS

Uma Integração ETL zero do Amazon RDS com o Amazon Redshift e o Amazon SageMaker permite analytics e machine learning (ML) quase em tempo real usando dados do RDS. Essa é uma solução totalmente gerenciada para disponibilizar dados transacionais no destino de analytics depois que eles são gravados em um banco dados do RDS. Extração, transformação e carregamento (ETL) é o processo de combinar dados de várias fontes em um grande data warehouse central.

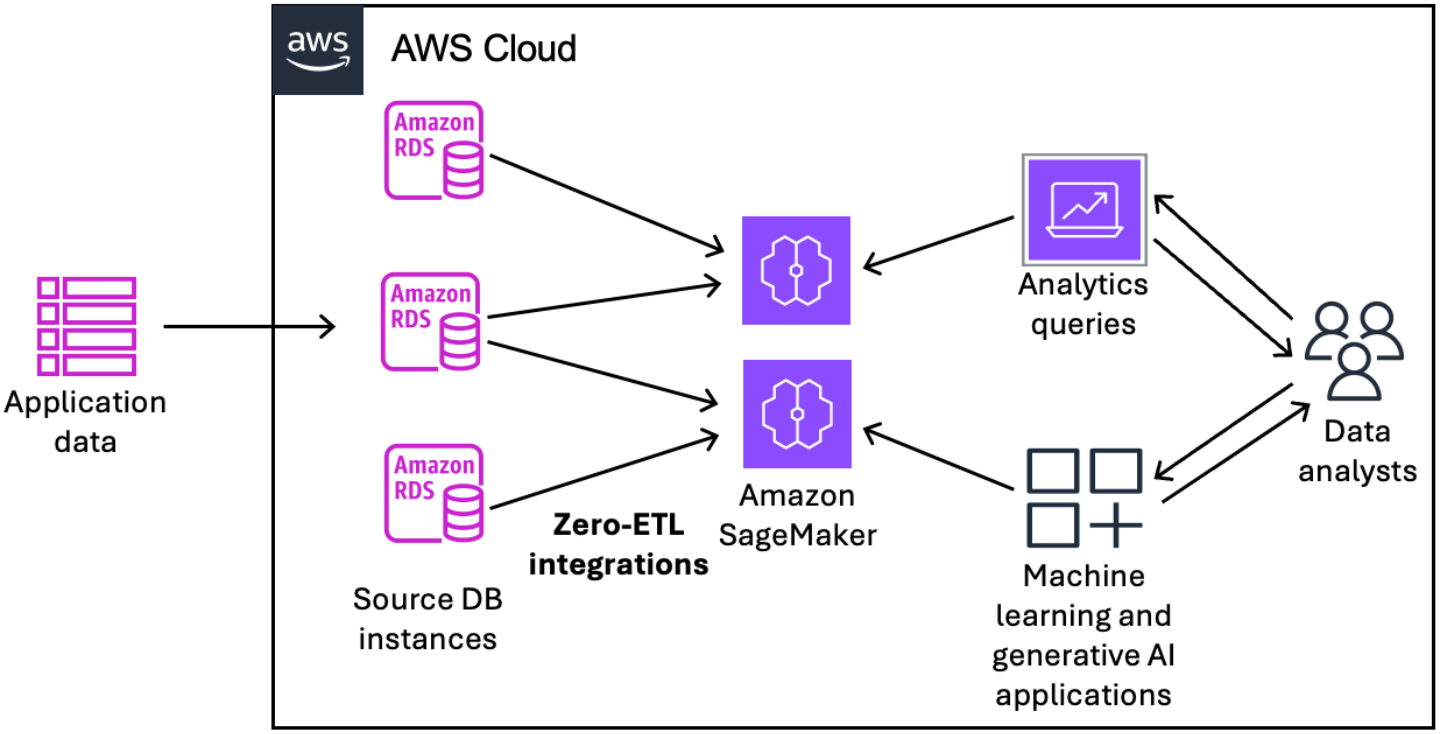
Uma Integração ETL zero torna os dados no banco de dados do RDS disponíveis no Amazon Redshift ou em um lakehouse do Amazon SageMaker quase em tempo real. Quando esses dados estiverem no data warehouse ou data lake de destino, você poderá potencializar suas workloads de analytics, ML e IA usando os recursos integrados, como machine learning, visões materializadas, compartilhamento de dados, acesso federado a vários datastores e data lakes e integrações com o Amazon SageMaker AI, o QuickSight e outros Serviços da AWS.

Para criar uma Integração ETL zero, especifique um banco de dados do Aurora RDS como a origem e um data warehouse ou lakehouse compatível como o destino. A integração replica os dados do banco de dados de origem no data warehouse ou lakehouse de destino.

O seguinte diagrama mostra essa funcionalidade de Integração ETL zero com o Amazon Redshift:



O seguinte diagrama mostra essa funcionalidade para uma Integração ETL zero com um lakehouse do Amazon SageMaker:



A integração monitora a integridade do pipeline de dados e se recupera de problemas quando possível. É possível criar integrações de vários bancos de dados do RDS em um único data warehouse ou lakehouse de destino, o que permite que você obtenha insights em várias aplicações.

Tópicos

- Benefícios
- Principais conceitos
- Limitações
- Cotas
- Regiões do compatíveis
- Conceitos básicos das integrações ETL zero do Amazon RDS com o Amazon Redshift
- Criar integrações ETL zero do Amazon Aurora com o Amazon Redshift
- Criar Integrações ETL zero entre o Amazon RDS e um lakehouse do Amazon SageMaker
- Filtragem de dados para Integrações ETL zero do Amazon RDS
- Adicionar dados a um banco de dados do RDS de origem e consultá-los no Amazon Redshift
- Visualizar e monitorar Integrações ETL zero do Amazon RDS
- Modificar Integrações ETL zero do Amazon RDS
- Excluir Integrações ETL zero do Amazon RDS

- Solução de problemas em integrações ETL zero do Amazon RDS com o Amazon Redshift

Benefícios

As Integrações ETL zero do RDS oferecem os seguintes benefícios:

- Ajudam você a obter insights holísticos de várias fontes de dados.
- Eliminam a necessidade de criar e manter canais de dados complexos que executam operações de extração, transformação e carregamento (ETL). As integrações ETL zero eliminam os desafios que surgem com a criação e o gerenciamento de pipelines, provisionando-os e gerenciando-os para você.
- Reduzem a carga e os custos operacionais para que você possa se concentrar em melhorar as aplicações.
- Permitem que você aproveite os recursos de analytics e ML do destino pretendido para obter insights de dados transacionais e outros dados e responder de forma eficaz a eventos críticos e urgentes.

Principais conceitos

Ao começar a usar integrações ETL zero, considere os seguintes conceitos:

Integração

Um pipeline de dados totalmente gerenciado que replica automaticamente dados e esquemas transacionais de um banco de dados do RDS em um data warehouse ou catálogo.

Banco de dados de origem

O banco de dados do RDS do qual os dados são replicados. É possível especificar uma instância de banco de dados single-AZ ou multi-AZ ou um cluster de banco de dados multi-AZ.

Alvo

O data warehouse ou lakehouse para o qual os dados são replicados. Há dois tipos de data warehouse: um data warehouse de cluster provisionado e um data warehouse sem servidor. Um data warehouse de cluster provisionado é um conjunto de recursos computacionais chamados nós, que são organizados em um grupo chamado cluster. Um data warehouse sem servidor é composto por um grupo de trabalho que armazena recursos computacionais e um namespace

que abriga os objetos e usuários do banco de dados. Ambos os data warehouses executam um mecanismo de analytics e contêm um ou mais bancos de dados.

Um lakehouse de destino consiste em catálogos, bancos de dados, tabelas e visualizações. Para ter mais informações sobre arquitetura de lakehouse, consulte Amazon SageMaker Lakehouse components no Guia do usuário do Amazon SageMaker Unified Studio.

Vários bancos de dados de origem podem gravar no mesmo destino.

Para obter mais informações, consulte Arquitetura do sistema de data warehouse no Guia do desenvolvedor do Amazon Redshift.

Limitações

As limitações a seguir se aplicam às Integrações ETL zero do RDS.

Tópicos

- Limitações gerais
- Limitações do RDS para MySQL
- Limitações do Amazon Redshift
- Limitações referentes a lakehouses do Amazon SageMaker

Limitações gerais

- O banco de dados de origem deve estar na mesma região que o destino.
- Não será possível renomear um banco de dados se o cluster tiver integrações existentes.
- Não é possível criar várias integrações entre os mesmos bancos de dados de origem e destino.
- Você não pode excluir um banco de dados que tenha integrações existentes. É necessário excluir todas as integrações correspondentes primeiro.
- Se você interromper o banco de dados de origem, as últimas transações provavelmente não serão replicadas no destino enquanto você não retomar o banco de dados.
- Você não poderá excluir uma integração se o banco de dados de origem for interrompido.
- Se o banco de dados for a origem de uma implantação azul/verde, os ambientes azul e verde não poderão ter integrações ETL zero existentes durante a transição. Você deve excluir a integração primeiro, alternar e, depois, recriá-la.

- Não é possível criar uma integração para um banco de dados de origem que tenha outra integração sendo criada ativamente.
- Quando você cria inicialmente uma integração ou quando uma tabela está sendo ressincronizada, a propagação de dados da origem para o destino pode levar de 20 a 25 minutos ou mais, dependendo do tamanho do banco de dados de origem. Esse atraso pode levar a um maior atraso na réplica.
- Alguns tipos de dados não compatíveis. Para obter mais informações, consulte the section called “Diferenças dos tipos de dados”.
- As tabelas do sistema, tabelas temporárias e visualizações não são replicadas nos warehouses de destino.

Limitações do RDS para MySQL

- O banco de dados de origem deve executar uma versão compatível do RDS para MySQL. Para conferir uma lista de versões compatíveis, consulte the section called “Integrações ETL zero ”.
- As integrações ETL zero dependem do registro em log binário (binlog) do MySQL para capturar alterações contínuas de dados. Não use a filtragem de dados baseada em log binário, pois isso pode causar inconsistências de dados entre os bancos de dados de origem e de destino.
- As integrações ETL zero são compatíveis apenas com bancos de dados configurados para usar o mecanismo de armazenamento InnoDB.
- Referências de chave externa com atualizações de tabelas predefinidas não são compatíveis. Especificamente, as regras `ON DELETE` e `ON UPDATE` não são compatíveis com as ações `CASCADE`, `SET NULL` e `SET DEFAULT`. A tentativa de criar ou atualizar uma tabela com essas referências a outra tabela colocará a tabela em um estado de falha.
- Operações de partição `ALTER TABLE` fazem com que a tabela seja ressincronizada para recarregar dados do RDS no warehouse de destino. A tabela não estará disponível para consulta durante a ressincronização. Para obter mais informações, consulte the section called “Uma ou mais das minhas tabelas do Amazon Redshift exigem ressincronização.”.

Limitações do Amazon Redshift

Para conferir uma lista das limitações do Amazon Redshift relacionadas às integrações ETL zero, consulte Considerações ao usar integrações ETL zero com o Amazon Redshift no Guia de gerenciamento do Amazon Redshift.

Limitações referentes a lakehouses do Amazon SageMaker

A seguir é apresentada uma limitação para Integrações ETL zero com lakehouses do Amazon SageMaker.

- Os nomes de catálogo podem ter no máximo 19 caracteres de comprimento.

Cotas

Sua conta tem as cotas a seguir relacionadas a Integrações ETL zero do RDS. Salvo indicação em contrário, cada cota aplica-se por região.

Name	Padrão	Descrição
Integrações	100	O número total de integrações em uma Conta da AWS.
Integrações por destino	50	O número de integrações que enviam dados a um único data warehouse ou lakehouse de destino.
de origem	5	O número de integrações que enviam dados de uma única instância de banco de dados de origem.

Além disso, o warehouse de destino impõe certos limites ao número de tabelas permitidas em cada instância de banco de dados ou nó de cluster. Para ter mais informações sobre cotas e limites do Amazon Redshift, consulte Cotas e limites no Amazon Redshift no Guida de gerenciamento do Amazon Redshift.

Regiões do compatíveis

As Integrações ETL zero do RDS estão disponíveis em um subconjunto de Regiões da AWS. Para obter uma lista de regiões compatíveis, consulte the section called “Integrações ETL zero ”.

Conceitos básicos das integrações ETL zero do Amazon RDS com o Amazon Redshift

Esta é a documentação de pré-lançamento das Integrações ETL zero do Amazon RDS com o Amazon Redshift, que está em versão de pré-visualização. A documentação e o atributo estão sujeitos a alterações. Recomendamos o uso desse atributo somente em ambientes de teste, e não em ambientes de produção. Para conferir os termos e condições da pré-visualização, consulte Betas e pré-visualizações nos Termos de serviços da AWS.

Antes de criar uma integração ETL zero com o Amazon Redshift, configure o banco de dados do RDS e o data warehouse do Amazon Redshift com os parâmetros e as permissões necessários. Durante a configuração, você realizará as seguintes etapas:

1. Criar um grupo de parâmetros de de banco de dados personalizado.
2. Crie um banco de dados de origem.
3. Criar um data warehouse de destino do Amazon Redshift

Depois de concluir essas tarefas, prossiga para the section called “Criar Integrações ETL zero com o Amazon Redshift”.

Etapa 1: Criar um grupo de parâmetros de de banco de dados personalizado

As integrações ETL zero do Amazon RDS com o Amazon Redshift exigem valores específicos para os parâmetros do banco de dados que controlam o registro em log binário (binlog). Para configurar o registro em log binário, primeiro é necessário criar um grupo de parâmetros de cluster de banco de dados personalizado e, depois, associá-lo ao banco de dados de origem.

Crie um grupo de parâmetros de de banco de dados personalizado com as seguintes configurações, . Para obter instruções de como criar um grupo de parâmetros, consulte the section called “Grupos de parâmetros do banco de dados”.

- `binlog_format=ROW`
- `binlog_row_image=full`
- `binlog_checksum=NONE`

Além disso, verifique se o parâmetro `binlog_row_value_options` não está definido como `PARTIAL_JSON`.

Etapa 2: Selecionar ou criar um banco de dados de origem

Depois de criar um grupo de parâmetros de de banco de dados personalizado, selecione ou crie uma instância de banco de dados de uma única zona de disponibilidade ou multi-AZ do RDS para MySQL. Esse banco de dados será a origem da replicação de dados para o Amazon Redshift.

O banco de dados deve estar executando o RDS para MySQL versão 8.0.32 ou posterior. Consulte instruções para criar uma instância de banco de dados de uma única zona de disponibilidade ou multi-AZ, consulte the section called “Criar uma instância de banco de dados”.

Em Configuração adicional, altere o grupo de parâmetros de de banco de dados padrão para o grupo de parâmetros personalizado que você criou na etapa anterior.

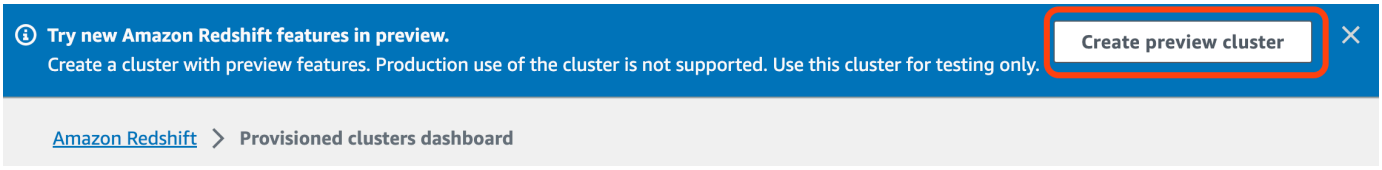
> **Note**
>
> se você associar o grupo de parâmetros ao banco de dados depois que o banco de dados for criado, você deverá reinicializar o banco de dados para aplicar as alterações antes de criar uma Integração ETL zero. Para obter instruções, consulte the section called “Reinicializar uma instância de banco de dados”.

Além disso, certifique-se de que os backups automatizados estejam habilitados no banco de dados. Para ter mais informações, consulte the section called “Ativar backups automáticos”.

Etapa 3: Criar um data warehouse de destino do Amazon Redshift

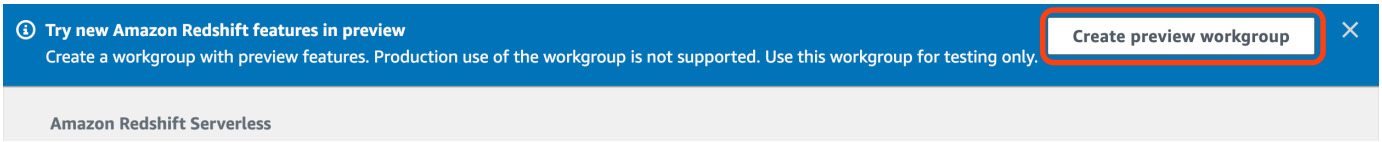
Depois de criar o banco de dados, será necessário criar e configurar um data warehouse de destino no Amazon Redshift. O data warehouse deve cumprir os seguintes requisitos:

- Criado na pré-visualização
  - Para criar um cluster provisionado em versão prévia, escolha Criar cluster de versão prévia no banner do painel de clusters provisionados. Para obter mais informações, consulte Criar um cluster de versão prévia.



Ao criar o cluster, defina a faixa de versão prévia como `preview_2023`.

- Para criar um grupo de trabalho de tecnologia sem servidor do Redshift na versão prévia, escolha Criar grupo de trabalho de versão prévia no banner do painel Tecnologia sem servidor. Para obter mais informações, consulte Criar um grupo de trabalho de versão prévia.



- Usar um tipo de nó RA3 (`ra3.xlplus`, `ra3.4xlarge` ou `ra3.16xlarge`) com pelo menos dois nós ou o Redshift sem servidor.
- Ser criptografado (se estiver usando um cluster provisionado). Para obter mais informações, consulte Criptografia de bancos de dados no Amazon Redshift.

Para obter instruções sobre como criar um data warehouse, consulte Criar um cluster para clusters provisionados ou Criar um grupo de trabalho com um namespace para o Redshift Serverless.

Ative a distinção entre maiúsculas e minúsculas no data warehouse

Para que a integração seja bem-sucedida, o parâmetro de diferenciação de maiúsculas e minúsculas (`enable_case_sensitive_identifier`) deve estar ativado para o data warehouse. Por padrão, a distinção entre maiúsculas e minúsculas é desativada em todos os clusters provisionados e grupos de trabalho do Redshift Serverless.

Para ativar a distinção entre maiúsculas e minúsculas, execute as seguintes etapas, dependendo do tipo de data warehouse:

- Cluster provisionado: para ativar a distinção entre maiúsculas e minúsculas em um cluster provisionado, crie um grupo de parâmetros personalizado com o parâmetro `enable_case_sensitive_identifier` ativado. Em seguida, associe o grupo de parâmetros ao cluster. Para obter instruções, consulte Gerenciar grupos de parâmetros usando o console ou Configurar valores de parâmetros usando a AWS CLI.

> **Note**
>
> Lembre-se de reinicializar o cluster depois de associar o grupo de parâmetros personalizado a ele.

- Grupo de trabalho de tecnologia sem servidor: para ativar a distinção entre maiúsculas e minúsculas em um grupo de trabalho do Redshift Serverless, você deve usar a AWS CLI. Atualmente, o console do Amazon Redshift não é compatível com a modificação dos valores dos parâmetros do Redshift Serverless. Envie a seguinte solicitação de atualização do grupo de trabalho:

```
aws redshift-serverless update-workgroup \
  --workgroup-name target-workgroup \
  --config-parameters
 parameterKey=enable_case_sensitive_identifier,parameterValue=true
```

  Não é necessário reinicializar um grupo de trabalho após modificar seus valores de parâmetros.

Configurar a autorização para o data warehouse

Depois de criar um data warehouse, você deve configurar o banco de dados do RDS de origem como uma origem de integração autorizada. Para obter instruções, consulte Configurar a autorização para o data warehouse do Amazon Redshift.

Próximas etapas

Com um banco de dados do RDS de origem e um data warehouse de destino do Amazon Redshift, agora você pode criar uma Integração ETL zero e começar a replicar dados. Para obter instruções, consulte the section called “Criar Integrações ETL zero com o Amazon Redshift”.

Criar integrações ETL zero do Amazon Aurora com o Amazon Redshift

Ao criar uma integração ETL zero do Amazon RDS, especifique o banco de dados RDS de origem e o data warehouse de destino do Amazon Redshift. Você também pode personalizar as configurações de criptografia e adicionar etiquetas. O Amazon RDS cria uma integração entre o banco de dados de

origem e o destino. Quando a integração está ativa, todos os dados inseridos no banco de dados de origem serão replicados no destino configurado do Amazon Redshift.

Pré-requisitos

Antes de criar uma integração ETL zero, é necessário criar um banco de dados de origem e um data warehouse de destino do Amazon Redshift. Também é necessário permitir a replicação no data warehouse adicionando o banco de dados como uma origem de integração autorizada.

Para obter instruções sobre como concluir cada uma dessas etapas, consulte the section called "Conceitos básicos das integrações ETL zero".

Permissões obrigatórias

Algumas permissões do IAM são necessárias para criar uma integração ETL zero. No mínimo, são necessárias permissões para executar as seguintes ações:

- Crie integrações ETL zero para o banco de dados do RDS de origem.
- Visualizar e excluir todas as integrações ETL zero.
- Criar integrações de entrada no data warehouse de destino.

As políticas de exemplo a seguir demonstram as permissões de privilégio mínimo necessárias para criar e gerenciar integrações. Talvez você não precise dessas permissões exatas se o usuário ou o perfil tiver permissões mais amplas, como uma política gerenciada `AdministratorAccess`.

> **Note**
>
> Os ARNs do Amazon Redshift têm o formato a seguir. Observe o uso de uma barra (`/`) em vez de dois pontos (`:`) antes do UUID do namespace sem servidor.
>
> - Cluster provisionado: `arn:aws:redshift:{region}:{account-id}:namespace:namespace-uuid`
> - Sem servidor: `arn:aws:redshift-serverless:{region}:{account-id}:namespace/namespace-uuid`

Exemplo de política para o destino do Redshift

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Action": [
            "rds:CreateIntegration"
        ],
        "Resource": [
            "arn:aws:rds:{region}:{account-id}:db:source-db",
            "arn:aws:rds:{region}:{account-id}:integration:*"
        ]
    },
    {
        "Effect": "Allow",
        "Action": [
            "rds:DescribeIntegrations"
        ],
        "Resource": ["*"]
    },
    {
        "Effect": "Allow",
        "Action": [
            "rds:DeleteIntegration",
            "rds:ModifyIntegration"
        ],
        "Resource": [
            "arn:aws:rds:{region}:{account-id}:integration:*"
        ]
    },
    {
        "Effect": "Allow",
        "Action": [
            "redshift:CreateInboundIntegration"
        ],
        "Resource": [
            "arn:aws:redshift:{region}:{account-id}:namespace:namespace-uuid"
        ]
    }]
```

```
}
```

Escolher um data warehouse de destino em uma conta diferente

Se você planeja especificar um data warehouse de destino do Amazon Redshift que esteja em outraConta da AWS, deverá criar um perfil que permita que os usuários da conta atual acessem os recursos na conta de destino. Para obter mais informações, consulte Fornecer acesso a um usuário do IAM em outra Conta da AWS de sua propriedade.

O perfil deve ter as permissões a seguir, que possibilitam ao usuário visualizar os clusters provisionados do Amazon Redshift e os namespaces do Redshift sem servidor disponíveis na conta de destino.

Permissões necessárias e política de confiança

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "redshift:DescribeClusters",
            "redshift-serverless:ListNamespaces"
         ],
         "Resource":[
            "*"
         ]
      }
   ]
}
```

O perfil deve ter a seguinte política de confiança, que especifica o ID da conta de destino.

JSON

```
{
```

```
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "AWS": "arn:aws:iam::{external-account-id}:root"
         },
         "Action":"sts:AssumeRole"
      }
   ]
}
```

Para obter instruções sobre como criar o perfil, consulte Criar um perfil usando políticas de confiança personalizadas.

Criar integrações ETL zero

É possível criar uma integração ETL zero usando o AWS Management Console, a AWS CLI ou a API do RDS.

> **Important**
>
> As Integrações ETL zero não permitem operações de atualização ou ressincronização. Se você encontrar problemas com uma integração após a criação, deverá excluir a integração e criar outra.

Por padrão, o RDS para MySQL limpa imediatamente os arquivos de log binários. Como as integrações ETL zero dependem de logs binários para replicar dados da origem para o destino, o período de retenção do banco de dados de origem deve ser de pelo menos uma hora. Assim que você cria uma integração, o Amazon RDS verifica o período de retenção do arquivo de log binário para o banco de dados de origem selecionado. Se o valor atual for 0 hora, o Amazon RDS o alterará automaticamente para 1 hora. Caso contrário, o valor permanecerá o mesmo.

Console do RDS

Como criar uma integração ETL zero

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.

2. No painel de navegação à esquerda, escolha Integrações ETL zero.
3. Escolha Criar integração ETL zero.
4. Em Nome da integração, insira um nome para a integração. O nome pode ter até 60 caracteres alfanuméricos e pode incluir hifens.

> **Important**
>
> Os nomes de catálogo podem ter no máximo 19 caracteres de comprimento. O identificador da integração deve atender a esse requisito se for usado como nome de catálogo.

5. Escolha Próximo.
6. Em Origem, selecione o banco de dados do RDS de onde os dados serão originados.

> **Note**
>
> O RDS notifica você se os parâmetros do de banco de dados não estiverem configurados corretamente. Se você receber essa mensagem, poderá escolher Corrigir para mim ou configurá-las manualmente. Para obter instruções sobre como corrigi-los manualmente, consultethe section called “Etapa 1: Criar um grupo de parâmetros de de banco de dados personalizado”.
>
> A modificação dos parâmetros de banco de dados requer uma reinicialização. Antes de criar a integração, a reinicialização deve ser concluída e os valores do novo parâmetro deve ser aplicado com êxito ao banco de dados.

7. Depois que o banco de dados de origem for configurado com êxito, selecione Próximo.
8. Em Destino 1, faça o seguinte:

    1. (Opcional) Para usar uma Conta da AWS diferente para o destino do Amazon Redshift, escolha Especificar uma conta diferente. Depois, insira o ID da e o nome de um perfil do IAM com permissões para exibir seus data warehouses. Para obter instruções sobre como criar um perfil do IAM, consulte the section called “Escolher um data warehouse de destino em uma conta diferente”.
    2. Em Data warehouse do Amazon Redshift, selecione o destino para os dados replicados do banco de dados de origem. Você pode escolher um cluster provisionado do Amazon Redshift ou um namespace do Redshift sem servidor como destino.

> **Note**
>
> O RDS notifica você se a política de recursos ou as configurações de diferenciação de maiúsculas e minúsculas do data warehouse especificado não estiverem configuradas corretamente. Se você receber essa mensagem, poderá escolher Corrigir para mim ou configurá-las manualmente. Para obter instruções sobre como corrigi-los manualmente, consulte Ativar a diferenciação de maiúsculas e minúsculas para seu armazém de dados e Configurar a autorização para seu armazém de dados no Guia de gerenciamento do Amazon Redshift.
> A modificação da distinção entre maiúsculas e minúsculas de um cluster provisionado do Redshift requer uma reinicialização. Antes de criar a integração, a reinicialização deve ser concluída e o novo valor do parâmetro deve ser aplicado com êxito ao cluster.
> Se a origem e o destino selecionados estiverem em valores diferentesContas da AWS, o Amazon RDS não poderá corrigir essas configurações para você. Você deve navegar até a outra conta e corrigi-la manualmente no Amazon Redshift.

9. Depois que seu data warehouse de destino estiver configurado corretamente, escolha Avançar.
10. (Opcional) Em Etiquetas, adicione uma ou mais etiquetas à integração. Para obter mais informações, consulte the section called “Marcar recursos do RDS”.
11. Em Criptografia, especifique como você deseja que sua integração seja criptografada. Por padrão, o RDS criptografa todas as integrações com uma Chave pertencente à AWS. Para escolher uma chave gerenciada pelo cliente em vez disso, habilite a opção Personalizar configurações de criptografia e escolha uma chave do KMS para criptografia. Para obter mais informações, consulte the section called “Criptografar recursos do Amazon RDS”.

    Como opção, adicione um contexto de criptografia. Para obter mais informações, consulte Contexto de criptografia no Guia do desenvolvedor AWS Key Management Service.

    > **Note**
    >
    > O Amazon RDS adiciona os seguintes pares de contexto de criptografia, além dos adicionados por você:
    >
    > - `aws:redshift:integration:arn` - `IntegrationArn`
    > - `aws:servicename:id` - `Redshift`

Isso reduz o número geral de pares que você pode adicionar de oito para seis e contribui com o limite geral de caracteres da restrição de concessão. Para ter mais informações, consulte Using grant constraints no Guia do desenvolvedor do AWS Key Management Service.

12. Escolha Próximo.
13. Revise suas configurações de integração e escolha Criar integração sem ETL.

    Se a criação falhar, consulte the section called “Não consigo criar uma integração ETL zero.” para conferir etapas de solução de problemas.

A integração tem um status de `Creating` enquanto está sendo criada, e o data warehouse de destino do Amazon Redshift tem um status de `Modifying`. Durante esse período, você não pode consultar o data warehouse nem alterar sua configuração.

Quando a integração tiver sido criada com sucesso, o status da integração e do data warehouse de destino do Amazon Redshift vão mudar para `Active`.

AWS CLI

Para criar uma integração sem ETL usando oAWS CLI, use o comando create-integration com as seguintes opções:

> **Note**
>
> Lembre-se de que os nomes de catálogo podem ter no máximo 19 caracteres. Escolha o nome da integração adequadamente se você pretende usá-lo como nome de catálogo.

- `--integration-name`: especifique um nome para a integração.
- `--source-arn`: especifique o ARN do banco de dados do RDS que será a origem da integração.
- `--target-arn`: especifique o ARN do data warehouse do Amazon Redshift que será o destino da integração.

Example

Para Linux, macOS ou Unix:

```
aws rds create-integration \
    --integration-name my-integration \
    --source-arn arn:aws:rds:{region}:{account-id}:my-db \
    --target-arn arn:aws:redshift:{region}:{account-id}:namespace:namespace-uuid
```

Para Windows:

```
aws rds create-integration ^
    --integration-name my-integration ^
    --source-arn arn:aws:rds:{region}:{account-id}:my-db ^
    --target-arn arn:aws:redshift:{region}:{account-id}:namespace:namespace-uuid
```

API do RDS

Para criar uma implantação azul/verde usando a API do Amazon RDS, use a operação `CreateIntegration` com os seguintes parâmetros:

> **Note**
>
> Os nomes de catálogo podem ter no máximo 19 caracteres. O parâmetro IntegrationName deve atender a esse requisito se for usado como nome de catálogo.

- `IntegrationName`: especifique um nome para a integração.
- `SourceArn`: especifique o ARN do banco de dados do RDS que será a origem da integração.
- `TargetArn`: especifique o ARN do data warehouse do Amazon Redshift que será o destino da integração.

Criptografar integrações com uma chave gerenciada pelo cliente

Se você especificar uma chave do KMS personalizada em vez de uma Chave pertencente à AWS ao criar uma integração, a política de chaves deverá fornecer ao serviço Amazon Redshift um acesso de entidade principal à ação `CreateGrant`. Além disso, ela deve permitir que o usuário atual execute as ações `DescribeKey` e `CreateGrant`.

O exemplo de política a seguir demonstra como conceder as permissões necessárias na política de chave. Ele inclui chaves de contexto para reduzir ainda mais o escopo das permissões.

Exemplo de política de chave

JSON

```
{
    "Version": "2012-10-17",
    "Id": "Key policy",
    "Statement": [
        {
            "Sid": "Enables IAM user permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::{account-ID}:root"
            },
            "Action": "kms:*",
            "Resource": "*"
        },
        {
            "Sid": "Allows the Redshift service principal to add a grant to a KMS
 key",
            "Effect": "Allow",
            "Principal": {
                "Service": "redshift.amazonaws.com"
            },
            "Action": "kms:CreateGrant",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "kms:EncryptionContext:{context-key}":"{context-value}"
                },
                "ForAllValues:StringEquals": {
                    "kms:GrantOperations": [
                        "Decrypt",
                        "GenerateDataKey",
                        "CreateGrant"
                    ]
                }
            }
        },
        {
            "Sid": "Allows the current user or role to add a grant to a KMS key",
            "Effect": "Allow",
            "Principal": {
```

```
                "AWS": "arn:aws:iam::{account-ID}:role/{role-name}"
            },
            "Action": "kms:CreateGrant",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "kms:EncryptionContext:{context-key}":"{context-value}",
                    "kms:ViaService": "rds.us-east-1.amazonaws.com"
                },
                "ForAllValues:StringEquals": {
                    "kms:GrantOperations": [
                        "Decrypt",
                        "GenerateDataKey",
                        "CreateGrant"
                    ]
                }
            }
        },
        {
            "Sid": "Allows the current uer or role to retrieve information about
 a KMS key",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::{account-ID}:role/{role-name}"
            },
            "Action": "kms:DescribeKey",
            "Resource": "*"
        }
    ]
}
```

Para obter mais informações, consulte Criar uma política de chave no Guia do desenvolvedor do AWS Key Management Service.

Próximas etapas

Depois de criar com sucesso uma integração ETL zero, você deve criar um banco de dados de destino em seu cluster ou grupo de trabalho de destino do Amazon Redshift. Depois, você poderá começar a adicionar dados ao banco de dados do RDS e consultá-los no Amazon Redshift. Para obter instruções, consulte Criar bancos de dados de destino no Amazon Redshift.

Criar Integrações ETL zero entre o Amazon RDS e um lakehouse do Amazon SageMaker

Ao criar uma Integração ETL zero entre o Amazon RDS e um lakehouse do Amazon SageMaker, você deve especificar o banco de dados do RDS de origem e o catálogo de destino gerenciado pelo AWS Glue. Você também pode personalizar as configurações de criptografia e adicionar etiquetas. O Amazon RDS cria uma integração entre o banco de dados de origem e o destino. Quando a integração estiver ativa, todos os dados inseridos no banco de dados de origem serão replicados no destino configurado.

Pré-requisitos

Antes de criar uma Integração ETL zero com um lakehouse do Amazon SageMaker, você deve criar um banco de dados de origem e um catálogo de destino gerenciado pelo AWS Glue. Também é necessário permitir a replicação no catálogo adicionando o banco de dados como uma origem de integração autorizada.

Para obter instruções sobre como concluir cada uma dessas etapas, consulte the section called “Conceitos básicos das integrações ETL zero”.

Permissões obrigatórias

Algumas permissões do IAM são necessárias para criar uma Integração ETL zero com o lakehouse do Amazon SageMaker. No mínimo, são necessárias permissões para executar as seguintes ações:

- Crie integrações ETL zero para o banco de dados do RDS de origem.
- Visualizar e excluir todas as integrações ETL zero.
- Criar integrações de entrada no catálogo de destino gerenciado pelo AWS Glue.
- Acessar os buckets do Amazon S3 usados pelo catálogo gerenciado pelo AWS Glue.
- Usar chaves do AWS KMS para criptografia se a criptografia personalizada estiver configurada.
- Registrar recursos com o Lake Formation.
- Inserir a política de recursos no catálogo gerenciado pelo AWS Glue para autorizar integrações de entrada.

O exemplo de política a seguir demonstra as permissões de privilégio mínimo necessárias para criar e gerenciar integrações com um lakehouse do Amazon SageMaker. Talvez você não precise

dessas permissões exatas se o usuário ou o perfil tiver permissões mais amplas, como uma política gerenciada `AdministratorAccess`.

Além disso, você deve configurar uma política de recursos no catálogo de destino gerenciado pelo AWS Glue para autorizar integrações de entrada. Use o comando da AWS CLI a seguir para aplicar a política de recursos.

Exemplo de comando da AWS CLI para autorizar integrações de entrada no catálogo de destino

```
aws glue put-resource-policy \
      --policy-in-json  '{
    "Version": "2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Principal": {
            "Service": "glue.amazonaws.com"
        },
        "Action": [
            "glue:AuthorizeInboundIntegration"
        ],
        "Resource": ["arn:aws:glue:region:account_id:catalog/catalog_name"],
        "Condition": {
            "StringEquals": {
                "aws:SourceArn": "arn:aws:rds:region:account_id:db:source_name"
            }
        }
    },
    {
        "Effect": "Allow",
        "Principal": {
            "AWS": "account_id"
        },
        "Action": ["glue:CreateInboundIntegration"],
        "Resource": ["arn:aws:glue:region:account_id:catalog/catalog_name"]
    }
    ]
}' \
      --region region
```

> **Note**
>
> Os nomes de recurso da Amazon (ARNs) do catálogo do Glue têm o seguinte formato:

- Catálogo do Glue: `arn:aws:glue:{region}:{account-id}:catalog/catalog-name`.

Escolher um catálogo de destino gerenciado pelo AWS Glue em uma conta diferente

Se você planeja especificar um catálogo de destino gerenciado pelo AWS Glue que esteja em outra Conta da AWS, deverá criar um perfil que permita que os usuários da conta atual acessem os recursos na conta de destino. Para obter mais informações, consulte Fornecer acesso a um usuário do IAM em outra Conta da AWS de sua propriedade.

O perfil deve ter as permissões a seguir, que possibilitam ao usuário visualizar os catálogos disponíveis do AWS Glue na conta de destino.

Permissões necessárias e política de confiança

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "glue:GetCatalog"
         ],
         "Resource":[
            "*"
         ]
      }
   ]
}
```

O perfil deve ter a seguinte política de confiança, que especifica o ID da conta de destino.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "AWS": "arn:aws:iam::{external-account-id}:root"
         },
```

```
            "Action":"sts:AssumeRole"
        }
    ]
}
```

Para obter instruções sobre como criar o perfil, consulte Criar um perfil usando políticas de confiança personalizadas.

Criar Integrações ETL zero com um lakehouse do Amazon SageMaker

É possível criar uma Integração ETL zero com um lakehouse do Amazon SageMakerusando o AWS Management Console, a AWS CLI ou a API do RDS.

> **Important**
>
> Em Integrações ETL zero com um lakehouse do Amazon SageMaker, não é possível usar operações de atualização ou ressincronização. Se você encontrar problemas com uma integração após a criação, deverá excluir a integração e criar outra.

Por padrão, o RDS para MySQL limpa imediatamente os arquivos de log binários. Como as integrações ETL zero dependem de logs binários para replicar dados da origem para o destino, o período de retenção do banco de dados de origem deve ser de pelo menos uma hora. Assim que você cria uma integração, o Amazon RDS verifica o período de retenção do arquivo de log binário para o banco de dados de origem selecionado. Se o valor atual for 0 hora, o Amazon RDS o alterará automaticamente para 1 hora. Caso contrário, o valor permanecerá o mesmo.

Console do RDS

Como criar Integrações ETL zero com um lakehouse do Amazon SageMaker

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação à esquerda, escolha Integrações ETL zero.
3. Escolha Criar integração ETL zero.
4. Em Nome da integração, insira um nome para a integração. O nome pode ter até 60 caracteres alfanuméricos e pode incluir hifens.
5. Escolha Próximo.

6. Em Origem, selecione o banco de dados do RDS de onde os dados serão originados.

> **Note**
>
> O RDS notifica você se os parâmetros do de banco de dados não estiverem configurados corretamente. Se você receber essa mensagem, poderá escolher Corrigir para mim ou configurá-las manualmente. Para obter instruções sobre como corrigi-los manualmente, consultethe section called “Etapa 1: Criar um grupo de parâmetros de de banco de dados personalizado”.
>
> A modificação dos parâmetros de banco de dados requer uma reinicialização. Antes de criar a integração, a reinicialização deve ser concluída e os valores do novo parâmetro deve ser aplicado com êxito ao banco de dados.

7. Depois que o banco de dados de origem for configurado com êxito, selecione Próximo.
8. Em Destino 1, faça o seguinte:

   1. (Opcional) Para usar uma Conta da AWS diferente para o lakehouse de destino do Amazon SageMaker, escolha Especificar uma conta diferente. Depois, insira o ARN de um perfil do IAM com permissões para exibir seus catálogos do AWS Glue. Para obter instruções sobre como criar um perfil do IAM, consulte the section called “Escolher um catálogo de destino gerenciado pelo AWS Glue em uma conta diferente”.
   2. Em Catálogo do AWS Glue, selecione o destino para os dados replicados do banco de dados de origem. Você pode escolher um catálogo existente gerenciado pelo AWS Glue como destino.
   3. O perfil do IAM de destino precisa de permissões de descrição no catálogo de destino e deve ter as seguintes permissões:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": "glue:GetCatalog",
            "Resource": [
                "arn:aws:glue:region:account-id:catalog/*",
                "arn:aws:glue:region:account-id:catalog"
            ]
        }
```

```
    ]
}
```

O perfil do IAM deve ter a seguinte relação de confiança:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "glue.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

4. Você deve conceder ao perfil do IAM de destino as permissões de descrição referentes ao catálogo de destino gerenciado pelo AWS Glue com o perfil de administrador do Lake Formation criado na seção ???.

> **Note**
>
> O RDS notificará você se a política de recursos ou as configurações do catálogo gerenciado pelo AWS Glue especificado não estiverem configuradas corretamente. Se você receber essa mensagem, poderá escolher Corrigir para mim ou configurá-las manualmente.
>
> Se a origem e o destino selecionados estiverem em valores diferentesContas da AWS, o Amazon RDS não poderá corrigir essas configurações para você. Você deve navegar até a outra conta e corrigi-la manualmente no SageMaker Unified Studio.

9. Depois que o catálogo de destino gerenciado pelo AWS Glue estiver configurado corretamente, escolha Próximo.
10. (Opcional) Em Etiquetas, adicione uma ou mais etiquetas à integração. Para obter mais informações, consulte the section called “Marcar recursos do RDS”.
11. Em Criptografia, especifique como você deseja que sua integração seja criptografada. Por padrão, o RDS criptografa todas as integrações com uma Chave pertencente à AWS. Para

escolher uma chave gerenciada pelo cliente em vez disso, habilite a opção Personalizar configurações de criptografia e escolha uma chave do KMS para criptografia. Para obter mais informações, consulte the section called “Criptografar recursos do Amazon RDS”.

Como opção, adicione um contexto de criptografia. Para obter mais informações, consulte Contexto de criptografia no Guia do desenvolvedor AWS Key Management Service.

> **Note**
>
> O Amazon RDS adiciona os seguintes pares de contexto de criptografia, além dos adicionados por você:
>
> - `aws:glue:integration:arn` - `IntegrationArn`
> - `aws:servicename:id` - `glue`
>
> Isso reduz o número geral de pares que você pode adicionar de oito para seis e contribui com o limite geral de caracteres da restrição de concessão. Para ter mais informações, consulte Using grant constraints no Guia do desenvolvedor do AWS Key Management Service.

12. Escolha Próximo.
13. Revise suas configurações de integração e escolha Criar integração sem ETL.

    Se a criação falhar, consulte the section called “Solução de problemas em integrações ETL zero” para conferir etapas de solução de problemas.

A integração tem o status `Creating` enquanto está sendo criada, e o lakehouse de destino do Amazon SageMaker tem o status `Modifying`. Durante esse período, não é possível consultar o catálogo nem alterar sua configuração.

Quando a integração é criada com êxito, o status da integração e o lakehouse de destino do Amazon SageMaker mudam para `Active`.

AWS CLI

Para preparar um catálogo de destino gerenciado pelo AWS Glue para Integração ETL zero usando a AWS CLI, você deve primeiro usar o comando create-integration-resource-property com as seguintes opções:

- `--resource-arn`: especifique o ARN do catálogo gerenciado pelo AWS Glue que será o destino da integração.
- `--target-processing-properties`: especifique o ARN do perfil do IAM para acessar o catálogo de destino gerenciado pelo AWS Glue.

```
aws glue create-integration-resource-property --region us-east-1
 --resource-arn arn:aws:glue:region:account_id:catalog/catalog_name \
 --target-processing-properties '{"RoleArn" : "arn:aws:iam::account_id:role/
TargetIamRole"}'
```

Para criar uma Integração ETL zero com um lakehouse do Amazon SageMaker usando a AWS CLI, use o comando create-integration com as seguintes opções:

- `--integration-name`: especifique um nome para a integração.
- `--source-arn`: especifique o ARN do banco de dados do RDS que será a origem da integração.
- `--target-arn`: especifique o ARN do catálogo gerenciado pelo AWS Glue que será o destino da integração.

Example

Para Linux, macOS ou Unix:

```
aws rds create-integration \
    --integration-name my-sagemaker-integration \
    --source-arn arn:aws:rds:{region}:{account-id}:my-db \
    --target-arn arn:aws:glue:{region}:{account-id}:catalog/catalog-name
```

Para Windows:

```
aws rds create-integration ^
    --integration-name my-sagemaker-integration ^
    --source-arn arn:aws:rds:{region}:{account-id}:my-db ^
    --target-arn arn:aws:glue:{region}:{account-id}:catalog/catalog-name
```

API do RDS

Para criar uma Integração ETL zero como Amazon SageMaker usando a API do Amazon RDS, use a operação `CreateIntegration` com os seguintes parâmetros:

> **Note**
>
> Os nomes de catálogo podem ter no máximo 19 caracteres. O parâmetro IntegrationName deve atender a esse requisito se for usado como nome de catálogo.

- `IntegrationName`: especifique um nome para a integração.
- `SourceArn`: especifique o ARN do banco de dados do RDS que será a origem da integração.
- `TargetArn`: especifique o ARN do catálogo gerenciado pelo AWS Glue que será o destino da integração.

Criptografar integrações com uma chave gerenciada pelo cliente

Se você especificar uma chave do KMS personalizada em vez de uma Chave pertencente à AWS ao criar uma integração com o Amazon SageMaker, a política de chave deverá fornecer à entidade principal do serviço SageMaker Unified Studio acesso à ação `CreateGrant`. Além disso, ela deve permitir que o usuário atual execute as ações `DescribeKey` e `CreateGrant`.

O exemplo de política a seguir demonstra como conceder as permissões necessárias na política de chave. Ele inclui chaves de contexto para reduzir ainda mais o escopo das permissões.

Exemplo de política de chave

```
{
    "Version": "2012-10-17",
    "Id": "Key policy",
    "Statement": [
        {
            "Sid": "Enables IAM user permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::{account-ID}:root"
            },
            "Action": "kms:*",
            "Resource": "*"
        },
        {
            "Sid": "Allows the Glue service principal to add a grant to an AWS KMS
 key",
            "Effect": "Allow",
```

```
            "Principal": {
                "Service": "glue.amazonaws.com"
            },
            "Action": "kms:CreateGrant",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "kms:EncryptionContext:{context-key}":"{context-value}"
                },
                "ForAllValues:StringEquals": {
                    "kms:GrantOperations": [
                        "Decrypt",
                        "GenerateDataKey",
                        "CreateGrant"
                    ]
                }
            }
        },
        {
            "Sid": "Allows the current user or role to add a grant to a KMS key",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::{account-ID}:role/{role-name}"
            },
            "Action": "kms:CreateGrant",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "kms:EncryptionContext:{context-key}":"{context-value}",
                    "kms:ViaService": "rds.us-east-1.amazonaws.com"
                },
                "ForAllValues:StringEquals": {
                    "kms:GrantOperations": [
                        "Decrypt",
                        "GenerateDataKey",
                        "CreateGrant"
                    ]
                }
            }
        },
        {
            "Sid": "Allows the current uer or role to retrieve information about a KMS
key",
            "Effect": "Allow",
```

```
            "Principal": {
                "AWS": "arn:aws:iam::{account-ID}:role/{role-name}"
            },
            "Action": "kms:DescribeKey",
            "Resource": "*"
        }
    ]
}
```

Para obter mais informações, consulte Criar uma política de chave no Guia do desenvolvedor do AWS Key Management Service.

Próximas etapas

Depois de criar com sucesso uma Integração ETL zero com o Amazon SageMaker, você poderá começar a adicionar dados ao banco de dados do RDS de origem e consultá-los no lakehouse do Amazon SageMaker. Os dados serão replicados automaticamente e disponibilizados para workloads de analytics e machine learning.

Filtragem de dados para Integrações ETL zero do Amazon RDS

As Integrações ETL zero do Amazon RDS permitem filtragem de dados, o que possibilita controlar quais dados são replicados do banco de dados do Amazon RDS de origem para o data warehouse de destino. Em vez de replicar todo o banco de dados, você pode aplicar um ou mais filtros para incluir ou excluir seletivamente tabelas específicas. Isso ajuda a otimizar o desempenho do armazenamento e das consultas, garantindo que somente os dados relevantes sejam transferidos. No momento, a filtragem é limitada aos níveis de banco de dados e tabela. Não é possível aplicar filtragem em nível de coluna e linha.

A filtragem de dados pode ser útil quando você deseja:

- Una determinadas tabelas de dois ou mais bancos de dados de origem diferentes para não precisar de dados completos de nenhum banco de dados.
- Economize custos realizando análises usando apenas um subconjunto de tabelas em vez de uma frota inteira de bancos de dados.
- Filtre informações confidenciais, como números de telefone, endereços ou detalhes do cartão de crédito, de determinadas tabelas.

É possível adicionar filtros de dados a uma integração ETL zero usando o AWS Management Console, a AWS Command Line Interface (AWS CLI) ou a API do Amazon RDS.

Se a integração tiver um cluster provisionado como destino, o cluster deverá estar no patch 180 ou posterior para usar a filtragem de dados.

Tópicos

- Formato de um filtro de dados
- Lógica de filtros
- Precedência do filtro
- Exemplos
- Adicionar filtros de dados a uma integração
- Remover filtros de dados de uma integração

Formato de um filtro de dados

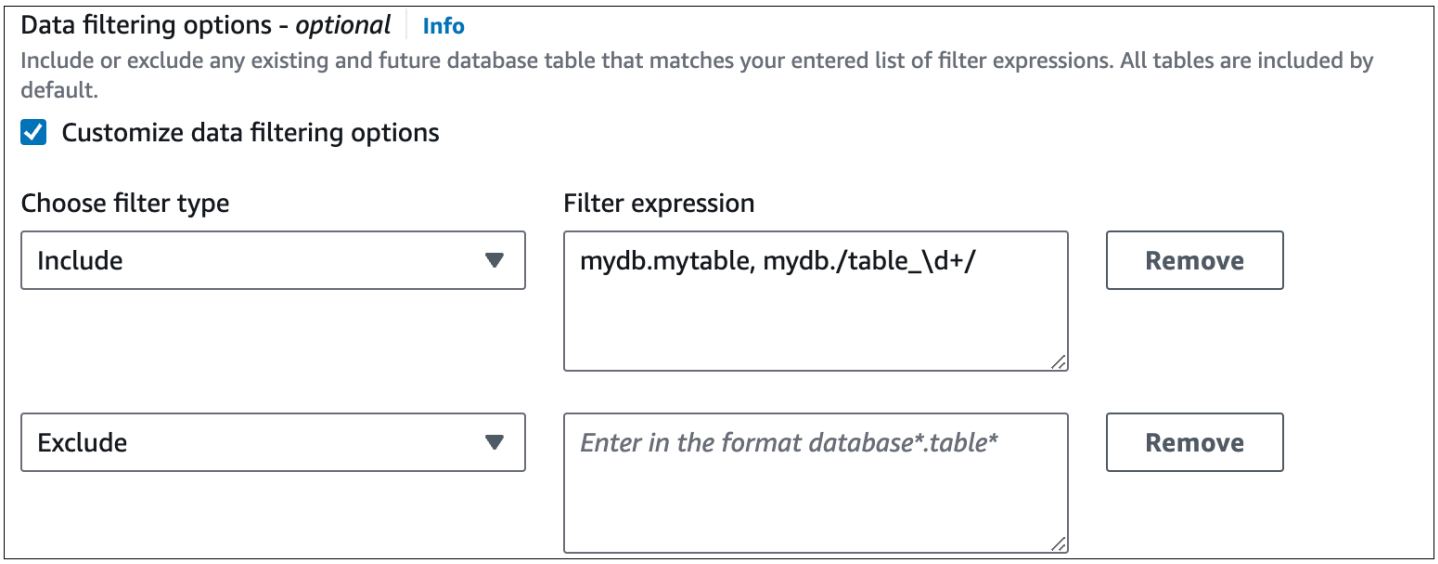
É possível definir vários filtros para uma única integração. Cada filtro inclui ou exclui qualquer tabela de banco de dados existente e futura que corresponda a um dos padrões na expressão do filtro. As integrações ETL zero do Amazon RDS usam a sintaxe de filtro Maxwell para a filtragem de dados.

Cada filtro tem os seguintes elementos:

Elemento	Descrição
Tipo de filtro	Um tipo de filtro `Include` inclui todas as tabelas que correspondem a um dos padrões na expressão do filtro. Um tipo de filtro `Exclude` exclui todas as tabelas que correspondem a um dos padrões.
Expressão de filtro	Uma lista separada por vírgulas de padrões. As expressões devem usar a sintaxe de filtro Maxwell.
Padrão	Um padrão de filtro no formato *`database`*`.`*`table`*. É possível especific

Elemento	Descrição
	ar nomes literais ou definir expressões regulares. Não é possível incluir filtros ao nível da coluna nem listas de bloqueio. Uma única integração pode ter um máximo de 99 padrões no total. No console, é possível inserir padrões em uma única expressão de filtro ou distribuí-los entre várias expressões. Um único padrão não pode exceder 256 caracteres.

A imagem a seguir mostra a estrutura dos filtros de dados do no console:



> **Important**
>
> Não inclua informações pessoais, confidenciais ou sigilosas em seus padrões de filtragem.

Filtros de dados na AWS CLI

Ao usar a AWS CLI para adicionar um filtro de dados, a sintaxe difere um pouco da sintaxe do console. Você deve atribuir um tipo de filtro (`Include` ou `Exclude`) a cada padrão individualmente; portanto, não é possível agrupar vários padrões sob um tipo de filtro.

Por exemplo, no console, é possível agrupar os seguintes padrões separados por vírgula em uma única instrução `Include`:

```
mydb.mytable, mydb./table_\d+/
```

No entanto, ao usar a AWS CLI, o mesmo filtro de dados deve estar no seguinte formato:

```
'include: mydb.mytable, include: mydb./table_\d+/'
```

Lógica de filtros

Se você não especificar nenhum filtro de dados na integração, o Amazon RDS assumirá o filtro padrão `include:*.*`, o que replica todas as tabelas no data warehouse de destino. Entretanto, se você adicionar pelo menos um filtro, a lógica padrão mudará para `exclude:*.*`, o que exclui todas as tabelas por padrão. Isso permite que você defina explicitamente quais bancos de dados e tabelas incluir na replicação.

Por exemplo, se você definir o seguinte filtro:

```
'include: db.table1, include: db.table2'
```

O Amazon RDS avaliará o filtro da seguinte forma:

```
'exclude:*.*, include: db.table1, include: db.table2'
```

Portanto, o Amazon RDS somente replica a `table1` e a `table2` do banco de dados denominado `db` para o data warehouse de destino.

Precedência do filtro

O Amazon RDS avalia os filtros de dados na ordem em que você especificar. No AWS Management Console, ele processa expressões de filtro da esquerda para a direita e de cima para baixo. Um segundo filtro ou um padrão individual subsequente pode substituir o primeiro.

Por exemplo, se o primeiro filtro for `Include books.stephenking`, ele incluirá somente a tabela `stephenking` do banco de dados `books`. No entanto, se você adicionar um segundo filtro, `Exclude books.*`, ele substituirá o primeiro filtro. Isso impede que qualquer tabela do índice `books` seja replicada para o data warehouse de destino.

Quando você especifica pelo menos um filtro, a lógica primeiro assume `exclude:*.*`, o que significa que todas as tabelas são automaticamente excluídas da replicação. Como prática recomendada, defina filtros do mais amplo ao mais específico. Comece com uma ou mais instruções `Include` para especificar os dados a serem replicados e, em seguida, adicione filtros `Exclude` para remover seletivamente determinadas tabelas.

O mesmo princípio se aplica aos filtros que você define usando a AWS CLI. O Amazon RDS avalia esses padrões de filtro na ordem em que você os especifica; portanto, um padrão pode substituir um especificado antes dele.

Exemplos

Os exemplos a seguir demonstram como a filtragem de dados funciona para integrações ETL zero do :

- Inclua todos os bancos de dados e todas as tabelas:

```
'include: *.*'
```

- Inclua todas as tabelas no banco de dados `books`:

```
'include: books.*'
```

- Exclua todas as tabelas chamadas `mystery`:

```
'include: *.*, exclude: *.mystery'
```

- Inclua duas tabelas específicas no banco de dados `books`:

```
'include: books.stephen_king, include: books.carolyn_keene'
```

- Inclua todas as tabelas no banco de dados `books`, exceto aquelas que contêm a substring `mystery`:

```
'include: books.*, exclude: books./.*mystery.*/'
```

- Inclua todas as tabelas no banco de dados `books`, exceto aquelas que começam com `mystery`:

```
'include: books.*, exclude: books./mystery.*/'
```

- Inclua todas as tabelas no banco de dados `books`, exceto aquelas que terminam com `mystery`:

```
'include: books.*, exclude: books./.*mystery/'
```

- Inclua todas as tabelas no banco de dados `books` que começam com `table_`, exceto aquela chamada `table_stephen_king`. Por exemplo, `table_movies` ou `table_books` seria replicada, mas não `table_stephen_king`.

```
'include: books./table_.*/, exclude: books.table_stephen_king'
```

Adicionar filtros de dados a uma integração

É possível configurar a filtragem de dados usando o AWS Management Console, a AWS CLI ou a API do Amazon RDS.

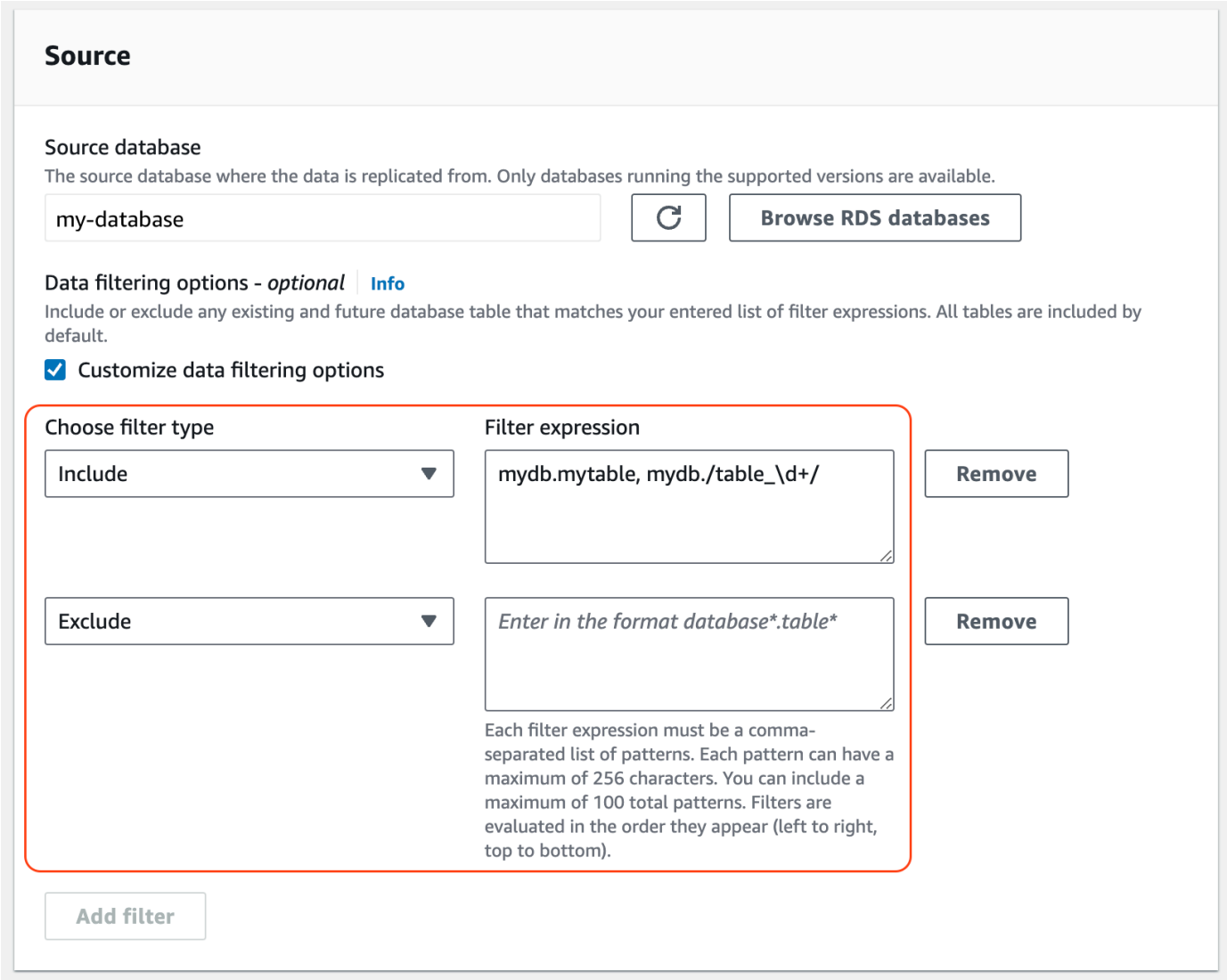
> **Important**
>
> Se você adicionar um filtro depois de criar uma integração, o Amazon RDS o tratará como se ele sempre tivesse existido. Ele remove quaisquer dados no data warehouse de destino que não correspondam aos novos critérios de filtragem e ressincroniza todas as tabelas afetadas.

Console do RDS

Para adicionar filtros de dados para uma integração ETL zero

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Integrações ETL zero. Selecione a integração à qual você deseja adicionar filtros de dados e escolha Modificar.
3. Em Fonte, adicione uma ou mais instruções `Exclude` e `Include`.

   A seguinte imagem mostra um exemplo de filtros de dados para uma integração do MySQL:



4. Quando as alterações estiverem adequadas para você, escolha Continuar e Salvar alterações.

AWS CLI

Para adicionar filtros de dados a uma integração ETL zero usando a AWS CLI, chame o comando modify-integration. Além do identificador de integração, especifique o parâmetro `--data-filter` com uma lista separada por vírgulas dos filtros Maxwell `Include` e `Exclude`.

Example

O exemplo a seguir adiciona padrões de filtro a `my-integration`.

Para Linux, macOS ou Unix:

```
aws rds modify-integration \
```

```
    --integration-identifier my-integration \
    --data-filter 'include: foodb.*, exclude: foodb.tbl, exclude: foodb./table_\d+/'
```

Para Windows:

```
aws rds modify-integration ^
    --integration-identifier my-integration ^
    --data-filter 'include: foodb.*, exclude: foodb.tbl, exclude: foodb./table_\d+/'
```

API do RDS

Para modificar uma integração ETL zero usando a API do RDS, chame a operação ModifyIntegration. Especifica o identificador de integração e fornece uma lista separada por vírgulas de padrões de filtro.

Remover filtros de dados de uma integração

Quando você remove um filtro de dados de uma integração, o Amazon RDS reavalia os filtros restantes como se o filtro removido nunca tivesse existido. Em seguida, ele replica quaisquer dados anteriormente excluídos que agora atendem aos critérios no data warehouse de destino. Isso aciona uma ressincronização de todas as tabelas afetadas.

Adicionar dados a um banco de dados do RDS de origem e consultá-los no Amazon Redshift

> Esta é a documentação de pré-lançamento das Integrações ETL zero do Amazon RDS com o Amazon Redshift, que está em versão de pré-visualização. A documentação e o atributo estão sujeitos a alterações. Recomendamos o uso desse atributo somente em ambientes de teste, e não em ambientes de produção. Para conferir os termos e condições da pré-visualização, consulte Betas e pré-visualizações nos Termos de serviços da AWS.

Para criar uma integração ETL zero que replica dados do Amazon RDS no Amazon Redshift, você deve criar um banco de dados de destino no Amazon Redshift.

Primeiro, conecte-se ao cluster ou grupo de trabalho do Amazon Redshift e crie um banco de dados com uma referência ao identificador de integração. Depois, é possível adicionar dados ao banco de dados do RDS de origem e vê-los replicados no Amazon Redshift.

Tópicos

- Criação de um banco de dados de destino no Amazon Redshift
- Adicionar dados ao banco de dados de origem
- Consultar os dados do Amazon RDS no Amazon Redshift
- Diferenças de tipos de dados entre os bancos de dados RDS e Amazon Redshift

Criação de um banco de dados de destino no Amazon Redshift

Antes de começar a replicar dados no Amazon Redshift, após você criar uma integração, você deve criar um banco de dados de destino no data warehouse de destino. Esse banco de dados de destino deve incluir uma referência ao identificador de integração. Você pode usar o console do Amazon Redshift ou o Editor de Consultas v2 para criar o banco de dados.

Para obter instruções sobre como criar um banco de dados de destino, consulte Criar um banco de dados de destino no Amazon Redshift.

Adicionar dados ao banco de dados de origem

Depois de configurar a integração, será possível adicionar alguns dados ao banco de dados do RDS que você deseja replicar no data warehouse do Amazon Redshift.

> **Note**
>
> Há diferenças entre os tipos de dados no Amazon RDS e Amazon Redshift. Para obter uma tabela de mapeamentos de tipos de dados, consulte. the section called “Diferenças dos tipos de dados”

Primeiro, conecte-se ao banco de dados de origem usando o cliente do MySQL de sua escolha. Para obter instruções, consulte the section called “Conexão a uma instância de banco de dados executando MySQL”.

Depois, crie uma tabela e insira uma linha de dados de exemplo.

> **Important**
>
> Verifique se a tabela tem uma chave primária. Caso contrário, ela não poderá ser replicada no data warehouse de destino.

O exemplo a seguir usa o utilitário MySQL Workbench.

```
CREATE DATABASE my_db;

USE my_db;

CREATE TABLE books_table (ID int NOT NULL, Title VARCHAR(50) NOT NULL, Author
 VARCHAR(50) NOT NULL,
Copyright INT NOT NULL, Genre VARCHAR(50) NOT NULL, PRIMARY KEY (ID));

INSERT INTO books_table VALUES (1, 'The Shining', 'Stephen King', 1977, 'Supernatural
 fiction');
```

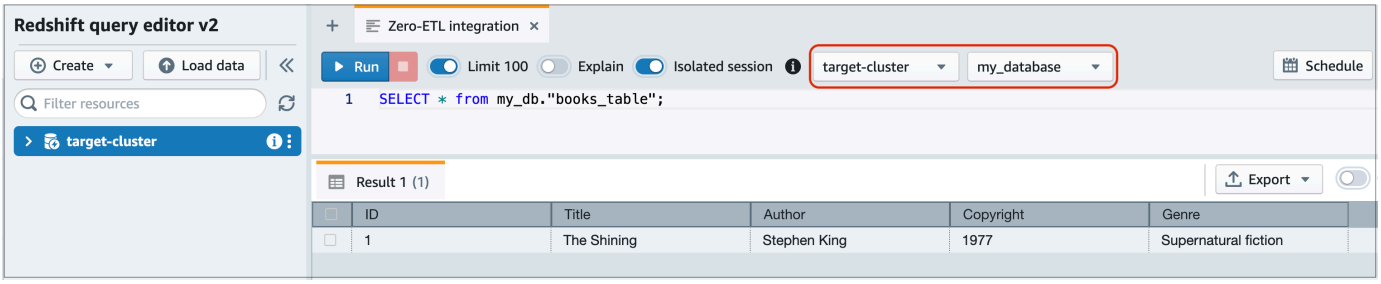
Consultar os dados do Amazon RDS no Amazon Redshift

Depois de adicionar dados ao banco de dados do RDS, eles são replicados no Amazon Redshift e ficam prontos para serem consultados.

Como consultar os dados replicados

1. Navegue até o console do Amazon Redshift e escolha Editor de Consultas v2 no painel de navegação esquerdo.
2. Conecte-se ao cluster ou grupo de trabalho e escolha o banco de dados de destino (criado na integração) no menu suspenso (destination_database neste exemplo). Para obter instruções sobre como criar um banco de dados de destino, consulte Criar um banco de dados de destino no Amazon Redshift.
3. Use uma declaração SELECT para consultar dados. Neste exemplo, é possível executar o comando a seguir para selecionar todos os dados da tabela que você criou no banco de dados do RDS de origem:

   ```
   SELECT * from my_db."books_table";
   ```



- *my_db* é o nome do esquema do banco de dados RDS.
- *books_table* é o nome da tabela do RDS.

Também é possível consultar os dados usando o cliente de linha de comandos. Por exemplo:

```
destination_database=# select * from my_db."books_table";

 ID |       Title |        Author |   Copyright |                 Genre |  txn_seq |
 txn_id
----+-------------+---------------+-------------+-----------------------+----------
+--------+
  1 | The Shining |  Stephen King |        1977 |   Supernatural fiction |        2 |
 12192
```

> **Note**
>
> Para diferenciar letras maiúsculas de minúsculas, use aspas duplas (“ ”) para nomes de esquemas, tabelas e colunas. Para obter mais informações, consulte enable_case_sensitive_identifier.

Diferenças de tipos de dados entre os bancos de dados RDS e Amazon Redshift

A tabela a seguir mostra o mapeamento de um tipo de dados do RDS para MySQL para um tipo de dados correspondente do Amazon Redshift. No momento, o Amazon RDS é compatível somente com esses tipos de dados para integrações ETL zero.

Se uma tabela no banco de dados de origem incluir um tipo de dado incompatível, a tabela ficará fora de sincronia e não poderá ser consumida pelo destino do Amazon Redshift. O streaming

da origem para o destino continuará, mas a tabela com o tipo de dados não compatível não estará disponível. Para corrigir a tabela e disponibilizá-la no Amazon Redshift, você deve reverter manualmente a alteração significativa e, em seguida, atualizar a integração executando `ALTER DATABASE...INTEGRATION REFRESH`.

RDS para MySQL

Tipo de dados do RDS para MySQL	Tipo de dados do Amazon Redshift	Descrição	Limitações
INT	INTEGER	Número inteiro de quatro bytes assinado	
SMALLINT	SMALLINT	Número inteiro de dois bytes assinado	
TINYINT	SMALLINT	Número inteiro de dois bytes assinado	
MEDIUMINT	INTEGER	Número inteiro de quatro bytes assinado	
BIGINT	BIGINT	Número inteiro de oito bytes assinado	
INT UNSIGNED	BIGINT	Número inteiro de oito bytes assinado	
TINYINT UNSIGNED	SMALLINT	Número inteiro de dois bytes assinado	

Tipo de dados do RDS para MySQL	Tipo de dados do Amazon Redshift	Descrição	Limitações
MEDIUMINT UNSIGNED	INTEGER	Número inteiro de quatro bytes assinado	
BIGINT UNSIGNED	DECIMAL(20,0)	Numérico exato com precisão selecionável	
DECIMAL(p,s) = NUMERIC(p,s)	DECIMAL (p,s)	Numérico exato com precisão selecionável	Precisão maior que 38 e escala maior que 37 não são aceitas
DECIMAL(p,s) UNSIGNED = NUMERIC(p,s) UNSIGNED	DECIMAL (p,s)	Numérico exato com precisão selecionável	Precisão maior que 38 e escala maior que 37 não são aceitas
FLOAT4/REAL	REAL	Número de ponto flutuante de precisão simples	
FLOAT4/REAL UNSIGNED	REAL	Número de ponto flutuante de precisão simples	
DOUBLE/REAL/FLOAT8	DOUBLE PRECISION	Número de ponto flutuante de precisão dupla	

Tipo de dados do RDS para MySQL	Tipo de dados do Amazon Redshift	Descrição	Limitações
DOUBLE/REAL/FLOAT8 UNSIGNED	DOUBLE PRECISION	Número de ponto flutuante de precisão dupla	
BIT (n)	VARBYTE(8)	Valor binário de comprimento variável	
BINARY(n)	VARBYTE(n)	Valor binário de comprimento variável	
VARBINARY (n)	VARBYTE(n)	Valor binário de comprimento variável	
CHAR(n)	VARCHAR(n)	Valor de string de comprimento variável	
VARCHAR(n)	VARCHAR(n)	Valor de string de comprimento variável	
TEXT	VARCHAR(65535)	Valor de string de comprimento variável de até 65535 bytes	
TINYTEXT	VARCHAR(255)	Valor de string de comprimento variável de até 255 bytes	

Tipo de dados do RDS para MySQL	Tipo de dados do Amazon Redshift	Descrição	Limitações
ENUM	VARCHAR(1020)	Valor de string de comprimento variável de até 1020 bytes	
SET	VARCHAR(1020)	Valor de string de comprimento variável de até 1020 bytes	
DATA	DATA	Data de calendário (ano, mês, dia)	
DATETIME	TIMESTAMP	Data e hora (sem fuso horário)	
TIMESTAMP(p)	TIMESTAMP	Data e hora (sem fuso horário)	
TIME	VARCHAR(18)	Valor de string de comprimento variável de até 18 bytes	
YEAR	VARCHAR(4)	Valor de string de comprimento variável de até 4 bytes	

Tipo de dados do RDS para MySQL	Tipo de dados do Amazon Redshift	Descrição	Limitações
JSON	SUPER	Dados ou documentos semiestruturados como valores	

Visualizar e monitorar Integrações ETL zero do Amazon RDS

Você pode ver os detalhes de uma integração ETL zero do Amazon RDS para ver suas informações de configuração e status atual. Também é possível monitorar o status da integração consultando visualizações específicas do sistema no Amazon Redshift. Além disso, o Amazon Redshift publica determinadas métricas relacionadas à integração no Amazon CloudWatch, que você pode visualizar no console do Amazon Redshift.

Tópicos

- Visualizar integrações
- Monitorar integrações usando tabelas de sistema para o Amazon Redshift
- Monitorar integrações com o Amazon EventBridge para o Amazon Redshift

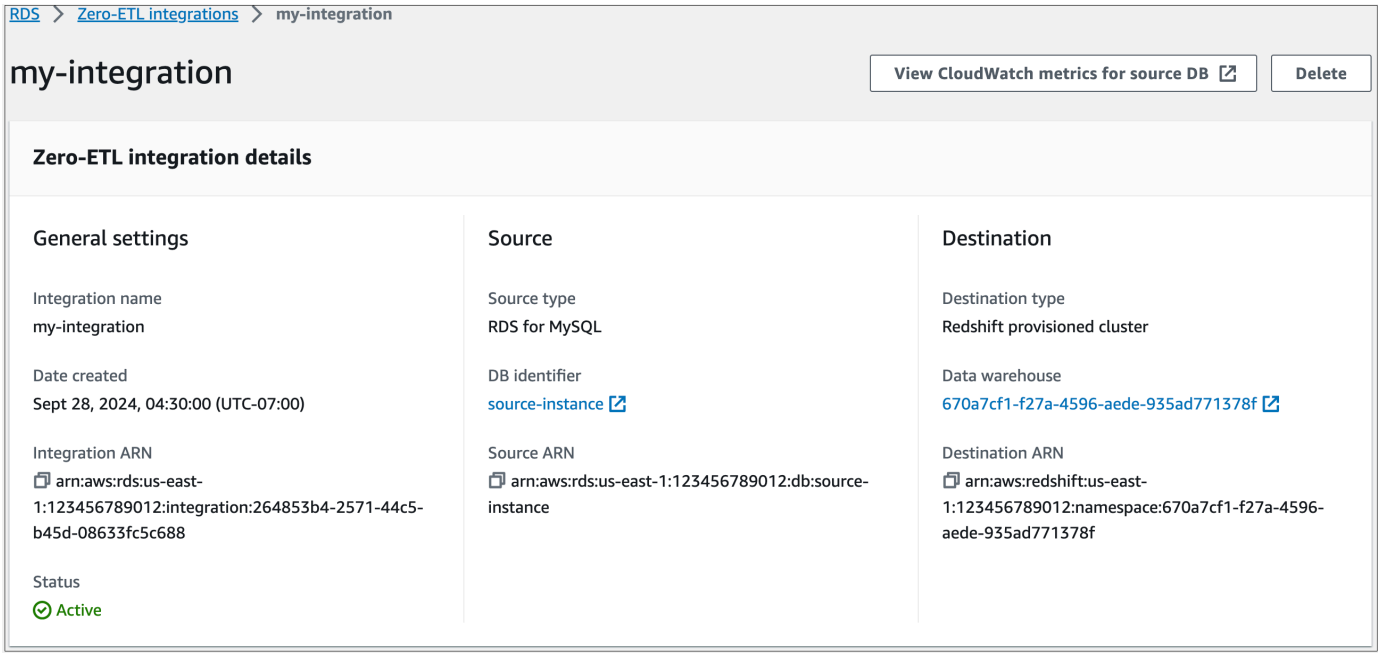
Visualizar integrações

Você pode visualizar Integrações ETL zero do Amazon RDS usando o AWS Management Console, a AWS CLI ou a API do RDS.

Console

Como visualizar os detalhes de uma integração ETL zero

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação à esquerda, escolha Integrações ETL zero.
3. Selecione uma integração para ver mais detalhes sobre ela, como o banco de dados de origem e o data warehouse de destino.



Uma integração pode ter os seguintes status:

- `Creating`: a integração está sendo criada.
- `Active`: a integração está enviando dados transacionais para o data warehouse de destino.
- `Syncing`: a integração encontrou um erro recuperável e precisa reprocessar os dados. As tabelas afetadas não estarão disponíveis para consulta enquanto a ressincronização não for concluída.
- `Needs attention`: a integração encontrou um evento ou erro que requer intervenção manual para resolvê-lo. Para corrigir o problema, siga as instruções na mensagem de erro na página de detalhes da integração.
- `Failed`: a integração encontrou um evento ou erro irrecuperável que não pode ser corrigido. Você precisa excluir e recriar a integração.
- `Deleting`: a integração está sendo excluída.

AWS CLI

Para visualizar todas as integrações ETL zero na conta atual usando a AWS CLI, use o comando describe-integrations e especifique a opção `--integration-identifier`.

Example

Para Linux, macOS ou Unix:

```
aws rds describe-integrations \
    --integration-identifier ee605691-6c47-48e8-8622-83f99b1af374
```

Para Windows:

```
aws rds describe-integrations ^
    --integration-identifier ee605691-6c47-48e8-8622-83f99b1af374
```

API do RDS

Para visualizar a integração ETL zero usando a API do Amazon RDS, use a operação `DescribeIntegrations` com o parâmetro `IntegrationIdentifier`.

Monitorar integrações usando tabelas de sistema para o Amazon Redshift

O Amazon Redshift tem visualizações e tabelas do sistema que contêm informações sobre como o sistema está funcionando. Você pode consultar essas visualizações e tabelas do sistema da mesma maneira como consultaria qualquer outra tabela de banco de dados. Para ter mais informações sobre visualizações e tabelas do sistema no Amazon Redshift, consulte Referência de visualizações e tabelas do sistema no Guia do desenvolvedor de banco de dados do Amazon Redshift.

Você pode consultar as seguintes visualizações e tabelas de sistema para obter informações sobre Integrações ETL zero do :

- SVV_INTEGRATION: fornece detalhes de configuração para suas integrações.
- SVV_INTEGRATION_TABLE_STATE: descreve o estado de cada tabela em uma integração.
- SYS_INTEGRATION_TABLE_STATE_CHANGE: exibe os logs de alteração do estado de uma integração.
- SYS_INTEGRATION_ACTIVITY: fornece informações sobre execuções de integração concluídas.

Todas as métricas do Amazon CloudWatch relacionadas à integração se originam no Amazon Redshift. Para ter mais informações, consulte Métricas para integrações ETL zero no Guia de gerenciamento do Amazon Redshift. No momento, o Amazon RDS não publica nenhuma métrica relacionada à integração no Amazon CloudWatch.

Monitorar integrações com o Amazon EventBridge para o Amazon Redshift

O Amazon Redshift envia eventos relacionados à integração ao Amazon EventBridge. Para obter uma lista de eventos e seus IDs de eventos correspondentes, consulte Notificações de eventos de integração ETL zero com o Amazon EventBridge no Guia de gerenciamento do Amazon Redshift.

Modificar Integrações ETL zero do Amazon RDS

É possível modificar somente o nome, a descrição e as opções de filtragem de dados de uma Integração ETL zero em um data warehouse compatível. Não é possível modificar a chave do AWS KMS usada para criptografar a integração ou os bancos de dados de origem ou de destino.

Se você adicionar um filtro de dados a uma integração existente, o Amazon RDS reavaliará o filtro como se ele sempre tivesse existido. Ele remove todos os dados que no momento estão no data warehouse de destino e não correspondem aos novos critérios de filtragem. Se você remover um filtro de dados de uma integração, ele replicará todos os dados que anteriormente não correspondiam aos critérios de filtragem (mas agora correspondem) no data warehouse de destino. Para obter mais informações, consulte the section called “Filtragem de dados para integrações ETL zero”.

É possível modificar uma integração ETL zero usando o AWS Management Console, a AWS CLI ou a API do Amazon RDS.

Console do RDS

Para modificar uma integração ETL zero

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Integrações ETL zero e a integração que você deseja modificar.
3. Escolha Modificar e altere as configurações disponíveis.
4. Quando todas as alterações estiverem conforme desejado, escolha Modificar.

AWS CLI

Para modificar uma integração ETL zero usando a AWS CLI, chame o comando modify-integration. Junto com o `--integration-identifier`, especifique qualquer uma das seguintes opções:

- `--integration-name`: especifique um novo nome para a integração.
- `--description`: especifique uma nova descrição para a integração.
- `--data-filter`: especifique as opções de filtragem de dados para a integração. Para obter mais informações, consulte the section called “Filtragem de dados para integrações ETL zero”.

Example

A solicitação a seguir modifica uma integração existente.

Para Linux, macOS ou Unix:

```
aws rds modify-integration \
    --integration-identifier ee605691-6c47-48e8-8622-83f99b1af374 \
    --integration-name my-renamed-integration
```

Para Windows:

```
aws rds modify-integration ^
    --integration-identifier ee605691-6c47-48e8-8622-83f99b1af374 ^
    --integration-name my-renamed-integration
```

API do RDS

Para modificar uma integração ETL zero usando a API do RDS, chame a operação ModifyIntegration. Especifique o identificador de integração e os parâmetros que você deseja modificar.

Excluir Integrações ETL zero do Amazon RDS

Quando você exclui uma integração ETL zero, o Amazon RDS a remove do banco de dados de origem. Os dados transacionais não são excluídos do Amazon RDS ou do destino de analytics, mas o Amazon RDS não envia dados novos ao Amazon Redshift ou ao Amazon SageMaker.

Só será possível excluir uma integração quando ela tiver o status `Active`, `Failed`, `Syncing` ou `Needs attention`.

É possível excluir as integrações ETL zero usando o AWS Management Console, a AWS CLI ou a API do RDS.

Console

Como excluir uma integração ETL zero

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação à esquerda, escolha Integrações ETL zero.
3. Selecione a integração ETL zero que você deseja excluir.
4. Escolha Ações e Excluir, depois confirme a exclusão.

AWS CLI

Para excluir uma integração ETL zero, use o comando delete-integration e especifique a opção `--integration-identifier`.

Example

Para Linux, macOS ou Unix:

```
aws rds delete-integration \
    --integration-identifier ee605691-6c47-48e8-8622-83f99b1af374
```

Para Windows:

```
aws rds delete-integration ^
    --integration-identifier ee605691-6c47-48e8-8622-83f99b1af374
```

API do RDS

Para excluir uma integração ETL zero usando a API do Amazon RDS, use a operação `DeleteIntegration` com o parâmetro `IntegrationIdentifier`.

Solução de problemas em integrações ETL zero do Amazon RDS com o Amazon Redshift

Esta é a documentação de pré-lançamento das Integrações ETL zero do Amazon RDS com o Amazon Redshift, que está em versão de pré-visualização. A documentação e o atributo estão

sujeitos a alterações. Recomendamos o uso desse atributo somente em ambientes de teste, e não em ambientes de produção. Para conferir os termos e condições da pré-visualização, consulte Betas e pré-visualizações nos Termos de serviços da AWS.

Você pode verificar o estado de uma integração ETL zero consultando a tabela do sistema SVV_INTEGRATION no Amazon Redshift. Se a coluna `state` tiver um valor de `ErrorState`, isso significa que há algo errado. Para ter mais informações, consulte the section called “Monitorar usando tabelas do sistema”.

Use as informações a seguir para solucionar problemas comuns em integrações ETL zero do Amazon RDS com o Amazon Redshift.

Tópicos

- Não consigo criar uma integração ETL zero.
- Minha integração está travada em um estado de Syncing.
- Minhas tabelas não estão sendo replicadas para o Amazon Redshift.
- Uma ou mais das minhas tabelas do Amazon Redshift exigem ressincronização.

Não consigo criar uma integração ETL zero.

Se você não consegue criar uma integração ETL zero, verifique se as informações a seguir estão corretas para sua instância de banco de dados de origem:

- O banco de dados de origem está executando o RDS para MySQL versão 8.0.32 ou posterior. Para confirmar a versão do mecanismo, escolha a guia Configuração do banco de dados e confira a versão do mecanismo.
- Você configurou corretamente os parâmetros do de banco de dados. Se os parâmetros necessários forem definidos incorretamente ou não estiverem associados à instância de banco de dados, a criação falhará. Consulte the section called “Etapa 1: Criar um grupo de parâmetros de de banco de dados personalizado”.

Além disso, confira se o indicado a seguir está correto para seu data warehouse de destino:

- Ter a diferenciação entre maiúsculas e minúsculas habilitada. Consulte Ativar a diferenciação entre maiúsculas e minúsculas em um data warehouse.

- Você adicionou a entidade principal autorizada e a origem de integração corretas. Consulte Configurar a autorização para o data warehouse do Amazon Redshift.
- O data warehouse é criptografado (se for um cluster provisionado). Consulte Criptografia de banco de dados do Amazon Redshift.

Minha integração está travada em um estado de **`Syncing`**.

Sua integração poderá mostrar consistentemente um status de `Syncing` se você alterar o valor de um dos parâmetros obrigatórios do banco de dados.

Para corrigir esse problema, confira os valores dos parâmetros no grupo de parâmetros associado ao banco de dados de origem e verifique se eles correspondem aos valores obrigatórios. Para ter mais informações, consulte the section called “Etapa 1: Criar um grupo de parâmetros de de banco de dados personalizado”.

Se você modificar algum parâmetro, reinicialize o banco de dados para aplicar as alterações.

Minhas tabelas não estão sendo replicadas para o Amazon Redshift.

Caso não veja uma ou mais tabelas refletidas no Amazon Redshift, execute um dos seguintes comandos para ressincronizá-las:

```
ALTER DATABASE dbname INTEGRATION REFRESH TABLES table1, table2;
```

Para ter mais informações, consulte ALTER DATABASE na referência do Amazon Redshift SQL.

Talvez seus dados não estejam sendo replicados porque uma ou mais tabelas de origem não têm uma chave primária. O painel de monitoramento no Amazon Redshift exibe o status dessas tabelas como `Failed`, e o status da integração ETL zero geral muda para `Needs attention`. Para resolver esse problema, é possível identificar uma chave em sua tabela que pode se tornar uma chave primária ou adicionar uma chave primária sintética. Para conhecer as soluções detalhadas, consulte Handle tables without primary keys while creating Amazon Aurora MySQL or Amazon RDS for MySQL zero-ETL integrations with Amazon Redshift.

Uma ou mais das minhas tabelas do Amazon Redshift exigem ressincronização.

A execução de determinados comandos na instância de banco de dados de origem pode exigir que suas tabelas sejam ressincronizadas. Nesses casos, a visualização do sistema

SVV_INTEGRATION_TABLE_STATE mostra um `table_state` de `ResyncRequired`, o que significa que a integração deve recarregar completamente os dados dessa tabela específica do MySQL para o Amazon Redshift.

Quando a tabela começa a ser ressincronizada, ela entra em um estado de `Syncing`. Você não precisa realizar nenhuma ação manual para ressincronizar uma tabela. Enquanto os dados da tabela estiverem sendo sincronizados novamente, você não poderá acessá-los no Amazon Redshift.

A seguir estão alguns exemplos de operações que podem colocar uma tabela em um estado `ResyncRequired` e possíveis alternativas a serem consideradas.

Operation	Exemplo	Alternativa
Adicionar uma coluna em uma posição específica	`ALTER TABLE table_name ADD COLUMN column_name INTEGER NOT NULL first;`	O Amazon Redshift não oferece suporte à adição de colunas em posições específicas usando as palavras-chave `first` ou `afte` Se a ordem das colunas na tabela de destino não for crítica, adicione a coluna no final da tabela usando um comando mais simples:

Operation	Exemplo	Alternativa
		`ALTER TABLE table_name ADD COLUMN column_name column_type ;`

Operation	Exemplo	Alternativa
Adicionar uma coluna de carimbo de data/hora com o CURRENT_TIMESTAMP pad	`ALTER TABLE table_name ADD COLUMN column_name TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP;`	O valor CURRENT_TIMESTAMP das linhas da tabela é calculado pelo RDS para MySQL e não pode ser simulado no Amazon Redshift sem uma nova sincronização completa dos dados da tabela. Se possível, altere o valor padrão para uma constante literal, como 2023-01-01 00:00:15, a fim de evitar latência na disponibilidade da tabela.

Operation	Exemplo	Alternativa

Executar várias operações de coluna em um único comando	`ALTER TABLE table_name`
	`  ADD COLUMN column_1,`
	`  RENAME COLUMN column_2 TO column_3;`
		Considere dividir o comando em duas operações separadas, `ADD` e `RENAME`, que não exigirão ressincronização.

Amazon RDS para Db2

O Amazon RDS é compatível com as instâncias de banco de dados que executam as seguintes edições do IBM Db2:

- Db2 Standard Edition
- Db2 Advanced Edition

O Amazon RDS é compatível com instâncias de banco de dados que executam as seguintes versões do Db2:

- Db2 11.5

Para obter mais informações sobre o suporte a versões secundárias, consulte Db2 em versões do Amazon RDS.

Antes de criar uma instância de banco de dados, realize as etapas na seção Configuração do ambiente do Amazon RDS deste guia do usuário. Ao criar uma instância de banco de dados usando o usuário principal, o usuário recebe a autoridade `DBADM`, com algumas limitações. Use esse usuário para tarefas administrativas, como a criação de contas de banco de dados adicionais. Não é possível usar a autoridade em nível de instância `SYSADM`, `SYSCTRL`, `SYSMAINT` nem a autoridade em nível de banco de dados `SECADM`.

Você pode criar o seguinte:

- Instâncias de banco de dados
- DB snapshots
- Restaurações point-in-time
- Backups de armazenamento automatizados
- Backups de armazenamento manuais

É possível usar instâncias de banco de dados executando o Db2 em uma nuvem privada virtual (VPC). Também é possível adicionar recursos à instância de banco de dados do Amazon RDS para Db2 habilitando várias opções. O Amazon RDS comporta implantações multi-AZ para RDS para Db2 como uma solução de failover de alta disponibilidade.

> **Important**
>
> Para oferecer uma experiência de serviço gerenciada, o Amazon RDS não fornece acesso ao shell para as instâncias de banco de dados. Ele também restringe o acesso a determinados procedimentos e tabelas do sistema que necessitam de privilégios avançados. É possível acessar o banco de dados usando clientes SQL padrão, como o IBM Db2 CLP. No entanto, não é possível acessar o host diretamente usando Telnet ou Secure Shell (SSH).

Tópicos

- Visão geral do Db2 no Amazon RDS
- Pré-requisitos para criar uma instância de banco de dados do Amazon RDS para Db2
- Vários bancos de dados em uma instância de banco de dados do Amazon RDS para Db2
- Conectar-se à instância de banco de dados do Db2
- Proteger as conexões de instância de banco de dados do Amazon RDS para Db2
- Administrar a instância de banco de dados do Amazon RDS para Db2
- Integrar uma instância de banco de dados do Amazon RDS para Db2 ao Amazon S3
- Migrar dados para o Amazon RDS para Db2
- Federação do Amazon RDS para Db2
- Trabalhar com réplicas para o Amazon RDS para Db2
- Opções para instâncias de banco de dados do Amazon RDS para Db2
- Procedimentos armazenados externos do Amazon RDS para Db2
- Limitações e problemas conhecidos do Amazon RDS para Db2
- Referência de procedimentos armazenados do Amazon RDS para Db2
- Referência de função definida pelo usuário do Amazon RDS para Db2
- Solução de problemas do Amazon RDS para Db2

Visão geral do Db2 no Amazon RDS

É possível ler as seções a seguir para ter uma visão geral do Db2 no Amazon RDS.

Tópicos

- Recursos do Amazon RDS para Db2

- Db2 em versões do Amazon RDS
- Opções de licenciamento do Amazon RDS para Db2
- Classes de instância do Amazon RDS para Db2
- Perfis padrão do Amazon RDS para Db2
- Parâmetros do Amazon RDS para Db2
- Agrupamento EBCDIC para bancos de dados do Db2 no Amazon RDS
- Fuso horário local para instâncias de banco de dados do Amazon RDS para Db2

Recursos do Amazon RDS para Db2

O Amazon RDS for Db2 comporta a maioria dos recursos e das funcionalidades do banco de dados do IBM Db2. Alguns recursos podem ter suporte limitado ou privilégios restritos. Para ter mais informações sobre os recursos do banco de dados do Db2 para versões específicas do Db2, consulte a documentação do IBM Db2.

Você pode filtrar novos recursos do Amazon RDS na página What's New with Database? (Novidades sobre bancos de dados). Em Products (Produtos), escolha Amazon RDS. Depois, é possível pesquisar usando palavras-chave como **`Db2 2023`**.

> **Note**
>
> As listas a seguir não são completas.

Tópicos

- Recursos aceitos no RDS para Db2
- Recursos não aceitos no RDS para Db2

Recursos aceitos no RDS para Db2

O RDS para Db2 é compatível com recursos que incluem os nativos do IBM Db2 e os essenciais do Amazon RDS.

Recursos nativos do IBM Db2

O RDS para Db2 é compatível com os seguintes recursos do banco de dados do Db2:

- Criação de um banco de dados padrão que usa um conjunto de códigos, agrupamento, tamanho da página e território definidos pelo cliente. Use o procedimento armazenado rdsadmin.create_database do Amazon RDS.
- Adição, exclusão ou modificação de usuários e grupos locais. Use os procedimentos armazenados do Amazon RDS para Conceder e revogar privilégios.
- Criação de perfis com o procedimento armazenado rdsadmin.create_role do Amazon RDS.
- Compatibilidade com tabelas organizadas em linhas padrão.
- Compatibilidade com workload analítica para tabelas organizadas em colunas.
- Capacidade de definir recursos de compatibilidade com o Db2, como Oracle e MySQL.
- Support para procedimentos armazenados externos baseados no Java.
- Compatibilidade com a criptografia de dados em trânsito usando SSL/TLS.
- Monitorar o status de um banco de dados (`ALIVE`, `DOWN`, `STORAGE_FULL`, `UNKNOWN` e `STANDBY_CONNECTABLE`).
- Restauração de um banco de dados Linux (LE) on-line ou off-line fornecido pelo cliente. Use os procedimentos armazenados do Amazon RDS para Gerenciar bancos de dados.
- Aplicação de logs de arquivamento do Db2 fornecidos pelo cliente para manter o banco de dados sincronizado com os bancos de dados autogerenciados do Db2. Use os procedimentos armazenados do Amazon RDS para Gerenciar bancos de dados.
- Compatibilidade com auditoria no nível do banco de dados e da instância do Db2.
- Compatibilidade com federação homogênea.
- Capacidade de carregar uma tabela de arquivos de dados no Amazon Simple Storage Service (Amazon S3).
- Autorizações concedidas a usuários, grupos ou perfis, como `CONNECT`, `SYSMON`, `ACCESSCTRL`, `DATAACCESS`, `SQLADM`, `WLMADM`, `EXPLAIN`, `LOAD` ou `IMPLICIT_SCHEMA`.
- Criação de vários bancos de dados.

> **Note**
>
> Uma instância de banco de dados do RDS para Db2 pode conter até 50 bancos de dados. Para obter mais informações, consulte the section called “Vários bancos de dados do Db2”.

Características principais do Amazon RDS

O RDS para Db2 é compatível com os seguintes recursos principais do Amazon RDS:

- Grupos de parâmetros personalizados para atribuir às instâncias de banco de dados
- Criação, modificação e exclusão de instâncias de banco de dados
- Restauração de um backup de banco de dados do Linux (LE) off-line ou on-line do Db2 autogerenciado
- Compatibilidade com os tipos de armazenamento gp3, io2 e io1
- Uso de AWS Managed Microsoft AD para autenticação Kerberos e autorização de grupos LDAP para o RDS para Db2
- Modificação de grupos de segurança, portas, tipos de instância, armazenamento, períodos de retenção de backup e outras configurações para instâncias do Db2 existentes
- Proteção contra exclusão de instâncias de banco de dados
- Recuperação para um ponto no tempo (PITR) entre regiões
- Uso de AWS Key Management Service (AWS KMS) para criptografia de armazenamento e criptografia em repouso
- Instâncias de banco de dados multi-AZ com modo de espera para alta disponibilidade
- Reinicializações de instâncias de banco de dados
- Atualizações nas senhas mestras
- Restauração de instâncias de banco de dados para um horário específico
- Backup e restauração de instâncias de banco de dados usando backups no nível de armazenamento
- Iniciar e interromper instâncias de banco de dados
- Manutenção de instâncias de banco de dados

Recursos não aceitos no RDS para Db2

O RDS para Db2 não é compatível com os seguintes recursos do banco de dados do Db2:

- Acesso a `SYSADM`, `SECADM` e `SYSMAINT` para o usuário principal.
- Procedimentos armazenados externos escritos em C, C++ ou Cobol.
- Várias instâncias de banco de dados do Db2 em um único host.
- Plug-ins GSS-API externos para autenticação.

- Plug-ins externos de terceiros para fazer backup ou restaurar bancos de dados do Db2.
- Processamento paralelo maciço de vários nós (MPP), como IBM Db2 Warehouse.
- IBM Db2 pureScale.
- Configuração manual da Recuperação de Desastres de Alta Disponibilidade (HADR) para RDS para Db2.

> **Note**
>
> O Amazon RDS oferece suporte e gerencia o HADR para RDS para Db2 por meio de réplicas. Para obter mais informações, consulte Trabalhar com réplicas para o Amazon RDS para Db2.
> O RDS para Db2 suporta implantações multi-AZ, backups automatizados entre regiões e replicação. Para obter mais informações, consulte Implantações de instâncias de banco de dados multi-AZ para o Amazon RDS e Replicar backups automatizados para outra Região da AWS.

- Criptografia nativa do banco de dados.
- Federação heterogênea para Informix, Sybase e Teradata. Para obter mais informações, consulte the section called “Federação”.
- Recuperação para um ponto no tempo (PITR) entre regiões para backups criptografados.
- Criação de rotinas não protegidas e migração de rotinas não protegidas existentes por meio de backup e restauração de dados. Para obter mais informações, consulte Rotinas não protegidas.
- Criação de espaços de tabela de armazenamento não automático. Para obter mais informações, consulte Espaços de tabela de armazenamento não automáticos durante a migração.
- Tabelas externas.

Db2 em versões do Amazon RDS

Para o Db2, os números de versão assumem o formato major.minor.build.revision, por exemplo, 11.5.9.0.sb00000000.r1. Nossa implementação de versão corresponde à do Db2.

principal

O número da versão principal é o número inteiro e a primeira parte fracionária do número da versão, por exemplo, 11.5. Uma alteração da versão será considerada principal se o número da versão principal for alterado, por exemplo, da versão 11.5 para a 12.1.

secundária

O número da versão secundária é a terceira e a quarta partes do número da versão, por exemplo, 9.0 em 11.5.9.0. A terceira parte indica o modpack do Db2, por exemplo, 9 em 9.0. A quarta parte indica o fixpack do Db2, por exemplo, 0 em 9.0. Uma alteração de versão será considerada secundária se o modpack do Db2 ou o fixpack do Db2 forem alterados, por exemplo, passando da versão 11.5.9.0 para a 11.5.9.1 ou da 11.5.9.0 para a 11.5.10.0, exceto para fornecer atualizações da tabela do catálogo. (O Amazon RDS trata essas exceções.)

build

O número da compilação é a quinta parte do número da versão, por exemplo, sb00000000 em 11.5.9.0.sb00000000. Um número de compilação em que a parte numérica é toda zerada indica uma compilação padrão. Um número de compilação em que a parte numérica não é toda zerada indica uma compilação padrão. Um número de compilação será alterado se houver uma correção de segurança ou uma compilação especial de uma versão existente do Db2. Uma alteração no número da versão também indica que o Amazon RDS aplicou automaticamente uma nova versão secundária.

revisão

O número da revisão é a sexta parte do número da versão, por exemplo, r1 in 11.5.9.0.sb00000000.r1. Revisão é uma revisão do Amazon RDS para uma versão existente do Db2. Uma alteração no número da revisão indica que o Amazon RDS aplicou automaticamente uma nova versão secundária.

Tópicos

- Versões secundárias do Db2 compatíveis com o Amazon RDS
- Versões principais do Db2 compatíveis com o Amazon RDS

Versões secundárias do Db2 compatíveis com o Amazon RDS

No momento, a tabela a seguir mostra as versões secundárias do Db2 11.5 compatíveis com o Amazon RDS.

> **Note**
>
> As datas com apenas mês e ano são aproximadas e são atualizadas com uma data exata quando ela é conhecida.

Versão do mecanismo do Db2	Data de lançamento da IBM	Data de lançamento do RDS	
11.5.9.0	15 de novembro de 2023	27 de novembro de 2023	

É possível especificar qualquer versão atualmente compatível do Db2 ao criar uma instância de banco de dados. É possível especificar a versão principal (como o Db2 11.5) e qualquer versão secundária compatível para a versão principal especificada. Se nenhuma versão for especificada, o Amazon RDS usará uma versão compatível por padrão, geralmente a versão mais recente. Se uma versão principal for especificada, mas uma versão secundária não, o Amazon RDS usará por padrão um lançamento recente da versão principal que você especificou. Para ver uma lista de versões compatíveis, bem como os padrões para instâncias de banco de dados recém-criadas, use o comando describe-db-engine-versions da AWS Command Line Interface (AWS CLI).

Por exemplo, para listar as versões de mecanismo compatíveis do Amazon RDS para Db2, execute o comando da AWS CLI a seguir. Substitua *region* pela Região da AWS.

Para Linux, macOS ou Unix:

```
aws rds describe-db-engine-versions \
    --filters Name=engine,Values=db2-ae,db2-se \
    --query "DBEngineVersions[].{Engine:Engine, EngineVersion:EngineVersion,
 DBParameterGroupFamily:DBParameterGroupFamily}" \
    --region region
```

Para Windows:

```
aws rds describe-db-engine-versions ^
    --filters Name=engine,Values=db2-ae,db2-se ^
    --query "DBEngineVersions[].{Engine:Engine, EngineVersion:EngineVersion,
 DBParameterGroupFamily:DBParameterGroupFamily}" ^
    --region region
```

Esse comando vai produzir uma saída semelhante ao seguinte exemplo:

```
[
    {
    "Engine": "db2-ae",
```

```
    "EngineVersion": "11.5.9.0.sb00000000.r1",
    "DBParameterGroupFamily": "db2-ae-11.5"
    },
    {
    "Engine": "db2-se",
    "EngineVersion": "11.5.9.0.sb00000000.r1",
    "DBParameterGroupFamily": "db2-se-11.5"
    }
]
```

A versão padrão do Db2 pode variar de acordo com a Região da AWS. Para criar uma instância de banco de dados com uma versão secundária específica, especifique a versão secundária durante a criação da instância de banco de dados. É possível determinar a versão padrão de uma Região da AWS para `db2-ae` e de mecanismo de banco de dados `db2-se` executando o comando `describe-db-engine-versions`. O exemplo a seguir exibe a versão padrão para a `db2-ae` do Leste dos EUA (Norte da Virgínia).

Para Linux, macOS ou Unix:

```
aws rds describe-db-engine-versions \
    --default-only --engine db2-ae \
    --query "DBEngineVersions[].{Engine:Engine, EngineVersion:EngineVersion,
 DBParameterGroupFamily:DBParameterGroupFamily}" \
    --region us-east-1
```

Para Windows:

```
aws rds describe-db-engine-versions ^
    --default-only --engine db2-ae ^
    --query "DBEngineVersions[].{Engine:Engine, EngineVersion:EngineVersion,
 DBParameterGroupFamily:DBParameterGroupFamily}" ^
    --region us-east-1
```

Esse comando vai produzir uma saída semelhante ao seguinte exemplo:

```
[
    {
    "Engine": "db2-ae",
    "EngineVersion": "11.5.9.0.sb00000000.r1",
    "DBParameterGroupFamily": "db2-ae-11.5"
    }
```

```
]
```

Com o Amazon RDS, você controla quando vai atualizar a instância do Db2 para uma nova versão principal compatível com o Amazon RDS. É possível manter a compatibilidade com versões específicas do Db2, testar novas versões com a aplicação antes de implantar na produção e realizar atualizações da versão principal no melhor período para a programação.

Com a atualização automática de versão secundária habilitada, o Amazon RDS atualiza automaticamente as instâncias de banco de dados para as novas versões secundárias do Db2, pois elas são compatíveis com o Amazon RDS. Essa correção ocorre durante a janela de manutenção programada. É possível modificar uma instância de banco de dados para habilitar ou desabilitar atualizações automáticas de versões secundárias.

Com exceção das versões 11.5.9.1 e 11.5.10.0 do Db2, as atualizações automáticas para a nova versão secundária do Db2 incluem atualizações automáticas para novas compilações e revisões. Para 11.5.9.1 e 11.5.10.0, atualize manualmente as versões secundárias.

Se você cancelar os upgrades automaticamente agendados, poderá fazer upgrade manualmente para um lançamento de versão secundária com suporte seguindo o mesmo procedimento de um upgrade de versão principal. Para ter mais informações, consulte Atualizar a versão de mecanismo de uma instância de banco de dados.

Versões principais do Db2 compatíveis com o Amazon RDS

As versões principais do RDS para Db2 estarão disponíveis pelo suporte padrão pelo menos até o fim da vida útil (base) da IBM para a versão da IBM correspondente. A tabela a seguir mostra as datas que você pode usar para planejar os ciclos de teste e atualização. Se a Amazon estender o suporte para uma versão do RDS para Db2 por mais tempo do que o planejado originalmente, planejamos atualizar essa tabela para refletir a data posterior.

É possível usar as seguintes datas para planejar seus ciclos de teste e atualização.

> **Note**
>
> As datas com apenas mês e ano são aproximadas e são atualizadas com uma data exata quando ela é conhecida.
> Você pode visualizar as versões principais dos bancos de dados Db2 executando o comando describe-db-major-engine-versions da AWS CLI ou usando a operação de API DescribeDBMajorEngineVersions do RDS.

Versão principal do Db2	Data de lançamento da IBM	Data de lançamento do RDS	Fim do suporte da IBM (Standard e Advanced Edition)	Fim do suporte da IBM (estendido)	
Db2 11.5	27 de junho de 2019	27 de novembro de 2023	Ainda não anunciado pela IBM[1]	Quatro anos após o fim do suporte	

[1] Para obter as informações mais recentes, consulte Db2 product end-of-support dates na documentação do IBM Db2.

Opções de licenciamento do Amazon RDS para Db2

O Amazon RDS para Db2 tem duas opções de licenciamento: traga a sua própria licença (BYOL) e licença do Db2 por meio do AWS Marketplace.

Tópicos

- Traga a sua própria licença para Db2
- Licença do Db2 por meio do AWS Marketplace
- Alternar entre licenças do Db2

Traga a sua própria licença para Db2

No modelo BYOL, é possível usar as licenças existentes do banco de dados do Db2 para implantar bancos de dados no Amazon RDS. Verifique se você tem a licença apropriada do banco de dados do Db2 para a classe da instância de banco de dados e a edição do Db2 que deseja executar. Também é necessário seguir as políticas da IBM para o licenciamento do software de banco de dados da IBM no ambiente de computação em nuvem.

> **Note**
>
> As instâncias de banco de dados multi-AZ estão em espera fria porque o banco de dados do Db2 está instalado, mas não está em execução. Os modos de não podem ser lidos,

executados nem atender a solicitações. Consulte mais informações as informações de licença do IBM Db2 no site da IBM.

Nesse modelo, você deve continuar a usar a conta de suporte ativa da IBM e entrar em contato com a IBM diretamente para solicitações de serviço de banco de dados do Db2. Caso tenha uma conta do Suporte com suporte a caso, você pode entrar em contato com o Suporte para solucionar problemas do Amazon RDS. A Amazon Web Services e a IBM têm um processo de suporte de vários fornecedores para casos que necessitem de assistência das duas organizações.

O Amazon RDS é compatível com o modelo BYOL para Db2 Standard Edition e Db2 Advanced Edition.

Tópicos

- IDs da IBM para traga a sua própria licença para o Db2
- Adicionar IDs da IBM a um grupo de parâmetros para instâncias de banco de dados do RDS para Db2
- Integração com o AWS License Manager

IDs da IBM para traga a sua própria licença para o Db2

No modelo BYOL, é necessário ter o IBM Customer ID e o IBM Site ID para criar, modificar ou restaurar instâncias de banco de dados do RDS para Db2. É necessário criar um grupo de parâmetros personalizado com o IBM Customer ID e o IBM Site ID antes de criar uma instância de banco de dados do RDS para Db2. Para ter mais informações, consulte Adicionar IDs da IBM a um grupo de parâmetros para instâncias de banco de dados do RDS para Db2. É possível executar várias instâncias de banco de dados do RDS para Db2 com diferentes IBM Customer IDs e IBM Site IDs na mesma Conta da AWS ou Região da AWS.

Important

Se não conseguirmos verificar a licença pelo IBM Customer ID e pelo IBM Site ID, poderemos encerrar qualquer instância de banco de dados em execução com essas licenças não verificadas.

Se você for um cliente novo do IBM Db2, será necessário primeiro comprar uma licença de software Db2 da IBM. Depois de comprar uma licença de software Db2, você receberá um comprovante de titularidade da IBM, que indica o IBM Customer ID e o IBM Site ID.

Se você já é cliente do IBM Db2, é possível encontrar o IBM Customer ID e o IBM Site ID no certificado de comprovante de titularidade da IBM.

Você também pode encontrar o IBM Customer ID e o IBM Site ID na sua conta IBM Passport Advantage Online. Depois de fazer login, você pode ver os dois IDs na página principal ou na página de downloads de software.

Adicionar IDs da IBM a um grupo de parâmetros para instâncias de banco de dados do RDS para Db2

Como não é possível modificar grupos de parâmetros padrão, é necessário criar um grupo de parâmetros personalizado e depois modificá-lo para incluir os valores de IBM Customer ID e IBM Site ID. Para obter informações sobre grupos de parâmetros, consulte Grupos de parâmetros de banco de dados para instâncias de banco de dados do Amazon RDS.

> **Important**
>
> É necessário criar um grupo de parâmetros personalizado com o IBM Customer ID e o IBM Site ID antes de criar uma instância de banco de dados do RDS para Db2.

Use as configurações de parâmetros na tabela a seguir.

Parameter	Valor
`rds.ibm_customer_id`	`<your IBM Customer ID>`
`rds.ibm_site_id`	`<your IBM Site ID>`
`ApplyMethod`	`immediate` , `pending-reboot`

Esses parâmetros são dinâmicos, o que significa que qualquer alteração neles tem efeito imediatamente e que não é necessário reinicializar a instância de banco de dados. Se não quiser que as alterações tenham efeito imediatamente, poderá definir `ApplyMethod` e `pending-reboot` para programar essas alterações para serem feitas durante uma janela de manutenção.

É possível criar e modificar um grupo de parâmetros personalizado usando o AWS Management Console, a AWS CLI ou a API do Amazon RDS.

Console

Como adicionar o IBM Customer ID e o IBM Site ID a um grupo de parâmetros

1. Crie um grupo de parâmetros de banco de dados. Para obter mais informações sobre como criar um parameter group de banco de dados, consulte Criar um grupo de parâmetros de banco de dados no Amazon RDS.
2. Modifique o grupo de parâmetros criado. Para ter mais informações sobre como modificar um grupo de parâmetros, consulte Modificar parâmetros em um grupo de parâmetros de banco de dados no Amazon RDS.

AWS CLI

Como adicionar o IBM Customer ID e o IBM Site ID a um grupo de parâmetros

1. Crie um grupo de parâmetros personalizado executando o comando create-db-parameter-group.

   Inclua as seguintes opções necessárias:

   - `--db-parameter-group-name`: um nome para o grupo de parâmetros que você está criando.
   - `--db-parameter-group-family`: a edição do mecanismo do Db2 e a versão principal. Valores válidos: `db2-se-11.5`, `db2-ae-11.5`.
   - `--description`: a descrição desse grupo de parâmetros.

   Para obter mais informações sobre como criar um parameter group de banco de dados, consulte Criar um grupo de parâmetros de banco de dados no Amazon RDS.
2. Modifique os parâmetros no grupo de parâmetros personalizado que você criou executando o comando modify-db-parameter-group.

   Inclua as seguintes opções necessárias:

   - `--db-parameter-group-name`: o nome do grupo de parâmetros criado.
   - `--parameters`: uma matriz de nomes de parâmetros, valores e os métodos de aplicação da atualização de parâmetros.

Para ter mais informações sobre como modificar um grupo de parâmetros, consulte Modificar parâmetros em um grupo de parâmetros de banco de dados no Amazon RDS.

API do RDS

Como adicionar o IBM Customer ID e o IBM Site ID a um grupo de parâmetros

1. Crie um grupo de parâmetros de banco de dados personalizado usando a operação da API CreateDBParameterGroup do Amazon RDS.

   Inclua os seguintes parâmetros necessários:

   - `DBParameterGroupName`
   - `DBParameterGroupFamily`
   - `Description`

   Para obter mais informações sobre como criar um parameter group de banco de dados, consulte Criar um grupo de parâmetros de banco de dados no Amazon RDS.
2. Modifique os parâmetros no grupo de parâmetros personalizado que você criou usando a operação ModifyDBParameterGroup da API do RDS.

   Inclua os seguintes parâmetros necessários:

   - `DBParameterGroupName`
   - `Parameters`

   Para ter mais informações sobre como modificar um grupo de parâmetros, consulte Modificar parâmetros em um grupo de parâmetros de banco de dados no Amazon RDS.

Agora está tudo pronto para criar uma instância de banco de dados e associar o grupo de parâmetros personalizado à instância de banco de dados. Para ter mais informações, consulte Criar uma instância de banco de dados do Amazon RDS e Associar um grupo de parâmetros de banco de dados a uma instância de banco de dados no Amazon RDS.

Integração com o AWS License Manager

Para ajudar no monitoramento do uso da licença do RDS para Db2 no modelo BYOL, o AWS License Manager integra-se ao RDS para Db2. O License Manager é compatível com o rastreamento de edições de mecanismos do RDS para Db2 com base em CPUs virtuais (vCPUs). Você também pode usar o License Manager com o AWS Organizations para gerenciar centralmente todas as suas contas organizacionais.

Para monitorar o uso da licença das instâncias de banco de dados do RDS para Db2, é possível criar licenças autogerenciadas. Você pode criar licenças autogerenciadas usando o AWS Management Console, a CLI do AWS License Manager e a API do AWS License Manager. Ou você pode automatizar a criação de licenças autogerenciadas com modelos do AWS CloudFormation e do Terraform.

Os recursos do RDS para Db2 que correspondem ao filtro de informações do produto são automaticamente associados à licença autogerenciada. A detecção de instâncias de banco de dados do RDS para Db2 pode levar até 24 horas.

A tabela a seguir mostra os valores disponíveis do filtro de informações de produtos da Engine Edition do RDS para Db2.

Valor	Descrição
`db2-se`	Db2 edição Standard
`db2-ae`	Db2 Advanced Edition

Tópicos

- Terminologia
- Criar uma licença autogerenciada no AWS License Manager
- Automatização da criação de licenças autogerenciadas no AWS License Manager com modelos
- Configurações para criar licenças autogerenciadas

Terminologia

Essa página usa a terminologia a seguir ao abordar a integração do Amazon RDS com o AWS License Manager.

Licença autogerenciada

Licença autogerenciada é um termo usado no AWS License Manager. O console do Amazon RDS se refere à licença como uma configuração do AWS License Manager. Uma licença autogerenciada contêm regras de licenciamento com base nos termos de seus contratos empresariais. As regras que você cria determinam como a AWS processa comandos que consomem licenças. Ao criar uma licença autogerenciada, trabalhe em conjunto com a equipe de conformidade da sua organização para analisar seus contratos empresariais. Consulte mais informações em Self-managed licenses in License Manager.

Criar uma licença autogerenciada no AWS License Manager

Você pode criar uma licença autogerenciada usando o AWS Management Console, a CLI do AWS License Manager e a API do AWS License Manager.

> **Note**
>
> Se criar uma instância de banco de dados do RDS para Db2 usando o AWS Management Console, você criará uma licença autogerenciada ao inserir um nome para a licença. Depois, o Amazon RDS associa a instância de banco de dados a essa licença. (O console do Amazon RDS se refere à licença como uma configuração do AWS License Manager.) Se quiser criar uma instância de banco de dados do RDS para Db2 usando a CLI do AWS License Manager ou a API do AWS License Manager, é necessário primeiro criar uma licença autogerenciada com as etapas a seguir. A mesma situação se aplica à restauração de uma instância de banco de dados do RDS para Db2 para determinado ponto no tempo ou de um snapshot.

Console

Como criar uma licença autogerenciada a fim de rastrear o uso da licença das instâncias de banco de dados do RDS para Db2

1. Acesse https://console.aws.amazon.com/license-manager/.
2. Crie uma licença autogerenciada.

   Consulte instruções em Create a self-managed license no Guia do usuário do AWS License Manager.

Adicione uma regra para um RDS Product Information Filter (Filtro de informações de produto do RDS) no painel Product Information (Informações do produto).

Para obter mais informações, consulte ProductInformation na Referência da API do AWS License Manager.

AWS License Manager CLI

> **Note**
>
> Esse procedimento usa um comando da CLI do AWS License Manager.

Para criar uma licença autogerenciada usando a AWS CLI, execute o comando create-license-configuration do AWS License Manager. Use as opções `--cli-input-json` ou `--cli-input-yaml` para passar as opções para o comando.

Para ter mais informações, consulte the section called “Configurações para criar licenças autogerenciadas”.

O comando a seguir cria uma licença autogerenciada para o Db2 Standard Edition.

```
aws license-manager create-license-configuration --cli-input-json file://rds-db2-se.json
```

O JSON a seguir é o conteúdo do arquivo `rds-db2-se.json` usado no comando anterior.

```
{
    "Name": "rds-db2-se",
    "Description": "RDS Db2 Standard Edition",
    "LicenseCountingType": "vCPU",
    "LicenseCountHardLimit": false,
    "ProductInformationList": [
        {
            "ResourceType": "RDS",
            "ProductInformationFilterList": [
                {
                    "ProductInformationFilterName": "Engine Edition",
                    "ProductInformationFilterValue": ["db2-se"],
```

```
                    "ProductInformationFilterComparator": "EQUALS"
                }
            ]
        }
    ]
}
```

Para obter mais informações sobre o produto, consulte Detecção automática do inventário de recursos no Guia do usuário do AWS License Manager.

Para obter mais informações sobre o parâmetro `--cli-input`, consulte Gerar um esqueleto AWS CLI e parâmetros de entrada usando um arquivo de entrada JSON ou YAML no AWS CLI Guia do usuário .

API do AWS License Manager

> **Note**
>
> Esse procedimento usa um comando da API do AWS License Manager.

Para criar uma licença autogerenciada, use a operação de API `CreateLicenseConfiguration` do AWS License Manager com os seguintes parâmetros obrigatórios:

- `Name`
- `LicenseCountingType`
- `ProductInformationList`
- `ResourceType`
- `ProductInformationFilterList`
- `ProductInformationFilterName`
- `ProductInformationFilterValue`
- `ProductInformationFilterComparator`

Para obter mais informações sobre os parâmetros, consulte the section called “Configurações para criar licenças autogerenciadas”.

Automatização da criação de licenças autogerenciadas no AWS License Manager com modelos

Você pode automatizar a criação de licenças autogerenciadas usando modelos do AWS CloudFormation e do Terraform.

O seguinte modelo de exemplo AWS CloudFormation cria licenças autogerenciadas para o Db2 Standard Edition no RDS para Db2. Para obter um modelo para o Db2 Advanced Edition, atualize os valores para `Name`, `Description` e `ProductInformationFilter`.

```
AWSTemplateFormatVersion: "2010-09-09"
Description: CloudFormation template to create a License Configuration for Db2 Standard
 Edition on RDS for Db2.

Resources:
  Db2LicenseConfiguration:
    Type: "AWS::LicenseManager::LicenseConfiguration"
    Properties:
      Name: "rds-db2-se"
      Description: "Db2 Standard Edition on RDS for Db2"
      LicenseCountingType: "vCPU"
      LicenseCountHardLimit: false
      ProductInformationList:
        - ResourceType: "RDS"
          ProductInformationFilterList:
            - ProductInformationFilterName: "Engine Edition"
              ProductInformationFilterValue:
                - "db2-se"
              ProductInformationFilterComparator: "EQUALS"
```

Para obter mais informações sobre como usar o AWS CloudFormation com o Amazon RDS, consulte Criar recursos do Amazon RDS com o AWS CloudFormation.

O seguinte modelo de exemplo Terraform cria licenças autogerenciadas para o Db2 Standard Edition no RDS para Db2. Substitua *`us-east-1`* por sua Região da AWS. Para obter um modelo para o Db2 Advanced Edition, atualize os valores para `name`, `description` e `product_information_filter`.

```
provider "aws" {
  region = "us-east-1"
}
```

```
resource "aws_licensemanager_license_configuration" "rds_db2_license_config" {
  name                     = "rds-db2-se"
  description              = "Db2 Standard Edition on RDS for Db2
  license_counting_type    = "vCPU"
  license_count_hard_limit = false

  product_information_list {
    resource_type = "RDS"

    product_information_filter {
      name       = "Engine Edition"
      comparator = "EQUALS"
      value      = ["db2-se"]
    }
  }
}
```

Para obter mais informações sobre o uso do Terraform e do Amazon RDS, consulte Uso do Terraform como uma ferramenta de IaC para a Nuvem AWS e Práticas recomendadas para usar o Terraform AWS Provider na Orientação prescritiva da AWS.

Configurações para criar licenças autogerenciadas

Na tabela a seguir, é possível encontrar detalhes sobre as configurações para criar licenças autogerenciadas usando a CLI do AWS License Manager, a API do AWS License Manager, um modelo do AWS CloudFormation e um modelo do Terraform. O nome do parâmetro na tabela a seguir aparece no formato do nome usado na API do AWS License Manager e no modelo do AWS CloudFormation.

Nome do parâmetro	Tipo de dados	Obrigatório	Descrição
Name	string	Sim	O nome da configuração da licença.
Descrição	string	Não	A descrição da configuração da licença.
LicenseCountingType	string	Sim	A dimensão usada para monitorar o

Nome do parâmetro	Tipo de dados	Obrigatório	Descrição
			inventário de licenças. Valor válido: `vCPU`.
LicenseCountHardLimit	boolean	Não	Indica se é usada a aplicação de licenças rígida ou flexível. Exceder um limite rígido bloqueia a iniciação de novas instâncias.
ProductInformationList	matriz de objetos	Sim	Uma lista de informações do produto para uma configuração de licença.
ResourceType	string	Sim	O tipo de recurso. Valor válido: `RDS`.
ProductInformationFilterList	matriz de objetos	Sim	Uma lista de filtros de informações do produto para uma configuração de licença.
ProductInformationFilterName	string	Sim	O nome do tipo de filtro que está sendo declarado. Valor válido: `Engine Edition`.

Nome do parâmetro	Tipo de dados	Obrigatório	Descrição
ProductInformation FilterValue	matriz de strings	Sim	O valor a ser filtrado. Você deve especific ar apenas um valor. Valores válidos: `db2-se` ou `db2-ae`.
ProductInformation FilterComparator	string	Sim	O operador lógico para `ProductInformation FilterName` . Valor válido: `EQUALS`.

Licença do Db2 por meio do AWS Marketplace

No modelo de licença do Db2 por meio do AWS Marketplace, você paga uma taxa horária para assinar as licenças do Db2. Esse modelo ajuda você a começar rapidamente a usar o RDS para Db2 sem precisar comprar licenças.

Para usar a licença do Db2 por meio do AWS Marketplace, você precisa de uma assinatura do AWS Marketplace ativa para a edição do IBM Db2 específica que deseja usar. Se ainda não tiver uma, assine o AWS Marketplace para essa edição do IBM Db2.

O Amazon RDS comporta a licença do Db2 por meio do AWS Marketplace para o IBM Db2 Standard Edition e o IBM Db2 Advanced Edition.

Tópicos

- Terminologia
- Pagamentos e faturamento
- Assinar ofertas do Db2 Marketplace e cadastrá-las na IBM
- Obtendo uma oferta privada

Terminologia

Essa página usa a terminologia a seguir ao abordar a integração do Amazon RDS com o AWS Marketplace.

Assinatura de SaaS

No AWS Marketplace, os produtos de software como serviço (SaaS), como o modelo de licença pagamento conforme o uso, adotam um modelo de assinatura baseado no uso. A IBM, que vende o software para Db2, monitora o uso e você só paga pelo que usar.

Oferta pública

As ofertas públicas permitem que você compre produtos do AWS Marketplace diretamente do AWS Management Console.

Oferta privada

As ofertas privadas são um programa de compras que permite que vendedores e compradores negociem preços personalizados e termos do contrato de licenciamento de usuário final (EULA) para compras em AWS Marketplace.

Tarifas do Db2 Marketplace

Taxas cobradas pelo uso da licença do software Db2 pela IBM. Essas taxas de serviço são calculadas por meio do AWS Marketplace e aparecem em sua fatura da AWS na seção AWS Marketplace.

Taxas do Amazon RDS

Taxas cobradas pela AWS pelos serviços do RDS para Db2, que excluem as licenças ao usar o AWS Marketplace para licenças do Db2. As taxas são calculadas por meio do serviço do Amazon RDS que está sendo usado e aparecem na fatura da AWS.

Pagamentos e faturamento

O RDS para Db2 integra-se ao AWS Marketplace para oferecer licenças horárias e pagas conforme o uso para o Db2. As taxas do Db2 Marketplace cobrem os custos de licença do software Db2, ao passo que as taxas do Amazon RDS cobrem os custos do uso da instância de banco de dados do RDS para Db2. Para ter mais informações sobre preço, consulte Preço do Amazon RDS para Db2.

Para interromper essas taxas, é necessário excluir todas as instâncias de banco de dados do RDS para Db2. Além disso, é possível remover suas assinaturas das licenças do AWS Marketplace para Db2. Se você remover suas assinaturas sem excluir as instâncias de banco de dados, o Amazon RDS continuará cobrando pelo uso das instâncias de banco de dados. Para ter mais informações, consulte the section called “Excluir uma instância de banco de dados”.

É possível visualizar as faturas e gerenciar os pagamentos das instâncias de banco de dados do RDS para Db2 que usam a licença do Db2 por meio do AWS Marketplace no console do AWS Billing. Suas faturas incluem duas cobranças: uma pelo uso da licença do Db2 por meio do AWS Marketplace e outra pelo uso do Amazon RDS. Para ter mais informações sobre faturamento, consulte Viewing your bill no Guia do usuário do Gerenciamento de Faturamento e Custos da AWS.

Assinar ofertas do Db2 Marketplace e cadastrá-las na IBM

Para usar a licença do Db2 por meio do AWS Marketplace, é necessário usar o AWS Management Console para concluir as duas tarefas a seguir. Você não pode concluir essas tarefas por meio da AWS CLI ou da API do RDS.

> **Note**
>
> Se você quiser criar as instâncias de banco de dados usando a AWS CLI ou a API do RDS, primeiro conclua as duas tarefas a seguir.

Tópicos

- Tarefa 1: assinar o Db2 no AWS Marketplace
- Tarefa 2: cadastrar sua assinatura na IBM

Tarefa 1: assinar o Db2 no AWS Marketplace

Para usar a licença do Db2 com o AWS Marketplace, você precisa ter uma assinatura do AWS Marketplace ativa para o Db2. Como as assinaturas estão associadas a uma edição específica do IBM Db2, você precisa assinar o Db2 no AWS Marketplace para cada edição do Db2 que deseja usar: IBM Db2 Advanced Edition, IBM Db2 Standard Edition. Para ter informações sobre assinaturas do AWS Marketplace, consulte Assinaturas baseadas no uso de Saas no Guia do comprador do AWS Marketplace.

Recomendamos que você assine o Db2 no AWS Marketplace antes de começar a criar uma instância de banco de dados.

Tarefa 2: cadastrar sua assinatura na IBM

Depois de assinar o Db2 no AWS Marketplace, conclua o cadastro do seu pedido IBM na página do AWS Marketplace para o tipo de assinatura do Db2 que você escolheu. Na página AWS

Marketplace, escolha Ver opções de compra e, depois, selecione Configure sua conta. É possível fazer o cadastro com a conta da IBM existente ou criando uma conta da IBM gratuita.

Obtendo uma oferta privada

É possível solicitar uma oferta AWS Marketplace privada para o Db2 à IBM. Para ter mais informações, consulte Ofertas privadas no Guia do comprador do AWS Marketplace.

> **Note**
>
> Se você é um usuário do AWS Organizations e recebeu uma oferta privada emitida para suas contas de pagador e membro, siga o procedimento abaixo para se inscrever no Db2 diretamente em cada conta da sua organização.

Como receber uma oferta privada do Db2

1. Depois que uma oferta privada for emitida, faça login no console do AWS Marketplace.
2. Abra o e-mail com um link de oferta privada do Db2.
3. Siga o link para acessar diretamente a oferta privada.

   > **Note**
   >
   > Seguir esse link antes de fazer login na conta correta resultará em um erro de Página não encontrada (404).

4. Revise todos os Termos e Condições.
5. Ecolha Aceitar termos.

   > **Note**
   >
   > Se uma oferta privada do AWS Marketplace não for aceita, as taxas de serviço do Db2 do AWS Marketplace continuarão sendo cobradas de acordo com a tarifa pública por hora.

6. Para verificar os detalhes da oferta, selecione Mostrar detalhes na lista de produtos.

Depois de concluir o procedimento, é possível criar sua instância de banco de dados seguindo as etapas em the section called “Criar uma instância de banco de dados”. No AWS Management Console, em Licença, escolha Por meio de AWS Marketplace.

Alternar entre licenças do Db2

É possível alternar entre licenças do Db2 no RDS para Db2. Por exemplo, é possível começar com traga a sua própria licença e depois mudar para a licença do Db2 por meio do AWS Marketplace.

> **Important**
>
> Se você quiser alternar para a licença do Db2 por meio do AWS Marketplace, é necessário ter uma assinatura do AWS Marketplace ativa para a edição do IBM Db2 que deseja usar. Caso contrário, primeiro assine o Db2 no AWS Marketplace para essa edição do Db2 e, depois, conclua o procedimento de restauração.

Console

Como alternar entre licenças do Db2

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Automated backups (Backups automatizados).

   Os backups automatizados são exibidos na guia Current Region (região atual).
3. Escolha a instância de banco de dados que você quer restaurar.
4. Em Actions (Ações), escolha Restore to point in time (Restaurar para point-in-time).

   A janela Restore to point in time (Restaurar para point-in-time) é exibida.
5. Escolha Latest restorable time (Hora da última restauração) para restaurar no último horário possível ou escolha Custom (Personalizar) para escolher um horário.

   Se você escolher Personalizar, insira a data e a hora para as quais deseja restaurar a instância.

> **Note**
>
> Os horários são mostrados no fuso horário local, que é indicado por um deslocamento do Tempo Universal Coordenado (UTC). Por exemplo, UTC-5 é a Hora Padrão do Leste dos EUA/Horário de Verão Central.

6. Em Mecanismo de banco de dados, escolha a licença do Db2 que você deseja usar.
7. Em Identificador de instância de banco de dados, digite o nome da instância de banco de dados restaurada de destino. O nome deve ser exclusivo.
8. Escolha outras opções conforme necessário, como classe de instância de banco de dados, armazenamento e se você quer usar a escalabilidade automática de armazenamento.

   Para obter informações sobre cada configuração, consulte Configurações para instâncias de banco de dados.
9. Escolha Restore to point in time (Restaurar para point-in-time).

Para ter mais informações, consulte Restaurar uma instância de banco de dados para um momento especificado no Amazon RDS.

AWS CLI

Para alternar entre licenças do Db2, execute o comando restore-db-instance-to-point-in-time. O exemplo a seguir restaura a versão para um ponto no tempo mais recente, define o mecanismo de banco de dados como IBM Db2 Advanced Edition e define o modelo de licença como a licença do Db2 por meio do AWS Marketplace.

Você pode especificar outras configurações. Para obter informações sobre cada configuração, consulte Configurações para instâncias de banco de dados.

Example

Para Linux, macOS ou Unix:

```
aws rds restore-db-instance-to-point-in-time \
    --source-db-instance-identifier my_source_db_instance \
    --target-db-instance-identifier my_target_db_instance \
    --use-latest-restorable-time \
    --engine db2-ae \
```

```
    --license-model marketplace-license
```

Para Windows:

```
aws rds restore-db-instance-to-point-in-time ^
    --source-db-instance-identifier my_source_db_instance ^
    --target-db-instance-identifier my_target_db_instance ^
    --use-latest-restorable-time ^
    --engine db2-ae ^
    --license-model marketplace-license
```

Para ter mais informações, consulte Restaurar uma instância de banco de dados para um momento especificado no Amazon RDS.

API do RDS

Para alternar entre licenças do Db2, chame a operação `RestoreDBInstanceToPointInTime` da API do Amazon RDS com os seguintes parâmetros:

- `SourceDBInstanceIdentifier`
- `TargetDBInstanceIdentifier`
- `RestoreTime`
- `Engine`
- `LicenseModel`

Para ter mais informações, consulte Restaurar uma instância de banco de dados para um momento especificado no Amazon RDS.

Classes de instância do Amazon RDS para Db2

A capacidade de computação e memória de uma instância de banco de dados é determinada de acordo com sua classe de instância. A classe de instância de banco de dados da qual você precisa depende dos requisitos de energia e memória de processamento.

Classes de instância compatíveis com o RDS para Db2

As classes de instância compatíveis com o Amazon RDS para Db2 são um subconjunto das classes de instância de banco de dados do Amazon RDS. Para ter a lista completa de classes de instância do Amazon RDS, consulte Classes de instâncias de banco de dados .

Tópicos

- Classes de instância compatíveis do RDS para Db2 para Db2 Standard Edition
- Classes de instância compatíveis do RDS para Db2 para Db2 Advanced Edition

Classes de instância compatíveis do RDS para Db2 para Db2 Standard Edition

A tabela a seguir lista todas as classes de instâncias compatíveis com Db2 Standard Edition do banco de dados do Db2 11.5.9.0. Essas classes de instância estão disponíveis para licença traga a sua própria licença (BYOL) e do Db2 por meio do AWS Marketplace.

Tipo de classe de instância	Classe de instância
Classes de instância de uso geral com processadores Intel Xeon Scalable de terceira geração, armazenamento SSD e otimização de rede	db.m6idn.large–db.m6idn.8xlarge
Classes instâncias de uso geral com processadores Intel Xeon Scalable de terceira geração	db.m6in.large–db.m6in.8xlarge
Classes de instâncias de uso geral	db.m7i.large–db.m7i.8xlarge db.m6i.large–db.m6i.8xlarge
Classes de instância otimizada para memória com SSDs baseados em NVMe com processadores Intel Xeon Scalable de terceira geração	db.x2iedn.xlarge
Classes de instância otimizada para memória com processadores Intel Xeon Scalable de terceira geração	db.r6idn.large–db.r6idn.4xlarge db.r6in.large–db.r6in.4xlarge
Classes de instância otimizada para memória	db.r7i.large–db.r7i.8xlarge db.r6i.large–db.r6i.4xlarge
Classes de instância expansível	db.t3.small–db.t3.2xlarge

Classes de instância compatíveis do RDS para Db2 para Db2 Advanced Edition

A tabela a seguir lista todas as classes de instâncias compatíveis com Db2 Advanced Edition do banco de dados do Db2 11.5.9.0. Essas classes de instância estão disponíveis para licença traga a sua própria licença (BYOL) e do Db2 por meio do AWS Marketplace.

Tipo de classe de instância	Classe de instância
Classes de instância de uso geral com processadores Intel Xeon Scalable de terceira geração, armazenamento SSD e otimização de rede	db.m6idn.large–db.m6idn.32xlarge
Classes instâncias de uso geral com processadores Intel Xeon Scalable de terceira geração	db.m6in.large–db.m6in.32xlarge
Classes de instâncias de uso geral	db.m6i.large–db.m7i.48xlarge db.m7i.large–db.m7i.48xlarge
Classes de instância otimizada para memória com SSDs baseados em NVMe com processadores Intel Xeon Scalable de terceira geração	db.x2iedn.xlarge–db.x2iedn.32xlarge
Classes de instância otimizada para memória com processadores Intel Xeon Scalable de terceira geração	db.r6idn.large–db.r6idn.32xlarge db.r6in.8xlarge–db.r6in.32xlarge
Classes de instância otimizada para memória	db.r6i.large–db.r7i.48xlarge db.r7i.large–db.r7i.48xlarge
Classes de instância expansível	db.t3.small–db.t3.2xlarge

Perfis padrão do Amazon RDS para Db2

O RDS para Db2 adiciona os três perfis a seguir e concede a eles `master_user_role` com a opção `ADMIN`. Quando o banco de dados é provisionado, o RDS para Db2 concede

`master_user_role` ao usuário mestre. O usuário mestre, por sua vez, pode conceder esses perfis a outros usuários, grupos ou perfis com declarações `GRANT` nativas conectando-se ao banco de dados.

- DBA: o RDS para Db2 cria esse perfil vazio com autorização `DATAACCESS`. O usuário mestre pode adicionar mais autorizações ou privilégios a esse perfil e depois atribuir o perfil a outros usuários, grupos ou perfis.
- DBA_RESTRICTED: o RDS para Db2 cria esse perfil vazio. O usuário mestre pode adicionar privilégios a esse perfil e depois atribuir o perfil a outros usuários, grupos ou perfis.
- DEVELOPER: o RDS para Db2 cria esse perfil vazio com a autorização `DATAACCESS`. O usuário mestre pode adicionar mais autorizações ou privilégios a esse perfil e depois atribuir o perfil a outros usuários, grupos ou perfis.
- ROLE_NULLID_PACKAGES: o RDS para Db2 concede privilégios `EXECUTE` a esse perfil em pacotes `ALL NULLID` que foram vinculados pelo Db2 quando `CREATE DATABASE` foi executado.
- ROLE_PROCEDURES: o RDS para Db2 concede privilégios `EXECUTE` a esse perfil em todos os procedimentos `SYSIBM`.
- ROLE_TABLESPACES: o RDS para Db2 concede privilégios `USAGE` em espaços de tabela criados pelo comando `CREATE DATABASE`.

Parâmetros do Amazon RDS para Db2

O Amazon RDS para Db2 usa três tipos de parâmetros: parâmetros de configuração do gerenciador de banco de dados, variáveis do registro e parâmetros de configuração do banco de dados. É possível gerenciar os dois primeiros tipos por meio de grupos de parâmetros e o último tipo por meio do procedimento armazenado rdsadmin.update_db_param.

Por padrão, uma instância de banco de dados do RDS para Db2 usa um grupo de parâmetros de banco de dados específico de um banco de dados do Db2 e uma instância de banco de dados. Esse grupo contém parâmetros para o mecanismo de banco de dados IBM Db2, especificamente os parâmetros de configuração do gerenciador de banco de dados e variáveis de registro. Para ter informações sobre como trabalhar com grupos de parâmetros, consulte Grupos de parâmetros para Amazon RDS.

Os parâmetros de configuração do banco de dados do RDS para Db2 são definidos como os valores padrão do mecanismo de armazenamento selecionado. Para ter mais informações sobre

os parâmetros do Db2, consulte parâmetros de configuração do banco de dados do Db2 na documentação do IBM Db2.

Tópicos

- Visualizar os parâmetros em grupos de parâmetros
- Visualizar todos os parâmetros com comandos do Db2
- Modificar os parâmetros no grupo de parâmetros
- Modificar os parâmetros de configuração do banco de dados com comandos do Db2

Visualizar os parâmetros em grupos de parâmetros

Os parâmetros de configuração do gerenciador de banco de dados e as variáveis do registro são definidos em grupos de parâmetros. É possível visualizar os parâmetros de configuração do gerenciador de banco de dados e as variáveis do registro para uma versão específica do Db2 usando o AWS Management Console, a AWS CLI ou a API do RDS.

Console

Para visualizar os valores de parâmetros para um grupo de parâmetros de banco de dados

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, selecione Parameter groups.

   Os grupos de parâmetros de banco de dados aparecem em uma lista.
3. Escolha o nome do grupo de parâmetros para ver sua lista de parâmetros.

AWS CLI

É possível visualizar os parâmetros de configuração do gerenciador de banco de dados e as variáveis de registro de uma versão do Db2 executando o comando describe-engine-default-parameters. Especifique um destes valores a seguir para a opção `--db-parameter-group-family`:

- `db2-ae-11.5`
- `db2-se-11.5`

Por exemplo, para visualizar os parâmetros do Db2 Standard Edition 11.5, execute o seguinte comando:

```
aws rds describe-engine-default-parameters --db-parameter-group-family db2-se-11.5
```

Esse comando vai produzir uma saída semelhante ao seguinte exemplo:

```
{
    "EngineDefaults": {
        "Parameters": [
            {
                "ParameterName": "agent_stack_sz",
                "ParameterValue": "1024",
                "Description": "You can use this parameter to determine the amount of
 memory that is allocated by Db2 for each agent thread stack.",
                "Source": "engine-default",
                "ApplyType": "static",
                "DataType": "integer",
                "AllowedValues": "256-32768",
                "IsModifiable": false
            },
            {
                "ParameterName": "agentpri",
                "ParameterValue": "-1",
                "Description": "This parameter controls the priority given to all
 agents and to other database manager instance processes and threads by the operating
 system scheduler. This priority determines how CPU time is allocated to the database
 manager processes, agents, and threads relative to other processes and threads running
 on the machine.",
                "Source": "engine-default",
                "ApplyType": "static",
                "DataType": "integer",
                "AllowedValues": "1-99",
                "IsModifiable": false
            },
            ...
        ]
    }
}
```

Para listar somente os parâmetros modificáveis do Db2 Standard Edition 11.5, execute o seguinte comando:

Para Linux, macOS ou Unix:

```
aws rds describe-engine-default-parameters \
    --db-parameter-group-family db2-se-11.5 \
    --query 'EngineDefaults.Parameters[?IsModifiable==`true`].
{ParameterName:ParameterName, DefaultValue:ParameterValue}'
```

Para Windows:

```
aws rds describe-engine-default-parameters ^
    --db-parameter-group-family db2-se-11.5 ^
    --query 'EngineDefaults.Parameters[?IsModifiable==`true`].
{ParameterName:ParameterName, DefaultValue:ParameterValue}'
```

API do RDS

Para visualizar os valores de parâmetros para um grupo de parâmetros de banco de dados, use a operação `DescribeDBParameters` com o seguinte parâmetro obrigatório.

- `DBParameterGroupName`

Visualizar todos os parâmetros com comandos do Db2

É possível visualizar as configurações dos parâmetros de configuração do gerenciador de banco de dados, dos parâmetros de configuração do banco de dados e das variáveis do registro usando os comandos do Db2.

Como visualizar as configurações

1. Conecte-se ao banco de dados Db2. No exemplo a seguir, substitua *`database_name`*, *`master_username`* e *`master_password`* por suas próprias informações.

   ```
   db2 "connect to database_name user master_username using master_password"
   ```

2. Encontre a versão compatível do Db2.

   ```
   db2 "select service_level, fixpack_num from table(sysproc.env_get_inst_info()) as
    instanceinfo"
   ```

3. Visualize os parâmetros de uma versão específica do Db2.

- Visualize os parâmetros de configuração do gerenciador de banco de dados executando o seguinte comando:

```
db2 "select cast(substr(name,1,24) as varchar(24)) as name, case
    when value_flags = 'NONE' then '' else value_flags end flags,
    cast(substr(value,1,64) as varchar(64)) as current_value
    from sysibmadm.dbmcfg
    order by name asc with UR"
```

- Visualize todos os parâmetros de configuração do banco de dados executando o seguinte comando:

```
db2 "select cast(substr(name,1,24) as varchar(24)) as name, case
    when value_flags = 'NONE' then '' else value_flags end flags,
    cast(substr(value,1,64) as varchar(64)) as current_value
    from table(db_get_cfg(null)) order by name asc, member asc with UR"
```

- Visualize as variáveis de registro definidas no momento executando o seguinte comando:

```
db2 "select cast(substr(reg_var_name,1,50) as varchar(50)) as reg_var_name,
    cast(substr(reg_var_value,1,50) as varchar(50)) as reg_var_value,
    level from table(env_get_reg_variables(null))
    order by reg_var_name,member with UR"
```

Modificar os parâmetros no grupo de parâmetros

É possível modificar os parâmetros de configuração do gerenciador de banco de dados e as variáveis do registro em grupos de parâmetros personalizados usando o AWS Management Console, a AWS CLI a ou a API do RDS. Para ter mais informações, consulte Grupos de parâmetros de banco de dados para instâncias de banco de dados do Amazon RDS.

Console

Como modificar os parâmetros de configuração do gerenciador de banco de dados e as variáveis do registro

1. Crie um grupo de parâmetros personalizado. Para obter mais informações sobre como criar um parameter group de banco de dados, consulte Criar um grupo de parâmetros de banco de dados no Amazon RDS.

2. Modifique os parâmetros nesse grupo de parâmetros de banco de dados. Para ter mais informações sobre como modificar um grupo de parâmetros, consulte Modificar parâmetros em um grupo de parâmetros de banco de dados no Amazon RDS.

AWS CLI

Como modificar os parâmetros de configuração do gerenciador de banco de dados e as variáveis do registro

1. Crie um grupo de parâmetros personalizado executando o comando create-db-parameter-group.

   Inclua as seguintes opções necessárias:

   - `--db-parameter-group-name`: um nome para o grupo de parâmetros que você está criando.
   - `--db-parameter-group-family`: a edição do mecanismo do Db2 e a versão principal. Valores válidos: `db2-se-11.5`, `db2-ae-11.5`.
   - `--description`: a descrição desse grupo de parâmetros.

   Para obter mais informações sobre como criar um parameter group de banco de dados, consulte Criar um grupo de parâmetros de banco de dados no Amazon RDS.
2. Modifique os parâmetros no grupo de parâmetros personalizado que você criou executando o comando modify-db-parameter-group.

   Inclua as seguintes opções necessárias:

   - `--db-parameter-group-name`: o nome do grupo de parâmetros criado.
   - `--parameters`: uma matriz de nomes de parâmetros, valores e os métodos de aplicação da atualização de parâmetros.

   Para ter mais informações sobre como modificar um grupo de parâmetros, consulte Modificar parâmetros em um grupo de parâmetros de banco de dados no Amazon RDS.

API do RDS

Como modificar os parâmetros de configuração do gerenciador de banco de dados e as variáveis do registro

1. Crie um grupo de parâmetros de banco de dados personalizado usando a operação CreateDBParameterGroup.

   Inclua os seguintes parâmetros necessários:

   - `DBParameterGroupName`
   - `DBParameterGroupFamily`
   - `Description`

   Para obter mais informações sobre como criar um parameter group de banco de dados, consulte Criar um grupo de parâmetros de banco de dados no Amazon RDS.
2. Modifique os parâmetros no grupo de parâmetros personalizado que você criou usando a operação ModifyDBParameterGroup.

   Inclua os seguintes parâmetros necessários:

   - `DBParameterGroupName`
   - `Parameters`

   Para ter mais informações sobre como modificar um grupo de parâmetros, consulte Modificar parâmetros em um grupo de parâmetros de banco de dados no Amazon RDS.

Modificar os parâmetros de configuração do banco de dados com comandos do Db2

É possível modificar os parâmetros de configuração do banco de dados com os comandos do Db2.

Como modificar os parâmetros de configuração do banco de dados

1. Conecte-se ao banco de dados `rdsadmin`. No exemplo a seguir, substitua *`master_username`* e *`master_password`* por suas próprias informações.

```
db2 "connect to rdsadmin user master_username using master_password"
```

2. Altere os parâmetros de configuração do banco de dados chamando o procedimento armazenado `rdsadmin.update_db_param`. Para ter mais informações, consulte rdsadmin.update_db_param.

```
db2 "call rdsadmin.update_db_param(
    'database_name',
    'parameter_to_modify',
    'changed_value')"
```

Agrupamento EBCDIC para bancos de dados do Db2 no Amazon RDS

O Amazon RDS para Db2 aceita o agrupamento EBCDIC para bancos de dados do Db2. Só é possível especificar uma sequência de agrupamento EBCDIC para um banco de dados ao criar o banco de dados usando o procedimento armazenado the section called “rdsadmin.create_database” do Amazon RDS.

Ao criar uma instância de banco de dados do RDS para Db2 usando o AWS Management Console, a AWS CLI ou a API do RDS, é possível especificar um nome de banco de dados. Se você especificar um nome de banco de dados, o Amazon RDS criará um banco de dados com o agrupamento padrão de `SYSTEM`. Se você precisar criar um banco de dados com agrupamento EBCDIC, não especifique um nome de banco de dados ao criar uma instância de banco de dados.

O agrupamento de um banco de dados no RDS para Db2 é definido no momento da criação e é imutável.

Como criar um banco de dados do Db2 com agrupamento EBCDIC

1. Se você não tiver uma instância de banco de dados do RDS para Db2, crie uma instância de banco de dados, mas não especifique um nome de banco de dados. É possível criar uma instância de banco de dados usando o AWS Management Console, a AWS CLI ou a API do RDS. Para ter mais informações, consulte Criar uma instância de banco de dados.
2. Crie um banco de dados do Db2 e defina a opção de agrupamento como um valor EBCDIC chamando o procedimento armazenado `rdsadmin.create_database`. Para ter mais informações, consulte rdsadmin.create_database.

> **Important**
>
> Após a criação de um banco de dados usando o procedimento armazenado, não é possível alterar a sequência de agrupamento. Se você quiser que um banco de dados use uma sequência de agrupamento diferente, descarte o banco de dados chamando o procedimento armazenado the section called “rdsadmin.drop_database”. Depois, crie um banco de dados com a sequência de agrupamento necessária.

Fuso horário local para instâncias de banco de dados do Amazon RDS para Db2

O fuso horário de uma instância de banco de dados do Amazon RDS que executa o Db2 é definida por padrão. O padrão é o Tempo Universal Coordenado (UTC). Para combinar o fuso horário de suas aplicações, você pode definir o fuso horário de sua instância de banco de dados como um fuso horário local.

Você define o fuso horário ao criar sua instância de banco de dados. É possível criar uma instância de banco de dados usando o AWS Management Console, a API do RDS ou a AWS CLI. Para ter mais informações, consulte Criar uma instância de banco de dados.

Se a instância de banco de dados fizer parte de uma implantação multi-AZ, durante o failover, o fuso horário permanecerá como o fuso horário local definido.

É possível restaurar a instância de banco de dados para um ponto no tempo que você especifica. A hora é mostrada no fuso horário local. Para ter mais informações, consulte Restaurar uma instância de banco de dados para um momento especificado no Amazon RDS.

Definir o fuso horário local em sua instância de banco de dados tem as seguintes limitações:

- Não é possível modificar o fuso horário de uma instância de banco de dados existente do Amazon RDS para Db2.
- Não é possível restaurar um snapshot de uma instância de banco de dados em um fuso horário para uma instância de banco de dados em um fuso horário diferente.
- Recomendamos que não restaure um arquivo de backup de um fuso horário em um fuso horário diferente. Se você restaurar um arquivo de backup de um fuso horário para outro, será preciso auditar as consultas e as aplicações quanto à alteração de fuso horário.

Fusos horários disponíveis

É possível usar os valores a seguir para a configuração de fuso horário.

Zona	Time zone (Fuso horário)
África	África/Cairo, África/Casablanca, África/Harare, África/Lagos, África/Luanda, África/Monrovia, África/Nairobi, África/Tripoli, África/Windhoek
América	América/Araguaina, América/Argentina/Buenos_Aires, América/Asuncion, América/Bogota, América/Caracas, América/Chicago, América/Chihuahua, América/Cuiaba, América/Denver, América/Detroit, América/Fortaleza, América/Godthab, América/Guatemala, América/Halifax, América/Lima, América/Los_Angeles, América/Manaus, América/Matamoros, América/Mexico_City, América/Monterrey, América/Montevideo, América/New_York, América/Phoenix, América/Santiago, América/Sao_Paulo, América/Tijuana, América/Toronto
Ásia	Ásia/Amman, Ásia/Ashgabat, Ásia/Baghdad, Ásia/Baku, Ásia/Bangkok, Ásia/Beirut, Ásia/Calcutta, Ásia/Damascus, Ásia/Dhaka, Ásia/Hong_Kong, Ásia/Irkutsk, Ásia/Jakarta, Ásia/Jerusalem, Ásia/Kabul, Ásia/Karachi, Ásia/Kathmandu, Ásia/Kolkata, Ásia/Krasnoyarsk, Ásia/Magadan, Ásia/Manila, Ásia/Muscat, Ásia/Novosibirsk, Ásia/Rangoon, Ásia/Riyadh, Ásia/Seoul, Ásia/Shanghai, Ásia/Singapore, Ásia/Taipei, Ásia/Tehran, Ásia/Tokyo, Ásia/Ulaanbaatar, Ásia/Vladivostok, Ásia/Yakutsk, Ásia/Yerevan
Atlântico	Atlântico/Açores, Atlântico/Cape_Verde
Austrália	Austrália/Adelaide, Austrália/Brisbane, Austrália/Darwin, Austrália/Eucla, Austrália/Hobart, Austrália/Lord_Howe, Austrália/Perth, Austrália/Sydney
Brasil	Brasil/DeNoronha, Brasil/Leste
Canadá	Canadá/Newfoundland, Canadá/Saskatchewan
Etc	Etc/GMT-3
Europa	Europa/Amsterdam, Europa/Athens, Europa/Berlin, Europa/Dublin, Europa/Helsinki, Europa/Kaliningrad, Europa/London, Europa/Madrid, Europa/Moscow,

Zona	Time zone (Fuso horário)
	Europa/Paris, Europa/Prague, Europa/Rome, Europa/Sarajevo, Europa/Stockholm
Pacífico	Pacífico/Apia, Pacífico/Auckland, Pacífico/Chatham, Pacífico/Fiji, Pacífico/Guam, Pacífico/Honolulu, Pacífico/Kiritimati, Pacífico/Marquesas, Pacífico/Samoa, Pacífico/Tongatapu, Pacífico/Wake
EUA	EUA/Alasca, EUA/Central, EUA/East-Indiana, EUA/Eastern, EUA/Pacífico
UTC	UTC

Pré-requisitos para criar uma instância de banco de dados do Amazon RDS para Db2

Os itens a seguir são pré-requisitos antes da criação de uma instância de banco de dados.

Tópicos

- Conta de administrador
- Considerações adicionais

Conta de administrador

Ao criar uma instância de banco de dados, é necessário designar uma conta de administrador para a instância. O Amazon RDS concede a autoridade `DBADM` a essa conta de administrador de banco de dados local.

A conta do administrador tem as seguintes características, recursos e limitações:

- É um usuário local e não uma Conta da AWS.
- Não tem autoridades em nível de instância do Db2, como `SYSADM`, `SYSMAINT` ou `SYSCTRL`.
- Não pode parar nem iniciar uma instância do Db2.
- Não será possível descartar um banco de dados do Db2 se você especificou o nome ao criar a instância de banco de dados.
- Tem acesso total ao banco de dados do Db2, incluindo tabelas e visualizações do catálogo.
- Pode criar usuários e grupos locais usando procedimentos armazenados do Amazon RDS.
- Pode conceder e revogar autoridades e privilégios.

A conta do administrador pode executar as seguintes tarefas:

- Criar, modificar ou excluir instâncias de banco de dados.
- Criar um snapshot de banco de dados.
- Iniciar restaurações para um ponto no tempo.
- Criar backups automatizados de snapshots de banco de dados.
- Criar backups manuais de snapshots de banco de dados.
- Usar outros recursos do Amazon RDS.

Considerações adicionais

Antes de criar uma instância de banco de dados, pense nos seguintes itens:

- Cada instância de banco de dados do Amazon RDS para Db2 pode hospedar até 50 bancos de dados do Db2. Para ter mais informações, consulte Vários bancos de dados em uma instância de banco de dados do Amazon RDS para Db2.
- Nome do banco de dados inicial
  - Se você não fornecer um nome de banco de dados ao criar uma instância de banco de dados, o Amazon RDS não criará um banco de dados.
  - Não forneça um nome de banco de dados nas seguintes circunstâncias:
    - Você deseja modificar o parâmetro `db2_compatibility_vector`. Para ter mais informações, consulte Configuração do parâmetro db2_compatibility_vector.
- No modelo traga a sua própria licença (BYOL), primeiro é necessário criar um grupo de parâmetros personalizado que contenha o IBM Customer ID e o IBM Site ID. Para ter mais informações, consulte Traga a sua própria licença para Db2.
- No modelo de licença do Db2 por meio do AWS Marketplace, você precisa de uma assinatura do AWS Marketplace ativa para a edição do IBM Db2 específica que deseja usar. Se ainda não tiver uma, assine o Db2 no AWS Marketplace para a edição IBM Db2 que deseja usar. Para ter mais informações, consulte Licença do Db2 por meio do AWS Marketplace.

Vários bancos de dados em uma instância de banco de dados do Amazon RDS para Db2

Você pode criar vários bancos de dados em uma única instância de banco de dados do RDS para Db2 chamando o procedimento armazenado rdsadmin.create_database. Uma única instância de banco de dados do RDS para Db2 é limitada a 50 bancos de dados. Esse número inclui bancos de dados nos estados ativado e desativado.

> **Note**
>
> Se você criar vários bancos de dados em uma instância de banco de dados do RDS para Db2 criada antes de 15 de novembro de 2024, deverá reiniciar a instância de banco de dados para habilitar o suporte a vários bancos de dados.

Por padrão, o Amazon RDS ativa os bancos de dados quando são criados. Para otimizar os recursos de memória, você pode desativar bancos de dados que usa com pouca frequência e ativá-los posteriormente, quando necessário. Para ter mais informações, consulte the section called “Desativação de um banco de dados” e the section called “Ativação de um banco de dados”.

O número de bancos de dados ativados em uma instância de banco de dados depende dos recursos de memória disponíveis no servidor. Os recursos de memória diferem com base na classe da instância de banco de dados e na quantidade de memória configurada para o banco de dados. Para obter informações sobre classes de instância de banco de dados, consulte the section called “Classes da instância de banco de dados”. Para obter informações sobre como atualizar a memória de um banco de dados do RDS para Db2, consulte the section called “rdsadmin.update_db_param”.

Recomendamos que você escolha uma classe de instância de banco de dados que tenha 2 GB de memória para tarefas comuns de banco de dados, requisitos de sistema operacional e outras tarefas de automação do Amazon RDS, como backups. Para obter mais informações sobre como alterar a classe de instância de banco de dados, consulte the section called “Modificação de uma instância de banco de dados”.

Além disso, a IBM recomenda no mínimo 1 GB de memória para cada banco de dados ativo. Para obter mais informações, consulte Requisitos de disco e memória na documentação da IBM.

Você pode calcular o número máximo de bancos de dados ativos que uma instância de banco de dados pode ter com a seguinte fórmula:

```
Active database limit = (total server memory - 2 GB) / 1 GB
```

O exemplo a seguir mostra o número máximo de bancos de dados ativos para uma instância de banco de dados com uma classe de instância de banco de dados db.m6i.xlarge:

```
Active database limit = (total server memory - 2 GB) / 1 GB
                      = (16 GB - 2 GB) / 1 GB
                      = 14 databases
```

Quando o Amazon RDS recupera um banco de dados após uma falha, ele ativa o banco de dados caso estivesse ativo anteriormente. Em determinados casos, como quando você modifica uma classe de instância de banco de dados para uma configuração de memória menor, pode haver memória insuficiente para ativar todos os bancos de dados na instância de banco de dados. Nesses casos, o Amazon RDS ativa os bancos de dados na ordem em que foram criados.

> **Note**
>
> Os bancos de dados que o Amazon RDS não puder ativar devido à memória insuficiente permanecerão em um estado desativado.

Conectar-se à instância de banco de dados do Db2

Depois que o Amazon RDS provisiona a instância de banco de dados do Amazon RDS para Db2, é possível usar qualquer aplicação cliente SQL padrão para se conectar à instância de banco de dados. Como o Amazon RDS é um serviço gerenciado, não é possível fazer login como `SYSADM`, `SYSCTRL`, `SECADM` e `SYSMAINT`.

É possível se conectar a uma instância de banco de dados que esteja executando o mecanismo de banco de dados IBM Db2 usando IBM Db2 CLP, IBM CLPPlus, DBeaver ou IBM Db2 Data Management Console.

> **Note**
>
> A conexão com um banco de dados do Db2 pode falhar se a instância de banco de dados do RDS para Db2 não tiver memória suficiente. Para ter mais informações, consulte the section called “Erro de conexão de banco de dados”.

Tópicos

- Encontrar o endpoint da instância de banco de dados do Amazon RDS para Db2
- Conectar-se à instância de banco de dados do Amazon RDS para Db2 com o IBM Db2 CLP
- Conectar-se à instância de banco de dados do Amazon RDS para Db2 com o IBM CLPPlus
- Conectar-se à instância de banco de dados do Amazon RDS para Db2 com o DBeaver
- Conectar-se à instância de banco de dados do Amazon RDS para Db2 com o IBM Db2 Data Management Console
- Considerações sobre grupos de segurança com o Amazon RDS para Db2

Encontrar o endpoint da instância de banco de dados do Amazon RDS para Db2

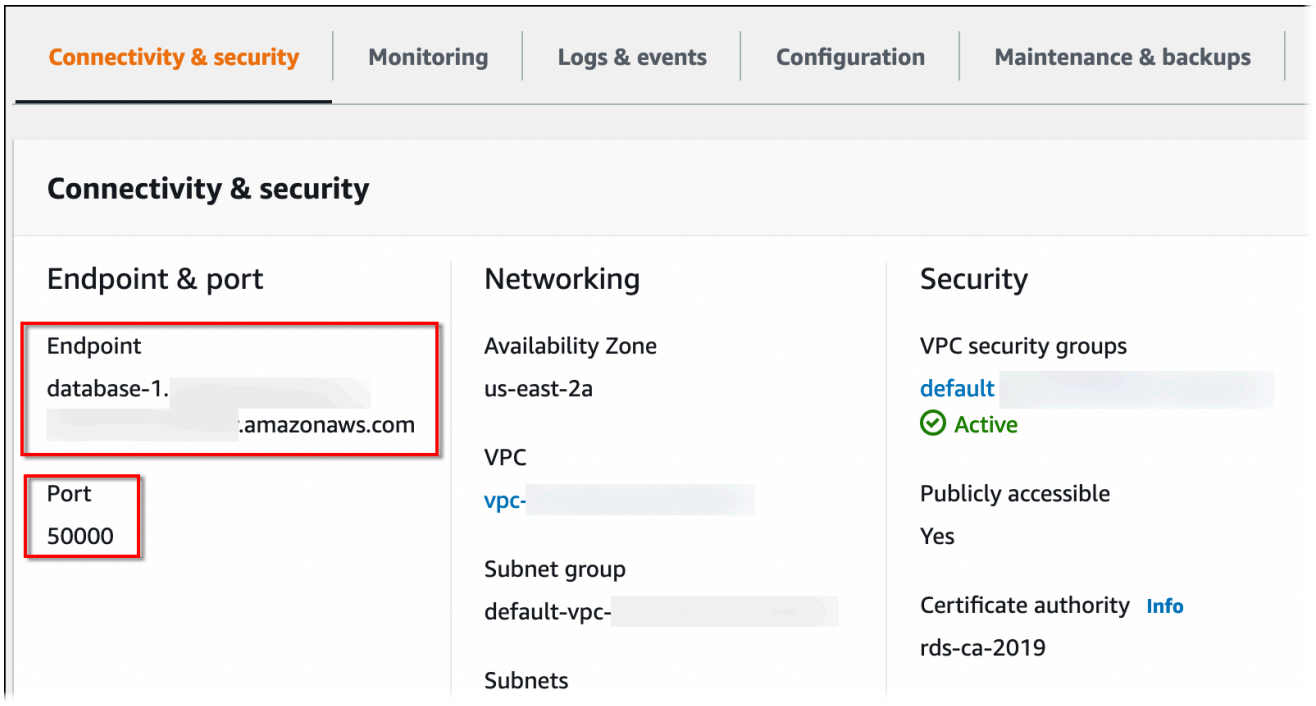
Cada instância de banco de dados do Amazon RDS tem um endpoint, e cada ponto final possui o nome DNS e o número da porta para a instância de banco de dados. Para se conectar à instância de banco de dados do Amazon RDS para Db2 com uma aplicação cliente do SQL, é necessário ter o nome do DNS e o número da porta da instância de banco de dados.

É possível encontrar o endpoint de uma instância de banco de dados usando o AWS Management Console ou a AWS CLI.

Console

Como encontrar o endpoint de uma instância de banco de dados do RDS para Db2

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No canto superior direito do console, selecione a Região da AWS da instância de banco de dados.
3. Encontre o nome do DNS e o número da porta da instância de banco de dados do RDS para Db2.
   a. Escolha Databases (Bancos de dados) para exibir a lista das instâncias de banco de dados.
   b. Escolha o nome da instância de banco de dados do RDS para Db2 para exibir os detalhes da instância.
   c. Na guia Connectivity & security (Conectividade e segurança), copie o endpoint. Além disso, anote o número da porta. Você precisará do endpoint e do número da porta para conectar-se à instância de banco de dados.



AWS CLI

Para encontrar o endpoint de uma instância de banco de dados do RDS para Db2, execute o comando describe-db-instances. No exemplo a seguir, substitua *database-1* pelo nome da instância de banco de dados.

Para Linux, macOS ou Unix:

```
aws rds describe-db-instances \
    --db-instance-identifier database-1 \
    --query 'DBInstances[].
{DBInstanceIdentifier:DBInstanceIdentifier,DBName:DBName,Endpoint:Endpoint}' \
    --output json
```

Para Windows:

```
aws rds describe-db-instances ^
    --db-instance-identifier database-1 ^
    --query 'DBInstances[].
{DBInstanceIdentifier:DBInstanceIdentifier,DBName:DBName,Endpoint:Endpoint}' ^
    --output json
```

O comando vai produzir uma saída semelhante ao exemplo a seguir. A linha `Address` na saída contém o nome DNS.

```
[
    {
        "DBInstanceIdentifier": "database-1",
        "DBName": "DB2DB",
        "Endpoint": {
            "Address": "database-1.123456789012.us-east-2.amazonaws.com",
            "Port": 50000,
            "HostedZoneId": "Z2OC4A7DETW6VH"
        }
    }
]
```

Conectar-se à instância de banco de dados do Amazon RDS para Db2 com o IBM Db2 CLP

É possível usar um utilitário de linha de comandos, como IBM Db2 CLP, para se conectar a instâncias de banco de dados do Amazon RDS para Db2. Esse utilitário faz parte do IBM Data Server Runtime Client. Para baixar o cliente do IBM Fix Central, consulte IBM Data Server Client Packages Version 11.5 Mod 8 Fix Pack 0 no IBM Support.

Tópicos

- Terminologia
- Instalar o cliente
- Conexão a uma instância de banco de dados
- Solução de problema de conexão com a instância de banco de dados do RDS para Db2

Terminologia

Os termos a seguir ajudam a explicar os comandos usados ao se conectar à instância de banco de dados do RDS para Db2.

catalog tcpip node

> Esse comando registra um nó de banco de dados remoto com um cliente Db2 local, o que torna o nó acessível à aplicação cliente. Para catalogar um nó, você vai fornecer informações, como nome do host, número da porta e protocolo de comunicação do servidor. O nó catalogado então representa um servidor de destino onde residem um ou mais bancos de dados remotos. Para ter mais informações, consulte CATALOG TCPIP/TCPIP4/TCPIP6 NODE command na documentação do IBM Db2.

catalog database

> Esse comando registra um banco de dados remoto com um cliente Db2 local, o que torna o banco de dados acessível à aplicação cliente. Para catalogar um banco de dados, você vai fornecer informações, como o alias do banco de dados, o nó no qual ele reside e o tipo de autenticação necessário para se conectar ao banco de dados. Para ter mais informações, consulte CATALOG DATABASE command na documentação do IBM Db2.

Instalar o cliente

Depois de downloading the package for Linux, instale o cliente usando privilégios raiz ou de administrador.

> **Note**
>
> Para instalar o cliente no AIX ou no Windows, siga o mesmo procedimento, mas modifique os comandos do sistema operacional.

Como instalar o cliente no Linux

1. Execute **`./db2_install -f sysreq`** e escolha **`yes`** para aceitar a licença.
2. Selecione o local para instalar o cliente.
3. Execute **`clientInstallDir/instance/db2icrt -s client`** ***`instance_name`***. Substitua *`instance_name`* por um usuário válido do sistema operacional no Linux. Em Linux, o nome da instância de banco de dados do Db2 está vinculado ao nome de usuário do sistema operacional.

   Esse comando cria um diretório **`sqllib`** no diretório inicial do usuário designado no Linux.

Conexão a uma instância de banco de dados

Para se conectar à instância de banco de dados do RDS para Db2, você precisa do nome do DNS e do número da porta. Para ter informações sobre como descobri-los, consulte Encontrar o endpoint. Também é necessário saber o nome do banco de dados, o nome de usuário principal e a senha mestra que você definiu ao criar a instância de banco de dados do RDS para Db2. Para ter mais informações sobre como descobri-los, consulte Criar uma instância de banco de dados.

Como se conectar a uma instância de banco de dados do RDS para Db2 com IBM Db2 CLP

1. Faça login com o nome de usuário que você especificou durante a instalação do cliente IBM Db2 CLP.
2. Catalogue a instância de banco de dados do RDS para Db2. No exemplo a seguir, substitua *`node_name`*, *`dns_name`* e *`port`* por um nome para o nó no catálogo local, o nome do DNS da instância de banco de dados e o número da porta.

   ```
   db2 catalog TCPIP node node_name remote dns_name server port
   ```

   Exemplo

```
db2 catalog TCPIP node remnode remote database-1.123456789012.us-
east-1.amazonaws.com server 50000
```

3. Catalogue o banco de dados `rdsadmin` e seu banco de dados. Isso permitirá que você se conecte ao banco de dados `rdsadmin` para realizar algumas tarefas administrativas usando procedimentos armazenados do Amazon RDS. Para ter mais informações, consulte Administrar a instância de banco de dados do RDS para Db2.

   No exemplo a seguir, substitua *`database_alias`*, *`node_name`* e *`database_name`* por aliases para esse banco de dados, o nome do nó definido na etapa anterior e o nome do banco de dados. O `server_encrypt` criptografa o nome de usuário e a senha na rede.

```
db2 catalog database rdsadmin [ as database_alias ] at node node_name
 authentication server_encrypt

db2 catalog database database_name [ as database_alias ] at node node_name
 authentication server_encrypt
```

   Exemplo

```
db2 catalog database rdsadmin at node remnode authentication server_encrypt

db2 catalog database testdb as rdsdb2 at node remnode authentication server_encrypt
```

4. Conecte-se ao banco de dados do RDS para Db2. No exemplo a seguir, substitua *`rds_database_alias`*, *`master_username`* e *`master_password`* pelo nome do banco de dados, o nome de usuário principal e a senha mestra da instância de banco de dados do RDS para Db2.

```
db2 connect to rds_database_alias user master_username using master_password
```

   O comando vai produzir uma saída semelhante ao seguinte exemplo:

```
Database Connection Information

    Database server        = DB2/LINUXX8664 11.5.9.0
    SQL authorization ID   = ADMIN
    Local database alias   = TESTDB
```

5. Execute consultas e visualize os resultados. O exemplo a seguir mostra uma declaração SQL que seleciona o banco de dados que você criou.

```
db2 "select current server from sysibm.dual"
```

O comando vai produzir uma saída semelhante ao seguinte exemplo:

```
1
    ------------------
    TESTDB

    1 record(s) selected.
```

Solução de problema de conexão com a instância de banco de dados do RDS para Db2

O erro `NULLID` a seguir geralmente indica que as versões do cliente e do servidor do RDS para Db2 não coincidem. Para ver as versões compatíveis do cliente do Db2, consulte Supported combinations of clients, drivers and server levels na documentação do IBM Db2.

```
db2 "select * from syscat.tables"
SQL0805N Package "NULLID.SQLC2O29 0X4141414141454A69" was not found.
SQLSTATE=51002
```

Depois de receber esse erro, você deverá vincular pacotes do cliente do Db2 antigo a uma versão do servidor do Db2 compatível com o RDS para Db2.

Como vincular pacotes de um cliente do Db2 mais antigo a um servidor do Db2 mais recente

1. Localize os arquivos de vinculação na máquina cliente. Normalmente, esses arquivos estão localizados no diretório bnd do caminho de instalação do cliente do Db2 e têm a extensão .bnd.
2. Conecte-se ao servidor do Db2. No exemplo a seguir, substitua *database_name* pelo nome do servidor do Db2. Substitua *master_username* e *master_password* por suas informações. Esse usuário tem autoridade `DBADM`.

```
db2 connect to database_name user master_username using master_password
```

3. Execute o comando `bind` para vincular os pacotes.

a. Navegue até o diretório onde estão os arquivos de vinculação na máquina cliente.

b. Execute o comando `bind` para cada arquivo.

   São necessárias as seguintes opções:

   - `blocking all`: vincula todos os pacotes no arquivo de associação em uma única solicitação de banco de dados.
   - `grant public`: concede permissão a `public` para executar o pacote.
   - `sqlerror continue`: especifica que o processo `bind` continue mesmo se ocorrerem erros.

   Para ter mais informações sobre o comando `bind`, consulte BIND command na documentação do IBM Db2.

4. Verifique se a associação foi bem-sucedida consultando a visualização do catálogo `syscat.package` ou verificando a mensagem exibida após o comando `bind`.

Para ter mais informações, consulte DB2 v11.5 Bind File and Package Name List em IBM Support.

Conectar-se à instância de banco de dados do Amazon RDS para Db2 com o IBM CLPPlus

É possível usar um utilitário, como o IBM CLPPlus, para se conectar a uma instância de banco de dados do Amazon RDS para Db2. Esse utilitário faz parte do IBM Data Server Runtime Client. Para baixar o cliente do IBM Fix Central, consulte IBM Data Server Client Packages Version 11.5 Mod 8 Fix Pack 0 no IBM Support.

> **Important**
>
> Recomendamos executar o IBM CLPPlus em um sistema operacional compatível com interfaces gráficas de usuário, como macOS, Windows ou Linux com Desktop. Se estiver executando o Linux headless, use a opção -nw com comandos do CLPPlus.

Tópicos

- Instalar o cliente

- Conexão a uma instância de banco de dados

Instalar o cliente

Depois de baixar o pacote para Linux, instale o cliente.

> **Note**
>
> Para instalar o cliente no AIX ou no Windows, siga o mesmo procedimento, mas modifique os comandos do sistema operacional.

Como instalar o cliente no Linux

1. Executar **`./db2_install`**.
2. Execute **`clientInstallDir/instance/db2icrt -s client`** ***`instance_name`***. Substitua *`instance_name`* por um usuário válido do sistema operacional no Linux. Em Linux, o nome da instância de banco de dados do Db2 está vinculado ao nome de usuário do sistema operacional.

   Esse comando cria um diretório **`sqllib`** no diretório inicial do usuário designado no Linux.

Conexão a uma instância de banco de dados

Para se conectar à instância de banco de dados do RDS para Db2, você precisa do nome do DNS e do número da porta. Para ter informações sobre como descobri-los, consulte Encontrar o endpoint. Também é necessário saber o nome do banco de dados, o nome de usuário principal e a senha mestra que você definiu ao criar a instância de banco de dados do RDS para Db2. Para ter mais informações sobre como descobri-los, consulte Criar uma instância de banco de dados.

Como se conectar a uma instância de banco de dados do RDS para Db2 com IBM CLPPlus

1. Revise a sintaxe do comando. No exemplo a seguir, substitua *`clientDir`* pelo local em que o cliente está instalado.

   ```
   cd clientDir/bin
       ./clpplus -h
   ```

2. Configure o servidor Db2. No exemplo a seguir, substitua *`dsn_name`*, *`database_name`*, *`endpoint`* e *`port`* pelo nome do DSN, nome do banco de dados, endpoint e porta da instância

de banco de dados do RDS para Db2. Para ter mais informações, consulte Encontrar o endpoint da instância de banco de dados do Amazon RDS para Db2.

```
db2cli writecfg add -dsn dsn_name -database database_name -host endpoint -port port
 -parameter "Authentication=SERVER_ENCRYPT"
```

3. Conecte-se à instância de banco de dados do RDS para Db2. No exemplo a seguir, substitua *master_username* e *dsn_name* pelo nome de usuário principal e pelo nome do DNS.

```
./clpplus -nw master_username@dsn_name
```

4. Uma janela Java Shell é aberta. Digite a senha mestra da instância de banco de dados do RDS para Db2.

> **Note**
>
> Se a janela Java Shell não for aberta, execute **`./clpplus -nw`** para usar a mesma janela da linha de comandos.

```
Enter password: *********
```

Uma conexão é estabelecida e produz uma saída semelhante ao seguinte exemplo:

```

Database Connection Information :
---------------------------------
Hostname = database-1.abcdefghij.us-east-1.rds.amazonaws.com
Database server = DB2/LINUXX8664  SQL110590
SQL authorization ID = admin
Local database alias = DB2DB
Port = 50000
```

5. Execute consultas e visualize os resultados. O exemplo a seguir mostra uma declaração SQL que seleciona o banco de dados que você criou.

```
SQL > select current server from sysibm.dual;
```

O comando vai produzir uma saída semelhante ao seguinte exemplo:

```
1
    --------------------
    DB2DB
    SQL>
```

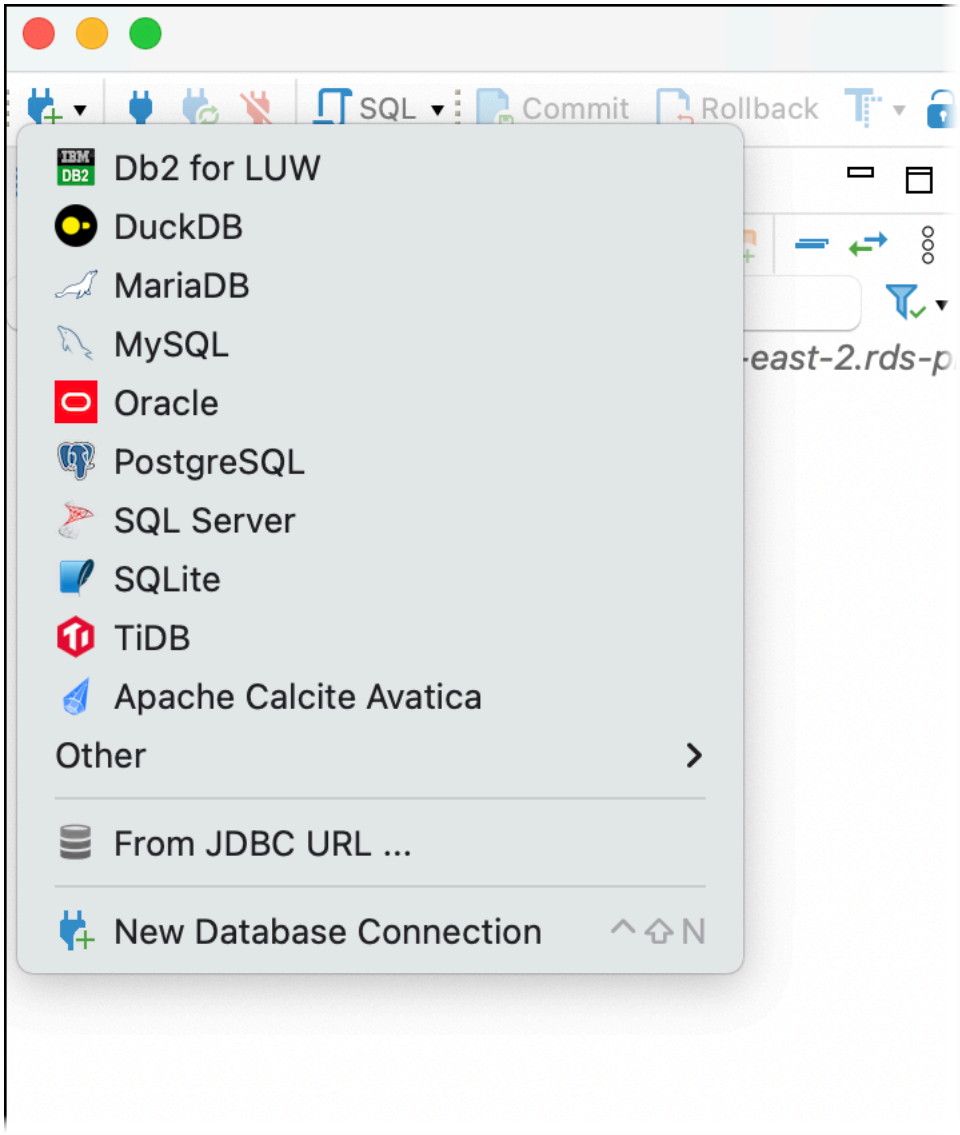
Conectar-se à instância de banco de dados do Amazon RDS para Db2 com o DBeaver

É possível usar ferramentas de terceiros, como o DBeaver, para se conectar a instâncias de banco de dados do Amazon RDS para Db2. Para baixar esse utilitário, consulte DBeaver Community.

Para se conectar à instância de banco de dados do RDS para Db2, você precisa do nome do DNS e do número da porta. Para ter informações sobre como descobri-los, consulte Encontrar o endpoint. Também é necessário saber o nome do banco de dados, o nome de usuário principal e a senha mestra que você definiu ao criar a instância de banco de dados do RDS para Db2. Para ter mais informações sobre como descobri-los, consulte Criar uma instância de banco de dados.

Como se conectar a uma instância de banco de dados do RDS para Db2 com DBeaver

1. Inicie DBeaver.
2. Selecione o ícone Nova conexão na barra de ferramentas e, depois, escolha Db2 para LUW.



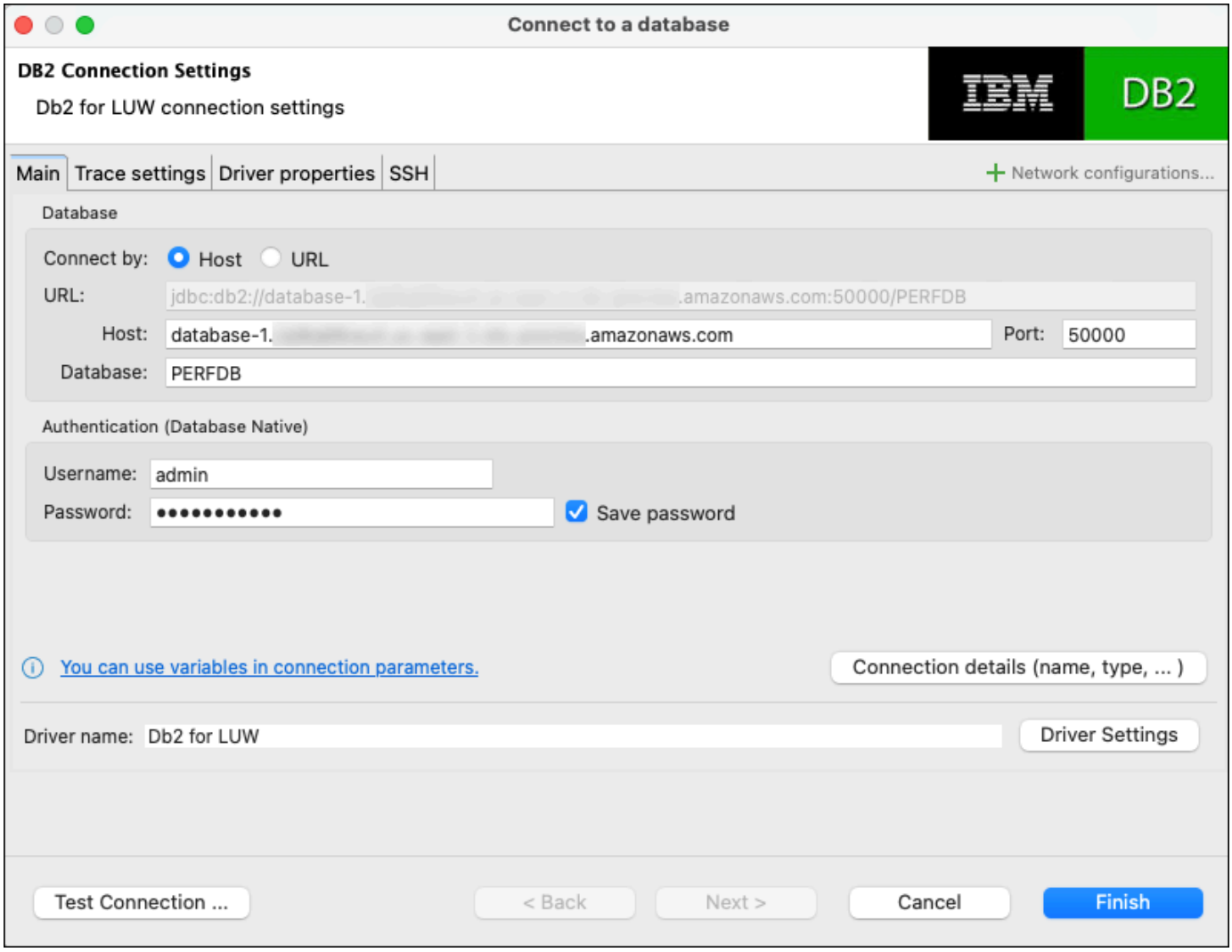
3. Na janela Conectar-se a um banco de dados, forneça informações da instância de banco de dados do RDS para Db2.

   a. Insira as seguintes informações:

   - Em Host, digite o nome do DNS da instância de banco de dados.
   - Em Porta, insira o número da porta da instância de banco de dados.
   - Em Banco de dados, insira o nome do banco de dados.
   - Em Username (Nome do usuário), digite o nome do administrador do banco de dados da instância de banco de dados.
   - Em Senha, digite a senha do administrador do banco de dados da instância de banco de dados.

b. Selecione Salvar senha.

c. Selecione Configurações do driver.



4. Na janela Editar driver, especifique propriedades de segurança adicionais.

a. Selecione a guia Propriedades do driver.

b. Adicione duas Propriedades do usuário.

i. Abra o menu de contexto (clique com o botão direito do mouse) e selecione Adicionar nova propriedade.

ii. Em Nome da propriedade, adicione encryptionAlgorithm e, depois, escolha OK.

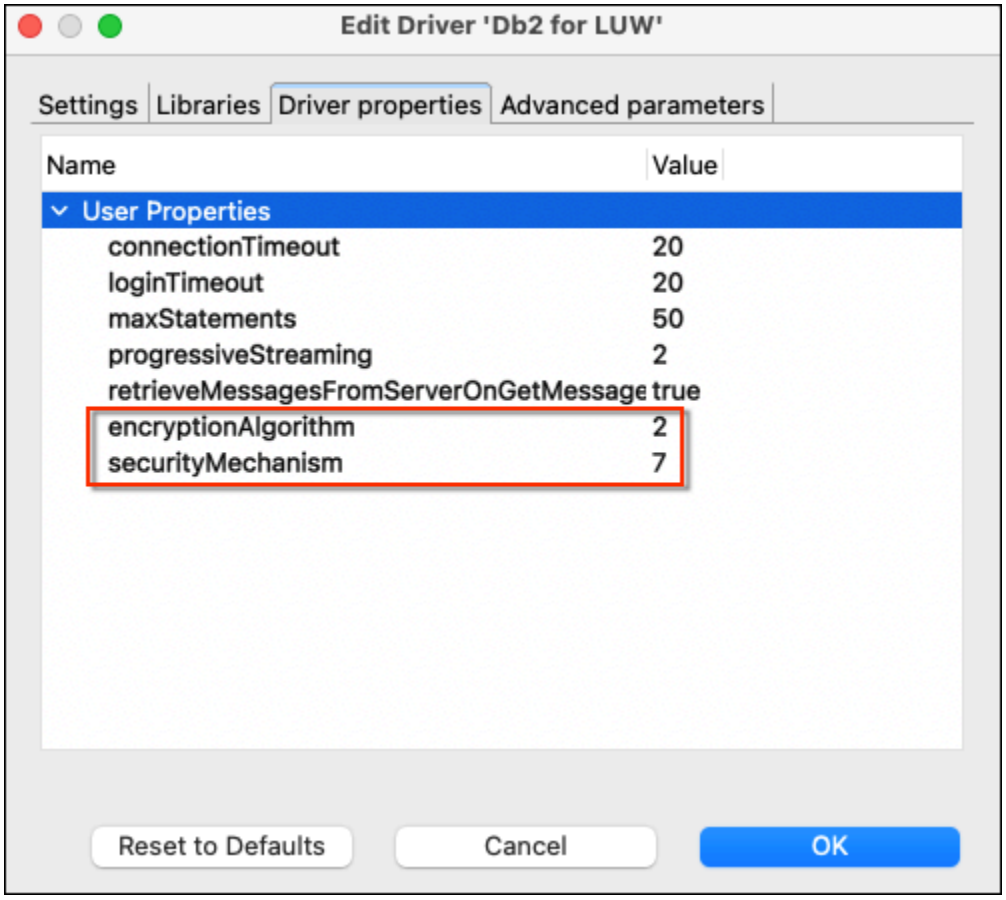
iii. Com a linha encryptionAlgorithm selecionada, escolha a coluna Valor e adicione 2.

iv. Abra o menu de contexto (clique com o botão direito do mouse) e selecione Adicionar nova propriedade.

v. Em Nome da propriedade, adicione securityMechanism e escolha OK.

vi. Com a linha securityMechanism selecionada, escolha a coluna Valor e adicione 7.

c. Escolha OK.

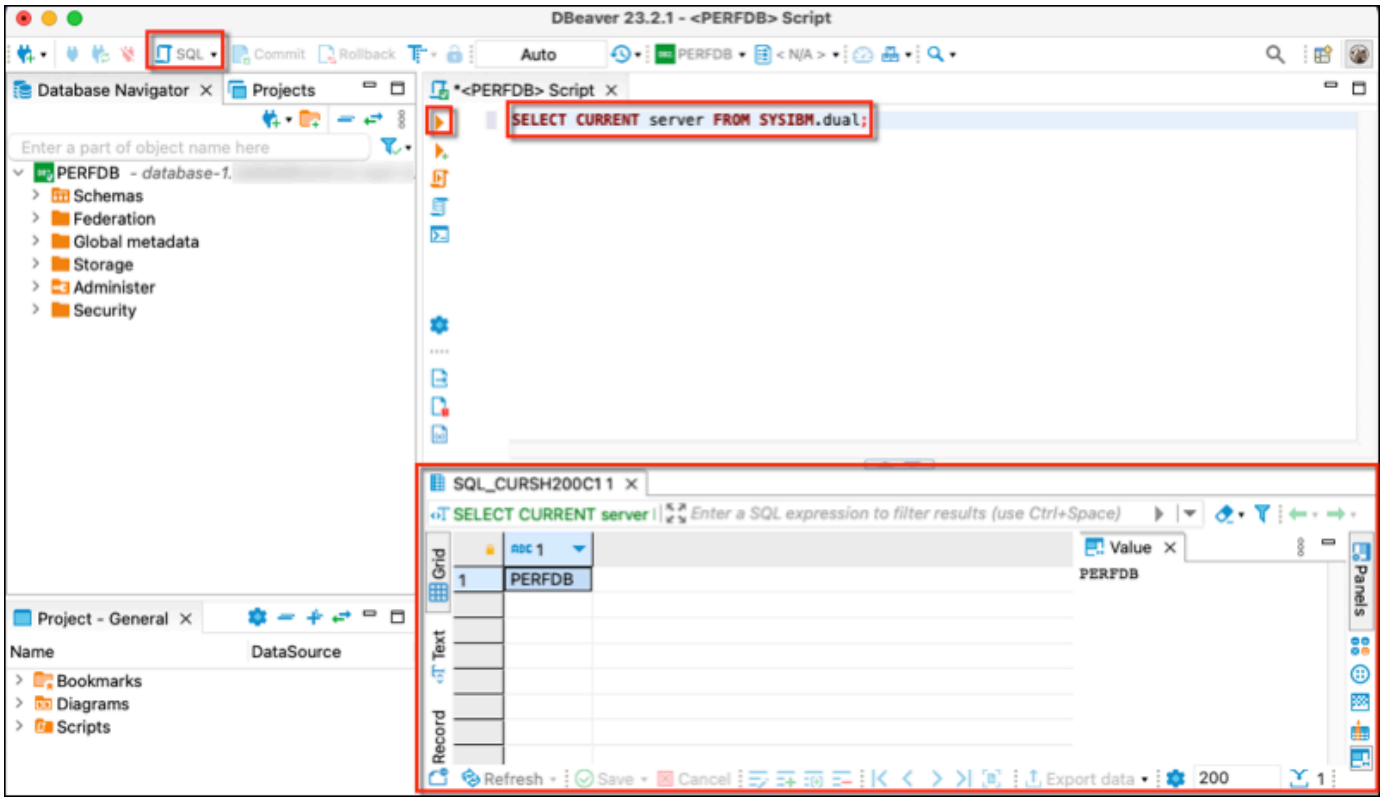


5. Na janela Conectar-se a um banco de dados, selecione Testar conexão. Se você não tiver um driver DB2 JBDC instalado no computador, o driver será baixado automaticamente.
6. Escolha OK.
7. Escolha Terminar.
8. Na guia Navegação no banco de dados, selecione o nome do banco de dados. Agora, é possível explorar objetos.

Agora, está tudo pronto para executar comandos SQL.

Como executar comandos SQL e visualizar os resultados

1. No menu superior, selecione SQL. Um painel de script SQL será aberto.
2. No painel Script, insira um comando SQL.
3. Para executar o comando, selecione o botão Executar consulta SQL.
4. No painel de resultados SQL, visualize os resultados das consultas SQL.



Conectar-se à instância de banco de dados do Amazon RDS para Db2 com o IBM Db2 Data Management Console

É possível se conectar à instância de banco de dados do Amazon RDS para Db2 com o IBM Db2 Data Management Console. O IBM Db2 Data Management Console pode administrar e monitorar várias instâncias de banco de dados do RDS para Db2.

> **Note**
>
> Você deve ter uma máquina Linux ou Windows do Amazon EC2 que esteja na mesma VPC e grupo de segurança da instância de banco de dados do RDS para Db2. A VPC e o grupo

de segurança controlam a conexão com a instância de banco de dados por meio da rede interna.

O IBM Db2 Data Management Console requer um banco de dados do Db2 do repositório para armazenar metadados e métricas de desempenho, mas não pode criar automaticamente um banco de dados de repositório para o RDS para Db2. Em vez disso, é necessário criar um banco de dados de repositório para monitorar uma ou mais instâncias de banco de dados do RDS para Db2. Depois, é possível instalar o IBM Db2 Data Management Console e conectar-se à instância de banco de dados do RDS para Db2 com o IBM Db2 Data Management Console.

Tópicos

- Etapa 1: criar um banco de dados de repositório para monitorar instâncias de banco de dados
- Etapa 2: instalar e configurar o IBM Db2 Data Management Console
- Etapa 3: configurar o banco de dados do repositório e se conectar a instâncias de banco de dados do RDS para Db2
- Utilizar o IBM Db2 Data Management Console

Etapa 1: criar um banco de dados de repositório para monitorar instâncias de banco de dados

É possível usar uma instância de banco de dados do RDS para Db2 de tamanho adequado como um repositório para o IBM Db2 Data Management Console para monitorar outras instâncias de banco de dados do RDS para Db2. No entanto, como o usuário administrador não tem a autoridade `SYSCTRL` para criar grupos de buffer e espaços de tabela, o uso da criação do repositório IBM Db2 Data Management Console para criar um banco de dados do repositório falha. Em vez disso, é necessário criar um banco de dados de repositório. Esse banco de dados de repositório monitora as instâncias de banco de dados do RDS para Db2.

É possível criar um banco de dados de repositório de duas maneiras diferentes. É possível criar um banco de dados do RDS para Db2 e criar manualmente um grupo de buffer, um espaço de tabela de usuário e um espaço de tabela temporário do sistema. Ou é possível criar uma instância separada do Amazon EC2 para hospedar um banco de dados de repositório do IBM Db2 Data Management Console.

Tópicos

- Criar manualmente um grupo de buffer, um espaço de tabela de usuário e um espaço de tabela temporário do sistema
- Criar uma instância do Amazon EC2 para hospedar um repositório do IBM Db2 Data Management Console

Criar manualmente um grupo de buffer, um espaço de tabela de usuário e um espaço de tabela temporário do sistema

Como criar um grupo de buffer, um espaço de tabela de usuário e um espaço de tabela temporário do sistema

1. Conecte-se ao banco de dados `rdsadmin`. No exemplo a seguir, substitua *`master_username`* e *`master_password`* por suas próprias informações.

```
db2 connect to rdsadmin user master_username using master_password
```

2. Crie um grupo de buffer pool para IBM Db2 Data Management Console. No exemplo a seguir, substitua *`database_name`* pelo nome do repositório que você criou para o IBM Db2 Data Management Console monitorar as instâncias de banco de dados do RDS para Db2.

```
db2 "call rdsadmin.create_bufferpool('database_name',
     'BP4CONSOLE', 1000, 'Y', 'Y', 32768)"
```

3. Crie um espaço de tabela do usuário para o IBM Db2 Data Management Console. No exemplo a seguir, substitua *`database_name`* pelo nome do repositório que você criou para o IBM Db2 Data Management Console monitorar as instâncias de banco de dados do RDS para Db2.

```
db2 "call rdsadmin.create_tablespace('database_name',
     'TS4CONSOLE', 'BP4CONSOLE', 32768)"
```

4. Crie um espaço de tabela temporário do sistema para o IBM Db2 Data Management Console. No exemplo a seguir, substitua *`database_name`* pelo nome do repositório que você criou para o IBM Db2 Data Management Console monitorar as instâncias de banco de dados do RDS para Db2.

```
db2 "call rdsadmin.create_tablespace('database_name',
    'TS4CONSOLE_TEMP', 'BP4CONSOLE', 32768, 0, 0, 'S')"
```

Agora está tudo pronto para instalar o IBM Db2 Data Management Console. Consulte mais informações sobre instalação e configuração em Etapa 2: instalar e configurar o IBM Db2 Data Management Console.

Criar uma instância do Amazon EC2 para hospedar um repositório do IBM Db2 Data Management Console

É possível criar uma instância separada do Amazon Elastic Compute Cloud (Amazon EC2) para hospedar um repositório do IBM Db2 Data Management Console. Consulte informações sobre como criar uma instância do Amazon EC2 em Tutorial: Comece a usar instâncias do Linux do Amazon EC2 no Guia do usuário do Amazon EC2.

Etapa 2: instalar e configurar o IBM Db2 Data Management Console

Depois de criar um grupo de buffer, um espaço de tabela do usuário e um espaço de tabela temporário do sistema, estará tudo pronto para instalar e configurar o IBM Db2 Data Management Console.

> **Important**
>
> Você deve ter uma máquina Linux ou Windows do Amazon EC2 que esteja na mesma VPC e grupo de segurança da instância de banco de dados do RDS para Db2. A VPC e o grupo de segurança controlam a conexão com a instância de banco de dados por meio da rede interna. Além disso, você já deve ter criado um banco de dados de repositório para o IBM Db2 Data Management Console.

Como instalar e configurar o IBM Db2 Data Management Console

1. Baixe o IBM Db2 Data Management Console do IBM Db2 Data Management Console versão 3.1x no site do Suporte IBM.
2. Instalar o IBM Db2 Data Management Console.
3. Abra o IBM Db2 Data Management Console e use o endereço IP da máquina do Amazon EC2 e o número da porta que você usou para a conexão HTTP ou HTTPS com a instância do Amazon EC2. Por exemplo, use `http://xx.xx.xx.xx:11080` ou `https://xx.xx.xx.xx.11081`. Substitua *xx.xx.xx.xx* pelo endereço IP da máquina do Amazon EC2. `11080` e `11081` são as portas padrão para conexões HTTP e HTTPS.

4. (Opcional) Se quiser usar a porta 80 ou 443 na instância do Amazon EC2, você poderá usar o Apache httpd ou um servidor HTTP Nginx para fazer o proxy da porta IBM Db2 Data Management Console para a porta 80 ou 443. Consulte mais informações em Apache HTTP Server Project e o site do nginx.

   Para permitir a conexão com o IBM Db2 Data Management Console, você deve editar as regras de entrada no grupo de segurança. Se você usa um proxy, altere a porta TCP/IP 80 ou 443 para redirecionar para as portas do IBM Db2 Data Management Console. Se você não estiver usando um proxy, altere a porta TCP/IP 80 ou 443 para as portas padrão 11080 (HTTP) ou 11081 (HTTPS).

Agora, é possível fazer login no IBM Db2 Data Management Console para configurar o banco de dados do repositório e estabelecer conexão com instâncias de banco de dados do RDS para Db2. Para ter mais informações, consulte Configurar o banco de dados do repositório e se conectar a instâncias de banco de dados.

Etapa 3: configurar o banco de dados do repositório e se conectar a instâncias de banco de dados do RDS para Db2

Quando você se conecta ao banco de dados do repositório pela primeira vez, o IBM Db2 Data Management Console configura o repositório automaticamente. Depois que o banco de dados do repositório estiver configurado, você poderá adicionar conexões de banco de dados ao IBM Db2 Data Management Console.

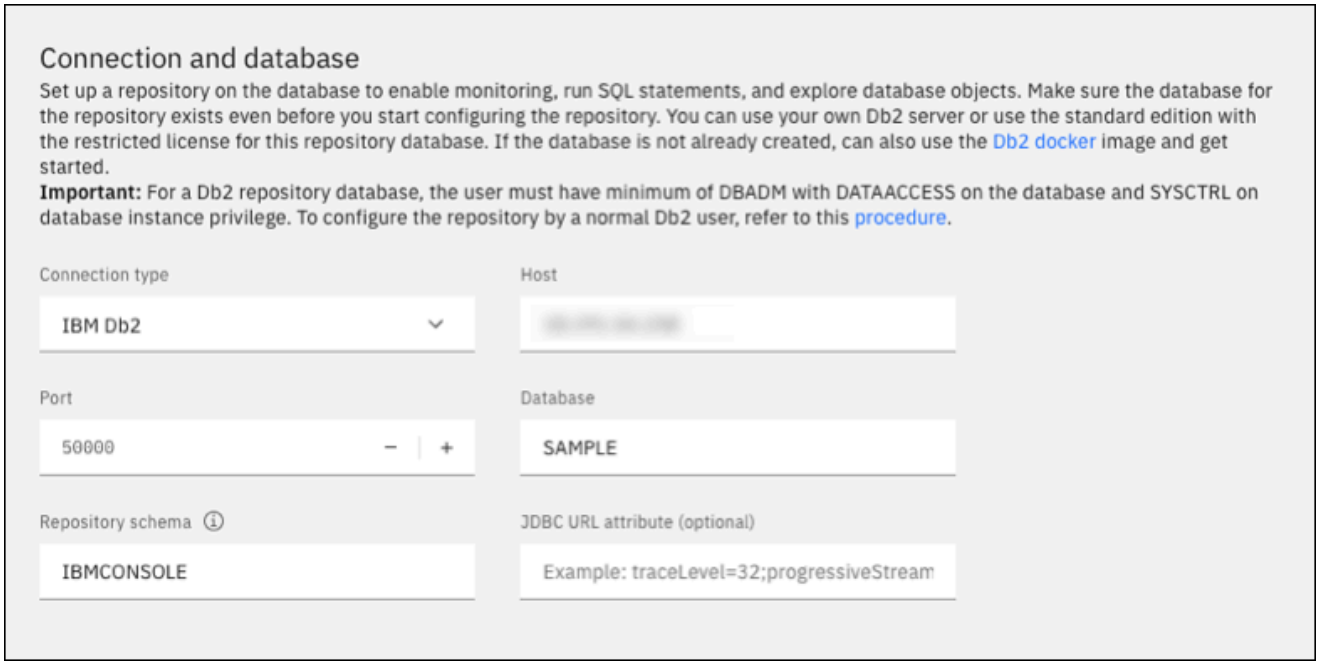
Para se conectar à instância de banco de dados do RDS para Db2, você precisa do nome do DNS e do número da porta. Para ter informações sobre como descobri-los, consulte Encontrar o endpoint. Também é necessário saber o nome do banco de dados, o nome de usuário principal e a senha mestra que você definiu ao criar a instância de banco de dados do RDS para Db2. Para ter mais informações sobre como descobri-los, consulte Criar uma instância de banco de dados. Se você estiver se conectando pela Internet, permita o tráfego para a porta do banco de dados. Para ter mais informações, consulte Criar uma instância de banco de dados.

Como se conectar a instâncias de banco de dados do RDS para Db2 com IBM Db2 Data Management Console

1. Faça login no IBM Db2 Data Management Console com as credenciais que você definiu durante a instalação.
2. Configure o repositório.

a. Na seção Conexão e banco de dados, insira as seguintes informações para a instância de banco de dados do RDS para Db2:

- Em Host, digite o nome do DNS da instância de banco de dados.
- Em Porta, insira o número da porta da instância de banco de dados.
- Em Banco de dados, insira o nome do banco de dados.



b. Na seção Segurança e credencial, insira as seguintes informações para a instância de banco de dados do RDS para Db2:

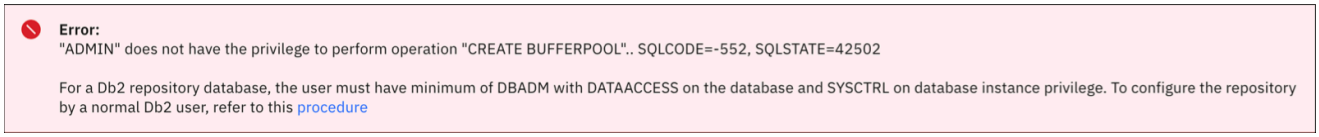
- Em Tipo de segurança, selecione Usuário e senha criptografados.
- Em Username (Nome do usuário), digite o nome do administrador do banco de dados da instância de banco de dados.
- Em Senha, digite a senha do administrador do banco de dados da instância de banco de dados.

c. Selecione Test connection (Testar conexão).

> **Note**
>
> Se a conexão não for bem-sucedida, confirme se a porta do banco de dados está aberta por meio das regras de entrada do grupo de segurança. Para ter mais

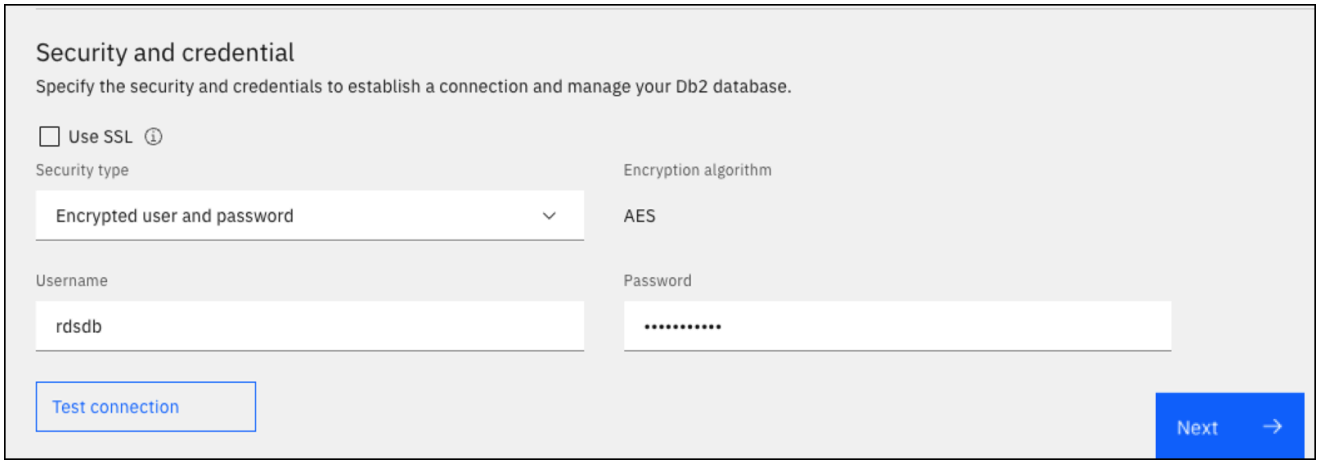
informações, consulte Considerações sobre grupos de segurança com o Amazon RDS para Db2.

Se você não criou manualmente um grupo de buffer, um espaço de tabela de usuário e um espaço de tabela temporário do sistema no RDS para Db2, talvez receba a seguinte mensagem de erro:



Você deve ter criado uma tabela de buffer, um espaço de tabela e objetos para um repositório do IBM Db2 Data Management Console para monitorar a instância de banco de dados do RDS para Db2. Ou é possível usar uma instância de banco de dados Db2 do Amazon EC2 para hospedar um repositório do IBM Db2 Data Management Console e monitorar a instância de banco de dados do RDS para Db2. Para ter mais informações, consulte Etapa 1: criar um banco de dados de repositório para monitorar instâncias de banco de dados.

d. Depois de testar a conexão com êxito, selecione Próximo.

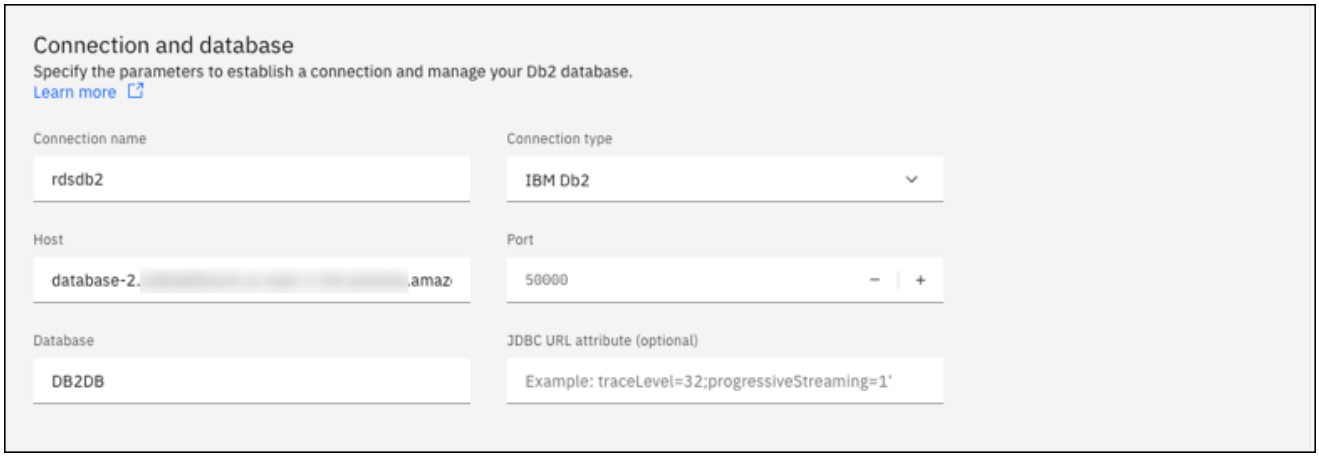


Se o IBM Db2 Data Management Console encontrar o grupo de buffer, o espaço de tabela do usuário e o espaço de tabela temporário do sistema na instância de banco de dados do RDS para Db2, o IBM Db2 Data Management Console configurará automaticamente o banco de dados do repositório. Se você usar a instância do Db2 na instância do Amazon EC2 como banco de dados do repositório, o IBM Db2 Data Management Console criará automaticamente o grupo buffer e outros objetos.

3. Na janela Definir aceitação do monitor de eventos de estatísticas, escolha Próximo.
4. (Opcional) Adicione uma nova conexão. Se você quiser usar uma instância de banco de dados do RDS para Db2 diferente para administração e monitoramento, adicione uma conexão a uma instância de banco de dados do RDS para Db2 que não seja do repositório.

   a. Na seção Conexão e banco de dados, insira as seguintes informações para a instância de banco de dados do RDS para Db2 para administração e monitoramento:

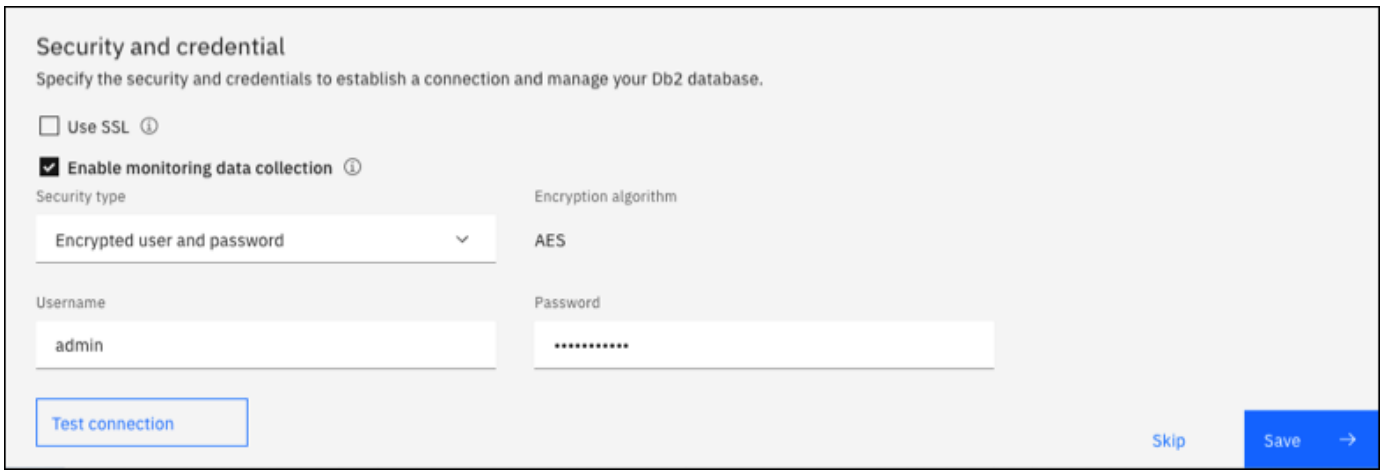
      - Em Nome da conexão, insira o identificador do banco de dados do Db2.
      - Em Host, digite o nome do DNS da instância de banco de dados.
      - Em Porta, insira o número da porta da instância de banco de dados.
      - Em Banco de dados, insira o nome do banco de dados.



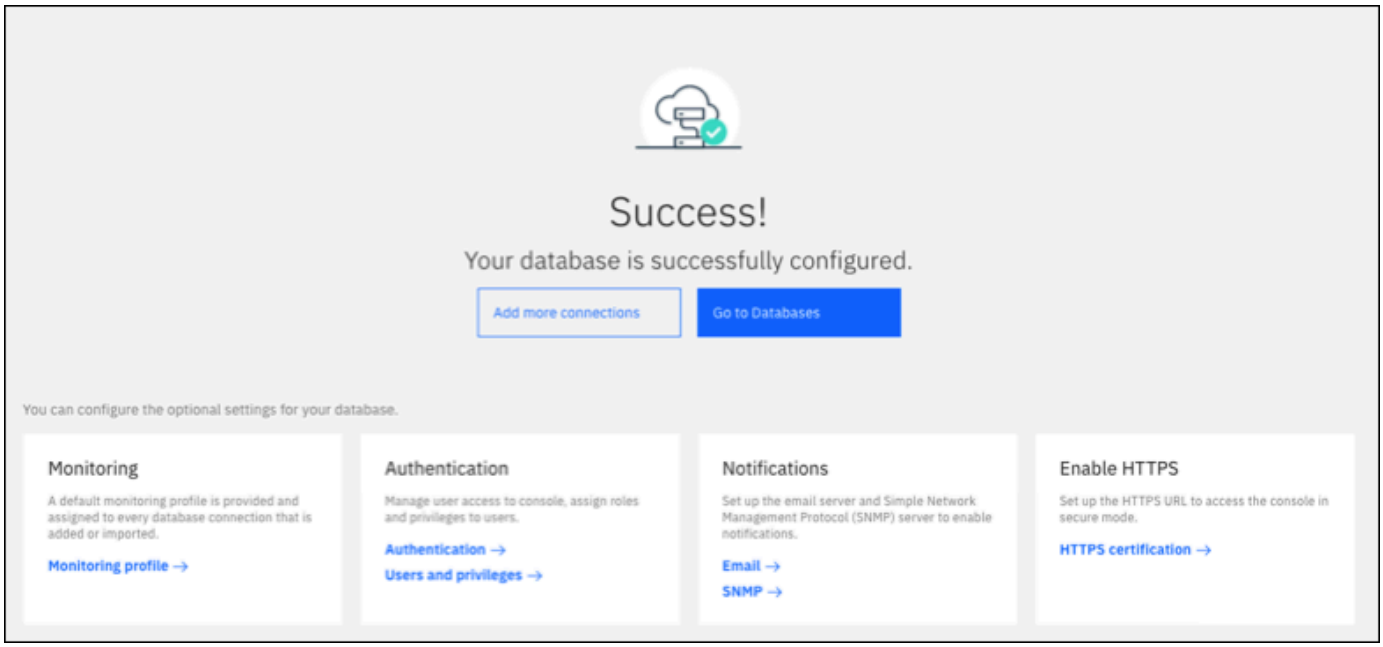
   b. Na seção Segurança e credencial , selecione Habilitar coleta de dados de monitoramento.
   c. Insira as seguintes informações da instância de banco de dados do RDS para Db2:

      - Em Username (Nome do usuário), digite o nome do administrador do banco de dados da instância de banco de dados.
      - Em Senha, digite a senha do administrador do banco de dados da instância de banco de dados.

   d. Selecione Test connection (Testar conexão).
   e. Depois de testar a conexão com êxito, selecione Salvar.



Depois que a conexão é adicionada, uma janela semelhante à seguinte é exibida. Essa janela indica que o banco de dados foi configurado com êxito.



5. Selecione Ir para bancos de dados. Uma janela Bancos de dados semelhante ao seguinte é exibida. Essa janela é um painel que mostra métricas, status e conexões.



Agora você pode começar a usar o IBM Db2 Data Management Console.

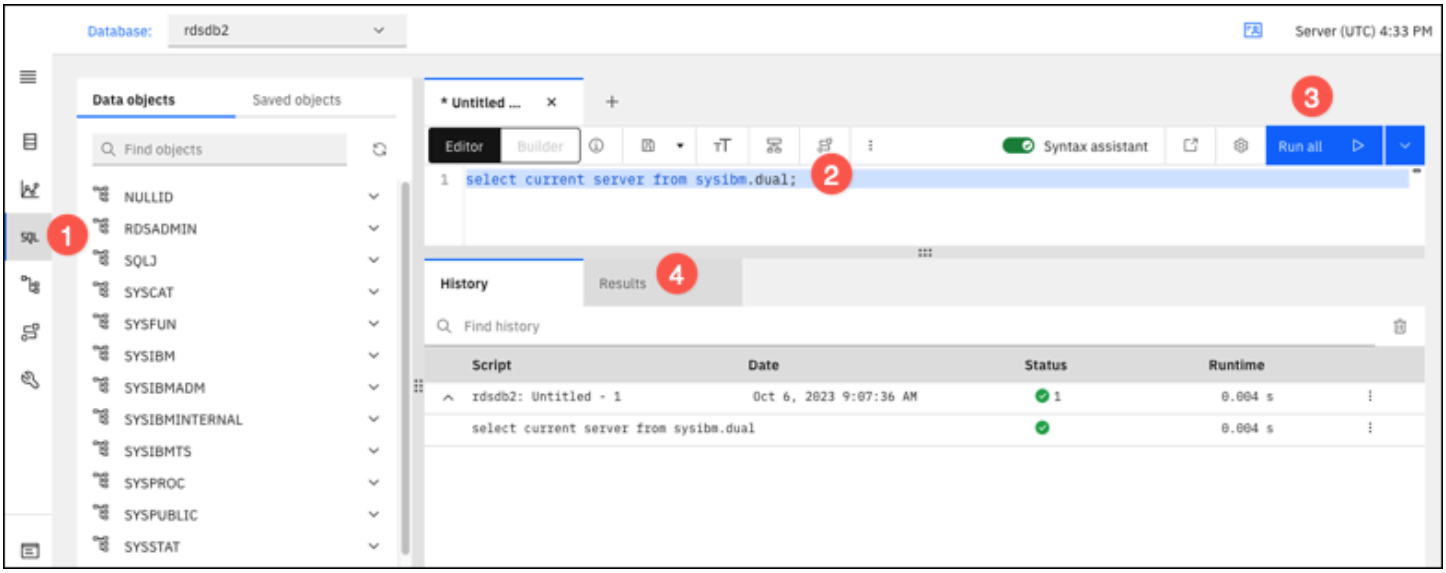
Utilizar o IBM Db2 Data Management Console

É possível usar o IBM Db2 Data Management Console para fazer os seguintes tipos de tarefa:

- Gerenciar várias instâncias de banco de dados do RDS para Db2.
- Executar comandos SQL.
- Examinear, criar ou alterar dados e objetos de banco de dados.
- Criar declarações `EXPLAIN PLAN` em SQL.
- Ajustar as consultas.

Como executar comandos SQL e visualizar os resultados

1. Na barra de navegação esquerda, selecione SQL.
2. Insira um comando SQL.
3. Selecione Executar tudo.
4. Para visualizar os resultados, selecione a guia Resultados.



Considerações sobre grupos de segurança com o Amazon RDS para Db2

Para se conectar à instância de banco de dados do Amazon RDS para Db2, ela deve estar associada a um grupo de segurança que contenha a configuração de rede e os endereços IP necessários. A instância de banco de dados do RDS para Db2 pode usar o grupo de segurança padrão. Se você atribuiu um grupo de segurança padrão não configurado ao criar a instância de banco de dados do RDS para Db2, o firewall impedirá conexões com a Internet. Para obter informações sobre como criar um novo security group, consulte Controlar acesso com grupos de segurança.

Depois de criar o novo security group, você modifica sua instância de banco de dados para associá-la a esse security group. Para obter mais informações, consulte Modificar uma instância de banco de dados do Amazon RDS.

Você pode aprimorar a segurança usando o SSL para criptografar conexões com sua a instância de banco de dados. Para ter mais informações, consulte Usar SSL/TLS com uma instância de banco de dados do Amazon RDS para Db2.

Proteger as conexões de instância de banco de dados do Amazon RDS para Db2

O Amazon RDS para Db2 é compatível com formas de melhorar a segurança da instância de banco de dados do RDS para Db2.

Tópicos

- Usar SSL/TLS com uma instância de banco de dados do Amazon RDS para Db2
- Usar a autenticação do Kerberos para o Amazon RDS para Db2

Usar SSL/TLS com uma instância de banco de dados do Amazon RDS para Db2

O SSL é um protocolo padrão do setor para proteger as conexões de rede entre o cliente e o servidor. Após o SSL versão 3.0, o nome mudou para TLS, mas ainda costuma ser chamado de SSL. O Amazon RDS é compatível com criptografia SSL para instâncias de banco de dados do Amazon RDS para Db2. Usando o SSL/TLS, é possível criptografar uma conexão entre o cliente de aplicação e a instância de banco de dados do RDS para Db2. A compatibilidade com SSL/TLS está disponível em todas as Regiões da AWS para RDS para Db2.

Para habilitar a criptografia SSL/TLS para uma instância de banco de dados do RDS para Db2, adicione a opção Db2 SSL ao grupo de parâmetros associado à instância de banco de dados. O Amazon RDS usa uma segunda porta, conforme exigido pelo Db2, para conexões SSL/TLS. Isso permite que comunicações de texto sem formatação e de texto com criptografia SSL ocorram ao mesmo tempo entre uma instância de banco de dados e um cliente Db2. Por exemplo, você pode usar a porta com comunicação de texto simples para se comunicar com outros recursos dentro de uma VPC enquanto usa a porta com comunicação criptografada em SSL para se comunicar com recursos fora da VPC.

Tópicos

- Criar uma conexão SSL/TLS
- Conectar-se ao servidor de banco de dados do Db2

Criar uma conexão SSL/TLS

Para criar uma conexão SSL/TLS, selecione uma autoridade de certificação (CA), baixe um pacote de certificados para todas as Regiões da AWS e adicione parâmetros a um grupo de parâmetros personalizado.

Etapa 1: Selecionar uma CA e baixar um certificado

Selecione uma autoridade de certificação (CA) e baixe um pacote de certificados para todas as Regiões da AWS. Para ter mais informações, consulte Usar SSL/TLS para criptografar uma conexão com uma instância ou um cluster de banco de dados.

Etapa 2: Atualizar os parâmetros em um grupo de parâmetros personalizado

> **Important**
>
> Se você estiver usando o modelo traga a sua própria licença (BYOL) para RDS para Db2, modifique o grupo de parâmetros personalizado criado para o IBM Customer ID e o IBM Site ID. Se você estiver usando um modelo de licenciamento diferente para o RDS para Db2, siga o procedimento para adicionar parâmetros a um grupo de parâmetros personalizado. Para ter mais informações, consulte Opções de licenciamento do Amazon RDS para Db2.

Não é possível modificar grupos de parâmetros padrão para instâncias de banco de dados do RDS para Db2. Portanto, é necessário criar um grupo de parâmetros personalizado, modificá-lo e associá-lo às instâncias de banco de dados do RDS para Db2. Para obter informações sobre grupos de parâmetros, consulte Grupos de parâmetros de banco de dados para instâncias de banco de dados do Amazon RDS.

Use as configurações de parâmetros na tabela a seguir.

Parâmetro	Valor
`DB2COMM`	`TCPIP,SSL` ou `SSL`
`SSL_SVCENAME`	`<any port number except the number used for the non-SSL port>`

Como atualizar os parâmetros em um grupo de parâmetros personalizado

1. Crie um grupo de parâmetros personalizado executando o comando create-db-parameter-group.

   Inclua as seguintes opções necessárias:

   - `--db-parameter-group-name`: um nome para o grupo de parâmetros que você está criando.
   - `--db-parameter-group-family`: a edição do mecanismo do Db2 e a versão principal. Valores válidos: `db2-se-11-5`, `db2-ae-11.5`.
   - `--description`: a descrição desse grupo de parâmetros.

   Para obter mais informações sobre como criar um parameter group de banco de dados, consulte Criar um grupo de parâmetros de banco de dados no Amazon RDS.
2. Modifique os parâmetros no grupo de parâmetros personalizado que você criou executando o comando modify-db-parameter-group.

   Inclua as seguintes opções necessárias:

   - `--db-parameter-group-name`: o nome do grupo de parâmetros criado.
   - `--parameters`: uma matriz de nomes de parâmetros, valores e os métodos de aplicação da atualização de parâmetros.

   Para ter mais informações sobre como modificar um grupo de parâmetros, consulte Modificar parâmetros em um grupo de parâmetros de banco de dados no Amazon RDS.
3. Associe o grupo de parâmetros à sua instância de banco de dados do RDS para Db2. Para ter mais informações, consulte Associar um grupo de parâmetros de banco de dados a uma instância de banco de dados no Amazon RDS.

Conectar-se ao servidor de banco de dados do Db2

As instruções para se conectar ao servidor de banco de dados do Db2 são específicas do idioma.

Java

Como se conectar ao servidor de banco de dados do Db2 usando Java

1. Baixe o driver JDBC. Para ter mais informações, consulte DB2 JDBC Driver Versions and Downloads na documentação do IBM Support.
2. Crie um arquivo de script de shell com o conteúdo a seguir. Esse script adiciona todos os certificados do pacote a um Java KeyStore.

> **Important**
>
> Verifique se há uma `keytool` no caminho do script para que o script possa localizá-la. Se você usa um cliente Db2, pode localizar a `keytool` em `~sqlib/java/jdk64/jre/bin`.

```
#!/bin/bash
PEM_FILE=$1
PASSWORD=$2
KEYSTORE=$3
# number of certs in the PEM file
CERTS=$(grep 'END CERTIFICATE' $PEM_FILE| wc -l)
for N in $(seq 0 $(($CERTS - 1))); do
    ALIAS="${PEM_FILE%.*}-$N"
    cat $PEM_FILE |
    awk "n==$N { print }; /END CERTIFICATE/ { n++ }" |
    keytool -noprompt -import -trustcacerts -alias $ALIAS -keystore $KEYSTORE -
storepass $PASSWORD
done
```

3. Para executar o script de shell e importar o arquivo PEM com o pacote de certificados em um Java KeyStore, execute o comando a seguir. Substitua *`shell_file_name.sh`* pelo nome do arquivo de script de shell, e *`password`* pela senha do Java KeyStore.

```
./shell_file_name.sh global-bundle.pem password truststore.jks
```

4. Para se conectar ao servidor Db2, execute o comando a seguir. Substitua os seguintes espaços reservados no exemplo pelas informações da instância de banco de dados do RDS para Db2.

- *ip_address*: o endereço IP do endpoint de instância de banco de dados.
- *port*: o número da porta para a conexão SSL. Pode ser qualquer número de porta, exceto o número usado para a porta não SSL.
- *database_name*: o nome do banco de dados na instância de banco de dados.
- *master_username*: o nome de usuário principal da sua instância de banco de dados.
- *master_password*: a senha mestra da instância de banco de dados.

```
export trustStorePassword=MyPassword
java -cp ~/dsdriver/jdbc_sqlj_driver/linuxamd64/db2jcc4.jar \
com.ibm.db2.jcc.DB2Jcc -url \
"jdbc:db2://ip_address:port/database_name:\
sslConnection=true;sslTrustStoreLocation=\
~/truststore.jks;\
sslTrustStorePassword=${trustStorePassword};\
sslVersion=TLSv1.2;\
encryptionAlgorithm=2;\
securityMechanism=7;" \
-user master_username -password master_password
```

Node.js

Como se conectar ao servidor de banco de dados do Db2 usando Node.js

1. Instale o driver node-ibm_db. Para ter mais informações, consulte Installing the node-ibm_db driver on Linux and UNIXsystems na documentação do IBM Db2.
2. Crie um arquivo JavaScript com base no conteúdo a seguir. Substitua os seguintes espaços reservados no exemplo pelas informações da instância de banco de dados do RDS para Db2.

   - *ip_address*: o endereço IP do endpoint de instância de banco de dados.
   - *master_username*: o nome de usuário principal da sua instância de banco de dados.
   - *master_password*: a senha mestra da instância de banco de dados.
   - *database_name*: o nome do banco de dados na instância de banco de dados.
   - *port*: o número da porta para a conexão SSL. Pode ser qualquer número de porta, exceto o número usado para a porta não SSL.

```
var ibmdb = require("ibm_db");
const hostname = "ip_address";
const username = "master_username";
const password = "master_password";
const database = "database_name";
const port = "port";
const certPath = "/root/qa-bundle.pem";
ibmdb.open("DRIVER={DB2};DATABASE=" + database + ";HOSTNAME=" +
 hostname + ";UID=" + username + ";PWD=" + password + ";PORT=" + port +
 ";PROTOCOL=TCPIP;SECURITY=SSL;SSLServerCertificate=" + certPath + ";", function
 (err, conn){
 if (err) return console.log(err);
 conn.close(function () {
 console.log('done');
 });
});
```

3. Para alterar o arquivo JavaScript, execute o comando a seguir.

```
node ssl-test.js
```

Python

Como se conectar ao servidor de banco de dados do Db2 usando Python

1. Crie um arquivo Python com o conteúdo a seguir. Substitua os seguintes espaços reservados no exemplo pelas informações da instância de banco de dados do RDS para Db2.

   - *port*: o número da porta para a conexão SSL. Pode ser qualquer número de porta, exceto o número usado para a porta não SSL.
   - *master_username*: o nome de usuário principal da sua instância de banco de dados.
   - *master_password*: a senha mestra da instância de banco de dados.
   - *database_name*: o nome do banco de dados na instância de banco de dados.
   - *ip_address*: o endereço IP do endpoint de instância de banco de dados.

```
import click
import ibm_db
```

```
import sys

port = port;
master_user_id = "master_username" # Master id used to create your DB instance
master_password = "master_password" # Master password used to create your DB
 instance
db_name = "database_name" # If not given "db-name'
vpc_customer_private_ip = "ip_address" # Hosts end points - Customer private IP
 Addressicert_path = "/root/ssl/global-bundle.pem" # cert path

@click.command()
@click.option("--path", help="certificate path")
def db2_connect(path):

    try:
        conn =
 ibm_db.connect(f"DATABASE={db_name};HOSTNAME={vpc_customer_private_ip};PORT={port};

 PROTOCOL=TCPIP;UID={master_user_id};PWD={master_password};SECURITY=ssl;SSLServerCertifi
 "", "")
        try:
            ibm_db.exec_immediate(conn, 'create table tablename (a int);')
            print("Query executed successfully")
        except Exception as e:
            print(e)
        finally:
            ibm_db.close(conn)
            sys.exit(1)
    except Exception as ex:
        print("Trying to connect...")

if __name__ == "__main__":
    db2_connect()
```

2. Crie o script de shell a seguir, que executa o arquivo Python criado. Substitua *python_file_name.py* pelo nome do arquivo de script Python.

```
#!/bin/bash
PEM_FILE=$1
# number of certs in the PEM file
CERTS=$(grep 'END CERTIFICATE' $PEM_FILE| wc -l)

for N in $(seq 0 $(($CERTS - 1))); do
```

```
    ALIAS="${PEM_FILE%.*}-$N"
    cert=`cat $PEM_FILE | awk "n==$N { print }; /END CERTIFICATE/ { n++ }"`
    cat $PEM_FILE | awk "n==$N { print }; /END CERTIFICATE/ { n++ }" >
 $ALIAS.pem
    python3 <python_file_name.py> --path $ALIAS.pem
    output=`echo $?`
    if [ $output == 1 ]; then
        break
    fi
done
```

3. Para importar o arquivo PEM com o pacote de certificados e executar o script de shell, execute o comando a seguir. Substitua *shell_file_name.sh* pelo nome do arquivo de script de shell.

```
./shell_file_name.sh global-bundle.pem
```

Usar a autenticação do Kerberos para o Amazon RDS para Db2

É possível usar a autenticação Kerberos para autenticar usuários quando se conectam à instância de banco de dados do Amazon RDS para Db2. A instância de banco de dados funciona com AWS Directory Service for Microsoft Active Directory (AWS Managed Microsoft AD) para habilitar a autenticação Kerberos. Quando os usuários são autenticados com uma instância de banco de dados do RDS para Db2 unida ao domínio confiável, as solicitações de autenticação são encaminhadas ao diretório criado com o AWS Directory Service. Para ter mais informações, consulte O que é AWS Directory Service? no Guia de administração do AWS Directory Service.

Primeiro, crie um diretório AWS Managed Microsoft AD para armazenar credenciais de usuário. Depois, adicione o domínio e outras informações do diretório AWS Managed Microsoft AD à instância de banco de dados do RDS para Db2. Quando os usuários são autenticados com a instância de banco de dados do RDS para Db2, as solicitações de autenticação são encaminhadas ao diretório AWS Managed Microsoft AD.

Manter todas as suas credenciais no mesmo diretório pode economizar tempo e esforço. Com essa abordagem, você tem um lugar centralizado para armazenar e gerenciar credenciais para várias instâncias de banco de dados. O uso de um diretório também pode melhorar o perfil de segurança geral.

Consulte informações sobre a autenticação Kerberos nos tópicos a seguir.

Tópicos

- Configurar a autenticação Kerberos para instâncias de banco de dados do Amazon RDS para Db2
- Conectar-se ao Amazon RDS para Db2 com a autenticação Kerberos

Disponibilidade de regiões e versões

A disponibilidade e a compatibilidade de recursos variam entre versões específicas de cada mecanismo de banco de dados e entre Regiões da AWS. Para ter mais informações sobre a disponibilidade de versões e regiões do RDS para Db2 com a autenticação Kerberos, consulte Regiões e mecanismos de banco de dados compatíveis com a autenticação Kerberos no Amazon RDS.

> **Note**
>
> A autenticação Kerberos não é compatível com classes de instância de banco de dados obsoletas para instâncias de banco de dados do RDS para Db2. Para obter mais informações, consulte Classes de instância do Amazon RDS para Db2.

Visão geral da autenticação Kerberos para instâncias de banco de dados do RDS para Db2

Para configurar a autenticação Kerberos para uma instância de banco de dados do RDS para Db2, conclua as seguintes etapas gerais, descritas em mais detalhes posteriormente:

1. Use AWS Managed Microsoft AD para criar um diretório do AWS Managed Microsoft AD. É possível usar o AWS Management Console, a AWS Command Line Interface (AWS CLI) ou o AWS Directory Service para criar o diretório. Para ter mais informações, consulte Create your AWS Managed Microsoft AD directory no Guia de administração do AWS Directory Service.
2. Crie uma função do AWS Identity and Access Management (IAM) que use a politica gerenciada `AmazonRDSDirectoryServiceAccess` do IAM. O perfil do IAM permite que o Amazon RDS faça chamadas para o diretório.

   Para o perfil do IAM permitir o acesso, o endpoint do AWS Security Token Service (AWS STS) deve estar ativado na Região da AWS correta da Conta da AWS. Os endpoints do AWS STS são ativos por padrão em todas as Regiões da AWS e é possível usá-los sem precisar tomar medidas

adicionais. Para ter mais informações, consulte Ativar e desativar o AWS STS em uma Região da AWS no Guia do usuário do IAM.

3. Crie ou modifique uma instância de banco de dados do RDS para Db2 usando o AWS Management Console, a AWS CLI ou a API do RDS com um dos seguintes métodos:
   - Crie uma instância de banco de dados do RDS para Db2 usando o console, o comando create-db-instance ou a operação da API CreateDBInstance. Para instruções, consulte Criar uma instância de banco de dados do Amazon RDS.
   - Modifique uma instância de banco de dados do RDS para Db2 existente usando o console, o comando modify-db-instance ou a operação da API ModifyDBInstance. Para instruções, consulte Modificar uma instância de banco de dados do Amazon RDS.
   - Restaure uma instância de banco de dados do RDS para Db2 de um snapshot de banco de dados usando o console, o comando restore-db-instance-from-db-snapshot ou a operação da API RestoreDBInstanceFromDBSnapshot. Para instruções, consulte Restaurar uma instância de banco de dados.
   - Restaure uma instância de banco de dados do RDS para Db2 para um ponto no tempo usando o console, o comando restore-db-instance-to-point-in-time ou a operação da API RestoreDBInstanceToPointInTime. Para instruções, consulte Restaurar uma instância de banco de dados para um momento especificado no Amazon RDS.

   É possível localizar a instância de banco de dados na mesma Amazon Virtual Private Cloud (VPC) do diretório ou em uma Conta da AWS ou uma VPC diferente. Ao criar ou modificar a instância de banco de dados do RDS para Db2, realize as seguintes tarefas:
   - Forneça o identificador de domínio (identificador `d-*`) que foi gerado quando você criou seu diretório.
   - Forneça o nome do perfil do IAM criado.
   - Verifique se o grupo de segurança da instância de banco de dados pode receber o tráfego de entrada do grupo de segurança do diretório.
4. Configure o cliente do Db2 e verifique se o tráfego pode fluir entre o host do cliente e o AWS Directory Service para as seguintes portas:
   - Porta TCP/UDP 53: DNS
   - TCP 88: autenticação Kerberos
   - TCP 389: LDAP
   - TCP 464: autenticação Kerberos

Gerenciamento de uma instância de banco de dados em um domínio

É possível usar o AWS Management Console, a AWS CLI ou a API do RDS para gerenciar a instância de banco de dados e a respectiva relação no Microsoft Active Directory. Por exemplo, é possível associar um Active Directory para habilitar a autenticação Kerberos. Também é possível remover a associação de um Active Directory para desabilitar a autenticação Kerberos. Também é possível mover uma instância de banco de dados para ser autenticada externamente por um Microsoft Active Directory para outro.

Por exemplo, executando o comando da CLI modify-db-instance, é possível realizar as seguintes ações:

- Tente habilitar novamente a autenticação Kerberos para uma associação com falha, especificando o ID do diretório atual da associação para a `--domain` opção.
- Desabilite a autenticação Kerberos em uma instância de banco de dados especificando `none` para a opção `--domain`.
- Mova uma instância de banco de dados de um domínio para outro especificando o identificador do novo domínio para a opção `--domain`.

Compreensão da associação de domínio

Depois que você cria ou modifica a instância de banco de dados, ela se torna membro do domínio. É possível visualizar o status da associação do domínio no console ou executando o comando describe-db-instances. O status da instância de banco de dados pode ser um dos seguintes:

- `kerberos-enabled`: a instância de banco de dados tem a autenticação Kerberos habilitada.
- `enabling-kerberos`: a AWS está no processo de habilitar a autenticação Kerberos nessa instância de bancos de dados.
- `pending-enable-kerberos`: a habilitação da autenticação Kerberos está pendente nessa instância de banco de dados.
- `pending-maintenance-enable-kerberos`: a AWS tentará habilitar a autenticação Kerberos na instância de bancos de dados durante a próxima janela de manutenção programada.
- `pending-disable-kerberos`: a desativação da autenticação Kerberos está pendente nessa instância de banco de dados.
- `pending-maintenance-disable-kerberos`: a AWS tentará desabilitar a autenticação Kerberos na instância de banco de dados durante a próxima janela de manutenção programada.

- `enable-kerberos-failed`: um problema de configuração impediu que AWS habilitasse a autenticação Kerberos na instância de banco de dados. Corrija o problema de configuração antes de emitir o comando novamente para modificar a instância de banco de dados.
- `disabling-kerberos`: a AWS está no processo de desabilitar a autenticação Kerberos nessa instância de bancos de dados.

Uma solicitação para habilitar a autenticação Kerberos pode falhar por conta de um problema de conectividade de rede ou de um perfil do IAM incorreto. Em alguns casos, a tentativa de habilitar a autenticação Kerberos pode falhar quando você cria ou modifica uma instância de banco de dados. Nesse caso, verifique se você está usando o perfil do IAM correto e modifique a instância de banco de dados para ingressar no domínio.

Configurar a autenticação Kerberos para instâncias de banco de dados do Amazon RDS para Db2

Você vai usar AWS Directory Service for Microsoft Active Directory (AWS Managed Microsoft AD) para configurar a autenticação Kerberos para uma instância de banco de dados do RDS para Db2 Para configurar a autenticação Kerberos, siga estas etapas:

Tópicos

- Etapa 1: Criar um diretório usando o AWS Managed Microsoft AD
- Etapa 2: Criar um perfil do IAM para o Amazon RDS acessar o AWS Directory Service
- Etapa 3: Criar e configurar usuários
- Etapa 4: Criar um grupo de administradores do RDS para Db2 no AWS Managed Microsoft AD
- Etapa 5: Criar ou modificar uma instância de banco de dados do RDS para Db2
- Etapa 6: Configurar um cliente Db2

Etapa 1: Criar um diretório usando o AWS Managed Microsoft AD

O AWS Directory Service cria um Active Directory totalmente gerenciado na Nuvem AWS. Quando você cria um diretório AWS Managed Microsoft AD, o AWS Directory Service cria dois controladores de domínio e servidores DNS para você. Os servidores do diretório são criados em sub-redes diferentes em uma VPC. Essa redundância ajuda a garantir que o diretório permaneça acessível mesmo se ocorrer uma falha.

Ao criar um diretório do AWS Managed Microsoft AD, o AWS Directory Service executa as seguintes tarefas em seu nome:

- Configura um Active Directory na VPC.
- Cria uma conta de administrador do diretório com o nome de usuário `Admin` e a senha especificada. Use essa conta para gerenciar seu diretório.

  > **Important**
  >
  > Certifique-se de salvar essa senha. O AWS Directory Service não armazena essa senha e não é possível recuperá-la ou redefini-la.

- Cria um grupo de segurança para os controladores do diretório. O grupo de segurança deve permitir a comunicação com a instância de banco de dados do RDS para Db2.

Ao iniciar o AWS Directory Service for Microsoft Active Directory, a AWS cria uma unidade organizacional (UO) que contém todos os objetos do diretório. Essa UO, que tem o nome de NetBIOS que você digitou ao criar o diretório, está localizada na raiz do domínio. A raiz do domínio é controlada e de propriedade da AWS.

A conta `Admin`, que foi criada com o diretório AWS Managed Microsoft AD, tem permissões para as atividades administrativas mais comuns da UO:

- Crie, atualize ou exclua usuários.
- Adicione recursos ao domínio, como servidores de arquivos ou de impressão, e atribua permissões para esses recursos aos usuários na UO.
- Criar OUs adicionais e contêineres.
- Delegar autoridade.
- Restaure objetos excluídos da lixeira do Active Directory.
- Execute os módulos Active Directory e Domain Name Service (DNS) para Windows PowerShell noAWS Directory Service.

A conta `Admin` também possui direitos para executar as seguintes atividades de domínio:

- Gerenciar configurações de DNS (adicionar, remover ou atualizar registros, zonas e encaminhadores).

- Visualizar logs de eventos de DNS.
- Visualizar logs de eventos de segurança.

Como criar um diretório com AWS Managed Microsoft AD

1. Faça login no AWS Management Console e abra o console do AWS Directory Service em https://console.aws.amazon.com/directoryservicev2/.
2. Selecione Configurar diretório.
3. Selecione AWS Managed Microsoft AD. O AWS Managed Microsoft AD é a única opção atualmente aceita para uso com o Amazon RDS.
4. Escolha Próximo.
5. Na página Enter directory information (Inserir informações do diretório), forneça as seguintes informações:

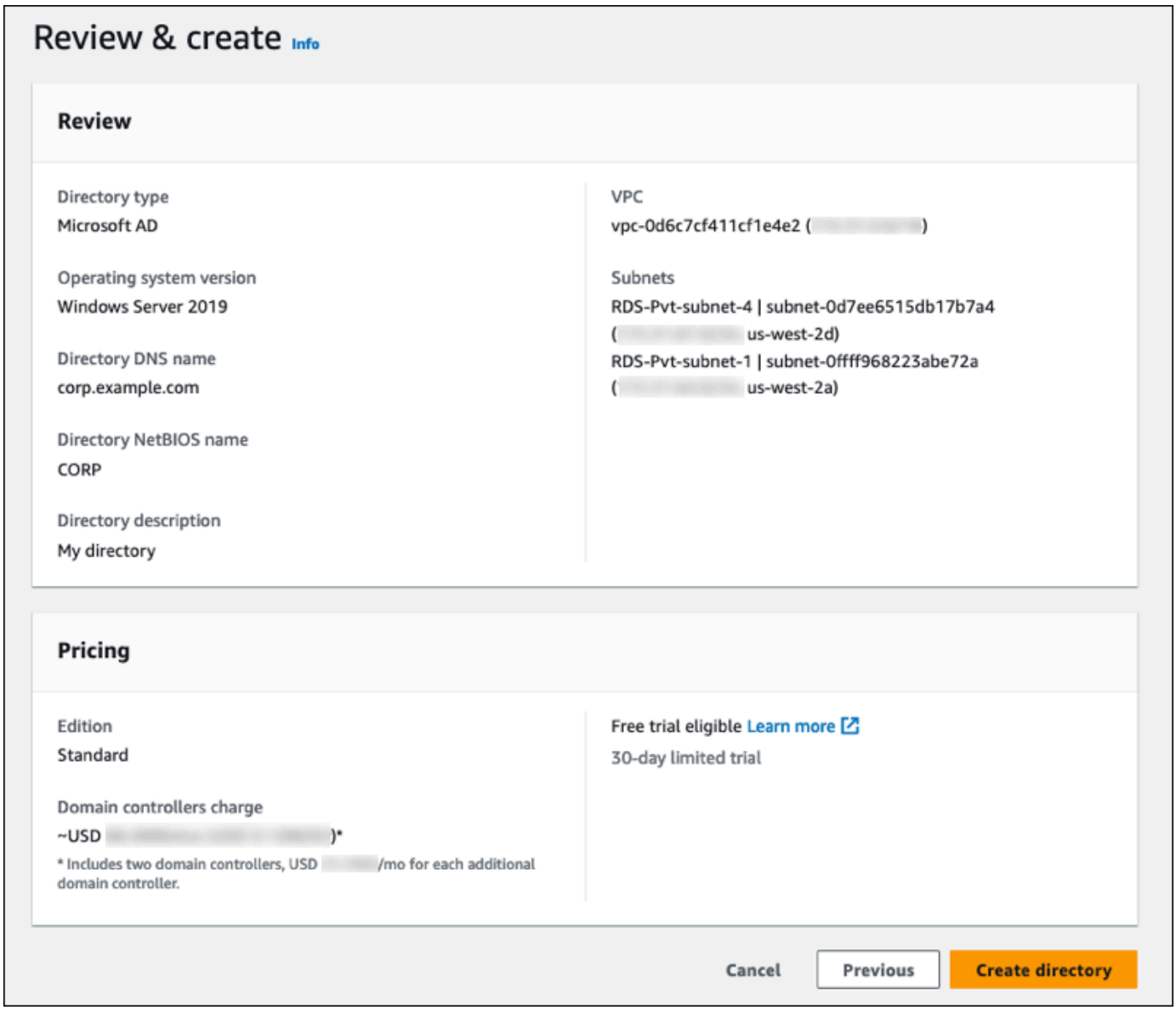
   - Edição: selecione a edição que atenda aos requisitos.
   - Nome do DNS do diretório : o nome totalmente qualificado do diretório, como `corp.example.com`.
   - Nome de NetBIOS do diretório : um nome curto opcional do diretório, como `CORP`.
   - Descrição do diretório: uma descrição opcional do diretório.
   - Senha do administrador : a senha do administrador do diretório. O processo de criação do diretório cria uma conta de administrador com o nome de usuário `Admin` e essa senha.

     A senha do administrador do diretório não pode incluir a palavra "admin". A senha diferencia letras maiúsculas de minúsculas e deve ter entre 8 e 64 caracteres. Ela também precisa conter pelo menos um caractere de três das quatro categorias a seguir:

     - Letras minúsculas (a–z)
     - Letras maiúsculas (A–Z)
     - Números (0–9)
     - Caracteres não alfanuméricos (~!@#$%^&*_-+=`|\(){}[]:;"'<>,.?/)
     - Confirmar senha: digite a senha do administrador novamente.

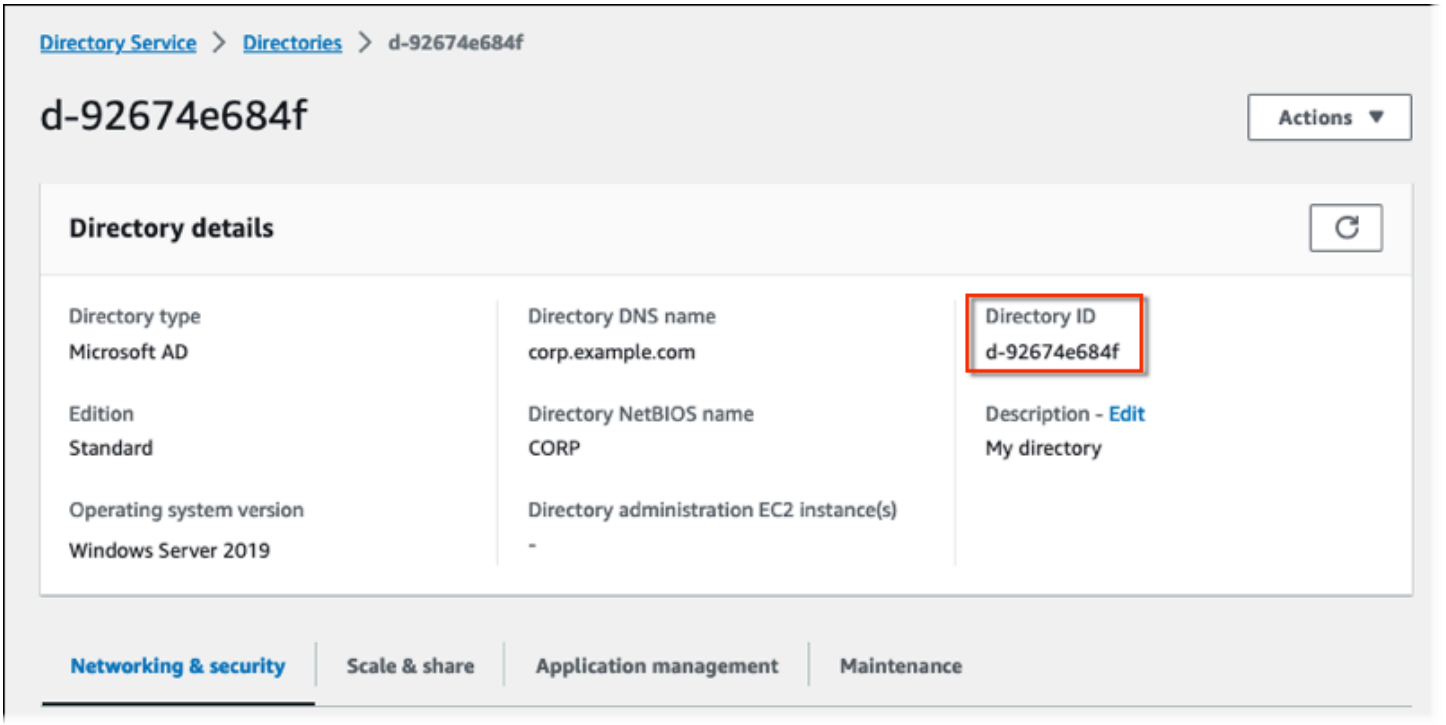
> **Important**
>
> Salve essa senha. O AWS Directory Service não armazena essa senha e não é possível recuperá-la ou redefini-la.

6. Escolha Next (Próximo).
7. Na página Choose VPC and subnets (Selecionar VPC e sub-redes), forneça as seguintes informações:
   - VPC: selecione a VPC do diretório. É possível criar a instância de banco de dados do RDS para Db2 nessa mesma VPC ou em outra VPC.
   - Sub-redes: selecione as sub-redes para os servidores do diretório. As duas sub-redes deve estar em diferentes zonas de disponibilidade.
8. Escolha Next (Próximo).
9. Analise as informações do diretório. Se alterações forem necessárias, escolha Previous (Anterior) e faça as alterações. Quando as informações estiverem corretas, escolha Create directory (Criar diretório).



A criação do diretório leva vários minutos. Depois que o diretório tiver sido criado com sucesso, o valor de Status muda para Active (Ativo).

Para ver as informações sobre o diretório, selecione o ID do diretório em ID do diretório. Anote o valor do Directory ID (ID do diretório). Esse valor será necessário ao criar ou modificar a instância de banco de dados do RDS para Db2.



Etapa 2: Criar um perfil do IAM para o Amazon RDS acessar o AWS Directory Service

Para que o Amazon RDS chame o AWS Directory Service para você, a Conta da AWS precisa de um perfil do IAM que use a política gerenciada do IAM `AmazonRDSDirectoryServiceAccess`. Esse perfil permite que o Amazon RDS faça chamadas ao AWS Directory Service.

Quando você cria uma instância de banco de dados usando o AWS Management Console e a conta do usuário do console tiver a permissão `iam:CreateRole`, o console criará o perfil do IAM necessário automaticamente. Nesse caso, o nome da função é `rds-directoryservice-kerberos-access-role`. Caso contrário, é necessário criar a função do IAM manualmente. Ao criar essa função do IAM, escolha `Directory Service` e associe a AWS política gerenciada da `AmazonRDSDirectoryServiceAccess` a ela.

Para ter mais informações sobre como criar funções do IAM para um serviço, consulte o tópico sobre como Criar uma função para delegar permissões a um serviço da AWS, no Guia do usuário do IAM.

> **Note**
>
> O perfil do IAM usado para a autenticação Windows para RDS para o Microsoft SQL Server não pode ser usado para o RDS para Db2.

Como alternativa ao uso da política gerenciada pelo `AmazonRDSDirectoryServiceAccess`, você pode criar políticas com as permissões exigidas. Nesse caso, o perfil do IAM deve ter a seguinte política de confiança do IAM:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "directoryservice.rds.amazonaws.com",
          "rds.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

O perfil também deve ter a seguinte política de perfil do IAM:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "ds:DescribeDirectories",
        "ds:AuthorizeApplication",
        "ds:UnauthorizeApplication",
        "ds:GetAuthorizedApplicationDetails"
      ],
    "Effect": "Allow",
    "Resource": "*"
    }
```

```
  ]
}
```

Etapa 3: Criar e configurar usuários

É possível criar usuários utilizando a ferramenta Active Directory Users and Computers. Essa é uma das ferramentas Active Directory Domain Services e Active Directory Lightweight Directory Services. Para ter mais informações, consulte Add Users and Computers to the Active Directory domain na documentação da Microsoft. Nesse caso, os usuários são indivíduos ou outras entidades, como computadores, que fazem parte do domínio e cujas identidades estão sendo mantidas no diretório.

Para criar usuários em um diretório do AWS Directory Service, é necessário estar conectado a uma instância do Amazon EC2 baseada no Windows que seja membro do diretório do AWS Directory Service. Ao mesmo tempo, é necessário estar conectado como um usuário que tenha privilégios para criar usuários. Para obter mais informações, consulte Criar um usuário no Guia de administração do AWS Directory Service.

Etapa 4: Criar um grupo de administradores do RDS para Db2 no AWS Managed Microsoft AD

O RDS para Db2 não é compatível com a autenticação Kerberos para o usuário principal ou para os dois usuários reservados `rdsdb` e `rdsadmin` do Amazon RDS. Em vez disso, você precisa criar um grupo chamado `masterdba` no AWS Managed Microsoft AD. Para ter mais informações, consulte Create a Group Account in Active Directory na documentação da Microsoft. Todos os usuários adicionados a esse grupo terão privilégios de usuário principal.

Depois de habilitar a autenticação Kerberos, o usuário principal perde o perfil `masterdba`. Como resultado, o usuário principal não conseguirá acessar a associação do grupo de usuários local da instância, a menos que você desabilite a autenticação Kerberos. Para continuar utilizando o usuário principal com login por senha, crie um usuário no AWS Managed Microsoft AD com o mesmo nome do usuário principal. Depois, adicione esse usuário ao grupo `masterdba`.

Etapa 5: Criar ou modificar uma instância de banco de dados do RDS para Db2

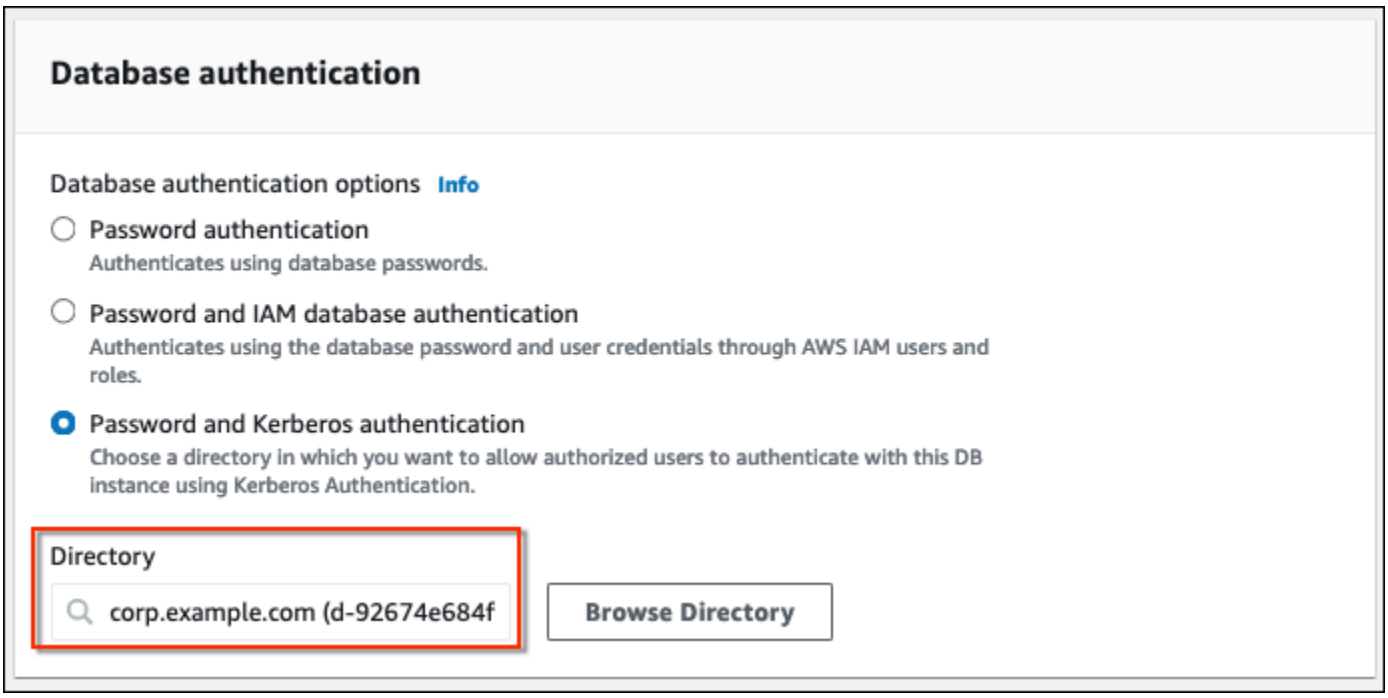
Crie ou modifique uma instância de banco de dados do RDS para Db2 para ser usada com o diretório. É possível usar o AWS Management Console, a AWS CLI ou a API do RDS para associar uma instância de banco de dados a um diretório. Você pode fazer isso por meio de uma das seguintes maneiras:

- Crie uma instância de banco de dados do RDS para Db2 usando o console, o comando create-db-instance ou a operação da API CreateDBInstance. Para instruções, consulte Criar uma instância de banco de dados do Amazon RDS.
- Modifique uma instância de banco de dados do RDS para Db2 existente usando o console, o comando modify-db-instance ou a operação da API ModifyDBInstance. Para instruções, consulte Modificar uma instância de banco de dados do Amazon RDS.
- Restaure uma instância de banco de dados do RDS para Db2 de um snapshot de banco de dados usando o console, o comando restore-db-instance-from-db-snapshot ou a operação da API RestoreDBInstanceFromDBSnapshot. Para instruções, consulte Restaurar uma instância de banco de dados.
- Restaure uma instância de banco de dados do RDS para Db2 para um ponto no tempo usando o console, o comando restore-db-instance-to-point-in-time ou a operação da API RestoreDBInstanceToPointInTime. Para instruções, consulte Restaurar uma instância de banco de dados para um momento especificado no Amazon RDS.

A autenticação Kerberos só é compatível com instâncias de banco de dados do RDS para Db2 em uma VPC. A instância de banco de dados pode estar na mesma VPC do diretório ou em uma VPC diferente. A instância de banco de dados deve usar um grupo de segurança que permita a entrada e a saída na VPC do diretório para que a instância de banco de dados possa se comunicar com o diretório.

Console

Ao usar o console para criar, modificar ou restaurar uma instância de banco de dados, selecione Senha e autenticação Kerberos na seção Autenticação de banco de dados. Em seguida, escolha Browse Directory (Procurar diretório). Selecione o diretório ou escolha Criar diretório para usar o Directory Service.



AWS CLI

Quando você usa a AWS CLI, são necessários os seguintes parâmetros para que a instância de banco de dados possa usar o diretório criado:

- Para o parâmetro `--domain`, use o identificador de domínio (identificador "`d-*`") gerado quando o diretório foi criado.
- Para o parâmetro `--domain-iam-role-name`, use a função criada que usa a política gerenciada `AmazonRDSDirectoryServiceAccess` do IAM.

O exemplo a seguir modifica uma instância de banco de dados para usar um diretório. Substitua os seguintes placeholders no exemplo pelos próprios valores:

- *`db_instance_name`*: o nome da instância de banco de dados do RDS para Db2.
- *`directory_id`*: o ID do diretório do AWS Directory Service for Microsoft Active Directory que você criou.
- *`role_name`*: o nome do perfil do IAM criado.

```
aws rds modify-db-instance --db-instance-identifier db_instance_name --domain
 d-directory_id --domain-iam-role-name role_name
```

> **Important**
>
> Se você modificar uma instância de banco de dados para habilitar a autenticação Kerberos, reinicialize a instância de banco de dados depois de fazer a alteração.

Etapa 6: Configurar um cliente Db2

Como configurar um cliente Db2

1. Crie um arquivo /etc/krb5.conf (ou equivalente) para apontar para o domínio.

   > **Note**
   >
   > Para sistemas operacionais Windows, crie um arquivo C:\windows\krb5.ini.

2. Verifique se o tráfego pode fluir entre o host do cliente e o AWS Directory Service. Use um utilitário de rede, como o Netcat, para as seguintes tarefas:

   a. Verifique o tráfego pelo DNS da porta 53.

   b. Verifique o tráfego pelo TCP/UDP da porta 53 e do Kerberos, que inclui as portas 88 e 464 do AWS Directory Service.

3. Verifique se o tráfego pode fluir entre o host do cliente e a instância de banco de dados pela porta do banco de dados. É possível usar o comando db2 para conectar e acessar o banco de dados.

O seguinte exemplo é o conteúdo do arquivo /etc/krb5.conf para o AWS Managed Microsoft AD:

```
[libdefaults]
default_realm = EXAMPLE.COM
[realms]
EXAMPLE.COM = {
kdc = example.com
admin_server = example.com
}
[domain_realm]
.example.com = EXAMPLE.COM
example.com = EXAMPLE.COM
```

Conectar-se ao Amazon RDS para Db2 com a autenticação Kerberos

Use o procedimento a seguir para se conectar à instância de banco de dados do Amazon RDS para Db2 com a autenticação Kerberos.

Para se conectar ao RDS para Db2 com autenticação Kerberos

1. Em um prompt de comando, execute o seguinte comando da . No exemplo a seguir, substitua *username* pelo nome do usuário Microsoft Active Directory.

   ```
   kinit username
   ```

2. Se a instância de banco de dados do RDS para Db2 estiver usando uma VPC acessível ao público, adicione o endereço IP do endpoint da instância de banco de dados ao arquivo `/etc/hosts` no cliente do Amazon EC2. O exemplo a seguir obtém o endereço IP e, depois, o adiciona ao arquivo `/etc/hosts`.

   ```
   % dig +short Db2-endpoint.AWS-Region.rds.amazonaws.com
   ;; Truncated, retrying in TCP mode.
   ec2-34-210-197-118.AWS-Region.compute.amazonaws.com.
   34.210.197.118

   % echo "34.210.197.118  Db2-endpoint.AWS-Region.rds.amazonaws.com" >> /etc/hosts
   ```

3. Use o comando a seguir para fazer login em uma instância de banco de dados do RDS para Db2 associada ao Active Directory. Substitua *database_name* pelo nome do banco de dados de do RDS para Db2.

   ```
   db2 connect to database_name
   ```

Administrar a instância de banco de dados do Amazon RDS para Db2

Este tópico abrange as tarefas de gerenciamento comuns que você realiza com uma instância de banco de dados do Amazon RDS para Db2. Algumas tarefas são as mesmas para todas as instâncias de banco de dados do Amazon RDS. Outras tarefas são específicas do RDS para Db2.

As tarefas a seguir são comuns a todos os bancos de dados do RDS. Também há tarefas específicas do RDS para Db2, como conectar-se a um banco de dados do RDS para Db2 com um cliente SQL padrão.

Área de tarefa	Documentação relevante
Classes de instância, armazenamento e PIOPS Se você estiver criando uma instância de produção, saiba como funcionam as classes de instância, os tipos de armazenamento e as IOPS provisionadas no Amazon RDS.	Classes de instâncias de banco de dados Tipos de armazenamento do Amazon RDS
Implantações multi-AZ Uma instância de banco de dados de produção deve usar implantações multi-AZ. As implantações multi-AZ oferecem maior disponibilidade, durabilidade de dados e tolerância a falhas para instâncias de banco de dados.	Configurar e gerenciar uma implantação multi-AZ para o Amazon RDS
Amazon VPC Se a Conta da AWS tiver uma nuvem privada virtual (VPC) padrão, a instância de banco de dados será criada automatic amente na VPC padrão. Se a sua conta não tiver uma VPC padrão e você quiser a instância de banco de dados em uma VPC, deverá criar os grupos de VPC e sub-redes antes de criar a instância de banco de dados.	Trabalhar com uma instância de banco de dados em uma VPC
Grupos de segurança Por padrão, as instâncias de banco de dados usam um firewall que impede o acesso. Crie um grupo de segurança com os	Controlar acesso com grupos de segurança

Área de tarefa	Documentação relevante
endereços IP corretos e uma configuração de rede para acessar a instância de banco de dados.	
Grupos de parâmetros Como a instância de banco de dados do RDS para Db2 exige que você adicione os parâmetros `rds.ibm_customer_id` e `rds.ibm_site_id` , crie um grupo de parâmetros antes de criar a instância de banco de dados. Se a instância de banco de dados exigir outros parâmetros de banco de dados específicos, adicione-os também a esse grupo de parâmetros antes de criar a instância de banco de dados.	Adicionar IDs da IBM a um grupo de parâmetros para instâncias de banco de dados do RDS para Db2 Grupos de parâmetros para Amazon RDS
Grupos de opções Se a sua instância de banco de dados exigir opções de banco de dados específicas, crie um grupo de opções antes de criar a instância de banco de dados.	Opções para instâncias de banco de dados do Amazon RDS para Db2
Conexão à instância de banco de dados Depois de criar um grupo de segurança e associá-lo a uma instância de banco de dados, será possível se conectar à instância de banco de dados com qualquer aplicação de cliente SQL padrão, como IBM Db2 CLP.	Conectar-se à instância de banco de dados do Db2
Backup e restauração É possível configurar a instância de banco de dados para fazer backups de armazenamento automatizados ou criar snapshots manuais e, depois, restaurar instâncias por meio dos backups ou dos snapshots.	Backup, restauração e exportação de dados

Área de tarefa	Documentação relevante
Monitoramento É possível monitorar uma instância de banco de dados do RDS para Db2 com o IBM Db2 Data Management Console. Também é possível monitorar uma instância de banco de dados do RDS para Db2 usando métricas, eventos e o monitoramento aprimorado do Amazon RDS do CloudWatch.	Conectar-se à instância de banco de dados do Amazon RDS para Db2 com o IBM Db2 Data Management Console Visualizar métricas no console do Amazon RDS Visualizar eventos do Amazon RDS Monitorar métricas do SO com o monitoramento avançado
Arquivos de log É possível acessar os arquivos de log da instância de banco de dados do RDS para Db2.	Monitorar arquivos de log do Amazon RDS

Tópicos

- Realizar tarefas comuns do sistema para instâncias de banco de dados do Amazon RDS para Db2
- Realizar tarefas comuns de banco de dados para instâncias de banco de dados do Amazon RDS para Db2

Realizar tarefas comuns do sistema para instâncias de banco de dados do Amazon RDS para Db2

É possível executar determinadas tarefas comuns de administrador de banco de dados relacionadas ao sistema nas instâncias de bancos de dados do Amazon RDS que executam o Db2. Para oferecer uma experiência de serviço gerenciado, o Amazon RDS não fornece acesso ao shell para instâncias de bancos de dados e restringe o acesso a determinados procedimentos e tabelas do sistema que exigem privilégios avançados.

Consulte informações sobre como conceder e revogar privilégios e anexar ao banco de dados remoto do RDS para Db2 nos tópicos a seguir.

Tópicos

- Conceder e revogar privilégios do RDS para Db2
- Conectar-se à instância de banco de dados remota do RDS para Db2

Criar um endpoint de banco de dados personalizado

Ao migrar para o Amazon RDS para Db2, é possível usar URLs de endpoint de banco de dados personalizados para minimizar as alterações na aplicação. Por exemplo, se você usar `db2.example.com` como o registro do DNS atual, poderá adicioná-lo ao Amazon Route 53. No Route 53, é possível usar zonas hospedadas privadas para associar o endpoint de banco de dados do DNS atual a um endpoint de banco de dados do RDS para Db2. Para adicionar um registro personalizado `A` ou `CNAME` ou um endpoint de banco de dados do Amazon RDS, consulte Registering and managing domains using Amazon Route 53 no Guia do desenvolvedor do Amazon Route 53.

> **Note**
>
> Se não for possível conseguir transferir o domínio para o Route 53, poderá usar o provedor de DNS para criar um registro `CNAME` para o URL do endpoint do banco de dados do RDS para Db2. Consulte a documentação do provedor de DNS.

Conceder e revogar privilégios do RDS para Db2

Os usuários recebem acesso aos bancos de dados por meio da participação em grupos associados aos bancos de dados.

Use os procedimentos a seguir para conceder e revogar privilégios a fim de controlar o acesso ao banco de dados.

Esses procedimentos usam IBM Db2 CLP em execução em uma máquina local para se conectar a uma instância de banco de dados do RDS para Db2. Catalogue o nó TCP/IP e o banco de dados para se conectar à instância de banco de dados do RDS para Db2 em execução na máquina local. Para ter mais informações, consulte Conectar-se à instância de banco de dados do Amazon RDS para Db2 com o IBM Db2 CLP.

Tópicos

- Conceder a um usuário acesso ao banco de dados

- Alterar a senha de um usuário
- Adicionar grupos a um usuário
- Remover grupos de um usuário
- Remover um usuário
- Listar usuários
- Criar uma função
- Conceder um perfil
- Revogar um perfil
- Eliminação de um perfil
- Conceder autorização de banco de dados
- Revogar autorização de banco de dados

Conceder a um usuário acesso ao banco de dados

Como conceder a um usuário acesso ao banco de dados

1. Conecte-se ao banco de dados `rdsadmin` usando o nome de usuário principal e a senha mestra da instância de banco de dados do RDS para Db2. No exemplo a seguir, substitua *`master_username`* e *`master_password`* por suas próprias informações.

   ```
   db2 connect to rdsadmin user master_username using master_password
   ```

   Esse comando vai produzir uma saída semelhante ao seguinte exemplo:

   ```
   Database Connection Information

   Database server        = DB2/LINUXX8664 11.5.8.0
   SQL authorization ID   = ADMIN
   Local database alias   = RDSADMIN
   ```

2. Adicione um usuário à lista de autorizações chamando `rdsadmin.add_user`. Para ter mais informações, consulte rdsadmin.add_user.

   ```
   db2 "call rdsadmin.add_user(
       'username',
       'password',
   ```

```
    'group_name,group_name')"
```

3. (Opcional) Adicione outros grupos ao usuário chamando `rdsadmin.add_groups`. Para ter mais informações, consulte rdsadmin.add_groups.

```
db2 "call rdsadmin.add_groups(
    'username',
    'group_name,group_name')"
```

4. Confirme as autoridades que estão disponíveis para o usuário. No exemplo a seguir, substitua *rds_database_alias*, *master_user* e *master_password* por suas próprias informações. Além disso, substitua *username* pelo nome do usuário.

```
db2 terminate
db2 connect to rds_database_alias user master_user using master_password
db2 "SELECT SUBSTR(AUTHORITY,1,20) AUTHORITY, D_USER, D_GROUP, D_PUBLIC
       FROM TABLE (SYSPROC.AUTH_LIST_AUTHORITIES_FOR_AUTHID ('username', 'U') ) AS
 T
       ORDER BY AUTHORITY"
```

Esse comando vai produzir uma saída semelhante ao seguinte exemplo:

```
AUTHORITY            D_USER D_GROUP D_PUBLIC
-------------------- ------ ------- --------
ACCESSCTRL           N      N       N
BINDADD              N      N       N
CONNECT              N      N       N
CREATETAB            N      N       N
CREATE_EXTERNAL_ROUT N      N       N
CREATE_NOT_FENCED_RO N      N       N
CREATE_SECURE_OBJECT N      N       N
DATAACCESS           N      N       N
DBADM                N      N       N
EXPLAIN              N      N       N
IMPLICIT_SCHEMA      N      N       N
LOAD                 N      N       N
QUIESCE_CONNECT      N      N       N
SECADM               N      N       N
SQLADM               N      N       N
SYSADM               *      N       *
SYSCTRL              *      N       *
SYSMAINT             *      N       *
```

```
SYSMON                  *       N       *
WLMADM                  N       N       N
```

5. Conceda os perfis do RDS para Db2 `ROLE_NULLID_PACKAGES`, `ROLE_TABLESPACES` e `ROLE_PROCEDURES` ao grupo ao qual você adicionou o usuário. Para ter mais informações, consulte Perfis padrão do Amazon RDS para Db2.

   > **Note**
   >
   > Criamos instâncias de banco de dados do RDS para Db2 no modo `RESTRICTIVE`. Portanto, os perfis do RDS para Db2 `ROLE_NULLID_PACKAGES`, `ROLE_TABLESPACES` e `ROLE_PROCEDURES` concedem privilégios de execução em pacotes `NULLID` para IBM Db2 CLP e Dynamic SQL. Esses perfis também concedem privilégios de usuário em espaços de tabela.

   a. Conecte-se ao banco de dados Db2. No exemplo a seguir, substitua *`database_name`*, *`master_user`* e *`master_password`* por suas próprias informações.

   ```
   db2 connect to database_name user master_user using master_password
   ```

   b. Conceda o perfil `ROLE_NULLED_PACKAGES` a um grupo. No exemplo a seguir, substitua *`group_name`* pelo nome do grupo ao qual você deseja adicionar o perfil.

   ```
   db2 "grant role ROLE_NULLID_PACKAGES to group group_name"
   ```

   c. Conceda o perfil `ROLE_TABLESPACES` ao mesmo grupo. No exemplo a seguir, substitua *`group_name`* pelo nome do grupo ao qual você deseja adicionar o perfil.

   ```
   db2 "grant role ROLE_TABLESPACES to group group_name"
   ```

   d. Conceda o perfil `ROLE_PROCEDURES` ao mesmo grupo. No exemplo a seguir, substitua *`group_name`* pelo nome do grupo ao qual você deseja adicionar o perfil.

   ```
   db2 "grant role ROLE_PROCEDURES to group group_name"
   ```

6. Conceda as autoridades `connect`, `bindadd`, `createtab` e `IMPLICIT_SCHEMA` ao grupo ao qual você adicionou o usuário. No exemplo a seguir, substitua *`group_name`* pelo nome do segundo grupo ao qual você adicionou o usuário.

```
db2 "grant usage on workload SYSDEFAULTUSERWORKLOAD to public"
db2 "grant connect, bindadd, createtab, implicit_schema on database to
 group group_name"
```

7. Repita as etapas 4 a 6 para cada grupo adicional ao qual o usuário foi adicionado.
8. Para testar o acesso do usuário, faça o seguinte: conecte-se como ele, crie uma tabela, insira valores nela e exiba os dados da tabela. No exemplo a seguir, substitua *rds_database_alias*, *username* e *password* pelo nome do banco de dados e pelo nome de usuário e senha do usuário.

```
db2 connect to rds_database_alias user username using password
db2 "create table t1(c1 int not null)"
db2 "insert into t1 values (1),(2),(3),(4)"
db2 "select * from t1"
```

Alterar a senha de um usuário

Para alterar a senha de um usuário

1. Conecte-se ao banco de dados `rdsadmin` usando o nome de usuário principal e a senha mestra da instância de banco de dados do RDS para Db2. No exemplo a seguir, substitua *master_username* e *master_password* por suas próprias informações.

```
db2 connect to rdsadmin user master_username using master_password
```

2. Altere a senha chamando `rdsadmin.change_password`. Para ter mais informações, consulte rdsadmin.change_password.

```
db2 "call rdsadmin.change_password(
    'username',
    'new_password')"
```

Adicionar grupos a um usuário

Como adicionar grupos a um usuário

1. Conecte-se ao banco de dados `rdsadmin` usando o nome de usuário principal e a senha mestra da instância de banco de dados do RDS para Db2. No exemplo a seguir, substitua *`master_username`* e *`master_password`* por suas próprias informações.

```
db2 connect to rdsadmin user master_username using master_password
```

2. Adicione grupos a um usuário chamando `rdsadmin.add_groups`. Para ter mais informações, consulte rdsadmin.add_groups.

```
db2 "call rdsadmin.add_groups(
    'username',
    'group_name,group_name')"
```

Remover grupos de um usuário

Como remover grupos a um usuário

1. Conecte-se ao banco de dados `rdsadmin` usando o nome de usuário principal e a senha mestra da instância de banco de dados do RDS para Db2. No exemplo a seguir, substitua *`master_username`* e *`master_password`* por suas próprias informações.

```
db2 connect to rdsadmin user master_username using master_password
```

2. Remova grupos chamando `rdsadmin.remove_groups`. Para ter mais informações, consulte rdsadmin.remove_groups.

```
db2 "call rdsadmin.remove_groups(
    'username',
    'group_name,group_name')"
```

Remover um usuário

Como remover um usuário da lista de autorizações

1. Conecte-se ao banco de dados `rdsadmin` usando o nome de usuário principal e a senha mestra da instância de banco de dados do RDS para Db2. No exemplo a seguir, substitua *`master_username`* e *`master_password`* por suas próprias informações.

   ```
   db2 connect to rdsadmin user master_username using master_password
   ```

2. Remova um usuário da lista de autorizações chamando `rdsadmin.remove_user`. Para ter mais informações, consulte rdsadmin.remove_user.

   ```
   db2 "call rdsadmin.remove_user('username')"
   ```

Listar usuários

Para listar usuários em uma lista de autorizações, chame o procedimento armazenado `rdsadmin.list_users`. Para ter mais informações, consulte rdsadmin.list_users.

```
db2 "call rdsadmin.list_users()"
```

Criar uma função

É possível usar o procedimento armazenado rdsadmin.create_role para criar um perfil.

Para criar uma função

1. Conecte-se ao banco de dados `rdsadmin`. No exemplo a seguir, substitua *`master_username`* e *`master_password`* por suas próprias informações.

   ```
   db2 connect to rdsadmin user master_username using master_password
   ```

2. Defina o Db2 para gerar conteúdo.

   ```
   db2 set serveroutput on
   ```

3. Crie uma função. Para ter mais informações, consulte the section called “rdsadmin.create_role”.

   ```
   db2 "call rdsadmin.create_role(
   ```

```
    'database_name',
    'role_name')"
```

4. Defina o Db2 para não gerar conteúdo.

```
db2 set serveroutput off
```

Conceder um perfil

É possível usar o procedimento armazenado rdsadmin.grant_role para atribuir um perfil a um perfil, um usuário ou um grupo.

Como atribuir um perfil

1. Conecte-se ao banco de dados `rdsadmin`. No exemplo a seguir, substitua *`master_username`* e *`master_password`* por suas próprias informações.

```
db2 connect to rdsadmin user master_username using master_password
```

2. Defina o Db2 para gerar conteúdo.

```
db2 set serveroutput on
```

3. Atribua um perfil. Para ter mais informações, consulte the section called “rdsadmin.grant_role”.

```
db2 "call rdsadmin.grant_role(
    'database_name',
    'role_name',
    'grantee',
    'admin_option')"
```

4. Defina o Db2 para não gerar conteúdo.

```
db2 set serveroutput off
```

Revogar um perfil

É possível usar o procedimento armazenado rdsadmin.revoke_role para revogar um perfil de um perfil, um usuário ou um grupo.

Como revogar um perfil

1. Conecte-se ao banco de dados `rdsadmin`. No exemplo a seguir, substitua *`master_username`* e *`master_password`* por suas próprias informações.

```
db2 connect to rdsadmin user master_username using master_password
```

2. Revogue um perfil. Para ter mais informações, consulte the section called "rdsadmin.revoke_role".

```
db2 "call rdsadmin.revoke_role(
    ?,
    'database_name',
    'role_name',
    'grantee')"
```

Eliminação de um perfil

É possível usar o procedimento armazenado ??? para descartar um perfil.

Como excluir um perfil

1. Conecte-se ao banco de dados `rdsadmin`. No exemplo a seguir, substitua *`master_username`* e *`master_password`* por suas próprias informações.

```
db2 connect to rdsadmin user master_username using master_password
```

2. Descarte um perfil. Para ter mais informações, consulte ???.

```
db2 "call rdsadmin.drop_role(
    ?,
    'database_name',
    'role_name')"
```

Conceder autorização de banco de dados

O usuário principal, que tem autorização `DBADM`, pode conceder a autorização `DBADM`, `ACCESSCTRL` ou `DATAACCESS` a um perfil, um usuário ou um grupo.

Como conceder autorização de banco de dados

1. Conecte-se ao banco de dados `rdsadmin` usando o nome de usuário principal e a senha mestra da instância de banco de dados do RDS para Db2. No exemplo a seguir, substitua *`master_username`* e *`master_password`* por suas próprias informações.

```
db2 connect to rdsadmin user master_username using master_password
```

2. Conceda acesso a um usuário chamando `rdsadmin.dbadm_grant`. Para ter mais informações, consulte rdsadmin.dbadm_grant.

```
db2 "call rdsadmin.dbadm_grant(
    ?,
    'database_name,
    'authorization',
    'grantee')"
```

Exemplo de caso de uso

O procedimento a seguir explica como criar um perfil, conceder autorização `DBADM` ao perfil, atribuir o perfil a um usuário e a um grupo.

1. Conecte-se ao banco de dados `rdsadmin` usando o nome de usuário principal e a senha mestra da instância de banco de dados do RDS para Db2. No exemplo a seguir, substitua *`master_username`* e *`master_password`* por suas próprias informações.

```
db2 connect to rdsadmin user master_username using master_password
```

2. Crie um perfil chamado `PROD_ROLE` para um banco de dados chamado `TESTDB`. Para ter mais informações, consulte rdsadmin.create_role.

```
db2 "call rdsadmin.create_role(
    'TESTDB',
    'PROD_ROLE')"
```

3. Atribua o perfil a um usuário denominado `PROD_USER`. O `PROD_USER` recebe autorização de administrador para atribuir perfis. Para ter mais informações, consulte rdsadmin.grant_role.

```
db2 "call rdsadmin.grant_role(
```

```
    ?,
    'TESTDB',
    'PROD_ROLE',
    'USER PROD_USER',
    'Y')"
```

4. (Opcional) Forneça autorização ou privilégios adicionais. O exemplo a seguir concede autorização DBADM a um perfil denominado PROD_ROLE para um banco de dados chamado FUNDPROD. Para ter mais informações, consulte rdsadmin.dbadm_grant.

```
db2 "call rdsadmin.dbadm_grant(
    ?,
    'FUNDPROD',
    'DBADM',
    'ROLE PROD_ROLE')"
```

5. Encerre a sessão.

```
db2 terminate
```

6. Conecte-se ao banco de dados TESTDB usando o nome de usuário principal e a senha mestra da instância de banco de dados do RDS para Db2. No exemplo a seguir, substitua *master_username* e *master_password* por suas próprias informações.

```
db2 connect to TESTDB user master_username using master_password
```

7. Adicione mais autorizações ao perfil.

```
db2 "grant connect, implicit_schema on database to role PROD_ROLE"
```

8. Conceda o perfil PROD_ROLE a um grupo.

```
db2 "grant role PROD_ROLE to group PRODGRP"
```

Os usuários que pertencem ao grupo PRODGRP agora podem realizar ações como conectar-se ao banco de dados TESTDB, criar tabelas ou esquemas.

Revogar autorização de banco de dados

O usuário principal, que tem a autorização DBADM, pode revogar a autorização DBADM, ACCESSCTRL ou DATAACCESS de um perfil, um usuário ou um grupo.

Como revogar autorização de banco de dados

1. Conecte-se ao banco de dados `rdsadmin` usando o nome de usuário principal e a senha mestra da instância de banco de dados do RDS para Db2. No exemplo a seguir, substitua *`master_username`* e *`master_password`* por suas próprias informações.

```
db2 connect to rdsadmin user master_username using master_password
```

2. Revogue o acesso do usuário chamando `rdsadmin.dbadm_revoke`. Para ter mais informações, consulte rdsadmin.dbadm_revoke.

```
db2 "call rdsadmin.dbadm_revoke(
    ?,
    'database_name,
    'authorization',
    'grantee')"
```

Conectar-se à instância de banco de dados remota do RDS para Db2

Use as etapas a seguir para se conectar à sua instância de banco de dados remota do RDS para Db2 e executar operações `get snapshot`.

Como se conectar à instância de banco de dados remota do RDS para Db2

1. Execute uma sessão IBM Db2 CLP do lado do cliente. Para ter informações sobre como catalogar a instância de banco de dados e banco de dados do RDS para Db2, consulte Conectar-se à instância de banco de dados do Amazon RDS para Db2 com o IBM Db2 CLP. Anote o nome de usuário principal e a senha mestra da instância de banco de dados do RDS para Db2.
2. Conecte-se à instância de banco de dados do RDS para Db2. No exemplo a seguir, substitua *`node_name`*, *`master_username`* e *`master_password`* pelo nome de nó TCPIP catalogado, além do nome de usuário principal e da senha mestra da instância de banco de dados do RDS para Db2.

```
db2 attach to node_name user master_username using master_password
```

Depois de se conectar à instância de banco de dados remota do RDS para Db2, é possível executar os comandos a seguir e outros comandos `get snapshot`. Para ter mais informações, consulte GET SNAPSHOT command na documentação do IBM Db2.

```
db2 list applications
db2 get snapshot for all databases
db2 get snapshot for database manager
db2 get snapshot for all applications
```

Realizar tarefas comuns de banco de dados para instâncias de banco de dados do Amazon RDS para Db2

É possível realizar determinadas tarefas comuns de DBA relacionadas aos bancos de dados nas instâncias de bancos de dados do Amazon RDS para Db2. Para oferecer uma experiência de serviço gerenciada, o Amazon RDS não fornece acesso ao shell para as instâncias de banco de dados. Além disso, o usuário principal não pode executar comandos nem utilitários que exijam as autoridades `SYSADM`, `SYSMAINT` ou `SYSCTRL`.

Consulte informações sobre tarefas comuns para os grupos de buffer, bancos de dados e espaços de tabelas nos tópicos a seguir.

Tópicos

- Tarefas comuns para grupos de buffer
- Tarefas comuns para bancos de dados
- Tarefas comuns para espaços de tabela

Tarefas comuns para grupos de buffer

É possível criar, alterar ou descartar grupos de buffers para um banco de dados do RDS para Db2. Criar, alterar ou descartar grupos de buffers requer a autoridade `SYSADM` ou `SYSCTRL` de nível mais alto, que não está disponível para o usuário principal. Em vez disso, use procedimentos armazenados do Amazon RDS.

Também é possível limpar grupos de buffers.

Tópicos

- Criar um grupo de buffers

- Alterar um grupo de buffers
- Descartar um grupo de buffers
- Liberar os grupos de buffers

Criar um grupo de buffers

Para criar um grupo de buffers para o banco de dados do RDS para Db2, chame o procedimento armazenado `rdsadmin.create_bufferpool`. Para ter mais informações, consulte Declaração CREATE BUFFERPOOL na documentação do IBM Db2.

Como criar um grupo de buffers

1. Conecte-se ao banco de dados `rdsadmin` usando o nome de usuário principal e a senha mestra da instância de banco de dados do RDS para Db2. No exemplo a seguir, substitua *`master_username`* e *`master_password`* por suas próprias informações.

   ```
   db2 "connect to rdsadmin user master_username using master_password"
   ```

2. Crie um grupo de buffers chamando `rdsadmin.create_bufferpool`. Para obter mais informações, consulte rdsadmin.create_bufferpool.

   ```
   db2 "call rdsadmin.create_bufferpool(
       'database_name',
       'buffer_pool_name',
       buffer_pool_size,
       'immediate',
       'automatic',
       page_size,
       number_block_pages,
       block_size)"
   ```

Alterar um grupo de buffers

Para alterar um grupo de buffers para o banco de dados do RDS para Db2, chame o procedimento armazenado `rdsadmin.alter_bufferpool`. Para ter mais informações, consulte Declaração ALTER BUFFERPOOL na documentação do IBM Db2.

Como alterar um grupo de buffers

1. Conecte-se ao banco de dados `rdsadmin` usando o nome de usuário principal e a senha mestra da instância de banco de dados do RDS para Db2. No exemplo a seguir, substitua *`master_username`* e *`master_password`* por suas próprias informações.

```
db2 "connect to rdsadmin user master_username using master_password"
```

2. Altere um grupo de buffers chamando `rdsadmin.alter_bufferpool`. Para obter mais informações, consulte rdsadmin.alter_bufferpool.

```
db2 "call rdsadmin.alter_bufferpool(
    'database_name',
    'buffer_pool_name',
    buffer_pool_size,
    'immediate',
    'automatic',
    change_number_blocks,
    number_block_pages,
    block_size)"
```

Descartar um grupo de buffers

Para descartar um grupo de buffers para o banco de dados do RDS para Db2, chame o procedimento armazenado `rdsadmin.drop_bufferpool`. Para ter mais informações, consulte Dropping buffer pools na documentação do IBM Db2.

> **Important**
>
> Garanta que nenhum espaço de tabela esteja atribuído ao grupo de buffers que você deseja descartar.

Como descartar um grupo de buffers

1. Conecte-se ao banco de dados `rdsadmin` usando o nome de usuário principal e a senha mestra da instância de banco de dados do RDS para Db2. No exemplo a seguir, substitua *`master_username`* e *`master_password`* por suas próprias informações.

```
db2 "connect to rdsadmin user master_username using master_password"
```

2. Descarte um grupo de buffers chamando `rdsadmin.drop_bufferpool`. Para obter mais informações, consulte rdsadmin.drop_bufferpool.

```
db2 "call rdsadmin.drop_bufferpool(
    'database_name',
    'buffer_pool_name')"
```

Liberar os grupos de buffers

É possível liberar os grupos de buffers para forçar um ponto de verificação para que o RDS para Db2 grave páginas da memória para o armazenamento.

> **Note**
>
> Não é necessário liberar os grupos de buffers. O Db2 grava logs de forma síncrona antes de confirmar as transações. As páginas sujas ainda podem estar em um grupo de buffers, mas o Db2 as grava no armazenamento de forma assíncrona. Mesmo que o sistema seja desligado inesperadamente, ao reiniciar o banco de dados, o Db2 executa automaticamente a recuperação de falhas. Durante a recuperação de falhas, o Db2 grava as alterações confirmadas no banco de dados ou as reverte para transações não confirmadas.

Como liberar os grupos de buffers

1. Conecte-se ao banco de dados Db2 usando o nome de usuário principal e a senha principal da instância de banco de dados do RDS para Db2. No exemplo a seguir, substitua *rds_database_alias*, *master_username* e *master_password* por suas próprias informações.

```
db2 connect to rds_database_alias user master_username using master_password
```

2. Libere os grupos de buffers.

```
db2 flush bufferpools all
```

Tarefas comuns para bancos de dados

É possível criar, descartar ou restaurar bancos de dados na instância de banco de dados do RDS para Db2. Criar, descartar ou restaurar bancos de dados requer autoridade SYSADM de nível mais alto, que não está disponível para o usuário principal. Em vez disso, use procedimentos armazenados do Amazon RDS.

Também é possível realizar tarefas comuns de gerenciamento, como monitoramento, manutenção e coleta de informações sobre seus bancos de dados.

Tópicos

- Criação de um banco de dados
- Definir configurações para um banco de dados
- Modificar parâmetros de banco de dados
- Configurar a retenção de logs
- Listar informações de log
- Desativação de um banco de dados
- Ativação de um banco de dados
- Reativar um banco de dados
- Descartar um banco de dados
- Restaurar um banco de dados
- Listagem de bancos de dados
- Coleta de informações sobre bancos de dados
- Forçar a retirada de aplicações de bancos de dados
- Gerar relatórios de performance

Criação de um banco de dados

Para criar um banco de dados na instância de banco de dados do RDS para Db2, chame o procedimento armazenado `rdsadmin.create_database`. Para ter mais informações, consulte CREATE DATABASE command na documentação do IBM Db2.

> **Note**
>
> Se você planeja modificar o parâmetro `db2_compatibility_vector`, modifique-o antes de criar um banco de dados. Para obter mais informações, consulte Configuração do parâmetro db2_compatibility_vector.

Para criar um banco de dados

1. Conecte-se ao banco de dados `rdsadmin` usando o nome de usuário principal e a senha mestra da instância de banco de dados do RDS para Db2. No exemplo a seguir, substitua *`master_username`* e *`master_password`* por suas próprias informações.

   ```
   db2 "connect to rdsadmin user master_username using master_password"
   ```

2. Crie um banco de dados chamando `rdsadmin.create_database`. Para obter mais informações, consulte rdsadmin.create_database.

   ```
   db2 "call rdsadmin.create_database('database_name')"
   ```

3. (Opcional) Crie bancos de dados adicionais chamando `rdsadmin.create_database` para cada banco de dados que você deseja criar. Cada instância de banco de dados do Db2 pode conter até 50 bancos de dados. Para obter mais informações, consulte rdsadmin.create_database.

   ```
   db2 "call rdsadmin.create_database('database_name')"
   ```

4. (Opcional) Confirme se o seu banco de dados foi criado usando um dos seguintes métodos:

   - Chame `rdsadmin.list_databases`. Para obter mais informações, consulte rdsadmin.list_databases.
   - Execute o seguinte comando SQL:

     ```
     db2 "select varchar(r.task_type,25) as task_type, r.database_name,
         varchar(r.lifecycle,15) as lifecycle, r.created_at, r.database_name,
         varchar(bson_to_json(task_input_params),256) as input_params,
         varchar(r.task_output,1024) as task_output
         from table(rdsadmin.get_task_status(null,null,'create_database'))
         as r order by created_at desc"
     ```

Definir configurações para um banco de dados

Para definir as configurações de um banco de dados na instância de banco de dados do RDS para Db2, chame o procedimento armazenado `rdsadmin.set_configuration`. Por exemplo, é possível configurar o número de buffers ou manipuladores de buffer a serem criados durante uma operação de restauração.

Como definir as configurações de um banco de dados

1. Conecte-se ao banco de dados `rdsadmin` usando o nome de usuário principal e a senha mestra da instância de banco de dados do RDS para Db2. No exemplo a seguir, substitua *`master_username`* e *`master_password`* por suas próprias informações.

   ```
   db2 "connect to rdsadmin user master_username using master_password"
   ```

2. (Opcional) Confira suas configurações atuais chamando `rdsadmin.show_configuration`. Para obter mais informações, consulte the section called “rdsadmin.show_configuration”.

   ```
   db2 "call rdsadmin.show_configuration('name')"
   ```

3. Defina as configurações do banco de dados chamando `rdsadmin.set_configuration`. Para obter mais informações, consulte the section called “rdsadmin.set_configuration”.

   ```
   db2 "call rdsadmin.set_configuration(
       'name',
       'value')"
   ```

Modificar parâmetros de banco de dados

O Amazon RDS para Db2 usa três tipos de parâmetros: parâmetros de configuração do gerenciador de banco de dados, variáveis do registro e parâmetros de configuração do banco de dados. É possível atualizar os dois primeiros tipos por meio de grupos de parâmetros e o último tipo por meio do procedimento armazenado rdsadmin.update_db_param.

> **Note**
>
> É possível modificar apenas os valores dos parâmetros existentes. Não é possível adicionar novos parâmetros não aceitos pelo RDS para Db2.

Para ter mais informações sobre esses parâmetros e como modificar os valores, consulte the section called “Parâmetros do Db2”.

Configurar a retenção de logs

Para configurar por quanto tempo o Amazon RDS retém arquivos de log do banco de dados do RDS para Db2, chame o procedimento armazenado `rdsadmin.set_archive_log_retention`.

Como configurar a retenção de logs de um banco de dados

1. Conecte-se ao banco de dados `rdsadmin` usando o nome de usuário principal e a senha mestra da instância de banco de dados do RDS para Db2. No exemplo a seguir, substitua *`master_username`* e *`master_password`* por suas próprias informações.

   ```
   db2 "connect to rdsadmin user master_username using master_password"
   ```

2. (Opcional) Confira sua configuração atual para retenção de logs chamando `rdsadmin.show_archive_log_retention`. Para obter mais informações, consulte the section called “rdsadmin.show_archive_log_retention”.

   ```
   db2 "call rdsadmin.show_archive_log_retention(
       ?,
       'database_name')"
   ```

3. Configure a retenção de logs do banco de dados chamando `rdsadmin.set_archive_log_retention`. Para obter mais informações, consulte the section called “rdsadmin.set_archive_log_retention”.

   ```
   db2 "call rdsadmin.set_archive_log_retention(
       ?,
       'database_name',
       'archive_log_retention_hours')"
   ```

Listar informações de log

Para listar detalhes sobre arquivos de log de arquivamento, como o tamanho total de armazenamento usado, chame o procedimento armazenado `rdsadmin.list_archive_log_information`.

Como listar informações de log de um banco de dados

1. Conecte-se ao banco de dados `rdsadmin` usando o nome de usuário principal e a senha mestra da instância de banco de dados do RDS para Db2. No exemplo a seguir, substitua *`master_username`* e *`master_password`* por suas próprias informações.

```
db2 "connect to rdsadmin user master_username using master_password"
```

2. Obtenha uma lista de informações do arquivo de log chamando `rdsadmin.list_archive_log_information`. Para obter mais informações, consulte ???.

```
db2 "call rdsadmin.list_archive_log_information(
    ?,
    'database_name')"
```

Desativação de um banco de dados

Para desativar um banco de dados na instância de banco de dados do RDS para Db2, chame o procedimento armazenado `rdsadmin.deactivate_database`.

Por padrão, o Amazon RDS ativa um banco de dados ao criar um banco de dados na instância de banco de dados do RDS para Db2. Você pode desativar bancos de dados usados com pouca frequência para preservar recursos de memória.

Como desativar um banco de dados

1. Conecte-se ao banco de dados `rdsadmin` usando o nome de usuário principal e a senha mestra da instância de banco de dados do RDS para Db2. No exemplo a seguir, substitua *`master_username`* e *`master_password`* por suas próprias informações.

```
db2 "connect to rdsadmin user master_username using master_password"
```

2. Desative um banco de dados chamando `rdsadmin.deactivate_database`. Para obter mais informações, consulte ???.

```
db2 "call rdsadmin.deactivate_database(
    ?,
    'database_name')"
```

Ativação de um banco de dados

Para ativar um banco de dados em uma instância autônoma de banco de dados do RDS para Db2, chame o procedimento armazenado `rdsadmin.activate_database`.

Por padrão, o Amazon RDS ativa um banco de dados ao criar um banco de dados na instância de banco de dados do RDS para Db2. Você pode desativar bancos de dados usados com pouca frequência para preservar recursos de memória e, posteriormente, ativar um banco de dados desativado.

Como ativar um banco de dados

1. Conecte-se ao banco de dados `rdsadmin` usando o nome de usuário principal e a senha mestra da instância de banco de dados do RDS para Db2. No exemplo a seguir, substitua *`master_username`* e *`master_password`* por suas próprias informações.

   ```
   db2 "connect to rdsadmin user master_username using master_password"
   ```

2. Ative um banco de dados chamando `rdsadmin.activate_database`. Para obter mais informações, consulte ???.

   ```
   db2 "call rdsadmin.activate_database(
       ?,
       'database_name')"
   ```

Reativar um banco de dados

Para reativar um banco de dados, chame o procedimento armazenado `rdsadmin.reactivate_database`. Depois de fazer alterações nas configurações do banco de dados, talvez seja necessário reativar um banco de dados em uma instância de banco de dados do RDS para Db2. Para determinar se você precisa reativar um banco de dados, conecte-se a ele e execute `db2 get db cfg show detail`.

Também é possível chamar esse procedimento armazenado para reativar um banco de dados em uma instância autônoma de banco de dados do RDS para Db2 depois de fazer alterações nas configurações do banco de dados. Ou você pode reativar um banco de dados em uma instância autônoma de banco de dados RDS para Db2 chamando primeiro o procedimento armazenado `rdsadmin.deactivate_database` e depois o procedimento armazenado

`rdsadmin.activate_database`. Para obter mais informações, consulte Desativação de um banco de dados e Ativação de um banco de dados.

Como reativar um banco de dados

1. Conecte-se ao banco de dados `rdsadmin` usando o nome de usuário principal e a senha mestra da instância de banco de dados do RDS para Db2. No exemplo a seguir, substitua *`master_username`* e *`master_password`* por suas próprias informações.

```
db2 "connect to rdsadmin user master_username using master_password"
```

2. Reative um banco de dados chamando `rdsadmin.reactivate_database`. Para obter mais informações, consulte ???.

```
db2 "call rdsadmin.reactivate_database(
    ?,
    'database_name')"
```

Descartar um banco de dados

Para descartar um banco de dados da instância de banco de dados do RDS para Db2, chame o procedimento armazenado `rdsadmin.drop_database`. Para ter mais informações, consulte Dropping databases na documentação do IBM Db2.

> **Note**
>
> Será possível descartar um banco de dados chamando o procedimento armazenado somente se determinadas condições forem atendidas. Para obter mais informações, consulte the section called “Observações de uso” para `rdsadmin.drop_database`.

Como descartar um banco de dados

1. Conecte-se ao banco de dados `rdsadmin` usando o nome de usuário principal e a senha mestra da instância de banco de dados do RDS para Db2. No exemplo a seguir, substitua *`master_username`* e *`master_password`* por suas próprias informações.

```
db2 "connect to rdsadmin user master_username using master_password"
```

2. Descarte um banco de dados chamando `rdsadmin.drop_database`. Para obter mais informações, consulte rdsadmin.drop_database.

```
db2 "call rdsadmin.drop_database('database_name')"
```

Restaurar um banco de dados

Para mover um banco de dados de um bucket do Amazon S3 para a instância de banco de dados do RDS para Db2, chame o procedimento armazenado `rdsadmin.restore_database`. Para ter mais informações, consulte RESTORE DATABASE command na documentação do IBM Db2.

Para restaurar um banco de dados do

1. Conecte-se ao banco de dados `rdsadmin` usando o nome de usuário principal e a senha mestra da instância de banco de dados do RDS para Db2. No exemplo a seguir, substitua *`master_username`* e *`master_password`* por suas próprias informações.

```
db2 "connect to rdsadmin user master_username using master_password"
```

2. (Opcional) Confira suas configurações atuais para otimizar a operação de restauração chamando `rdsadmin.show_configuration`. Para obter mais informações, consulte the section called “rdsadmin.show_configuration”.

```
db2 "call rdsadmin.show_configuration('name')"
```

3. Defina as configurações para otimizar a operação de restauração chamando `rdsadmin.set_configuration`. Definir explicitamente esses valores pode melhorar a performance ao restaurar bancos de dados com grandes volumes de dados. Para obter mais informações, consulte the section called “rdsadmin.set_configuration”.

```
db2 "call rdsadmin.set_configuration(
    'name',
    'value')"
```

4. Restaure o banco de dados chamando `rdsadmin.restore_database`. Para obter mais informações, consulte the section called “rdsadmin.restore_database”.

```
db2 "call rdsadmin.restore_database(
    ?,
```

```
    'database_name',
    's3_bucket_name',
    's3_prefix',
    restore_timestamp,
    'backup_type')"
```

5. (Opcional) Confirme se seu banco de dados foi restaurado chamando `rdsadmin.list_databases` e verificando se o banco de dados restaurado está listado. Para obter mais informações, consulte rdsadmin.list_databases.
6. Reative o banco de dados e aplique logs de transações adicionais chamando `rdsadmin.rollforward_database`. Para obter mais informações, consulte the section called “rdsadmin.rollforward_database”.

```
db2 "call rdsadmin.rollforward_database(
    ?,
    'database_name',
    's3_bucket_name',
    s3_prefix,
    'rollforward_to_option',
    'complete_rollforward')"
```

7. Se você definiu `complete_rollforward` como `FALSE` na etapa anterior, termine de ativar o banco de dados chamando `rdsadmin.complete_rollforward`. Para obter mais informações, consulte the section called “rdsadmin.complete_rollforward”.

```
db2 "call rdsadmin.complete_rollforward(
    ?,
    'database_name')"
```

Listagem de bancos de dados

Você pode listar todos os seus bancos de dados em execução no Amazon RDS para Db2 chamando a função definida pelo usuário `rdsadmin.list_databases`.

Como listar seus bancos de dados

1. Conecte-se ao banco de dados `rdsadmin` usando o nome de usuário principal e a senha mestra da instância de banco de dados do RDS para Db2. No exemplo a seguir, substitua *master_username* e *master_password* por suas próprias informações.

```
db2 "connect to rdsadmin user master_username using master_password"
```

2. Liste seus bancos de dados chamando `rdsadmin.list_databases`. Para obter mais informações, consulte rdsadmin.list_databases.

```
db2 "select * from table(rdsadmin.list_databases())"
```

Coleta de informações sobre bancos de dados

Para coletar informações sobre uma instância de banco de dados do RDS para Db2, chame o procedimento armazenado `rdsadmin.db2pd_command`. Essas informações podem ajudar a monitorar os bancos de dados ou solucionar problemas.

Como coletar informações sobre um banco de dados

1. Conecte-se ao banco de dados `rdsadmin` usando o nome de usuário principal e a senha mestra da instância de banco de dados do RDS para Db2. No exemplo a seguir, substitua *master_username* e *master_password* por suas próprias informações.

```
db2 "connect to rdsadmin user master_username using master_password"
```

2. Colete informações sobre o banco de dados chamando `rdsadmin.db2pd_command`. Para obter mais informações, consulte rdsadmin.db2pd_command.

```
db2 "call rdsadmin.db2pd_command('db2pd_cmd')"
```

Forçar a retirada de aplicações de bancos de dados

Para forçar a retirada das aplicações de um banco de dados em sua instância de banco de dados do RDS para Db2, chame o procedimento armazenado `rdsadmin.force_application`. Antes de realizar a manutenção nos bancos de dados, retire as aplicações dos bancos de dados.

Como forçar a retirada das aplicações de um banco de dados

1. Conecte-se ao banco de dados `rdsadmin` usando o nome de usuário principal e a senha mestra da instância de banco de dados do RDS para Db2. No exemplo a seguir, substitua *master_username* e *master_password* por suas próprias informações.

```
db2 "connect to rdsadmin user master_username using master_password"
```

2. Force a retirada das aplicações de um banco de dados chamando `rdsadmin.force_application`. Para obter mais informações, consulte rdsadmin.force_application.

```
db2 "call rdsadmin.force_application(
    ?,
    'applications')"
```

Gerar relatórios de performance

É possível gerar relatórios de performance com um procedimento ou um script. Para ter informações sobre o uso de um procedimento, consulte DBSUMMARY procedure - Generate a summary report of system and application performance metrics na documentação do IBM Db2.

O Db2 inclui um arquivo `db2mon.sh` no `~sqllib/sample/perf` diretório. A execução do script produz um relatório de métricas SQL extenso e de baixo custo. Para baixar o arquivo `db2mon.sh` e os arquivos de script relacionados, consulte o diretório perf no repositório GitHub db2-samples da IBM.

Como gerar relatórios de performance com o script

1. Conecte-se ao banco de dados Db2 usando o nome de usuário principal e a senha principal da instância de banco de dados do RDS para Db2. No exemplo a seguir, substitua *`master_username`* e *`master_password`* por suas próprias informações.

```
db2 connect to rdsadmin user master_username using master_password
```

2. Crie um grupo de buffers chamado `db2monbp` com um tamanho de página de 4.096 chamando `rdsadmin.create_bufferpool`. Para obter mais informações, consulte rdsadmin.create_bufferpool.

```
db2 "call rdsadmin.create_bufferpool('database_name','db2monbp',4096)"
```

3. Crie um espaço de tabela temporário chamado `db2montmptbsp` que use o grupo de buffers `db2monbp` chamando `rdsadmin.create_tablespace`. Para obter mais informações, consulte rdsadmin.create_tablespace.

```
db2 "call rdsadmin.create_tablespace('database_name',\
    'db2montmptbsp','db2monbp',4096,1000,100,'T')"
```

4. Abra o script `db2mon.sh` e modifique a linha sobre a conexão com um banco de dados.

   a. Remova a linha a seguir.

   ```
   db2 -v connect to $dbName
   ```

   b. Substitua a linha na etapa anterior pela linha a seguir. No exemplo a seguir, substitua *`master_username`* e *`master_password`* pelo nome de usuário principal e a senha mestra da instância de banco de dados do RDS para Db2.

   ```
   db2 -v connect to $dbName user master_username using master_password
   ```

   c. Remova as linhas a seguir.

   ```
   db2 -v create bufferpool db2monbp

   db2 -v create user temporary tablespace db2montmptbsp bufferpool db2monbp

   db2 -v drop tablespace db2montmptbsp

   db2 -v drop bufferpool db2monbp
   ```

5. Execute o script `db2mon.sh` para gerar um relatório em intervalos especificados. No exemplo a seguir, substitua *`absolute_path`* pelo caminho completo do arquivo de script, *`rds_database_alias`* pelo nome do banco de dados e *`seconds`* pelo número de segundos (0 a 3.600) entre as gerações do relatório.

   ```
   absolute_path/db2mon.sh rds_database_alias seconds | tee -a db2mon.out
   ```

   Exemplos

   O exemplo a seguir mostra que o arquivo de script está localizado no diretório `perf` abaixo do diretório home.

   ```
   /home/db2inst1/sqllib/samples/perf/db2mon.sh rds_database_alias seconds | tee -a
    db2mon.out
   ```

6. Descarte o grupo de buffers e o espaço de tabela que foram criados para o arquivo `db2mon.sh`. No exemplo a seguir, substitua *`master_username`* e *`master_password`* pelo nome de usuário principal e a senha mestra da instância de banco de dados do RDS para Db2. Substitua *`database_name`* pelo nome do banco de dados. Para obter mais informações, consulte rdsadmin.drop_tablespace e rdsadmin.drop_bufferpool.

```
db2 connect to rdsadmin user master_username using master_password

db2 "call rdsadmin.drop_tablespace('database_name','db2montmptbsp')"

db2 "call rdsadmin.drop_bufferpool('database_name','db2monbp')"
```

Gerenciar o armazenamento

O Db2 usa armazenamento automático para gerenciar o armazenamento físico de objetos de banco de dados, como tabelas, índices e arquivos temporários. Em vez de alocar espaço de armazenamento manualmente e acompanhar quais caminhos de armazenamento estão sendo usados, o armazenamento automático permite que o sistema Db2 crie e gerencie caminhos de armazenamento conforme necessário. Isso pode simplificar a administração de bancos de dados do Db2 e reduzir a probabilidade de erros humanos. Para ter mais informações, consulte Armazenamento automático, na documentação do IBM Db2.

Com o RDS para Db2, é possível aumentar dinamicamente o tamanho do armazenamento com a expansão automática dos volumes lógicos e do sistema de arquivos. Para ter mais informações, consulte Trabalhar com armazenamento para instâncias de banco de dados do Amazon RDS.

Tarefas comuns para espaços de tabela

É possível criar, alterar, renomear ou descartar espaços de tabela de um banco de dados do RDS para Db2. Criar, alterar, renomear ou descartar espaços de tabela requer autoridade `SYSADM` de nível mais alto, que não está disponível para o usuário principal. Em vez disso, use procedimentos armazenados do Amazon RDS.

Tópicos

- Criar um espaço de tabela
- Alterar um espaço de tabela
- Renomear um espaço de tabela

- Descartar um espaço de tabela
- Conferir o status de um espaço de tabela
- Exibir informações detalhadas sobre espaços de tabela
- Listar o estado e o grupo de armazenamento de um espaço de tabela
- Listar os espaços de uma tabela
- Listar contêineres de espaço de tabela

Criar um espaço de tabela

Para criar um espaço de tabela do banco de dados do RDS para Db2, chame o procedimento armazenado `rdsadmin.create_tablespace`. Para ter mais informações, consulte Declaração CREATE TABLESPACE na documentação do IBM Db2.

> **Important**
>
> Para criar um espaço de tabela, é necessário ter um grupo de buffers do mesmo tamanho de página para associar ao espaço de tabela. Para ter mais informações, consulte Tarefas comuns para grupos de buffer.

Como criar um espaço de tabela

1. Conecte-se ao banco de dados `rdsadmin` usando o nome de usuário principal e a senha mestra da instância de banco de dados do RDS para Db2. No exemplo a seguir, substitua *`master_username`* e *`master_password`* por suas próprias informações.

   ```
   db2 "connect to rdsadmin user master_username using master_password"
   ```

2. Crie um espaço de tabela chamando `rdsadmin.create_tablespace`. Para ter mais informações, consulte rdsadmin.create_tablespace.

   ```
   db2 "call rdsadmin.create_tablespace(
       'database_name',
       'tablespace_name',
       'buffer_pool_name',
       tablespace_initial_size,
       tablespace_increase_size,
   ```

```
    'tablespace_type')"
```

Alterar um espaço de tabela

Para alterar um espaço de tabela do banco de dados do RDS para Db2, chame o procedimento armazenado `rdsadmin.alter_tablespace`. É possível usar esse procedimento armazenado para alterar o grupo de buffers de um espaço de tabela, diminuir a marca d'água alta ou colocar um espaço de tabela on-line. Para ter mais informações, consulte Declaração ALTER TABLESPACE na documentação do IBM Db2.

Como alterar um espaço de tabela

1. Conecte-se ao banco de dados `rdsadmin` usando o nome de usuário principal e a senha mestra da instância de banco de dados do RDS para Db2. No exemplo a seguir, substitua *`master_username`* e *`master_password`* por suas próprias informações.

   ```
   db2 "connect to rdsadmin user master_username using master_password"
   ```

2. Altere um espaço de tabela chamando `rdsadmin.alter_tablespace`. Para ter mais informações, consulte rdsadmin.alter_tablespace.

   ```
   db2 "call rdsadmin.alter_tablespace(
       'database_name',
       'tablespace_name',
       'buffer_pool_name',
       buffer_pool_size,
       tablespace_increase_size,
       'max_size', 'reduce_max',
       'reduce_stop',
       'reduce_value',
       'lower_high_water',
       'lower_high_water_stop',
       'switch_online')"
   ```

Renomear um espaço de tabela

Para alterar o nome de um espaço de tabela do banco de dados do RDS para Db2, chame o procedimento armazenado `rdsadmin.rename_tablespace`. Para ter mais informações, consulte Declaração RENAME TABLESPACE na documentação do IBM Db2.

Como renomear um espaço de tabela

1. Conecte-se ao banco de dados `rdsadmin` usando o nome de usuário principal e a senha mestra da instância de banco de dados do RDS para Db2. No exemplo a seguir, substitua *`master_username`* e *`master_password`* por suas próprias informações.

```
db2 "connect to rdsadmin user master_username using master_password"
```

2. Renomeie um espaço de tabela chamando `rdsadmin.rename_tablespace`. Para ter mais informações, bem como conhecer as restrições em relação a como nomear um espaço de tabela, consulte rdsadmin.rename_tablespace.

```
db2 "call rdsadmin.rename_tablespace(
    'database_name',
    'source_tablespace_name',
    'target_tablespace_name')"
```

Descartar um espaço de tabela

Para descartar um espaço de tabela do banco de dados do RDS para Db2, chame o procedimento armazenado `rdsadmin.drop_tablespace`. Antes de descartar um espaço de tabela, primeiro descarte qualquer objeto no espaço de tabela, como tabelas, índices ou objetos grandes (LOBs). Para ter mais informações, consulte Dropping table spaces na documentação do IBM Db2.

Como descartar um espaço de tabela

1. Conecte-se ao banco de dados `rdsadmin` usando o nome de usuário principal e a senha mestra da instância de banco de dados do RDS para Db2. No exemplo a seguir, substitua *`master_username`* e *`master_password`* por suas próprias informações.

```
db2 "connect to rdsadmin user master_username using master_password"
```

2. Altere um espaço de tabela chamando `rdsadmin.drop_tablespace`. Para ter mais informações, consulte rdsadmin.drop_tablespace.

```
db2 "call rdsadmin.drop_tablespace(
    'database_name',
    'tablespace_name')"
```

Conferir o status de um espaço de tabela

É possível conferir o status de um espaço de tabela usando a função `cast`.

Como conferir o status de um espaço de tabela

1. Conecte-se ao banco de dados Db2 usando o nome de usuário principal e a senha principal da instância de banco de dados do RDS para Db2. No exemplo a seguir, substitua *`rds_database_alias`*, *`master_username`* e *`master_password`* por suas próprias informações.

   ```
   db2 connect to rds_database_alias user master_username using master_password
   ```

2. Exiba uma saída resumida.

   Para exibir uma saída resumida:

   ```
   db2 "select cast(tbsp_id as smallint) as tbsp_id,
   cast(tbsp_name as varchar(35)) as tbsp_name,
   cast(tbsp_type as varchar(3)) as tbsp_type,
   cast(tbsp_state as varchar(10)) as state,
   cast(tbsp_content_type as varchar(8)) as contents from
    table(mon_get_tablespace(null,-1)) order by tbsp_id"
   ```

Exibir informações detalhadas sobre espaços de tabela

É possível exibir informações sobre um espaço de tabela para um membro ou todos os membros usando a função `cast`.

Como exibir informações detalhadas sobre espaços de tabela

1. Conecte-se ao banco de dados Db2 usando o nome de usuário principal e a senha principal da instância de banco de dados do RDS para Db2. No exemplo a seguir, substitua *`rds_database_alias`*, *`master_username`* e *`master_password`* por suas próprias informações.

   ```
   db2 connect to rds_database_alias user master_username using master_password
   ```

2. Exiba detalhes sobre todos os espaços de tabela no banco de dados para um ou todos os membros.

Para um membro:

```
db2 "select cast(member as smallint) as member,
cast(tbsp_id as smallint) as tbsp_id,
cast(tbsp_name as varchar(35)) as tbsp_name,
cast(tbsp_type as varchar(3)) as tbsp_type,
cast(tbsp_state as varchar(10)) as state,
cast(tbsp_content_type as varchar(8)) as contents,
cast(tbsp_total_pages as integer) as total_pages,
cast(tbsp_used_pages as integer) as used_pages,
cast(tbsp_free_pages as integer) as free_pages,
cast(tbsp_page_top as integer) as page_hwm,
cast(tbsp_page_size as integer) as page_sz,
cast(tbsp_extent_size as smallint) as extent_sz,
cast(tbsp_prefetch_size as smallint) as prefetch_sz,
cast(tbsp_initial_size as integer) as initial_size,
cast(tbsp_increase_size_percent as smallint) as increase_pct,
cast(storage_group_name as varchar(12)) as stogroup from
 table(mon_get_tablespace(null,-1)) order by member, tbsp_id "
```

Para todos os membros:

```
db2 "select cast(member as smallint) as member
cast(tbsp_id as smallint) as tbsp_id,
cast(tbsp_name as varchar(35)) as tbsp_name,
cast(tbsp_type as varchar(3)) as tbsp_type,
cast(tbsp_state as varchar(10)) as state,
cast(tbsp_content_type as varchar(8)) as contents,
cast(tbsp_total_pages as integer) as total_pages,
cast(tbsp_used_pages as integer) as used_pages,
cast(tbsp_free_pages as integer) as free_pages,
cast(tbsp_page_top as integer) as page_hwm,
cast(tbsp_page_size as integer) as page_sz,
cast(tbsp_extent_size as smallint) as extent_sz,
cast(tbsp_prefetch_size as smallint) as prefetch_sz,
cast(tbsp_initial_size as integer) as initial_size,
cast(tbsp_increase_size_percent as smallint) as increase_pct,
cast(storage_group_name as varchar(12)) as stogroup from
 table(mon_get_tablespace(null,-2)) order by member, tbsp_id "
```

Listar o estado e o grupo de armazenamento de um espaço de tabela

É possível listar o estado e o grupo de armazenamento de um espaço de tabela ao executar uma declaração SQL.

Para listar o estado e o grupo de armazenamento de um espaço de tabela, execute a seguinte declaração SQL:

```
db2 "SELECT varchar(tbsp_name, 30) as tbsp_name,
                  varchar(TBSP_STATE, 30) state,
                  tbsp_type,
                  varchar(storage_group_name,30) storage_group
FROM TABLE(MON_GET_TABLESPACE('',-2)) AS t"
```

Listar os espaços de uma tabela

É possível listar os espaços de tabela de uma tabela executando uma declaração SQL.

Para listar os espaços de uma tabela, execute a declaração SQL a seguir. No seguinte exemplo, substitua *SCHEMA_NAME* e *TABLE_NAME* pelo nome do esquema e da tabela:

```
db2 "SELECT
   VARCHAR(SD.TBSPACE,30) AS DATA_SPACE,
   VARCHAR(SL.TBSPACE,30) AS LONG_SPACE,
   VARCHAR(SI.TBSPACE,30) AS INDEX_SPACE
 FROM
   SYSCAT.DATAPARTITIONS P
   JOIN SYSCAT.TABLESPACES SD ON SD.TBSPACEID = P.TBSPACEID
   LEFT JOIN SYSCAT.TABLESPACES SL ON SL.TBSPACEID = P.LONG_TBSPACEID
   LEFT JOIN SYSCAT.TABLESPACES SI ON SI.TBSPACEID = P.INDEX_TBSPACEID
 WHERE
    TABSCHEMA = 'SCHEMA_NAME'
 AND TABNAME   = 'TABLE_NAME'"
```

Listar contêineres de espaço de tabela

É possível listar todos os contêineres de espaço de tabela ou contêineres de espaço de tabela específicos usando o comando `cast`.

Como listar os contêineres de um espaço de tabela

1. Conecte-se ao banco de dados Db2 usando o nome de usuário principal e a senha principal da instância de banco de dados do RDS para Db2. No seguinte exemplo, substitua *rds_database_alias*, *master_username* e *master_password* por suas próprias informações:

```
db2 connect to rds_database_alias user master_username using master_password
```

2. Exiba uma lista de todos os contêineres de espaço de tabela no banco de dados ou de contêineres de espaço de tabela específicos.

   Para todos os contêineres de espaço de tabela:

```
db2 "select cast(member as smallint) as member,
cast(tbsp_name as varchar(35)) as tbsp_name,
cast(container_id as smallint) as id,
cast(container_name as varchar(60)) as container_path, container_type as type from
 table(mon_get_container(null,-2)) order by member,tbsp_id,container_id"
```

   Para todos os contêineres de espaço de tabela específicos:

```
db2 "select cast(member as smallint) as member,
cast(tbsp_name as varchar(35)) as tbsp_name,
cast(container_id as smallint) as id,
cast(container_name as varchar(60)) as container_path, container_type as type from
 table(mon_get_container('TBSP_1',-2)) order by member, tbsp_id,container_id"
```

Integrar uma instância de banco de dados do Amazon RDS para Db2 ao Amazon S3

É possível transferir arquivos entre uma instância de banco de dados do Amazon RDS para Db2 e um bucket do Amazon Simple Storage Service (Amazon S3) com os procedimentos armazenados do Amazon RDS. Para ter mais informações, consulte Referência de procedimentos armazenados do Amazon RDS para Db2.

> **Note**
>
> A instância de banco de dados e o bucket do Amazon S3 devem estar na mesma Região da AWS.

Para que o RDS para Db2 se integre ao Amazon S3, a instância de banco de dados deve ter acesso a um bucket do Amazon S3 no qual se encontra o RDS para Db2. Se você não tiver um bucket do S3, crie um bucket.

Tópicos

- Etapa 1: criar uma política do IAM
- Etapa 2: Criar um perfil do IAM e associar a política do IAM
- Etapa 3: Adicionar o perfil do IAM à instância de banco de dados do RDS para Db2

Etapa 1: criar uma política do IAM

Nesta etapa, crie uma política do AWS Identity and Access Management (IAM) com as permissões necessárias para transferir arquivos do bucket do Amazon S3 para a instância de banco de dados do RDS. Esta etapa pressupõe que você já tenha criado um bucket do S3. Para obter mais informações, consulte Como criar um bucket no Guia do usuário do Amazon S3.

Antes de criar a política, observe as seguintes informações:

- Nome do recurso da Amazon (ARN) do bucket.
- O ARN da chave do AWS Key Management Service (AWS KMS), se o bucket usar criptografia SSE-KMS ou SSE-S3.

Crie uma política do IAM que inclua as seguintes permissões:

```
"kms:GenerateDataKey",
"kms:Decrypt",
"s3:PutObject",
"s3:GetObject",
"s3:AbortMultipartUpload",
"s3:ListBucket",
"s3:DeleteObject",
"s3:GetObjectVersion",
"s3:ListMultipartUploadParts"
```

É possível criar uma política do IAM usando o AWS Management Console ou a AWS Command Line Interface (AWS CLI).

Console

Como criar uma política do IAM para permitir acesso do Amazon RDS a um bucket do Amazon S3

1. Faça login no AWS Management Console e abra o console do IAM, em https://console.aws.amazon.com/iam/.
2. No painel de navegação, selecione Políticas.
3. Selecione Criar política e, depois, escolha a guia JSON.
4. Adicione ações por serviço. Para transferir arquivos de um bucket do Amazon S3 para o Amazon RDS, é necessário selecionar permissões de bucket e permissões de objeto.
5. Ampliar os Resources (Recursos). É necessário especificar os recursos de bucket e objeto.
6. Escolha Próximo.
7. Em Nome da política, digite um nome para essa política.
8. (Opcional) Em Descrição, digite uma descrição para essa política.
9. Escolha Criar política.

AWS CLI

Como criar uma política do IAM para permitir acesso do Amazon RDS a um bucket do Amazon S3

1. Execute o comando create-policy. No exemplo a seguir, substitua *iam_policy_name* e *s3_bucket_name* pelo nome da política do IAM e o nome do bucket do Amazon S3 no qual reside o banco de dados do RDS para Db2.

Para Linux, macOS ou Unix:

```
aws iam create-policy \
    --policy-name iam_policy_name \
    --policy-document '{
      "Version": "2012-10-17",
      "Statement": [
        {
          "Effect": "Allow",
          "Action": [
            "kms:GenerateDataKey",
            "kms:Decrypt",
            "s3:PutObject",
            "s3:GetObject",
            "s3:AbortMultipartUpload",
            "s3:ListBucket",
            "s3:DeleteObject",
            "s3:GetObjectVersion",
            "s3:ListMultipartUploadParts"
          ],
          "Resource": [
            "arn:aws:s3:::s3_bucket_name/*",
            "arn:aws:s3:::s3_bucket_name"
          ]
        }
      ]
    }'
```

Para Windows:

```
aws iam create-policy ^
    --policy-name iam_policy_name ^
    --policy-document '{
      "Version": "2012-10-17",
      "Statement": [
        {
          "Effect": "Allow",
            "Action": [
              "s3:PutObject",
              "s3:GetObject",
              "s3:AbortMultipartUpload",
              "s3:ListBucket",
```

```
            "s3:DeleteObject",
            "s3:GetObjectVersion",
            "s3:ListMultipartUploadParts"
          ],
          "Resource": [
            "arn:aws:s3:::s3_bucket_name/*",
            "arn:aws:s3:::s3_bucket_name"
          ]
        }
      ]
    }'
```

2. Depois de criar a política, anote o ARN da política. É necessário ter o ARN do Etapa 2: Criar um perfil do IAM e associar a política do IAM.

Para ter informações sobre como criar uma política do IAM, consulte Criação de políticas do IAM no Guia do usuário do IAM.

Etapa 2: Criar um perfil do IAM e associar a política do IAM

Esta etapa pressupõe que você tenha criado a política do IAM no Etapa 1: criar uma política do IAM. Nesta etapa, você vai criar um perfil do IAM para a instância de banco de dados do RDS para Db2 e associar a política ao perfil.

É possível criar um perfil do IAM para a instância de banco de dados usando o AWS Management Console ou a AWS CLI.

Console

Como criar um perfil do IAM e associar a política do IAM a ele

1. Faça login no AWS Management Console e abra o console do IAM, em https://console.aws.amazon.com/iam/.
2. No painel de navegação, selecione Perfis.
3. Escolha Criar Perfil.
4. Em Tipo de entidade confiável, selecione AWS service (Serviço da AWS).
5. Para Serviço ou caso de uso, selecione RDS e, depois, selecione RDS: Adicionar perfil ao banco de dados.
6. Escolha Próximo.

7. Em Políticas de permissões, pesquise e selecione o nome da política do IAM que você criou.
8. Escolha Próximo.
9. Em Role name (Nome da função), insira um nome.
10. (Opcional) Em Descrição da função, insira uma descrição para a nova função.
11. Selecione Criar perfil.

AWS CLI

Como criar um perfil do IAM e associar a política do IAM a ele

1. Execute o comando create-role. No exemplo a seguir, substitua *iam_role_name* por um nome para o perfil do IAM.

   Para Linux, macOS ou Unix:

```
aws iam create-role \
    --role-name iam_role_name \
    --assume-role-policy-document '{
      "Version": "2012-10-17",
      "Statement": [
        {
          "Effect": "Allow",
          "Principal": {
            "Service": "rds.amazonaws.com"
          },
          "Action": "sts:AssumeRole"
        }
      ]
    }'
```

   Para Windows:

```
aws iam create-role ^
    --role-name iam_role_name ^
    --assume-role-policy-document '{
      "Version": "2012-10-17",
      "Statement": [
        {
          "Effect": "Allow",
          "Principal": {
```

```
            "Service": "rds.amazonaws.com"
          },
          "Action": "sts:AssumeRole"
        }
      ]
    }'
```

2. Depois que a função for criada, observe seu ARN. É necessário ter o ARN do Etapa 3: Adicionar o perfil do IAM à instância de banco de dados do RDS para Db2.
3. Execute o comando attach-role-policy. No exemplo a seguir, substitua *iam_policy_arn* pelo ARN da política do IAM que você criou em Etapa 1: criar uma política do IAM. Substitua *iam_role_name* pelo nome do perfil do IAM que você acabou de criar.

   Para Linux, macOS ou Unix:

   ```
   aws iam attach-role-policy \
      --policy-arn iam_policy_arn \
      --role-name iam_role_name
   ```

   Para Windows:

   ```
   aws iam attach-role-policy ^
      --policy-arn iam_policy_arn ^
      --role-name iam_role_name
   ```

Para ter mais informações, consulte Criar um perfil para delegar permissões a um usuário do IAM no Guia do usuário do IAM.

Etapa 3: Adicionar o perfil do IAM à instância de banco de dados do RDS para Db2

Nesta etapa, você vai adicionar o perfil do IAM à instância de banco de dados do RDS para Db2. Observe os seguintes requisitos:

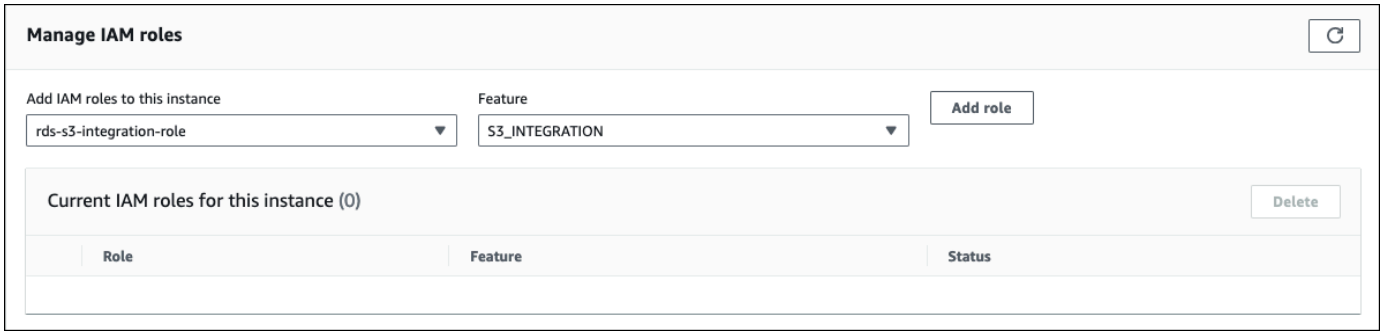
- Você precisa ter acesso a um perfil do IAM com a política de permissões do Amazon S3 necessária anexada a ele.
- Só é possível associar um perfil do IAM à instância de banco de dados do RDS para Db2 por vez.
- A instância de banco de dados do RDS para Db2 deve estar no estado Disponível.

É possível adicionar um perfil do IAM à instância de banco de dados usando o AWS Management Console ou a AWS CLI.

Console

Como adicionar um perfil do IAM à instância de banco de dados do RDS para Db2

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Bancos de dados.
3. Selecione o nome da instância de banco de dados do RDS para Db2.
4. Na guia Connectivity & security (Conectividade e segurança), role para baixo até a seção Manage IAM roles (Gerenciar funções do IAM) na parte inferior da página.
5. Em Adicionar perfis do IAM a essa instância, selecione o perfil que você criou em Etapa 2: Criar um perfil do IAM e associar a política do IAM.
6. Em Feature (Recurso), escolha S3_INTEGRATION.
7. Escolha Add role (adicionar função).



AWS CLI

Para adicionar um perfil do IAM à instância de banco de dados do RDS para Db2, execute o comando add-role-to-db-instance. No exemplo a seguir, substitua *db_instance_name* e *iam_role_arn* pelo nome da instância de banco de dados e pelo ARN do perfil do IAM criado em Etapa 2: Criar um perfil do IAM e associar a política do IAM.

Para Linux, macOS ou Unix:

```
aws rds add-role-to-db-instance \
    --db-instance-identifier db_instance_name \
    --feature-name S3_INTEGRATION \
```

```
    --role-arn iam_role_arn \
```

Para Windows:

```
aws rds add-role-to-db-instance ^
    --db-instance-identifier db_instance_name ^
    --feature-name S3_INTEGRATION ^
    --role-arn iam_role_arn ^
```

Para confirmar que o perfil foi adicionado com êxito à instância de banco de dados do RDS para Db2, execute o comando describe-db-instances. No exemplo a seguir, substitua *db_instance_name* pelo nome da instância de banco de dados.

Para Linux, macOS ou Unix:

```
aws rds describe-db-instances \
    --filters "Name=db-instance-id,Values=db_instance_name" \
    --query 'DBInstances[].AssociatedRoles'
```

Para Windows:

```
aws rds describe-db-instances ^
    --filters "Name=db-instance-id,Values=db_instance_name" ^
    --query 'DBInstances[].AssociatedRoles'
```

O comando gerará uma saída semelhante ao seguinte exemplo:

```
[
    [
        {
            "RoleArn": "arn:aws:iam::0123456789012:role/rds-db2-s3-role",
            "FeatureName": "S3_INTEGRATION",
            "Status": "ACTIVE"
        }
    ]
]
```

Migrar dados para o Amazon RDS para Db2

É possível migrar bancos de dados do Db2 autogerenciados para o Amazon RDS para Db2 usando ferramentas da AWS ou nativas do Db2.

Consulte informações sobre como migrar do banco de dados do Db2 para o Amazon RDS para Db2 usando serviços da AWS em Usar serviços da AWS para migrar dados do Db2 para o Amazon RDS para Db2.

Consulte informações sobre como migrar do banco de dados do Db2 para o Amazon RDS para Db2 usando ferramentas nativas do Db2 em Usar ferramentas nativas do Db2 para migrar dados do Db2 para o Amazon RDS para Db2.

Usar serviços da AWS para migrar dados do Db2 para o Amazon RDS para Db2

No Amazon RDS, há algumas maneiras de migrar dados de um banco de dados do Db2 para o Amazon RDS para Db2. É possível realizar uma migração única do banco de dados do Db2 de ambientes Linux, AIX ou Windows para o Amazon RDS para Db2. Para minimizar o tempo de inatividade, é possível realizar uma migração de tempo de inatividade quase zero. É possível migrar os dados salvando-os no Amazon S3 e carregando-os uma tabela por vez no banco de dados do Db2. Também é possível realizar uma migração síncrona por meio da replicação ou do uso do AWS Database Migration Service.

Para migrações únicas para bancos de dados do Db2 baseados no Linux, o Amazon RDS só permite backups off-line e on-line. O Amazon RDS não permite incrementos e backups Delta. Para migrações de tempo de inatividade quase nulas para bancos de dados do Db2 baseados no Linux, o Amazon RDS exige backups on-line. Recomendamos que você use backups on-line para migrações com tempo de inatividade quase zero e backups off-line para migrações que possam lidar com o tempo de inatividade.

Tópicos

- Migrar do Linux para Linux para o Amazon RDS para Db2
- Migrar do Linux para Linux com quase zero tempo de inatividade para o Amazon RDS para Db2
- Migrar de forma síncrona de Linux para Linux no Amazon RDS para Db2
- Migrar do AIX ou do Windows para Linux para o Amazon RDS para Db2

- Migrar dados do Db2 por meio do Amazon S3 para o Amazon RDS para Db2
- Migrar dados para o Amazon RDS para Db2 com o AWS Database Migration Service (AWS DMS)

Migrar do Linux para Linux para o Amazon RDS para Db2

Com essa abordagem de migração, é feito backup do banco de dados do Db2 autogerenciado em um bucket do Amazon S3. Depois, use os procedimentos armazenados do Amazon RDS para restaurar o banco de dados do Db2 em uma instância de banco de dados do Amazon RDS para Db2. Para ter mais informações sobre como usar o Amazon S3, consulte Integrar uma instância de banco de dados do Amazon RDS para Db2 ao Amazon S3.

O backup e a restauração do RDS para Db2 seguem os caminhos e as restrições de atualização compatíveis do IBM Db2. Para ter mais informações, consulte Supported upgrade paths for Db2 servers e Upgrade restrictions for Db2 servers na documentação do IBM Db2.

Tópicos

- Limitações e recomendações para usar a restauração nativa
- Fazer backup do banco de dados no Amazon S3
- Criar um grupo de armazenamento automático padrão
- Restaurar o bancos de dados do Db2

Limitações e recomendações para usar a restauração nativa

As seguintes limitações e recomendações se aplicam ao uso da restauração nativa:

- O Amazon RDS só comporta a migração de versões on-premises do Db2 que correspondam às versões compatíveis do RDS para Db2. Para ter mais informações sobre as versões compatíveis, consulte Versões secundárias do Db2 compatíveis com o Amazon RDS.
- O Amazon RDS só permite backups off-line e on-line para restauração nativa. O Amazon RDS não permite incrementos ou backups Delta.
- Não é possível restaurar de um bucket do Amazon S3 em uma Região da AWS diferente da região onde a instância de banco de dados do RDS para Db2 está localizada.
- O Amazon S3 limita o tamanho dos arquivos carregados para um bucket do Amazon S3 a 5 TB. Se um arquivo de backup de banco de dados exceder 5 TB, divida o arquivo de backup em arquivos menores.

- O Amazon RDS não comporta rotinas externas não protegidas, restaurações incrementais nem restaurações Delta.
- Não é possível restaurar de um banco de dados de origem criptografado, mas você pode restaurar para uma instância de banco de dados do Amazon RDS.

Ao restaurar o banco de dados, o backup é copiado e extraído na instância de banco de dados do RDS para Db2. Recomendamos provisionar espaço de armazenamento para a instância de banco de dados do RDS para Db2 igual ou maior que a soma do tamanho do backup, além do tamanho do banco de dados original no disco.

O tamanho máximo do banco de dados restaurado é o tamanho máximo do banco de dados compatível menos o tamanho do backup. Por exemplo, se o tamanho máximo do banco de dados compatível for de 64 TiB e o tamanho do backup for de 30 TiB, o tamanho máximo do banco de dados restaurado será de 34 TiB.

```
64 TiB - 30 TiB = 34 TiB
```

Fazer backup do banco de dados no Amazon S3

Para fazer backup do banco de dados no Amazon S3, são necessários os seguintes componentes da AWS:

- Um bucket do Amazon S3 para armazenar arquivos de backup: faça upload de todos os arquivos de backup que você deseja migrar para o Amazon RDS. Recomendamos que você use backups off-line para migrações que possam lidar com o tempo de inatividade. Se você já tiver um bucket do Amazon S3, poderá usá-lo. Se você não tiver um bucket do S3, consulte Criar um bucket no Guia do usuário do Amazon S3.

  > **Note**
  >
  > Se o banco de dados for grande e levar muito tempo para ser transferido para um bucket do S3, será possível solicitar um dispositivo AWS Snow Family e pedir para que a AWS realize o backup. Depois de copiar os arquivos para o dispositivo e devolvê-los à equipe da família Snow, a equipe vai transferir as imagens de backup para o bucket do S3. Para obter mais informações, consulte a documentação do AWS Snow Family.

- Um perfil do IAM para acessar o bucket do S3: se você já tem um perfil do IAM, pode usá-lo. Se não tiver um perfil, consulte Etapa 2: Criar um perfil do IAM e associar a política do IAM.

- Uma política do IAM com relações de confiança e permissões anexadas ao perfil do IAM: consulte mais informações em Etapa 1: criar uma política do IAM.
- O perfil do IAM adicionado à instância de banco de dados do RDS para Db2: consulte mais informações em Etapa 3: Adicionar o perfil do IAM à instância de banco de dados do RDS para Db2.

Criar um grupo de armazenamento automático padrão

O banco de dados de origem deve ter um grupo de armazenamento automático padrão. Se o banco de dados não tiver um grupo de armazenamento automático padrão, você deverá criar um.

Como criar um grupo de armazenamento automático padrão

1. Conecte-se ao seu banco de dados de origem. No exemplo a seguir, substitua *source_database* pelo nome do banco de dados.

```
db2 connect to source_database
```

2. Crie um grupo de armazenamento automático e defina-o como padrão. No exemplo a seguir, substitua *storage_path* pelo caminho absoluto de onde o grupo de armazenamento está localizado.

```
db2 "create stogroup IBMSTOGROUP ON storage_path set as default"
```

3. Encerre os processos de backend.

```
db2 terminate
```

4. Desative o banco de dados e interrompa todos os serviços dele. No exemplo a seguir, substitua *source_database* pelo nome do banco de dados para o qual você criou o grupo de armazenamento.

```
db2 deactivate db source_database
```

5. Faça backup do bancos de dados. No exemplo a seguir, substitua *source_database* pelo nome do banco de dados para o qual você criou o grupo de armazenamento. Substitua *file_system_path* pelo caminho absoluto para onde você deseja fazer backup do banco de dados.

```
db2 backup database source_database to file_system_path
```

Restaurar o bancos de dados do Db2

Depois de fazer backup do banco de dados no Amazon S3 e criar um grupo de armazenamento automático, estará tudo pronto para restaurar o banco de dados do Db2 para a instância de banco de dados do RDS para Db2.

Como restaurar o banco de dados do Db2 na instância de banco de dados do RDS para Db2

1. Conecte-se à instância de banco de dados do RDS para Db2. Para ter mais informações, consulte Conectar-se à instância de banco de dados do Db2.
2. (Opcional) Para garantir que o banco de dados tenha as configurações ideais definidas para a operação de restauração, é possível chamar the section called "rdsadmin.show_configuration" para conferir os valores de `RESTORE_DATABASE_PARALLELISM` e `RESTORE_DATABASE_NUM_BUFFERS`. Chame the section called "rdsadmin.set_configuration" para alterar esses valores, conforme necessário. Definir explicitamente esses valores pode melhorar a performance ao restaurar bancos de dados com grandes volumes de dados.
3. Restaure o banco de dados chamando `rdsadmin.restore_database`. Para ter mais informações, consulte rdsadmin.restore_database.

Migrar do Linux para Linux com quase zero tempo de inatividade para o Amazon RDS para Db2

Com essa abordagem de migração, você vai migrar um banco de dados do Db2 baseado em Linux de um banco de dados do Db2 autogerenciado (origem) para o Amazon RDS para Db2. Essa abordagem ocasiona mínima ou nenhuma interrupção ou tempo de inatividade da aplicação ou dos usuários. Essa abordagem faz backup do banco de dados e o restaura com a reprodução do log, o que ajuda a evitar interrupções nas operações contínuas e oferece alta disponibilidade do banco de dados.

Para ter uma migração com tempo de inatividade quase zero, o RDS para Db2 implementa a restauração com reprodução de logs. Essa abordagem faz backup do banco de dados do DB2 baseado em Linux autogerenciado e o restaura no servidor RDS para Db2. Com os procedimentos armazenados do Amazon RDS, você aplica os logs de transações subsequentes para atualizar o banco de dados.

Tópicos

- Limitações e recomendações de migração com tempo de inatividade quase zero
- Fazer backup do banco de dados no Amazon S3
- Criar um grupo de armazenamento automático padrão
- Migrar o bancos de dados do Db2

Limitações e recomendações de migração com tempo de inatividade quase zero

As seguintes limitações e recomendações se aplicam à utilização da migração de tempo de inatividade quase zero:

- O Amazon RDS exige um backup on-line para uma migração com tempo de inatividade quase zero. Isso ocorre porque o Amazon RDS mantém o banco de dados em um estado pendente de rollforward à medida que você faz upload dos logs de transações arquivados. Para ter mais informações, consulte the section called “Migrar o bancos de dados do Db2”.
- Não é possível restaurar de um bucket do Amazon S3 em uma Região da AWS diferente da região onde a instância de banco de dados do RDS para Db2 está localizada.
- O Amazon S3 limita o tamanho dos arquivos carregados para um bucket do S3 a 5 TB. Se um arquivo de backup de banco de dados exceder 5 TB, divida o arquivo de backup em arquivos menores.
- O Amazon RDS não comporta rotinas externas não protegidas, restaurações incrementais nem restaurações Delta.
- Não é possível restaurar de um banco de dados de origem criptografado, mas você pode restaurar para uma instância de banco de dados do Amazon RDS.

Ao restaurar o banco de dados, o Amazon RDS copia o backup e o extrai na instância de banco de dados do RDS para Db2. Recomendamos provisionar espaço de armazenamento para a instância de banco de dados do RDS para Db2 igual ou maior que a soma do tamanho do backup, além do tamanho do banco de dados original no disco.

O tamanho máximo do banco de dados restaurado é o tamanho máximo do banco de dados compatível menos o tamanho do backup. Por exemplo, se o tamanho máximo do banco de dados compatível for de 64 TiB e o tamanho do backup for de 30 TiB, o tamanho máximo do banco de dados restaurado será de 34 TiB.

```
64 TiB - 30 TiB = 34 TiB
```

Fazer backup do banco de dados no Amazon S3

Para fazer backup do banco de dados no Amazon S3, são necessários os seguintes componentes da AWS:

- Um bucket do Amazon S3 para armazenar arquivos de backup: faça upload de todos os arquivos de backup que você deseja migrar para o Amazon RDS. O Amazon RDS exige um backup on-line para uma migração com tempo de inatividade quase zero. Se você já tiver um bucket do Amazon S3, poderá usá-lo. Se você não tiver um bucket do S3, consulte Criar um bucket no Guia do usuário do Amazon S3.

  > **Note**
  >
  > Se o banco de dados for grande e levar muito tempo para ser transferido para um bucket do S3, será possível solicitar um dispositivo AWS Snow Family e pedir para que a AWS realize o backup. Depois de copiar os arquivos para o dispositivo e devolvê-los à equipe da família Snow, a equipe vai transferir as imagens de backup para o bucket do S3. Para obter mais informações, consulte a documentação do AWS Snow Family.

- Um perfil do IAM para acessar o bucket do S3: se você já tem um perfil do AWS Identity and Access Management (IAM), pode usá-lo. Se não tiver um perfil, consulte Etapa 2: Criar um perfil do IAM e associar a política do IAM.
- Uma política do IAM com relações de confiança e permissões anexadas ao perfil do IAM: consulte mais informações em Etapa 1: criar uma política do IAM.
- O perfil do IAM adicionado à instância de banco de dados do RDS para Db2: consulte mais informações em Etapa 3: Adicionar o perfil do IAM à instância de banco de dados do RDS para Db2.

Criar um grupo de armazenamento automático padrão

O banco de dados de origem deve ter um grupo de armazenamento automático padrão. Se o banco de dados não tiver um grupo de armazenamento automático padrão, você deverá criar um.

Como criar um grupo de armazenamento automático padrão

1. Conecte-se ao seu banco de dados de origem. No exemplo a seguir, substitua *source_database* pelo nome do banco de dados.

```
db2 connect to source_database
```

2. Crie um grupo de armazenamento automático e defina-o como padrão. No exemplo a seguir, substitua *storage_path* pelo caminho absoluto de onde o grupo de armazenamento está localizado.

```
db2 "create stogroup IBMSTOGROUP ON storage_path set as default"
```

3. Encerre os processos de backend.

```
db2 terminate
```

Migrar o bancos de dados do Db2

Depois de fazer backup do banco de dados no Amazon S3 e criar um grupo de armazenamento automático, estará tudo pronto para migrar o banco de dados do Db2 para a instância de banco de dados do RDS para Db2.

Como realizar uma migração com tempo de inatividade quase zero

1. Execute um backup on-line do banco de dados de origem. Para ter mais informações, consulte BACKUP DATABASE command na documentação do IBM Db2.
2. Copie o backup do banco de dados para um bucket do Amazon S3. Para ter informações sobre o Amazon S3, consulte o Guia do usuário do Amazon Simple Storage Service.
3. Conecte-se ao servidor `rdsadmin` com o *master_username* e a *master_password* da instância de banco de dados do RDS para Db2.

```
db2 connect to rdsadmin user master_username using master_password
```

4. (Opcional) Para garantir que o banco de dados tenha as configurações ideais definidas para a operação de restauração, é possível chamar the section called “rdsadmin.show_configuration” para conferir os valores de `RESTORE_DATABASE_PARALLELISM` e `RESTORE_DATABASE_NUM_BUFFERS`. Chame the section called “rdsadmin.set_configuration”

para alterar esses valores, conforme necessário. Definir explicitamente esses valores pode melhorar a performance ao restaurar bancos de dados com grandes volumes de dados.

5. Restaure o backup no servidor RDS para Db2 chamando `rdsadmin.restore_database`. Defina `backup_type` como `ONLINE`. Para ter mais informações, consulte rdsadmin.restore_database.
6. Copie os logs de arquivamento do servidor de origem para o bucket do S3. Para ter mais informações, consulte Archive logging na documentação do IBM Db2.
7. Aplique logs de arquivamento quantas vezes for necessário chamando `rdsadmin.rollforward_database`. Defina `complete_rollforward` como `FALSE` para manter o banco de dados em um estado `ROLL-FORWARD PENDING`. Para ter mais informações, consulte rdsadmin.rollforward_database.
8. Depois de aplicar todos os logs de arquivamento, coloque o banco de dados on-line chamando `rdsadmin.complete_rollforward`. Para ter mais informações, consulte rdsadmin.complete_rollforward.
9. Alterne as conexões da aplicação para o servidor RDS para Db2 atualizando os endpoints da aplicação para o banco de dados ou atualizando os endpoints do DNS para redirecionar o tráfego para o servidor RDS para Db2. Também é possível usar o recurso de redirecionamento automático do cliente Db2 no banco de dados do Db2 autogerenciado com o endpoint do banco de dados do RDS para Db2. Para ter mais informações, consulte Automatic client reroute description and setup na documentação do IBM Db2.
10. (Opcional) Encerre o banco de dados de origem.

Migrar de forma síncrona de Linux para Linux no Amazon RDS para Db2

Com essa abordagem de migração, é possível configurar a replicação entre o banco de dados do Db2 autogerenciado e a instância de banco de dados do Amazon RDS para Db2. As alterações feitas no banco de dados autogerenciado são replicadas na instância de banco de dados do RDS para Db2 quase em tempo real. Essa abordagem pode oferecer disponibilidade contínua e minimizar o tempo de inatividade durante o processo de migração.

Migrar do AIX ou do Windows para Linux para o Amazon RDS para Db2

Com essa abordagem de migração, é necessário ferramentas nativas do Db2 para fazer backup do banco de dados do Db2 autogerenciado em um bucket do Amazon S3. As ferramentas nativas do Db2 incluem o utilitário `export`, o comando `db2move` do sistema ou o comando do sistema `db2look`. O banco de dados do Db2 pode ser autogerenciado ou no Amazon Elastic Compute Cloud

(Amazon EC2). É possível mover dados do sistema AIX ou Windows para o bucket do Amazon S3. Depois, use um cliente Db2 para carregar dados diretamente do bucket do S3 para o banco de dados do Amazon RDS para Db2. O tempo de inatividade depende do tamanho do banco de dados. Para ter mais informações sobre como usar o Amazon S3, consulte Integrar uma instância de banco de dados do Amazon RDS para Db2 ao Amazon S3.

Como migrar o banco de dados do Db2 para o RDS para Db2

1. Prepare-se para fazer backup do banco de dados. Configure uma quantidade de armazenamento suficiente para manter o backup no sistema Db2 autogerenciado.
2. Faça backup do banco de dados.
    a. Execute o comando db2look do sistema para extrair o arquivo de linguagem de definição de dados (DDL) de todos os objetos.
    b. Execute o utilitário de exportação do Db2, o comando db2move do sistema ou uma declaração CREATE EXTERNAL TABLE para descarregar os dados da tabela do Db2 no armazenamento no sistema Db2.
3. Mova o backup para um bucket do Amazon S3. Para ter mais informações, consulte Integrar uma instância de banco de dados do Amazon RDS para Db2 ao Amazon S3.

    > **Note**
    >
    > Se o banco de dados for grande e levar muito tempo para ser transferido para um bucket do S3, será possível solicitar um dispositivo AWS Snow Family e pedir para que a AWS realize o backup. Depois de copiar os arquivos para o dispositivo e devolvê-los à equipe da família Snow, a equipe vai transferir as imagens de backup para o bucket do S3. Para obter mais informações, consulte a documentação do AWS Snow Family.

4. Depois, use um cliente Db2 para carregar dados diretamente do bucket do S3 para o banco de dados do RDS para Db2. Para ter mais informações, consulte Migrar com o Amazon S3.

Migrar dados do Db2 por meio do Amazon S3 para o Amazon RDS para Db2

Com essa abordagem de migração, primeiro salve dados de uma única tabela em um arquivo de dados que é colocado em um bucket do Amazon S3. Depois, use o comando LOAD para carregar os dados desse arquivo em uma tabela no banco de dados do Amazon RDS para Db2. Para ter mais

informações sobre como usar o Amazon S3, consulte Integrar uma instância de banco de dados do Amazon RDS para Db2 ao Amazon S3.

Tópicos

- Salvar dados no Amazon S3
- Carregar dados em tabelas do RDS para Db2

Salvar dados no Amazon S3

Para salvar dados de uma única tabela no Amazon S3, use um utilitário de banco de dados para extrair os dados do sistema de gerenciamento de banco de dados (DBMS) em um arquivo CSV. Depois, faça upload de cada arquivo de dados no Amazon S3.

Para armazenar arquivos de dados no Amazon S3, são necessários os seguintes componentes da AWS:

- Um bucket do Amazon S3 para armazenar arquivos de backup: se você já tiver um bucket do S3, poderá usá-lo. Se você não tiver um bucket do S3, consulte Criar um bucket no Guia do usuário do Amazon S3.
- Um perfil do IAM para acessar o bucket do S3: se você já tem um perfil do IAM, pode usá-lo. Se não tiver um perfil, consulte Etapa 2: Criar um perfil do IAM e associar a política do IAM.
- Uma política do IAM com relações de confiança e permissões anexadas ao perfil do IAM: consulte mais informações em Etapa 1: criar uma política do IAM.
- O perfil do IAM adicionado à instância de banco de dados do RDS para Db2: consulte mais informações em Etapa 3: Adicionar o perfil do IAM à instância de banco de dados do RDS para Db2.

Carregar dados em tabelas do RDS para Db2

Depois de salvar os arquivos de dados no Amazon S3, é possível carregar os dados desses arquivos em tabelas individuais na instância de banco de dados do RDS para Db2.

Como carregar dados da tabela do Db2 na tabela do banco de dados do RDS para Db2

1. Conecte-se ao banco de dados `rdsadmin` usando o nome de usuário principal e a senha mestra da instância de banco de dados do RDS para Db2. No exemplo a seguir, substitua *`master_username`* e *`master_password`* por suas próprias informações.

```
db2 connect to rdsadmin user master_username using master_password
```

2. Catalogue um alias de acesso ao armazenamento que aponte para o bucket do Amazon S3 onde os arquivos salvos são armazenados. Anote o nome desse alias para uso na próxima etapa. Só é necessário realizar essa etapa uma vez se planeja carregar várias tabelas de arquivos de dados armazenados no mesmo bucket do Amazon S3.

   O exemplo a seguir cataloga um alias chamado *my_s3_alias*, que concede a um usuário chamado *jorge_souza* acesso a um bucket denominado *amzn-s3-demo-bucket*.

```
db2 "call rdsadmin.catalog_storage_access(?, 'my_s3_alias', 'amzn-s3-demo-bucket',
 'USER', 'jorge_souza')"
```

   Para ter mais informações sobre esse procedimento armazenado, consulte the section called “rdsadmin.catalog_storage_access”.

3. Execute o comando LOAD usando o alias de acesso ao armazenamento que aponta para o bucket do Amazon S3.

   > **Note**
   >
   > Se o comando LOAD gerar um erro, talvez seja necessário criar um endpoint de gateway da VPC para o Amazon S3 e adicionar regras de saída ao grupo de segurança. Para ter mais informações, consulte the section called “Erro de E/S de arquivo”.

   O exemplo a seguir carrega dados de um arquivo chamado *my_s3_datafile.csv* em uma tabela chamada *my_db2_table*. O exemplo pressupõe que o arquivo de dados esteja no bucket do Amazon S3 para o qual o alias chamando *my_s3_alias* aponta.

```
db2 "load from db2remote://my_s3_alias//my_s3_datafile.csv of DEL insert
 into my_db2_table";
```

   O exemplo a seguir carrega LOBs de um arquivo de dados chamado *my_table1_export.ixf* em uma tabela denominada *my_db2_table*. O exemplo pressupõe que o arquivo de dados esteja no bucket do Amazon S3 para o qual o alias chamando *my_s3_alias* aponta.

```
db2 "call sysproc.admin_cmd('load from
 "db2remote://my_s3_alias//my_table1_export.ixf" of ixf
        lobs from "db2remote://my_s3_alias//" xml from "db2remote://my_s3_alias//"
        modified by lobsinfile implicitlyhiddeninclude identityoverride
 generatedoverride periodoverride transactionidoverride
        messages on server
        replace into "my_schema"."my_db2_table"
                                nonrecoverable
        indexing mode incremental allow no access')"
```

Repita essa etapa para cada arquivo de dados do bucket do Amazon S3 que você deseja carregar em uma tabela na instância de banco de dados do RDS para Db2.

Para ter mais informações sobre o comando LOAD, consulte Comando LOAD.

Migrar dados para o Amazon RDS para Db2 com o AWS Database Migration Service (AWS DMS)

É possível usar o AWS DMS para migrações únicas e, depois, sincronizar do Db2 no Linux, no Unix (como AIX) e no Windows para o Amazon RDS para Db2. Para obter mais informações, consulte O que é o AWS Database Migration Service?.

Usar ferramentas nativas do Db2 para migrar dados do Db2 para o Amazon RDS para Db2

É possível usar várias ferramentas, utilitários e comandos nativos do Db2 para mover dados diretamente de um banco de dados do Db2 para um banco de dados do Amazon RDS para Db2. Para usar essas ferramentas nativas do Db2, é necessário ser capaz de conectar a máquina cliente a uma instância de banco de dados do RDS para Db2. Para ter mais informações, consulte Conectar uma máquina cliente a uma instância de banco de dados do Amazon RDS para Db2.

> **Note**
>
> Outra forma de mover os dados é primeiro salvá-los em um bucket do Amazon S3 e, depois, usar o comando LOAD para transferir esses dados para uma tabela no banco de dados do RDS para Db2. Esse método fornece o melhor performance ao migrar um grande volume

de dados devido à boa conectividade de rede entre o RDS para Db2 e o S3. Para ter mais informações, consulte the section called “Migrar com o Amazon S3”.

Nome da ferramenta	Caso de uso	Limitações
db2look	Copiar metadados de um banco de dados do Db2 autogerenciado em um banco de dados do RDS para Db2.	• É necessário modificar a sintaxe para criar grupos de buffer, espaços de tabela e perfis para correspon der à sintaxe usada pelo Procedimentos armazenad os do RDS para Db2.
Comando da IMPORT	Migrar tabelas pequenas e tabelas com objetos grandes (LOBs) de uma máquina cliente para a instância de banco de dados do RDS para Db2.	• Mais lento que o utilitário `LOAD` devido às operações de registro em log `INSERT` e `DELETE`. • Baixa performance com largura de banda da rede limitada.
Utilitário INGEST	Transmitir de forma contínua dados de arquivos e canais sem objetos grandes (LOBs) na máquina cliente para a instância de banco de dados do RDS para Db2. É compatível com as operações `INSERT` e `MERGE`.	• Não é possível transmiti r arquivos de dados que contenham LOBs. Em vez disso, use o comando `IMPORT`. • Conectividade necessári a entre o banco de dados do Db2 autogerenciado e o banco de dados do RDS para Db2.
Comando da INSERT	Copiar dados em tabelas pequenas de um banco de dados do Db2 autogerenciado	• Conectividade necessári a entre o banco de dados do Db2 autogerenciado e

Nome da ferramenta	Caso de uso	Limitações
	em um banco de dados do RDS para Db2.	o banco de dados do RDS para Db2. • Baixa performance com largura de banda da rede limitada.
Comando da LOAD CLIENT	Migrar tabelas pequenas sem objetos grandes (LOBs) de uma máquina cliente para a instância de banco de dados do RDS para Db2.	• Não é possível migrar arquivos de dados que contenham LOBs. Em vez disso, use o comando `IMPORT`. • Baixa performance com largura de banda da rede limitada.

Conectar uma máquina cliente a uma instância de banco de dados do Amazon RDS para Db2

Para usar qualquer uma das ferramentas nativas do Db2 para mover dados de um banco de dados do Db2 para um banco de dados do Amazon RDS para Db2, primeiro é necessário conectar a máquina cliente a uma instância de banco de dados do RDS para Db2.

A máquina cliente pode ser qualquer uma das seguintes:

- Uma instância do Amazon Elastic Compute Cloud (Amazon EC2) no Linux, no Windows ou no macOS. Essa instância deve estar na mesma nuvem privada virtual (VPC) que a instância do banco de dados do RDS para Db2, AWS Cloud9 ou AWS CloudShell.
- Uma instância do Db2 autogerenciada em uma instância do Amazon EC2. As instâncias devem estar na mesma VPC.
- Uma instância do Db2 autogerenciada em uma instância do Amazon EC2. As instâncias podem estar em VPCs diferentes se você habilitou o emparelhamento de VPC. Para ter mais informações, consulte Create a VPC peering connection no Guia de emparelhamento de Amazon Virtual Private Cloud.

- Uma máquina local executando Linux, Windows ou macOS em um ambiente autogerenciado. É necessário ter conectividade pública com o RDS para Db2 ou habilitar a conectividade de VPN entre instâncias autogerenciadas do Db2 e a AWS.

Para conectar a máquina cliente à instância de banco de dados do RDS para Db2, faça login na máquina cliente com o IBM Db2 Data Management Console. Para ter mais informações, consulte Criar uma instância de banco de dados do Amazon RDS e IBM Db2 Data Management Console.

É possível usar AWS Database Migration Service (AWS DMS) para executar consultas no banco de dados, executar um plano de execução SQL e monitorar o banco de dados. Para ter mais informações, consulte What is AWS Database Migration Service? no Guia do usuário do AWS Database Migration Service.

Depois de conectar com êxito a máquina cliente à instância de banco de dados do RDS para Db2, estará tudo pronto para usar qualquer ferramenta nativa do Db2 para copiar dados. Para ter mais informações, consulte Usar ferramentas nativas do Db2 para migrar dados do Db2 para o Amazon RDS para Db2.

Copiar metadados do banco de dados do Db2 para o Amazon RDS para Db2 com db2look

`db2look` é uma ferramenta nativa do Db2 que extrai arquivos, objetos, autorizações, configurações, WLM e layouts de banco de dados da linguagem de definição de dados (DDL). É possível usar o `db2look` para copiar metadados de um banco de dados do Db2 autogerenciado para um banco de dados do Amazon RDS para Db2. Para ter mais informações, consulte Mimicking databases using db2look na documentação do IBM Db2.

Como copiar os metadados do banco de dados

1. Execute a ferramenta `db2look` no sistema do Db2 autogerenciado para extrair o arquivo DDL. No exemplo a seguir, substitua *`database_name`* pelo nome do banco de dados do Db2.

   ```
   db2look -d database_name -e -l -a -f -wlm -cor -createdb -printdbcfg -o db2look.sql
   ```

2. Se a máquina cliente tiver acesso ao banco de dados de origem (Db2 autogerenciado) e à instância de banco de dados do RDS para Db2, será possível criar o arquivo `db2look.sql` na máquina cliente conectando-se diretamente à instância remota. Depois, catalogue a instância remota autogerenciada do Db2.

a. Catalogue o nó. No exemplo a seguir, substitua *dns_ip_address* e *port* pelo nome do DNS ou pelo endereço IP e pelo número da porta do banco de dados do Db2 autogerenciado.

```
db2 catalog tcpip node srcnode REMOTE dns_ip_address server port
```

b. Catalogue o banco de dados. No exemplo a seguir, substitua *source_database_name* e *source_database_alias* pelo nome do banco de dados do Db2 autogerenciado e pelo alias a ser utilizado para esse banco de dados.

```
db2 catalog database source_database_name as source_database_alias at node
 srcnode \
    authentication server_encrypt
```

c. Associe ao banco de dados de origem. No exemplo a seguir, substitua *source_database_alias*, *user_id* e *user_password* pelo alias que você criou na etapa anterior e pelo ID de usuário e senha do banco de dados do Db2 autogerenciado.

```
db2look -d source_database_alias -i user_id -w user_password -e -l -a -f -wlm \
    -cor -createdb -printdbcfg -o db2look.sql
```

3. Se você não conseguir acessar o banco de dados do Db2 autogerenciado remoto da máquina cliente, copie o arquivo `db2look.sql` para a máquina cliente. Depois, catalogue a instância de banco de dados do RDS para Db2.

a. Catalogue o nó. No exemplo a seguir, substitua *dns_ip_address* e *port* pelo nome do DNS ou pelo endereço IP e pelo número da porta da instância de banco de dados do RDS para Db2.

```
db2 catalog tcpip node remnode REMOTE dns_ip_address server port
```

b. Catalogue o banco de dados. No exemplo a seguir, substitua *rds_database_name* e *rds_database_alias* pelo nome do banco de dados do RDS para Db2 e pelo alias a ser utilizado para esse banco de dados.

```
db2 catalog database rds_database_name as rds_database_alias at node remnode \
    authentication server_encrypt
```

c. Catalogue o banco de dados administrativo que gerencia o RDS para Db2. Não é possível usar esse banco de dados para armazenar dados.

```
db2 catalog database rdsadmin as rdsadmin at node remnode authentication
 server_encrypt
```

4. Crie grupos de buffer e espaços de tabela. O administrador não tem privilégios para criar grupos de buffer nem espaços de tabela. No entanto, é possível usar procedimentos armazenados do Amazon RDS para criá-los.

a. Encontre os nomes e as definições dos grupos de buffer e dos espaços de tabela no arquivo `db2look.sql`.

b. Conecte-se ao Amazon RDS usando o nome de usuário principal e a senha mestra da instância de banco de dados do RDS para Db2. No exemplo a seguir, substitua *`master_username`* e *`master_password`* por suas próprias informações.

```
db2 connect to rdsadmin user master_username using master_password
```

c. Crie um grupo de buffers chamando `rdsadmin.create_bufferpool`. Para ter mais informações, consulte rdsadmin.create_bufferpool.

```
db2 "call rdsadmin.create_bufferpool(
    'database_name',
    'buffer_pool_name',
    buffer_pool_size,
    'immediate',
    'automatic',
    page_size,
    number_block_pages,
    block_size)"
```

d. Crie um espaço de tabela chamando `rdsadmin.create_tablespace`. Para ter mais informações, consulte rdsadmin.create_tablespace.

```
db2 "call rdsadmin.create_tablespace(
    'database_name',
    'tablespace_name',
    'buffer_pool_name',
    tablespace_initial_size,
    tablespace_increase_size,
```

```
    'tablespace_type')"
```

e. Repita as etapas c ou d para cada grupo de buffers ou espaço de tabela extra que você deseja adicionar.

f. Encerre a conexão.

```
db2 terminate
```

5. Crie tabelas e objetos.

a. Conecte-se ao banco de dados do RDS para Db2 usando o nome de usuário principal e a senha mestra da instância de banco de dados do RDS para Db2. No exemplo a seguir, substitua *rds_database_name*, *master_username* e *master_password* por suas próprias informações.

```
db2 connect to rds_database_name user master_username using master_password
```

b. Execute o arquivo `db2look.sql`.

```
db2 -tvf db2look.sql
```

c. Encerre a conexão.

```
db2 terminate
```

Importar dados de uma máquina cliente para o Amazon RDS para Db2 com o comando IMPORT

É possível usar o comando `IMPORT` de uma máquina cliente para importar os dados para o servidor Amazon RDS para Db2.

> **Important**
>
> O método de comando `IMPORT` é útil para migrar tabelas pequenas e tabelas que incluem objetos grandes (LOBs). O comando `IMPORT` é mais lento do que o utilitário `LOAD` devido às operações de registro em log `INSERT` e `DELETE`. Se a largura de banda da rede entre a máquina cliente e o RDS para Db2 for limitada, recomendamos usar uma abordagem de

migração diferente. Para ter mais informações, consulte Usar ferramentas nativas do Db2 para migrar dados do Db2 para o Amazon RDS para Db2.

Como importar dados para o servidor RDS para Db2

1. Faça login na máquina cliente com o IBM Db2 Data Management Console. Para ter mais informações, consulte Conectar-se à instância de banco de dados do Amazon RDS para Db2 com o IBM Db2 Data Management Console.
2. Catalogue o banco de dados do RDS para Db2 na máquina cliente.

   a. Catalogue o nó. No exemplo a seguir, substitua *dns_ip_address* e *port* pelo nome do DNS ou pelo endereço IP e pelo número da porta do banco de dados do Db2 autogerenciado.

   ```
   db2 catalog tcpip node srcnode REMOTE dns_ip_address server port
   ```

   b. Catalogue o banco de dados. No exemplo a seguir, substitua *source_database_name* e *source_database_alias* pelo nome do banco de dados do Db2 autogerenciado e pelo alias a ser utilizado para esse banco de dados.

   ```
   db2 catalog database source_database_name as source_database_alias at node
    srcnode \
       authentication server_encrypt
   ```

3. Associe ao banco de dados de origem. No exemplo a seguir, substitua *source_database_alias*, *user_id* e *user_password* pelo alias que você criou na etapa anterior e pelo ID de usuário e senha do banco de dados do Db2 autogerenciado.

   ```
   db2look -d source_database_alias -i user_id -w user_password -e -l -a -f -wlm \
       -cor -createdb -printdbcfg -o db2look.sql
   ```

4. Gere o arquivo de dados usando o comando EXPORT no sistema Db2 autogerenciado. No exemplo a seguir, substitua *directory* pelo diretório da máquina cliente onde se encontra o arquivo de dados. Substitua *file_name* e *table_name* pelo nome do arquivo de dados e pelo nome da tabela.

   ```
   db2 "export to /directory/file_name.txt of del lobs to /directory/lobs/ \
       modified by coldel\| select * from table_name"
   ```

5. Conecte-se ao banco de dados do RDS para Db2 usando o nome de usuário principal e a senha mestra da instância de banco de dados do RDS para Db2. No exemplo a seguir, substitua *rds_database_alias*, *master_username* e *master_password* por suas próprias informações.

```
db2 connect to rds_database_alias user master_username using master_password
```

6. Use o comando IMPORT para importar dados de um arquivo na máquina cliente para o banco de dados remoto do RDS para Db2. Para ter mais informações, consulte IMPORT command na documentação do IBM Db2. No exemplo a seguir, substitua *directory* e *file_name* pelo diretório da máquina cliente onde se encontra o arquivo de dados e o nome do arquivo de dados. Substitua *SCHEMA_NAME* e *TABLE_NAME* pelo nome do esquema e da tabela.

```
db2 "IMPORT from /directory/file_name.tbl OF DEL LOBS FROM /directory/lobs/ \
    modified by coldel\| replace into SCHEMA_NAME.TABLE_NAME"
```

7. Encerre a conexão.

```
db2 terminate
```

Importar dados de uma máquina cliente para o Amazon RDS para Db2 com o comando LOAD

É possível usar o comando LOAD CLIENT para carregar dados de um arquivo em uma máquina cliente para o servidor do RDS para Db2. Como não existe conectividade SSH com o servidor do RDS para Db2, é possível usar o comando LOAD CLIENT no servidor do Db2 autogerenciado ou na máquina cliente do Db2.

> **Important**
>
> O método de comando LOAD CLIENT é útil para migrar tabelas pequenas. Se a largura de banda da rede entre o cliente e o RDS para Db2 for limitada, recomendamos usar uma abordagem de migração diferente. Para obter mais informações, consulte Usar ferramentas nativas do Db2 para migrar dados do Db2 para o Amazon RDS para Db2.
> Se o arquivo de dados incluir referências a nomes de arquivos de objetos grandes, o comando LOAD não funcionará porque objetos grandes (LOBs) precisam estar no servidor

Db2. Se você tentar carregar LOBs da máquina cliente para o servidor RDS para Db2, receberá um erro SQL3025N. Em vez disso, use o comando IMPORT.

Como carregar dados para o servidor RDS para Db2

1. Faça login na máquina cliente com o IBM Db2 Data Management Console. Para ter mais informações, consulte Conectar-se à instância de banco de dados do Amazon RDS para Db2 com o IBM Db2 Data Management Console.
2. Catalogue o banco de dados do RDS para Db2 na máquina cliente.

   a. Catalogue o nó. No exemplo a seguir, substitua *dns_ip_address* e *port* pelo nome do DNS ou pelo endereço IP e pelo número da porta do banco de dados do Db2 autogerenciado.

   ```
   db2 catalog tcpip node srcnode REMOTE dns_ip_address server port
   ```

   b. Catalogue o banco de dados. No exemplo a seguir, substitua *source_database_name* e *source_database_alias* pelo nome do banco de dados do Db2 autogerenciado e pelo alias a ser utilizado para esse banco de dados.

   ```
   db2 catalog database source_database_name as source_database_alias at node
    srcnode \
       authentication server_encrypt
   ```

3. Associe ao banco de dados de origem. No exemplo a seguir, substitua *source_database_alias*, *user_id* e *user_password* pelo alias que você criou na etapa anterior e pelo ID de usuário e senha do banco de dados do Db2 autogerenciado.

   ```
   db2look -d source_database_alias -i user_id -w user_password -e -l -a -f -wlm \
       -cor -createdb -printdbcfg -o db2look.sql
   ```

4. Gere o arquivo de dados usando o comando EXPORT no sistema Db2 autogerenciado. No exemplo a seguir, substitua *directory* pelo diretório da máquina cliente onde se encontra o arquivo de dados. Substitua *file_name* e *TABLE_NAME* pelo nome do arquivo de dados e pelo nome da tabela.

   ```
   db2 "export to /directory/file_name.txt of del modified by coldel\| \
       select * from TPCH.TABLE_NAME"
   ```

5. Conecte-se ao banco de dados do RDS para Db2 usando o nome de usuário principal e a senha mestra da instância de banco de dados do RDS para Db2. No exemplo a seguir, substitua *rds_database_alias*, *master_username* e *master_password* por suas próprias informações.

```
db2 connect to rds_database_alias user master_username using master_password
```

6. Use o comando LOAD para carregar dados de um arquivo na máquina cliente para o banco de dados remoto do RDS para Db2. Para ter mais informações, consulte LOAD command na documentação do IBM Db2. No exemplo a seguir, substitua *directory* pelo diretório da máquina cliente onde se encontra o arquivo de dados. Substitua *file_name* e *TABLE_NAME* pelo nome do arquivo de dados e pelo nome da tabela.

```
db2 "LOAD CLIENT from /directory/file_name.txt \
    modified by coldel\| replace into TPCH.TABLE_NAME \
    nonrecoverable without prompting"
```

7. Encerre a conexão.

```
db2 terminate
```

Importar dados do Db2 para o Amazon RDS para Db2 com o comando INSERT

É possível usar o comando INSERT de um servidor do Db2 autogerenciado para inserir os dados em um banco de dados do Amazon RDS para Db2. Com essa abordagem de migração, é usado um apelido para a instância de banco de dados do RDS para Db2 remota. O banco de dados do Db2 autogerenciado (origem) deve ser capaz de se conectar ao banco de dados do RDS para Db2 (destino).

> **Important**
>
> O método de comando INSERT é útil para migrar tabelas pequenas. Se a largura de banda da rede entre o banco de dados do Db2 autogerenciado e o banco de dados do RDS para Db2 for limitada, recomendamos usar uma abordagem de migração diferente. Para ter mais informações, consulte Usar ferramentas nativas do Db2 para migrar dados do Db2 para o Amazon RDS para Db2.

Como copiar dados de um banco de dados do Db2 autogerenciado em um banco de dados do RDS para Db2

1. Catalogue a instância de banco de dados do RDS para Db2 na instância do Db2 autogerenciado.

   a. Catalogue o nó. No exemplo a seguir, substitua *dns_ip_address* e *port* pelo nome do DNS ou pelo endereço IP e pelo número da porta do banco de dados do Db2 autogerenciado.

   ```
   db2 catalog tcpip node remnode REMOTE dns_ip_address SERVER port
   ```

   b. Catalogue o banco de dados. No exemplo a seguir, substitua *rds_database_name* pelo nome do banco de dados na instância de banco de dados do RDS para Db2.

   ```
   db2 catalog database rds_database_name as remdb at node remnode \
       authentication server_encrypt
   ```

2. Habilite a federação na instância do Db2 autogerenciado. No exemplo a seguir, substitua *source_database_name* pelo nome do banco de dados na instância de banco de dados do Db2 autogerenciado.

   ```
   db2 update dbm cfg using FEDERATED YES source_database_name
   ```

3. Crie tabelas na instância de banco de dados do RDS para Db2.

   a. Catalogue o nó. No exemplo a seguir, substitua *dns_ip_address* e *port* pelo nome do DNS ou pelo endereço IP e pelo número da porta do banco de dados do Db2 autogerenciado.

   ```
   db2 catalog tcpip node srcnode REMOTE dns_ip_address server port
   ```

   b. Catalogue o banco de dados. No exemplo a seguir, substitua *source_database_name* e *source_database_alias* pelo nome do banco de dados do Db2 autogerenciado e pelo alias a ser utilizado para esse banco de dados.

   ```
   db2 catalog database source_database_name as source_database_alias at node
    srcnode \
       authentication server_encrypt
   ```

4. Associe ao banco de dados de origem. No exemplo a seguir, substitua *source_database_alias*, *user_id* e *user_password* pelo alias que você criou na etapa anterior e pelo ID de usuário e senha do banco de dados do Db2 autogerenciado.

```
db2look -d source_database_alias -i user_id -w user_password -e -l -a -f -wlm \
    -cor -createdb -printdbcfg -o db2look.sql
```

5. Configure a federação e crie um apelido para a tabela de banco de dados do RDS para Db2 na instância do Db2 autogerenciado.

   a. Conecte-se ao banco de dados local. No exemplo a seguir, substitua *source_database_name* pelo nome do banco de dados na instância do Db2 autogerenciado.

```
db2 connect to source_database_name
```

   b. Crie um wrapper para acessar fontes de dados do Db2.

```
db2 create wrapper drda
```

   c. Defina uma fonte de dados em um banco de dados federado. No exemplo a seguir, substitua *admin* e *admin_password* pelas credenciais da instância do Db2 autogerenciado. Substitua *rds_database_name* pelo nome do banco de dados na instância de banco de dados do RDS para Db2.

```
db2 "create server rdsdb2 type DB2/LUW version '11.5.9.0' \
    wrapper drda authorization "admin" password "admin_password" \
    options( dbname 'rds_database_name', node 'remnode')"
```

   d. Associe os usuários nos dois bancos de dados. No exemplo a seguir, substitua *master_username* e *master_password* pelas credenciais da instância de banco de dados do RDS para Db2.

```
db2 "create user mapping for user server rdsdb2 \
    options (REMOTE_AUTHID 'master_username', REMOTE_PASSWORD
 'master_password')"
```

   e. Verifique a conexão com o servidor RDS para Db2.

```
db2 set passthru rdsdb2
```

f. Crie um apelido para a tabela no banco de dados remoto do RDS para Db2. No exemplo a seguir, substitua *NICKNAME* e *TABLE_NAME* por um apelido para a tabela e o nome da tabela.

```
db2 create nickname REMOTE.NICKNAME for RDSDB2.TABLE_NAME.NICKNAME
```

6. Insira dados na tabela no banco de dados remoto do RDS para Db2. Use o apelido em uma declaração `select` na tabela local na instância do Db2 autogerenciado. No exemplo a seguir, substitua *NICKNAME* e *TABLE_NAME* por um apelido para a tabela e o nome da tabela.

```
db2 "INSERT into REMOTE.NICKNAME select * from RDS2DB2.TABLE_NAME.NICKNAME"
```

Importar dados do Db2 para o Amazon RDS para Db2 com o utilitário INGEST

É possível usar o utilitário `INGEST` para transmitir continuamente dados de arquivos e canais em uma máquina cliente para uma instância de banco de dados do Amazon RDS para Db2 de destino. O utilitário `INGEST` aceita operações `INSERT` e `MERGE`. Para ter mais informações, consulte Ingest utility na documentação do IBM Db2.

Como o utilitário `INGEST` aceita apelidos, é possível usá-lo para transferir dados do banco de dados do Db2 autogerenciado para um banco de dados do RDS para Db2. Essa abordagem funciona desde que exista conectividade de rede entre os dois bancos de dados.

> **Important**
>
> O utilitário `INGEST` não comporta objetos grandes (LOBs). Em vez disso, use o comando `IMPORT`.

Para usar o recurso `RESTARTABLE` do utilitário `INGEST`, execute o comando a seguir no banco de dados do RDS para Db2.

```
db2 "call sysproc.sysinstallobjects('INGEST','C',NULL,NULL)"
```

Federação do Amazon RDS para Db2

É possível usar o banco de dados do Amazon RDS para Db2 como um banco de dados federado. Depois de configurar a federação para o RDS para Db2, você poderá acessar e consultar dados em vários bancos de dados do seu banco de dados do RDS para Db2. A federação evita que você precise migrar dados para o banco de dados do RDS para Db2 ou consolidar dados em um único banco de dados.

Ao usar o banco de dados do RDS para Db2 como um banco de dados federado, é possível continuar acessando todos os recursos do RDS para Db2 e aproveitar vários Serviços da AWS, além de manter os dados em bancos de dados diferentes. É possível configurar a federação homogênea, que conecta diferentes bancos de dados do mesmo tipo, ou a federação heterogênea, que conecta bancos de dados diferentes de tipos diferentes.

Primeiro, é necessário conectar o banco de dados do Db2 no RDS para Db2 aos bancos de dados remotos. Depois, é possível executar consultas em todos os bancos de dados conectados. Por exemplo, é possível executar uma declaração SQL JOIN que une tabelas no banco de dados do RDS para Db2 com tabelas em um banco de dados do Db2 remoto no z/OS.

Tópicos

- Federação homogênea
- Federação heterogênea

Federação homogênea

É possível configurar uma federação homogênea entre o banco de dados do RDS para Db2 e a seguinte família de produtos Db2:

- Db2 para Linux, UNIX, Windows (LUW)
- Db2 iSeries
- Db2 para z/OS

A federação homogênea do RDS para Db2 não é compatível com as seguintes ações:

- Executar comandos `CATALOG` para configurar um diretório de nós e um banco de dados remoto em um banco de dados host do RDS para Db2.

- Configurar o balanceamento de workload (WLB) ao federar para o Db2 no z/OS.
- Configurar o arquivo de configuração do driver do servidor de dados IBM (`db2dsdriver.cfg`).

A federação homogênea do RDS para Db2 tem os seguintes requisitos:

- É necessário criar o wrapper DRDA no modo `UNFENCED`. Caso contrário, a federação não funcionará no RDS para Db2.
- É necessário permitir o tráfego de entrada e de saída do banco de dados host do RDS para Db2 para os bancos de dados host remotos. Para ter mais informações, consulte Fornecer acesso à instância de banco de dados na VPC criando um grupo de segurança.

Tópicos

- Etapa 1: Criar um wrapper DRDA e um servidor federado
- Etapa 2: Criar um mapeamento do usuário
- Etapa 3: Conferir a conexão

Etapa 1: Criar um wrapper DRDA e um servidor federado

Em relação a uma federação homogênea, crie um wrapper DRDA e um servidor federado. A conexão com o host remoto usa `HOST`, `PORT` e `DBNAME`.

Escolha um dos métodos a seguir com base no tipo do banco de dados do Db2 remoto:

- Banco de dados do Db2 para Linux, UNIX e Windows (LUX): execute os comandos SQL a seguir. No exemplo a seguir, substitua *`server_name`* pelo nome do servidor que você usará para federação. Substitua *`db2_version`* pela versão do banco de dados do Db2 remoto. Substitua *`username`* e *`password`* por suas credenciais para o banco de dados do Db2 remoto ao qual você deseja se conectar. Substitua *`db_name`*, *`dns_name`* e *`port`* pelos valores apropriados para o banco de dados do Db2 remoto ao qual você deseja se conectar.

```
create wrapper drda options(DB2_FENCED 'N');
create server server_name type DB2/LUW wrapper drda version 'db2_version'
 authorization "master_username" password "master_password" options (add DBNAME
 'db_name',add HOST 'dns_name',add PORT 'port');
```

  Exemplo

```
create wrapper drda options(DB2_FENCED 'N');
create server SERVER1 type DB2/LUW wrapper drda version '11.5' authorization
 "sysuser" password "******" options (add DBNAME 'TESTDB2',add HOST
 'ip-123-45-67-899.us-west-1.compute.internal',add PORT '25010');
```

- Db2 iSeries: execute os comandos SQL a seguir. No exemplo a seguir, substitua *wrapper_name* e *library_name* por um nome para o wrapper DRDA e o arquivo de biblioteca de wrapper. Substitua *server_name* pelo nome do servidor que você usará para federação. Substitua *db2_version* pela versão do banco de dados do Db2 remoto. Substitua *username* e *password* por suas credenciais para o banco de dados do Db2 remoto ao qual você deseja se conectar. Substitua *dns_name*, *port* e *db_name* pelos valores apropriados para o banco de dados do Db2 remoto ao qual você deseja se conectar.

```
create wrapper wrapper_name library 'library name' options(DB2_FENCED 'N');
create server server_name type db2/mvs version db2_version wrapper wrapper_name
 authorization "sername" password "password" options (HOST 'dns_name', PORT 'port',
 DBNAME 'db_name');
```

  Exemplo

```
create wrapper WRAPPER1 library 'libdb2drda.so' options(DB2_FENCED 'N');
create server SERVER1 type db2/mvs version 11 wrapper WRAPPER1 authorization
 "sysuser" password "******" options (HOST 'test1.123.com', PORT '446', DBNAME
 'STLEC1');
```

- Db2 para z/OS: execute os comandos SQL a seguir. No exemplo a seguir, substitua *wrapper_name* e *library_name* por um nome para o wrapper DRDA e o arquivo de biblioteca de wrapper. Substitua *server_name* pelo nome do servidor que você usará para federação. Substitua *db2_version* pela versão do banco de dados do Db2 remoto. Substitua *username* e *password* por suas credenciais para o banco de dados do Db2 remoto ao qual você deseja se conectar. Substitua *dns_name*, *port* e *db_name* pelos valores apropriados para o banco de dados do Db2 remoto ao qual você deseja se conectar.

```
create wrapper wrapper_name library 'library_name' options(DB2_FENCED 'N');
create server server_name type db2/mvs version db2_version wrapper wrapper_name
 authorization "username" password "password" options (HOST 'dns_name', PORT 'port',
 DBNAME 'db_name');
```

Exemplo

```
create wrapper WRAPPER1 library 'libdb2drda.so' OPTIONS(DB2_FENCED 'N');
create server SERVER1 type db2/mvs version 11 wrapper WRAPPER1 authorization
 "sysuser" password "******" options (HOST 'test1.123.com', PORT '446', DBNAME
 'STLEC1');
```

Etapa 2: Criar um mapeamento do usuário

Crie um mapeamento de usuário para associar o servidor federado ao servidor de fonte de dados executando o comando SQL a seguir. No exemplo a seguir, substitua *server_name* pelo nome do servidor remoto no qual você deseja realizar as operações. Esse é o servidor que você criou na etapa 1. Substitua *username* e *password* pelas suas credenciais para esse servidor remoto.

```
create user mapping for user server server_name options (REMOTE_AUTHID 'username',
 REMOTE_PASSWORD 'password');
```

Para ter mais informações, consulte User mappings na documentação do IBM Db2.

Etapa 3: Conferir a conexão

Confirme se a configuração da federação foi bem-sucedida conferindo a conexão. Abra uma sessão para enviar comandos SQL nativos à sua fonte de dados remota usando o comando SET PASSTHRU e, depois, crie uma tabela no servidor de dados remoto.

1. Abra e feche uma sessão para enviar SQL a uma fonte de dados. No exemplo a seguir, substitua *server_name* pelo nome do servidor que você criou para federação na etapa 1.

   ```
   set passthru server_name;
   ```

2. Crie uma tabela. No exemplo a seguir, substitua *column_name*, *data_type* e *value* pelos itens apropriados para sua tabela.

   ```
   create table table_name
    ( column_name data_type(value), column_name data_type(value);
   ```

   Para ter mais informações, consulte CREATE TABLE statement na documentação do IBM Db2.

3. Crie um índice, insira valores para linhas na tabela e redefina a conexão. A redefinição da conexão interrompe a conexão, mas retém os processos de back-end. No exemplo a seguir, substitua *index_name*, *table_name*, *column_value* e *columnx_value* por suas informações.

```
create index index_name on table_name(column_name);
insert into table_name values(column1_value,column2_value,column3_value);
insert into table_name values(column1_value,column2_value,column3_value);
set passthru reset;

connect reset;
```

4. Conecte-se ao seu banco de dados do Db2 remoto, crie um apelido para seu servidor remoto e execute operações. Quando terminar de acessar os dados no banco de dados do Db2 remoto, redefina e encerre a conexão. No exemplo a seguir, substitua *database_name* pelo nome do banco de dados do Db2 remoto. Substitua *nickname* por um nome. Substitua *server_name* e *table_name* pelo nome do servidor remoto e da tabela no servidor em que você deseja realizar operações. Substitua *username* pelas informações do seu servidor remoto. Substitua *sql_command* pela operação a ser executada no servidor remoto.

```
connect to database_name;
create nickname nickname for server_name."username"."table_name";
select sql_command from nickname;
connect reset;
terminate;
```

Exemplo

O exemplo a seguir cria uma sessão de passagem para permitir operações no servidor federado `testdb10`.

Depois, ele cria a tabela `t1` com três colunas com diferentes tipos de dados.

Depois, o exemplo cria o índice `i1_t1` em três colunas na tabela `t1`. Depois, ele insere duas linhas com valores para essas três colunas e então se desconecta.

Por último, o exemplo se conecta ao banco de dados do Db2 remoto `testdb2` e cria um apelido para a tabela `t1` no servidor federado `testdb10`. Ele cria o apelido com o nome de usuário `TESTUSER` dessa fonte de dados. Um comando SQL gera todos os dados da tabela `t1`. O exemplo desconecta e encerra a sessão.

```
set passthru testdbl0;

create table t1 ( c1 decimal(13,0), c2 char(200), c3 int);

create index i1_t1 on t1(c3);
insert into t1 values(1,'Test',1);
insert into t1 values(2,'Test 2',2);
connect reset;

connect to testdb2;
create nickname remote_t1 for testdbl0."TESTUSER"."T1";
select * from remote_t1;
connect reset;
terminate;
```

Federação heterogênea

É possível configurar uma federação heterogênea entre o banco de dados do RDS para Db2 e outras fontes de dados, como Oracle e Microsoft SQL Server. Para ter uma lista completa das fontes de dados que o Db2 LUW comporta, consulte Data Source Support Matrix of Federation Bundled in Db2 LUW V11.5 no site de suporte da IBM.

A federação heterogênea do RDS para Db2 não é compatível com os seguintes itens:

- Wrappers nativos para as outras fontes de dados.
- Wrappers JDBC para as outras fontes de dados.
- Federação para fontes de dados Sybase, Informix e Teradata porque essas fontes de dados exigem instalação de software-cliente no RDS para Db2

A federação heterogênea do RDS para Db2 tem os seguintes requisitos:

- O RDS para Db2 é compatível com o método de wrapper ODBC.
- Se você criar uma definição explícita de um wrapper, deverá definir a opção `DB2_FENCED` como `'N'`. Para ter uma lista de opções de wrapper válidas para ODBC, consulte ODBC options na documentação do IBM Db2.
- É necessário permitir o tráfego de entrada e de saída do banco de dados host do RDS para Db2 para o banco de dados host remoto. Para ter mais informações, consulte Fornecer acesso à instância de banco de dados na VPC criando um grupo de segurança.

Para ter informações sobre federação para Oracle, consulte How to query Oracle by using Db2 Federation and the ODBC driver? no site de suporte da IBM.

Para ter mais informações sobre fontes de dados que comportam a federação, consulte Data Source Support Matrix of Federation Bundled in Db2 LUW V11.5 no site de suporte da IBM.

Tópicos

- Etapa 1: Criar um wrapper ODBC
- Etapa 2: Criar um servidor federado
- Etapa 3: Criar um mapeamento do usuário
- Etapa 4: Conferir a conexão

Etapa 1: Criar um wrapper ODBC

Crie um wrapper executando o seguinte comando:

```
db2 "create wrapper odbc options( module '/home/rdsdb/sqllib/federation/odbc/lib/
libodbc.so')"
```

Etapa 2: Criar um servidor federado

Crie um servidor federado executando o comando a seguir. No exemplo a seguir, substitua *server_name* pelo nome do servidor que você usará para federação. Substitua *wrapper_type* pelo wrapper apropriado. Substitua *db_version* pela versão do banco de dados remoto. Substitua *dns_name*, *port* e *service_name* pelos valores apropriados para o banco de dados remoto ao qual você deseja se conectar.

```
db2 "create server server_name type wrapper_type version db_version options (HOST
 'dns_name', PORT 'port', SERVICE_NAME 'service_name')“
```

Para ter informações sobre tipos de wrapper, consulte Data Source Support Matrix of Federation Bundled in Db2 LUW V11.5 no site de suporte da IBM.

Exemplo

O exemplo a seguir cria um servidor federado para um banco de dados Oracle remoto.

```
db2 "create server server1 type oracle_odbc version 12.1 options (HOST
 'test1.amazon.com', PORT '1521', SERVICE_NAME 'pdborcl.amazon.com')“
```

Etapa 3: Criar um mapeamento do usuário

Crie um mapeamento de usuário para associar o servidor federado ao servidor de fonte de dados executando o comando SQL a seguir. No exemplo a seguir, substitua *server_name* pelo nome do servidor remoto no qual você deseja realizar as operações. Esse é o servidor que você criou na etapa 2. Substitua *username* e *password* pelas suas credenciais para esse servidor remoto.

```
create user mapping for user server server_name options (REMOTE_AUTHID 'username',
 REMOTE_PASSWORD 'password');
```

Para ter mais informações, consulte User mappings na documentação do IBM Db2.

Etapa 4: Conferir a conexão

Confirme se a configuração da federação foi bem-sucedida conferindo a conexão. Abra uma sessão para enviar comandos SQL nativos à sua fonte de dados remota usando o comando SET PASSTHRU e, depois, crie uma tabela no servidor de dados remoto.

1. Abra e feche uma sessão para enviar SQL a uma fonte de dados. No exemplo a seguir, substitua *server_name* pelo nome do servidor que você criou para federação na etapa 2.

   ```
   set passthru server_name;
   ```

2. Crie uma tabela. No exemplo a seguir, substitua *column_name*, *data_type* e *value* pelos itens apropriados para sua tabela.

   ```
   create table table_name
    ( column_name data_type(value), column_name data_type(value);
   ```

   Para ter mais informações, consulte CREATE TABLE statement na documentação do IBM Db2.

3. Crie um índice, insira valores para linhas na tabela e redefina a conexão. A redefinição da conexão interrompe a conexão, mas retém os processos de back-end. No exemplo a seguir, substitua *index_name*, *table_name*, *column_value* e *columnx_value* por suas informações.

   ```
   create index index_name on table_name(column_name);
   ```

```
insert into table_name values(column1_value,column2_value,column3_value);
insert into table_name values(column1_value,column2_value,column3_value);
set passthru reset;

connect reset;
```

4. Conecte-se ao seu banco de dados do Db2 remoto, crie um apelido para seu servidor remoto e execute operações. Quando terminar de acessar os dados no banco de dados do Db2 remoto, redefina e encerre a conexão. No exemplo a seguir, substitua *database_name* pelo nome do banco de dados do Db2 remoto. Substitua *nickname* por um nome. Substitua *server_name* e *table_name* pelo nome do servidor remoto e da tabela no servidor em que você deseja realizar operações. Substitua *username* pelas informações do seu servidor remoto. Substitua *sql_command* pela operação a ser executada no servidor remoto.

```
connect to database_name;
create nickname nickname for server_name."username"."table_name";
select sql_command from nickname;
connect reset;
terminate;
```

Exemplo

O exemplo a seguir cria uma sessão de passagem para permitir operações no servidor federado `testdb10`.

Depois, ele cria a tabela `t1` com três colunas com diferentes tipos de dados.

Depois, o exemplo cria o índice `i1_t1` em três colunas na tabela `t1`. Depois, ele insere duas linhas com valores para essas três colunas e então se desconecta.

Por último, o exemplo se conecta ao banco de dados do Db2 remoto `testdb2` e cria um apelido para a tabela `t1` no servidor federado `testdb10`. Ele cria o apelido com o nome de usuário `TESTUSER` dessa fonte de dados. Um comando SQL gera todos os dados da tabela `t1`. O exemplo desconecta e encerra a sessão.

```
set passthru testdbl0;

create table t1 ( c1 decimal(13,0), c2 char(200), c3 int);

create index i1_t1 on t1(c3);
```

```
insert into t1 values(1,'Test',1);
insert into t1 values(2,'Test 2',2);
connect reset;

connect to testdb2;
create nickname remote_t1 for testdbl0."TESTUSER"."T1";
select * from remote_t1;
connect reset;
terminate;
```

Trabalhar com réplicas para o Amazon RDS para Db2

Para configurar a replicação entre instâncias de banco de dados RDS para Db2, você pode criar bancos de dados de réplica. Para ver um resumo das diferenças entre réplicas do RDS para Db2 e outros mecanismos de banco de dados, consulte Diferenças entre réplicas de leitura para mecanismos de banco de dados.

Um banco de dados de réplica do Db2 é uma cópia física do seu banco de dados primário. Uma réplica do Db2 no modo de espera é chamada de réplica em espera. O Db2 não permite leituras ou gravações em uma réplica, mas é possível promover uma réplica para torná-la gravável. A réplica promovida terá os dados replicados para o ponto quando a solicitação foi feita para promovê-la. O RDS para Db2 usa Recuperação de Desastres de Alta Disponibilidade (HADR) do Db2 para replicação. Para obter mais informações, consulte Recuperação de Desastres de Alta Disponibilidade (HADR) na IBM Db2documentação.

Réplicas em espera

Ao criar ou modificar uma réplica do Db2, você pode colocá-la no modo de espera. Para réplicas em espera, a variável de ambiente do Db2 `DB2_HADR_ROS` é definida como `OFF` de forma que o banco de dados de réplica não aceite conexões de usuário. O principal uso para réplicas em espera é a recuperação de desastres entre regiões.

Uma réplica em espera não pode atender a uma workload somente leitura. A réplica em espera não tem nenhum log de arquivo.

Você pode criar até três réplicas de uma instância de banco de dados de origem.

Ativações do banco de dados

O HADR do Db2 é configurado no nível do banco de dados. Depois de criar réplicas, o HADR é definido para todos os bancos de dados Db2, inclusive `rdsadmin`, que o RDS gerencia totalmente. Antes de criar réplicas do Db2, você deve ativar explicitamente todos os bancos de dados. Caso contrário, a criação de réplicas falhará e o Amazon RDS emitirá um evento. Depois que uma instância de banco de dados tiver uma ou mais réplicas, você não poderá ativar ou desativar nenhum banco de dados na instância de banco de dados usando os procedimentos armazenados `rdsadmin.activate_database` ou `rdsadmin.deactivate_database`. No entanto, você pode chamar o procedimento armazenado `rdsadmin.reactivate_database` para reativar bancos de dados. Para obter mais informações, consulte Gerenciar bancos de dados.

Configurações de HADR

Você pode ver todas as configurações de HADR de um banco de dados conectando-se ao banco de dados e depois executando `db2 get db cfg`.

Arquivar retenção de logs

O Amazon RDS limpará os logs de uma instância de banco de dados primária depois que as seguintes condições forem atendidas:

- Os logs têm pelo menos duas horas.
- A configuração para o período de retenção para os logs de arquivo expirou.
- Os logs de arquivamento foram replicados com sucesso em todas as instâncias de banco de dados de réplica. Essa condição se aplica tanto às instâncias de banco de dados na mesma Região da AWS quanto às instâncias de banco de dados entre regiões.

Para obter informações sobre como definir horas de retenção de log de arquivamento, consulte rdsadmin.set_archive_log_retention.

O Amazon RDS verifica e limpa cada banco de dados individualmente. Se um banco de dados perder a conexão HADR ou se as informações sobre a conexão não estiverem disponíveis, o Amazon RDS ignorará o banco de dados e não limpará os logs de arquivamento.

Interrupções durante a replicação do Db2

Ao criar uma réplica, o Amazon RDS faz um snapshot do banco de dados da sua instância de banco de dados de origem e inicia a replicação. Quando a operação de snapshot do banco de dados se inicia, a instância de banco de dados de origem sofre uma suspensão de E/S muito breve. A suspensão de E/S geralmente dura um segundo. No entanto, se a instância de banco de dados de origem for uma implantação Multi-AZ, a instância de banco de dados de origem não sofrerá nenhuma suspensão de E/S. Isso ocorre porque, com implantações Multi-AZ, o snapshot é obtido da instância de banco de dados secundária.

O snapshot de banco de dados torna-se a réplica do Db2. O Amazon RDS define as permissões e os parâmetros necessários para o banco de dados de origem e para a réplica sem interrupção do serviço. Da mesma forma, se você excluir uma réplica, nenhuma interrupção ocorrerá.

Requisitos e considerações sobre réplicas do RDS para Db2

Antes de criar uma réplica do Oracle, familiarize-se com os seguintes requisitos e considerações.

Requisitos de versão e licenciamento para réplicas do RDS para Db2

Antes de criar uma réplica do RDS para Db2, analise as seguintes informações:

- Tanto o Db2 Advanced Edition (AE) quanto o Standard Edition (SE) podem criar réplicas em espera tanto para o modelo traga a sua própria licença (BYOL) quanto para o modelo de licença do Db2 através da AWS Marketplace.
- É necessário especificar um grupo de parâmetros personalizado para a réplica de leitura. Para réplicas que usam o modelo BYOL, esse grupo de parâmetros personalizados deve incluir seu IBM Site ID e IBM Customer ID. Você pode especificar esse grupo de parâmetros personalizado para a réplica de leitura usando a AWS Management Console, a AWS CLIou a API do RDS. Para obter mais informações, consulte IDs da IBM para traga a sua própria licença para o Db2.
- Para réplicas em espera, a contagem de vCPUs é dois, independentemente do tamanho da instância de banco de dados. Os itens a seguir descrevem as diferenças entre os tipos de licença do RDS para Db2:
  - Para o modelo BYOL, as configurações de AWS License Manager mostram que as instâncias de banco de dados RDS para Db2 usam duas vCPUs.
  - Para o modelo de licença do Db2 por meio do AWS Marketplace, as faturas refletem os custos de licença de duas vCPUs.
- Todas as versões do Db2 11.5 são compatíveis com instâncias de banco de dados de réplicas. As réplicas do Db2 são compatíveis com atualizações de versões secundárias.

Considerações sobre backup e restauração de réplicas do RDS para Db2

Antes de criar uma réplica do RDS para Db2, considere os seguintes itens:

- Para criar snapshots de réplicas do RDS para Db2 ou ativar backups automáticos, defina o período de retenção de backup manualmente. Por padrão, os backups automáticos não estão ativados.
- Ao restaurar um backup de réplica, você restaura para a hora do banco de dados, não para a hora de criação do backup. A hora do banco de dados se refere ao horário de transação mais recente aplicado dos dados no backup. A diferença é significativa porque uma réplica pode apresentar atraso de minutos ou horas em relação ao banco de dados primário. Quando há vários bancos de dados, o RDS para Db2 usa o horário mais antigo do banco de dados.

Para descobrir a diferença, execute o comando AWS CLI describe-db-snapshots ou chame a operação DescribeDBSnapshots da API do RDS. Compare o valor `SnapshotDatabaseTime` com o valor `OriginalSnapshotCreateTime`. O valor `SnapshotDatabaseTime` é o horário do banco de dados do backup da réplica. O valor `OriginalSnapshotCreateTime` é a última transação aplicada no banco de dados primário.

Considerações sobre replicação para réplicas do RDS para Db2

Antes de criar uma réplica do RDS para Db2, considere os seguintes itens:

- A replicação usa Db2 HADR para todos os bancos de dados na instância de banco de dados do RDS para Db2.
- A replicação não suporta o comando `LOAD`. Se você executar o comando `LOAD` a partir da instância de banco de dados de origem, receberá dados inconsistentes.
- O RDS para Db2 não replica os seguintes itens:
  - Acesso ao armazenamento. Esteja ciente de dados, como tabelas externas, que dependem do acesso ao armazenamento.
  - LOBs não embutidos.
  - Binários de procedimentos armazenados externos (em C ou Java).
- Para réplicas em espera, o RDS para Db2 replica os seguintes itens:
  - Usuários locais, exceto usuários mestres
  - Parâmetros de configuração do banco de dados

Considerações diversas sobre réplicas do RDS para Db2

Antes de criar uma réplica do RDS para Db2, considere os seguintes itens:

- O RDS para Db2 replica as configurações do banco de dados para as réplicas. Quando o RDS para Db2 promove uma réplica, ele desativa e ativa cada banco de dados.
- O RDS para Db2 replica os usuários locais, mas não o usuário mestre, para as réplicas. Você pode modificar o usuário mestre na réplica. Para obter mais informações, consulte Modificar uma instância de banco de dados do Amazon RDS.
- Todos os bancos de dados devem estar em um estado ativo. Para obter informações sobre como ativar banco de dados, consulte Gerenciar bancos de dados.

- Todos os procedimentos armazenados para criar, eliminar, restaurar ou avançar bancos de dados devem ser concluídos antes da criação de uma réplica. Para obter informações sobre esses procedimentos armazenados, consulte Gerenciar bancos de dados.
- Quando a réplica é criada, o Amazon RDS define o parâmetro em nível de banco de dados `blocknonlogged` para todos os bancos de dados na instância de banco de dados de origem como `YES`. Quando a réplica de origem se torna novamente uma instância independente, o Amazon RDS redefine o valor para `NO`. Para obter mais informações, consulte blocknonlogged - Bloquear a criação de tabelas que permitem o parâmetro de configuração de atividades não registradas na documentação do IBM Db2.
- Quando a réplica é criada, o Amazon RDS define o parâmetro em nível de banco de dados `logindexbuild` para todos os bancos de dados na instância de banco de dados de origem como `YES`. Quando a réplica de origem se torna novamente uma instância independente, o Amazon RDS redefine o valor para `NO`. Para obter mais informações, consulte logindexbuild - Parâmetro de configuração de registro de páginas de índice criadas na documentação do IBM Db2.

Preparar-se para criar uma réplica do RDS para Db2

Antes de começar a usar a réplica, execute as tarefas a seguir.

Tarefas

- Habilitar backups automáticos
- Planejar recursos de computação e armazenamento
- Preparar bancos de dados

Habilitar backups automáticos

Antes que uma instância de banco de dados possa servir como uma instância de banco de dados de origem, você deve habilitar backups automáticos na instância de banco de dados de origem. Para saber como executar esse procedimento, consulte Ativar backups automáticos.

Planejar recursos de computação e armazenamento

Verifique se a instância de banco de dados de origem e suas réplicas estão dimensionadas corretamente, em termos de computação e armazenamento, para atender às suas cargas operacionais. Se uma réplica atingir a capacidade de recursos de computação, rede ou

armazenamento, a réplica deixará de receber ou aplicar alterações de sua origem. O RDS para Db2 não intervém para atenuar atrasos altos de replicação entre uma instância de banco de dados de origem e suas réplicas. É possível modificar os recursos de armazenamento e de CPU de uma réplica independentemente de sua origem e de outras réplicas.

Preparar bancos de dados

Antes da criação da réplica, verifique se a instância de banco de dados contém todos os bancos de dados que você deseja que estejam presentes na instância de banco de dados. Após a criação da réplica, você não pode criar, excluir ou restaurar nativamente um banco de dados na instância de banco de dados. Todas as chamadas para os procedimentos armazenados rdsadmin.create_database, rdsadmin.drop_database ou rdsadmin.restore_database falham.

Além disso, verifique se todos os bancos de dados na instância de banco de dados estão em um estado ativo. Caso contrário, a criação da réplica falhará. Para obter informações sobre como ativar banco de dados, consulte Gerenciar bancos de dados.

Criar uma réplica do RDS para Db2 no modo de espera

É possível criar uma réplica em modo de espera usando o AWS Management Console, o AWS CLI ou a API do RDS.

Console

Para criar uma réplica em espera a partir de uma instância de banco de dados de origem do RDS para Db2

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Bancos de dados.
3. Escolha a instância de banco de dados do RDS para Db2 que você deseja usar como a origem da réplica em espera.
4. Em Actions (Ações), escolha Create replica (Criar réplica).
5. Em Replica mode (Modo de réplica), escolha Standby (Em espera).
6. Escolha as configurações que deseja usar. Em DB instance identifier (Identificador da instância de banco de dados), insira um nome para a réplica de leitura. Ajuste outras configurações, conforme necessário.

7. Em Regions (Regiões), escolha Região da AWS onde a réplica em espera será iniciada.
8. Escolha o tamanho da instância e o tipo de armazenamento. Recomendamos que você use a classe de instância de banco de dados e o tipo de armazenamento iguais aos da instância de banco de dados de origem na réplica de leitura.
9. Em Multi-AZ deployment (Implantação Multi-AZ), escolha Create a standby instance (Criar uma instância de espera) para criar uma versão de espera de sua réplica em outra zona de disponibilidade a fim de oferecer suporte de failover para a réplica em espera.
10. Escolha as outras configurações que deseja usar.
11. Escolha Create replica (Criar réplica).

Na página Databases (Bancos de dados), a réplica em espera tem a função Replica (Réplica).

AWS CLI

Para criar uma réplica do Db2 no modo em espera, defina `--replica-mode` como `mounted` no comando create-db-instance-read-replica da AWS CLI.

Example

Para Linux, macOS ou Unix:

```
aws rds create-db-instance-read-replica \
    --db-instance-identifier my_read_replica \
    --source-db-instance-identifier my_db_instance \
    --replica-mode mounted
```

Para Windows:

```
aws rds create-db-instance-read-replica ^
    --db-instance-identifier my_read_replica ^
    --source-db-instance-identifier my_db_instance ^
    --replica-mode mounted
```

API do RDS

Para criar uma réplica do Db2 no modo em espera, especifique `ReplicaMode=mounted` na operação de API CreateDBInstanceReadReplica do RDS.

Como trabalhar com backups de réplicas do RDS para Db2

Você pode criar e restaurar backups de uma réplica do RDS para Db2. Tanto backups automáticos como snapshots manuais são compatíveis. Para obter mais informações, consulte Backup, restauração e exportação de dados. As seções a seguir descrevem as principais diferenças entre o gerenciamento de backups de um banco de dados primário e de uma réplica do RDS para Db2.

Tópicos

- Ativar backups de réplicas do RDS para Db2
- Restaurar um backup de réplica do RDS para Db2

Ativar backups de réplicas do RDS para Db2

Uma réplica do RDS para Db2 não tem o recurso de backups automatizados ativado por padrão. Ative o recurso de backups automáticos configurando o período de retenção de backup para um valor positivo diferente de zero.

Console

Para habilitar imediatamente backups automáticos

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Databases (Bancos de dados) e a instância de banco de dados que você deseja modificar.
3. Selecione Modify.
4. Em Backup Retention Period (Período de retenção de backup), escolha um valor positivo diferente de zero; por exemplo, 3 dias.
5. Escolha Continue.
6. Escolha Apply immediately (Aplicar imediatamente).
7. Escolha Modify DB Instance (Modificar instância de banco de dados) para salvar as alterações e ativar os backups automatizados.

AWS CLI

Para ativar os backups automatizados, use o comando AWS CLImodify-db-instance.

Inclua os seguintes parâmetros:

- `--db-instance-identifier`
- `--backup-retention-period`
- `--apply-immediately` ou `--no-apply-immediately`

O exemplo a seguir ativa backups automatizados definindo o período de retenção de backup como três dias. As alterações são aplicadas imediatamente.

Example

Para Linux, macOS ou Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier my_db_instance  \
    --backup-retention-period 3 \
    --apply-immediately
```

Para Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier my_db_instance  ^
    --backup-retention-period 3 ^
    --apply-immediately
```

API do RDS

Para ativar backups automatizados, use a operação ModifyDBInstance da API do RDS com os seguintes parâmetros obrigatórios:

- `DBInstanceIdentifier`
- `BackupRetentionPeriod`

Restaurar um backup de réplica do RDS para Db2

Você pode restaurar um backup de réplica do RDS para Db2 da mesma forma que pode restaurar um backup do banco de dados primário. Para obter mais informações, consulte os tópicos a seguir.

- Restaurar uma instância de banco de dados

- Restaurar uma instância de banco de dados para um momento especificado no Amazon RDS

A principal consideração ao restaurar um backup de réplica é determinar o momento para o qual está restaurando. A hora do banco de dados se refere ao horário de transação mais recente aplicado dos dados no backup. Ao restaurar um backup de réplica, você restaura para a hora do banco de dados, não para a hora em que o backup foi concluído. A diferença é significativa porque uma réplica pode apresentar atraso de minutos ou horas em relação ao banco de dados primário. Portanto, a hora do banco de dados de um backup de réplica e o momento para o qual vai restaurar podem ser muito antes do horário de criação do backup.

Para encontrar a diferença entre a hora do banco de dados e o horário de criação, execute o comando AWS CLIdescribe-db-snapshots ou chame a operação DescribeDBSnapshots da API do RDS. Compare o valor `SnapshotDatabaseTime` e o valor `OriginalSnapshotCreateTime`. O valor `SnapshotDatabaseTime` é o horário mais antigo do banco de dados entre todos os bancos de dados do backup da réplica. O valor `OriginalSnapshotCreateTime` é a última transação aplicada no banco de dados primário. Observe que os atrasos de replicação podem ser diferentes para vários bancos de dados, e o tempo do banco de dados pode estar entre esses dois tempos.

O exemplo AWS CLI a seguir mostra a diferença entre as duas horas:

```
aws rds describe-db-snapshots \
    --db-instance-identifier my_db2_replica
    --db-snapshot-identifier my_replica_snapshot

{
    "DBSnapshots": [
        {
            "DBSnapshotIdentifier": "my_replica_snapshot",
            "DBInstanceIdentifier": "my_db2_replica",
            "SnapshotDatabaseTime": "2022-07-26T17:49:44Z",
            ...
            "OriginalSnapshotCreateTime": "2021-07-26T19:49:44Z"
        }
    ]
}
```

Solução de problemas de replicação do RDS para Db2

Este conteúdo descreve possíveis problemas e soluções de replicação do RDS para Db2.

Tópicos

- Monitoramento de atraso da replicação do Db2
- erros de replicação no Db2

Monitoramento de atraso da replicação do Db2

Para monitorar o atraso de replicação no Amazon CloudWatch, visualize a métrica `ReplicaLag` do Amazon RDS. Para obter informações sobre o tempo de atraso de replicação, consulte Monitoramento da replicação de leitura e Métricas do Amazon CloudWatch para o Amazon RDS.

Para uma réplica em espera, se o tempo de atraso for muito longo, consulte a tabela `MON_GET_HADR` para saber o status da instância de banco de dados de origem. Não consulte a instância de banco de dados de réplica porque as instâncias de banco de dados de réplica não aceitam conexões de usuário.

erros de replicação no Db2

A replicação do Db2 pode estar em estado de erro por vários motivos. Monitore eventos e o estado da instância de banco de dados para garantir que a instância de banco de dados esteja replicando. Para confirmar se a instância de banco de dados está replicando, verifique os logs de diagnóstico da réplica no console do Amazon RDS. Nos logs, procure erros nas mensagens do HADR. Compare o número da sequência de log com o número de sequência principal. Para obter mais informações, consulte Arquivos de log do banco de dados do Amazon RDS para Db2.

Opções para instâncias de banco de dados do Amazon RDS para Db2

Veja a seguir as opções, ou recursos adicionais, disponíveis para instâncias do Amazon RDS que executam o mecanismo de banco de dados do Db2. Para habilitar essas opções, você pode adicioná-las a um grupo de opções personalizado e, em seguida, associe o grupo de opções à sua instância de banco de dados. Para mais informações sobre como trabalhar com grupos de opções, consulte Trabalhar com grupos de opções.

O Amazon RDS é compatível com as seguintes opções do Db2:

Opção	ID da opção
Registro em log de auditoria do Db2	`DB2_AUDIT`

Registro em log de auditoria do Db2

Com o registro em log de auditoria do Db2, o Amazon RDS registra a atividade do banco de dados, incluindo usuários que fazem login no banco de dados e as consultas ao banco de dados que são executadas. O RDS faz upload dos logs de auditoria concluídos no bucket do Amazon S3 usando o perfil do AWS Identity and Access Management (IAM) que você fornece.

Tópicos

- Configurar o registro em log de auditoria do Db2
- Gerenciar o registro em log de auditoria do Db2
- Visualizar logs de auditoria
- Solução de problemas do registro em log da auditoria do Db2

Configurar o registro em log de auditoria do Db2

Para habilitar o registro em log de auditoria para um banco de dados do Amazon RDS para Db2, você habilita a opção `DB2_AUDIT` na instância de banco de dados do RDS para Db2. Depois, configure uma política de auditoria para habilitar o recurso para o banco de dados específico. Para habilitar a opção na instância de banco de dados do RDS para Db2, você define as configurações da opção `DB2_AUDIT`. Você faz isso fornecendo os nomes de recursos da Amazon (ARNs) para o bucket do Amazon S3 e o perfil do IAM com permissões para acessar o bucket.

Para configurar o registro em log de auditoria do Db2 para um banco de dados do RDS para Db2, conclua as etapas a seguir.

Tópicos

- Etapa 1: Crie um bucket do Amazon S3
- Etapa 2: criar uma política do IAM
- Etapa 3: criar um perfil do IAM e associar a política do IAM
- Etapa 4: configurar um grupo de opções para o registro em log de auditoria do Db2
- Etapa 5: configurar a política de auditoria
- Etapa 6: verificar a configuração da auditoria

Etapa 1: Crie um bucket do Amazon S3

Se você ainda não tiver feito isso, crie um bucket do Amazon S3 no qual o Amazon RDS possa fazer upload dos arquivos de log de auditoria do banco de dados do RDS para Db2. As seguintes restrições se aplicam ao bucket do S3 que você usa como destino para arquivos de auditoria:

- Ele deve estar na mesma Região da AWS que a instância de banco de dados do RDS para Db2.
- Ele não deve estar aberto ao público.
- O proprietário do bucket também deve ser o proprietário da função do IAM.

Para saber como criar um bucket do Amazon S3, consulte Criação de um bucket no Guia do usuário do Amazon S3.

Depois de habilitar o registro em log de auditoria, o Amazon RDS envia automaticamente os logs da instância de banco de dados para os seguintes locais:

- Logs do nível da instância de banco de dados: *bucket_name*`/db2-audit-logs/`*dbi_resource_id*`/`*date_time_utc*`/`
- Logs do nível de banco de dados: *bucket_name*`/db2-audit-logs/`*dbi_resource_id*`/`*date_time_utc*`/`*db_name*`/`

Anote o nome do recurso da Amazon (ARN) do bucket. Essas informações são necessárias para concluir as etapas subsequentes.

Etapa 2: criar uma política do IAM

Crie uma política do IAM com as permissões exigidas para transferir arquivos de log de auditoria da instância de banco de dados para o bucket do Amazon S3. Essa etapa pressupõe que você tenha um bucket do S3.

Antes de criar a política, colete as seguintes informações:

- O ARN do bucket.
- O ARN da chave do AWS Key Management Service (AWS KMS), se o bucket usar criptografia SSE-KMS.

Crie uma política do IAM que inclua as seguintes permissões:

```
"s3:ListBucket",
 "s3:GetBucketAcl",
 "s3:GetBucketLocation",
 "s3:PutObject",
 "s3:ListMultipartUploadParts",
 "s3:AbortMultipartUpload",
 "s3:ListAllMyBuckets"
```

> **Note**
>
> O Amazon RDS precisa da ação `s3:ListAllMyBuckets` internamente para verificar se a mesma Conta da AWS tem o bucket do S3 e a instância de banco de dados do Db2.

Se o bucket usa criptografia SSE-KMS, inclua também as permissões a seguir para o perfil do IAM e a chave do AWS KMS.

Inclua as permissões a seguir na política para o perfil do IAM.

```
"kms:GenerateDataKey",
 "kms:Decrypt"
```

Inclua as permissões a seguir na política de chave para a sua chave AWS KMS. Substitua *111122223333* pelo número da conta e *AROA123456789EXAMPLE* pelo nome do perfil do IAM.

```
{
  "Sid": "Allow RDS role to use the key",
  "Effect": "Allow",
  "Principal": {
    "AWS": [
      "arn:aws:sts::111122223333:assumed-role/AROA123456789EXAMPLE/RDS-Db2Audit",
      "arn:aws:iam::111122223333:role/AROA123456789EXAMPLE"
    ]
  },
  "Action": [
    "kms:GenerateDataKey",
    "kms:Decrypt"
  ],
  "Resource": "*"
}
```

É possível criar uma política do IAM usando o AWS Management Console ou a AWS Command Line Interface (AWS CLI).

Console

Como criar uma política do IAM para permitir acesso do Amazon RDS a um bucket do Amazon S3

1. Faça login no AWS Management Console e abra o console do IAM, em https://console.aws.amazon.com/iam/.
2. No painel de navegação, selecione Políticas.
3. Selecione Criar política e, depois, escolha a guia JSON.
4. Em Adicionar ações, filtre por S3. Adicione acesso ListBucket, GetBucketAcl e GetBucketLocation.
5. Em Adicionar um recurso, escolha Adicionar. Em Tipo de recurso, escolha bucket e insira o nome do bucket. Depois, escolha Adicionar recurso.
6. Escolha Adicionar nova instrução.
7. Em Adicionar ações, filtre por S3. Adicione acesso PutObject, ListMultipartUploadParts e AbortMultipartUpload.
8. Em Adicionar um recurso, escolha Adicionar. Em Tipo de recurso, escolha objeto e insira *o nome do bucket/**. Depois, escolha Adicionar recurso.
9. Escolha Adicionar nova instrução.
10. Em Adicionar ações, filtre por S3. Adicione acesso ListAllMyBuckets.
11. Em Adicionar um recurso, escolha Adicionar. Em Tipo de recurso, escolha Todos os recursos. Depois, escolha Adicionar recurso.
12. Se estiver usando suas próprias chaves do KMS para criptografar os dados:
    1. Escolha Adicionar nova instrução.
    2. Em Adicionar ações, filtre por KMS. Adicione acesso GenerateDataKey e Decrypt.
    3. Em Adicionar um recurso, escolha Adicionar. Em Tipo de recurso, escolha Todos os recursos. Depois, escolha Adicionar recurso.
13. Escolha Próximo.
14. Em Nome da política, digite um nome para essa política.
15. (Opcional) Em Descrição, digite uma descrição para essa política.
16. Escolha Criar política.

AWS CLI

Como criar uma política do IAM para permitir acesso do Amazon RDS a um bucket do Amazon S3

1. Execute o comando create-policy. No exemplo a seguir, substitua *iam_policy_name* e *amzn-s3-demo-bucket* pelo nome da sua política do IAM e pelo nome do bucket de destino do Amazon S3.

   Para Linux, macOS ou Unix:

```
aws iam create-policy \
    --policy-name iam_policy_name \
    --policy-document '{
        "Version": "2012-10-17",
        "Statement": [
            {
                "Sid": "Statement1",
                "Effect": "Allow",
                "Action": [
                    "s3:ListBucket",
                    "s3:GetBucketAcl",
                    "s3:GetBucketLocation"
                ],
                "Resource": [
                    "arn:aws:s3:::amzn-s3-demo-bucket"
                ]
            },
            {
                "Sid": "Statement2",
                "Effect": "Allow",
                "Action": [
                    "s3:PutObject",
                    "s3:ListMultipartUploadParts",
                    "s3:AbortMultipartUpload"
                ],
                "Resource": [
                    "arn:aws:s3:::amzn-s3-demo-bucket/*"
                ]
            },
            {
                "Sid": "Statement3",
                "Effect": "Allow",
                "Action": [
```

```
                    "s3:ListAllMyBuckets"
                ],
                "Resource": [
                    "*"
                ]
            },
            {
                "Sid": "Statement4",
                "Effect": "Allow",
                "Action": [
                    "kms:GenerateDataKey",
                    "kms:Decrypt"
                ],
                "Resource": [
                    "*"
                ]
            }
        ]
  }'
```

Para Windows:

```
aws iam create-policy ^
    --policy-name iam_policy_name ^
    --policy-document '{
        "Version": "2012-10-17",
        "Statement": [
            {
                "Sid": "Statement1",
                "Effect": "Allow",
                "Action": [
                    "s3:ListBucket",
                    "s3:GetBucketAcl",
                    "s3:GetBucketLocation"
                ],
                "Resource": [
                    "arn:aws:s3:::amzn-s3-demo-bucket"
                ]
            },
            {
                "Sid": "Statement2",
                "Effect": "Allow",
                "Action": [
```

```
                    "s3:PutObject",
                    "s3:ListMultipartUploadParts",
                    "s3:AbortMultipartUpload"
                ],
                "Resource": [
                    "arn:aws:s3:::amzn-s3-demo-bucket/*"
                ]
            },
            {
                "Sid": "Statement3",
                "Effect": "Allow",
                "Action": [
                    "s3:ListAllMyBuckets"
                ],
                "Resource": [
                    "*"
                ]
            },
            {
                "Sid": "Statement4",
                "Effect": "Allow",
                "Action": [
                    "kms:GenerateDataKey",
                    "kms:Decrypt"
                ],
                "Resource": [
                    "*"
                ]
            }
        ]
    }'
```

2. Depois de criar a política, anote o ARN da política. É necessário ter o ARN do Etapa 3: criar um perfil do IAM e associar a política do IAM.

Para ter informações sobre como criar uma política do IAM, consulte Criação de políticas do IAM no Guia do usuário do IAM.

Etapa 3: criar um perfil do IAM e associar a política do IAM

Esta etapa pressupõe que você tenha criado a política do IAM no Etapa 2: criar uma política do IAM. Nesta etapa, você vai criar um perfil do IAM para a instância de banco de dados do RDS para Db2 e anexar a política do IAM ao perfil.

É possível criar um perfil do IAM para a instância de banco de dados usando o console ou a AWS CLI.

Console

Como criar um perfil do IAM e associar a política do IAM a ele

1. Faça login no AWS Management Console e abra o console do IAM, em https://console.aws.amazon.com/iam/.
2. No painel de navegação, selecione Perfis.
3. Escolha Criar Perfil.
4. Em Tipo de entidade confiável, selecione AWS service (Serviço da AWS).
5. Para Serviço ou caso de uso, selecione RDS e, depois, selecione RDS: adicionar perfil ao banco de dados.
6. Escolha Próximo.
7. Em Políticas de permissões, pesquise e selecione o nome da política do IAM que você criou.
8. Escolha Próximo.
9. Em Role name (Nome da função), insira um nome.
10. (Opcional) Em Descrição da função, insira uma descrição para a nova função.
11. Selecione Criar perfil.

AWS CLI

Como criar um perfil do IAM e associar a política do IAM a ele

1. Execute o comando create-role. No exemplo a seguir, substitua *iam_role_name* por um nome para o perfil do IAM.

   Para Linux, macOS ou Unix:

```
aws iam create-role \
    --role-name iam_role_name \
    --assume-role-policy-document '{
      "Version": "2012-10-17",
      "Statement": [
        {
          "Effect": "Allow",
          "Principal": {
```

```
            "Service": "rds.amazonaws.com"
          },
          "Action": "sts:AssumeRole"
        }
      ]
    }'
```

Para Windows:

```
aws iam create-role ^
    --role-name iam_role_name ^
    --assume-role-policy-document '{
      "Version": "2012-10-17",
      "Statement": [
        {
          "Effect": "Allow",
          "Principal": {
            "Service": "rds.amazonaws.com"
          },
          "Action": "sts:AssumeRole"
        }
      ]
    }'
```

2. Depois que o perfil for criado, observe o respectivo ARN. Você precisará desse ARN na próxima etapa, Etapa 4: configurar um grupo de opções para o registro em log de auditoria do Db2.
3. Execute o comando attach-role-policy. No exemplo a seguir, substitua *iam_policy_arn* pelo ARN da política do IAM que você criou em Etapa 2: criar uma política do IAM. Substitua *iam_role_name* pelo nome do perfil do IAM que você acabou de criar.

   Para Linux, macOS ou Unix:

   ```
   aws iam attach-role-policy \
      --policy-arn iam_policy_arn \
      --role-name iam_role_name
   ```

   Para Windows:

   ```
   aws iam attach-role-policy ^
      --policy-arn iam_policy_arn ^
      --role-name iam_role_name
   ```

Para ter mais informações, consulte Criar um perfil para delegar permissões a um usuário do IAM no Guia do usuário do IAM.

Etapa 4: configurar um grupo de opções para o registro em log de auditoria do Db2

O processo para adicionar a opção de registro em log de auditoria do Db2 a uma instância de banco de dados do RDS para Db2 é o seguinte:

1. Crie um novo grupo de opções, ou copie ou modifique um existente.
2. Adicione e configure todas as opções necessárias.
3. Associe o grupo de opções à instância de banco de dados.

Depois de adicionar a opção de registro em log de auditoria do Db2, você não precisará reiniciar a instância de banco de dados. Assim que o grupo de opções estiver ativo, você poderá criar auditorias e armazenar logs de auditoria no seu bucket do S3.

Para adicionar e configurar o registro em log de auditoria do Db2 em um grupo de opções da instância de banco de dados

1. Escolha uma das seguintes opções:
   - Use um grupo de opções existente.
   - Crie um grupo de opções de banco de dados personalizado e use esse grupo de opções. Para obter mais informações, consulte Criar um grupo de opções.
2. Adicione a opção DB2_AUDIT ao grupo de opções e defina as configurações da opção. Para ter mais informações sobre a adição de opções, consulte Adicionar uma opção a um grupo de opções.
   - Em IAM_ROLE_ARN, insira o ARN do perfil do IAM que você criou em the section called “Criar um perfil do IAM e associar a política do IAM”.
   - Para S3_BUCKET_ARN, insira o ARN do bucket do S3 a ser usado para os logs de auditoria do Db2. O bucket deve estar na mesma região que a instância de banco de dados do RDS para Db2. A política associada ao perfil do IAM que você inseriu deve permitir as operações necessárias nesse recurso.
3. Aplique o grupo de opções a uma instância de banco de dados nova ou existente. Escolha uma das seguintes opções:

- Se você estiver criando uma nova instância de banco de dados, aplique o grupo de opções ao executar a instância.
- Em uma instância de banco de dados existente, aplique o grupo de opções modificando a instância e anexando o novo grupo de opções. Para obter mais informações, consulte Modificar uma instância de banco de dados do Amazon RDS.

Etapa 5: configurar a política de auditoria

Para configurar a política de auditoria do banco de dados do RDS para Db2, conecte-se ao banco de dados `rdsadmin` usando o nome de usuário e a senha principais da instância de banco de dados do RDS para Db2. Depois, chame o procedimento armazenado `rdsadmin.configure_db_audit` com o nome do banco de dados e os valores dos parâmetros aplicáveis.

O exemplo a seguir se conecta ao banco de dados e configura uma política de auditoria `testdb` com as categorias AUDIT, CHECKING, OBJMAINT, SECMAINT, SYSADMIN e VALIDATE. O valor do status `BOTH` registra em log sucessos e falhas, e o `ERROR TYPE` é `NORMAL` por padrão. Consulte mais informações sobre como usar esse procedimento armazenado em the section called “rdsadmin.configure_db_audit”.

```
db2 "connect to rdsadmin user master_user using master_password"
db2 "call rdsadmin.configure_db_audit('testdb', 'ALL', 'BOTH', ?)"
```

Etapa 6: verificar a configuração da auditoria

Para garantir que a política de auditoria esteja configurada corretamente, verifique o status da configuração da auditoria.

Para verificar a conexão, conecte-se ao banco de dados `rdsadmin` usando o nome de usuário e a senha principais da instância de banco de dados do RDS para Db2. Depois, use a instrução SQL a seguir com o nome do banco de dados. No exemplo a seguir, o nome do banco de dados é *`testdb`*.

```
db2 "select task_id, task_type, database_name, lifecycle,
    varchar(bson_to_json(task_input_params), 500) as task_params,
    cast(task_output as varchar(500)) as task_output
    from table(rdsadmin.get_task_status(null,'testdb','CONFIGURE_DB_AUDIT'))"

Sample Output
```

```
TASK_ID              TASK_TYPE            DATABASE_NAME       LIFECYCLE
-------------------- -------------------- --------------- -------------
                   2  CONFIGURE_DB_AUDIT            DB2DB       SUCCESS

... continued ...
TASK_PARAMS
------------------------------------------------------------
{ "AUDIT_CATEGORY" : "ALL", "CATEGORY_SETTING" : "BOTH" }

... continued ...
                                          TASK_OUTPUT
-------------------------------------------------
2023-12-22T20:27:03.029Z Task execution has started.

2023-12-22T20:27:04.285Z Task execution has completed successfully.
```

Gerenciar o registro em log de auditoria do Db2

Depois de configurar o registro em log de auditoria do Db2, você pode modificar a política de auditoria para um banco de dados específico ou desabilitar o registro em log de auditoria no nível do banco de dados ou para toda a instância de banco de dados. Você também pode alterar o bucket do Amazon S3 para o qual os arquivos de log são enviados.

Tópicos

- Modificar uma política de auditoria do Db2
- Modificar o local dos arquivos de log
- Desabilitar o registro em log de auditoria do Db2

Modificar uma política de auditoria do Db2

Para modificar a política de auditoria de um banco de dados específico do RDS para Db2, execute o procedimento armazenado `rdsadmin.configure_db_audit`. Com esse procedimento armazenado, você pode alterar as categorias, as configurações da categoria e a configuração do tipo de erro da política de auditoria. Para obter mais informações, consulte the section called “rdsadmin.configure_db_audit”.

Modificar o local dos arquivos de log

Para alterar o bucket do Amazon S3 no qual os arquivos de log são carregados, realize uma das seguintes ações:

- Modifique o grupo de opções atual anexado à instância de banco de dados do RDS para Db2: atualize a configuração `S3_BUCKET_ARN` da opção `DB2_AUDIT` a fim de apontar para o novo bucket. Além disso, atualize a política do IAM anexada ao perfil do IAM especificado pela configuração `IAM_ROLE_ARN` no grupo de opções anexado. Essa política do IAM deve fornecer ao novo bucket as permissões de acesso necessárias. Consulte mais informações sobre as permissões exigidas na política do IAM em Criar uma política do IAM.
- Anexe a instância de banco de dados do RDS para Db2 a um grupo de opções diferente: modifique a instância de banco de dados para alterar o grupo de opções anexado a ela. O novo grupo de opções deve estar configurado com as definições `S3_BUCKET_ARN` e `IAM_ROLE_ARN` corretas. Consulte informações sobre como definir essas configurações para a opção `DB2_AUDIT` em Configurar um grupo de opções.

Ao modificar o grupo de opções, aplique as alterações imediatamente. Para obter mais informações, consulte the section called “Modificação de uma instância de banco de dados”.

Desabilitar o registro em log de auditoria do Db2

Para desabilitar o registro em log de auditoria do Db2, realize uma das seguintes ações:

- Desabilite o registro em log de auditoria para a instância de banco de dados do RDS para Db2: modifique a instância de banco de dados e remova o grupo de opções com a opção `DB2_AUDIT` da instância. Para obter mais informações, consulte the section called “Modificação de uma instância de banco de dados”.
- Desabilite o registro em log de auditoria para um banco de dados específico: interrompa o registro em log de auditoria e remova a política de auditoria chamando `rdsadmin.disable_db_audit` com o nome do banco de dados. Para obter mais informações, consulte the section called “rdsadmin.disable_db_audit”.

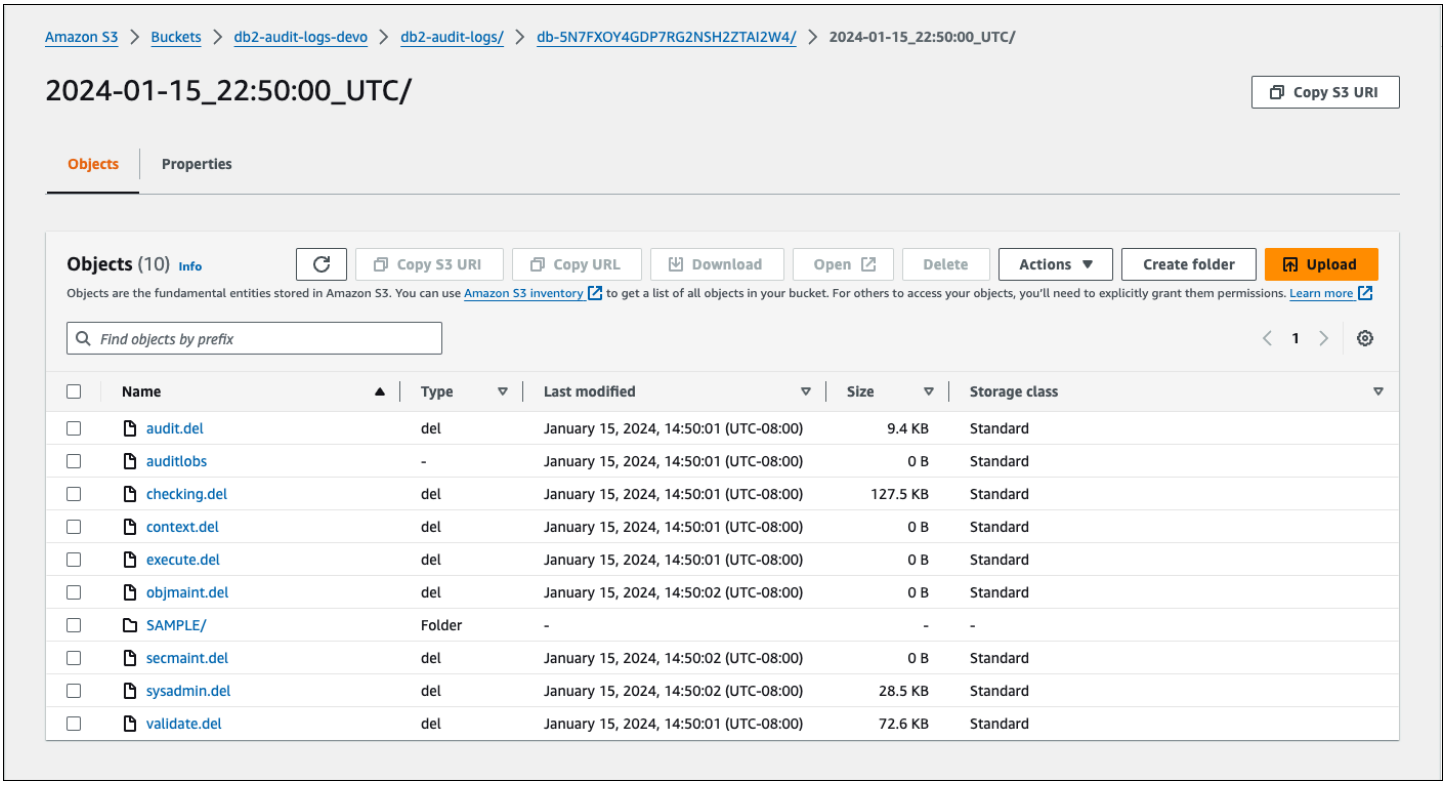
```
db2 "call rdsadmin.disable_db_audit(
    'db_name',
    ?)"
```

Visualizar logs de auditoria

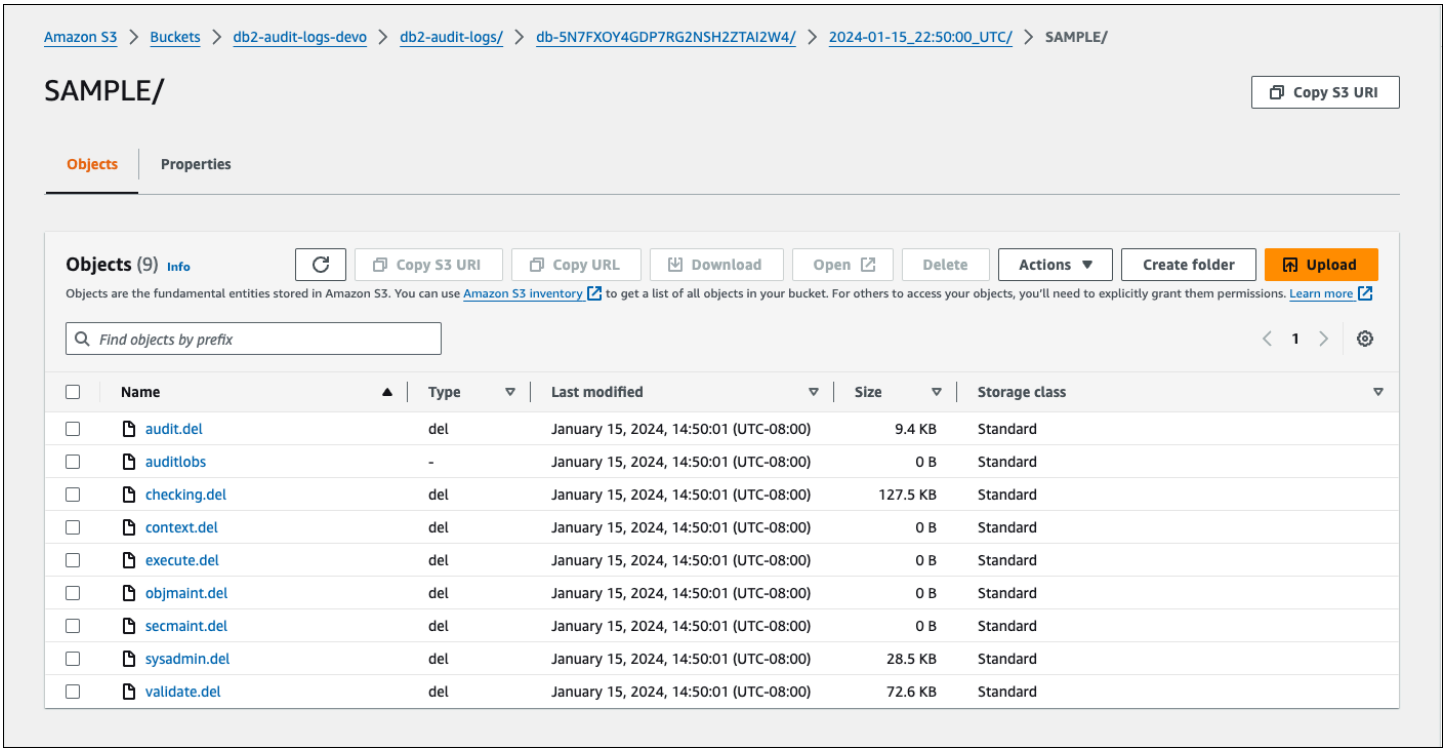
Depois de habilitar o registro em log de auditoria do Db2, aguarde pelo menos uma hora antes de visualizar os dados de auditoria no bucket do Amazon S3. O Amazon RDS envia automaticamente os logs da instância de banco de dados do RDS para Db2 aos seguintes locais:

- Logs do nível da instância de banco de dados: *bucket_name*/db2-audit-logs/*dbi_resource_id*/*date_time_utc*/
- Logs do nível de banco de dados: *bucket_name*/db2-audit-logs/*dbi_resource_id*/*date_time_utc*/*db_name*/

O exemplo de captura de tela a seguir do console do Amazon S3 mostra uma lista de pastas para arquivos de log no nível da instância de banco de dados do RDS para Db2.



O exemplo de captura de tela a seguir do console do Amazon S3 mostra arquivos de log no nível de banco de dados para a instância de banco de dados do RDS para Db2.



Solução de problemas do registro em log da auditoria do Db2

Use as informações a seguir para solucionar problemas comuns com o registro em log de auditoria do Db2.

Não é possível configurar a política de auditoria

Se chamar o procedimento armazenado `rdsadmin.configure_db_audit` retornar um erro, pode ser que o grupo de opções com a opção `DB2_AUDIT` não esteja associado à instância de banco de dados do RDS para Db2. Modifique a instância de banco de dados para adicionar o grupo de opções e tente chamar o procedimento armazenado novamente. Para obter mais informações, consulte Modificar uma instância de banco de dados do Amazon RDS.

Nenhum dado no bucket do Amazon S3

Se não houver nenhum dado de log no bucket do Amazon S3, verifique se:

- O bucket do Amazon S3 está na mesma região da instância de banco de dados do RDS para Db2.
- O perfil que você especificou na configuração da opção `IAM_ROLE_ARN` está configurado com as permissões necessárias para fazer upload de logs no bucket do Amazon S3. Para obter mais informações, consulte Criar uma política do IAM.

- Os ARNs para as configurações da opção `IAM_ROLE_ARN` e `S3_BUCKET_ARN` estão corretos no grupo de opções associado à instância de banco de dados do RDS para Db2. Para obter mais informações, consulte Configurar um grupo de opções.

Você pode verificar o status da tarefa da configuração do registro em log de auditoria conectando-se ao banco de dados e executando uma instrução SQL. Para obter mais informações, consulte Verificar a configuração da auditoria.

Também é possível verificar os eventos para saber mais sobre por que os logs podem estar ausentes. Consulte informações sobre como visualizar eventos em the section called “Visualizar logs, eventos e transmissões no console do Amazon RDS”.

Procedimentos armazenados externos do Amazon RDS para Db2

É possível criar rotinas externas e registrá-las em bancos de dados do Amazon RDS para Db2 como procedimentos armazenados externos. No momento, o RDS para Db2 permite somente rotinas baseadas em Java para procedimentos armazenados externos.

Procedimentos armazenados externos baseados em Java

Procedimentos armazenados externos baseados em Java são rotinas externas do Java que você registra no banco de dados do RDS para Db2 como procedimentos armazenados externos.

Tópicos

- Limitações para procedimentos armazenados externos baseados em Java
- Configurar procedimentos armazenados externos baseados em Java

Limitações para procedimentos armazenados externos baseados em Java

Antes de desenvolver a rotina externa, considere as seguintes limitações e restrições.

Para criar a rotina externa, use o Java Development Kit (JDK) fornecido pelo Db2. Consulte mais informações em Java software support for Db2 database products.

O programa Java pode criar arquivos somente no diretório `/tmp`, e o Amazon RDS não permite habilitar permissões executáveis ou de definição de ID de usuário (SUID) nesses arquivos. O programa Java também não pode usar chamadas de sistema de soquete nem as seguintes chamadas de sistema:

- _sysctl
- acct
- afs_syscall
- bpf
- capset
- chown
- chroot
- create_module

- delete_module
- fanotify_init
- fanotify_mark
- finit_module
- fsconfig
- fsopen
- fspick
- get_kernel_syms
- getpmsg
- init_module
- mount
- move_mount
- nfsservctl
- open_by_handle_at
- open_tree
- pivot_root
- putpmsg
- query_module
- quotactl
- reboot
- security
- setdomainname
- setfsuid
- sethostname
- sysfs
- tuxcall
- umount2
- uselib
- ustat

- vhangup
- vserver

Consulte as restrições adicionais sobre rotinas externas para o Db2 em Restrictions on external routines na documentação do IBM Db2.

Configurar procedimentos armazenados externos baseados em Java

Para configurar um procedimento armazenado externo, crie um arquivo .jar com a rotina externa, instale-o no banco de dados do RDS para Db2 e, depois, registre-o como um procedimento armazenado externo.

Tópicos

- Etapa 1: habilitar procedimentos armazenados externos
- Etapa 2: instalar o arquivo .jar com a rotina externa
- Etapa 3: registrar o procedimento armazenado externo
- Etapa 4: validar o procedimento armazenado externo

Etapa 1: habilitar procedimentos armazenados externos

Para habilitar procedimentos armazenados externos, em um grupo de parâmetros personalizado associado à instância de banco de dados, defina o parâmetro `db2_alternate_authz_behaviour` como um dos seguintes valores:

- `EXTERNAL_ROUTINE_DBADM`: concede implicitamente a permissão `CREATE_EXTERNAL_ROUTINE` a qualquer usuário, grupo ou perfil com a autoridade `DBADM`.
- `EXTERNAL_ROUTINE_DBAUTH`: permite que um usuário com a autoridade `DBADM` conceda a permissão `CREATE_EXTERNAL_ROUTINE` a qualquer usuário, grupo ou perfil. Nesse caso, nenhum usuário, grupo ou perfil recebe essa permissão implicitamente, nem mesmo um usuário com a autoridade `DBADM`.

Consulte mais informações sobre essa configuração em GRANT (database authorities) statement na documentação do IBM Db2.

É possível criar e modificar um grupo de parâmetros personalizado usando o AWS Management Console, a AWS CLI ou a API do Amazon RDS.

Console

Para configurar o parâmetro db2_alternate_authz_behaviour em um grupo de parâmetros personalizado

1. Se você quiser usar um grupo de parâmetros de banco de dados personalizado diferente do que o que a instância de banco de dados está usando, crie outro grupo de parâmetros de banco de dados. Se você estiver usando o modelo traga a sua própria licença (BYOL), garanta que o novo grupo de parâmetros personalizado inclua os IDs da IBM. Consulte informações sobre esses IDs em the section called “IDs da IBM para traga a sua própria licença para o Db2”. Para obter mais informações sobre como criar um parameter group de banco de dados, consulte Criar um grupo de parâmetros de banco de dados no Amazon RDS.
2. Defina o valor do parâmetro `db2_alternate_authz_behaviour` em seu grupo de parâmetros personalizado. Para ter mais informações sobre como modificar um grupo de parâmetros, consulte Modificar parâmetros em um grupo de parâmetros de banco de dados no Amazon RDS.

AWS CLI

Para configurar o parâmetro db2_alternate_authz_behaviour em um grupo de parâmetros personalizado

1. Se você quiser usar um grupo de parâmetros de banco de dados personalizado diferente do que o que a instância de banco de dados está usando, crie um grupo de parâmetros de personalizado executando o comando create-db-parameter-group. Se você estiver usando o modelo traga a sua própria licença (BYOL), garanta que o novo grupo de parâmetros personalizado inclua os IDs da IBM. Consulte informações sobre esses IDs em the section called “IDs da IBM para traga a sua própria licença para o Db2”.

   Inclua as seguintes opções necessárias:

   - `--db-parameter-group-name`: um nome para o grupo de parâmetros que você está criando.
   - `--db-parameter-group-family`: a edição do mecanismo do Db2 e a versão principal. Os valores válidos são `db2-se-11.5` e `db2-ae-11.5`.
   - `--description`: a descrição desse grupo de parâmetros.

Para obter mais informações sobre como criar um parameter group de banco de dados, consulte Criar um grupo de parâmetros de banco de dados no Amazon RDS.

O exemplo a seguir mostra como criar um grupo de parâmetros personalizado chamado `MY_EXT_SP_PARAM_GROUP` para a família do grupos de parâmetros `db2-se-11.5`.

Para Linux, macOS ou Unix:

```
aws rds create-db-parameter-group \
--region us-east-1 \
--db-parameter-group-name MY_EXT_SP_PARAM_GROUP \
--db-parameter-group-family db2-se-11.5 \
--description "test db2 external routines"
```

Para Windows:

```
aws rds create-db-parameter-group ^
--region us-east-1 ^
--db-parameter-group-name MY_EXT_SP_PARAM_GROUP ^
--db-parameter-group-family db2-se-11.5 ^
--description "test db2 external routines"
```

2. Modifique o parâmetro `db2_alternate_authz_behaviour` em seu grupo de parâmetros personalizado executando o comando modify-db-parameter-group.

   Inclua as seguintes opções necessárias:

   - `--db-parameter-group-name`: o nome do grupo de parâmetros criado.
   - `--parameters`: uma matriz de nomes de parâmetros, valores e os métodos de aplicação da atualização de parâmetros.

   Para ter mais informações sobre como modificar um grupo de parâmetros, consulte Modificar parâmetros em um grupo de parâmetros de banco de dados no Amazon RDS.

   O exemplo a seguir mostra como modificar o grupo de parâmetros `MY_EXT_SP_PARAM_GROUP` definindo o valor de `db2_alternate_authz_behaviour` como `EXTERNAL_ROUTINE_DBADM`.

   Para Linux, macOS ou Unix:

```
aws rds modify-db-parameter-group \
    --db-parameter-group-name MY_EXT_SP_PARAM_GROUP \
    --parameters
 "ParameterName='db2_alternate_authz_behaviour',ParameterValue='EXTERNAL_ROUTINE_DBADM',App
```

Para Windows:

```
aws rds modify-db-parameter-group ^
    --db-parameter-group-name MY_EXT_SP_PARAM_GROUP ^
    --parameters
 "ParameterName='db2_alternate_authz_behaviour',ParameterValue='EXTERNAL_ROUTINE_DBADM',App
```

API do RDS

Para configurar o parâmetro db2_alternate_authz_behaviour em um grupo de parâmetros personalizado

1. Se você quiser usar um grupo de parâmetros de banco de dados personalizado diferente do que o que a instância de banco de dados está usando, crie outro grupo de parâmetros de banco de dados usando a operação de API CreateDBParameterGroup do Amazon RDS. Se você estiver usando o modelo traga a sua própria licença (BYOL), garanta que o novo grupo de parâmetros personalizado inclua os IDs da IBM Db2. Consulte informações sobre esses IDs em the section called “IDs da IBM para traga a sua própria licença para o Db2”.

   Inclua os seguintes parâmetros necessários:

   - `DBParameterGroupName`
   - `DBParameterGroupFamily`
   - `Description`

   Para obter mais informações sobre como criar um parameter group de banco de dados, consulte Criar um grupo de parâmetros de banco de dados no Amazon RDS.
2. Modifique o parâmetro `db2_alternate_authz_behaviour` no grupo de parâmetros personalizado que você criou usando a operação de API ModifyDBParameterGroup do RDS.

   Inclua os seguintes parâmetros necessários:

- `DBParameterGroupName`
- `Parameters`

Para ter mais informações sobre como modificar um grupo de parâmetros, consulte Modificar parâmetros em um grupo de parâmetros de banco de dados no Amazon RDS.

Etapa 2: instalar o arquivo .jar com a rotina externa

Depois de criar a rotina do Java, crie o arquivo .jar e execute `db2 "call sqlj.install_jar('file:file_path',jar_ID)"` para instalá-lo no banco de dados do RDS para Db2.

O exemplo a seguir mostra como criar uma rotina do Java e instalá-la em um banco de dados do RDS para Db2. O exemplo inclui um exemplo de código para uma rotina simples que pode ser usada para testar o processo. Esse exemplo faz as seguintes suposições:

- O código Java é compilado em um servidor em que o Db2 está instalado. Essa é uma prática recomendada porque não compilar com o JDK fornecido pela IBM pode resultar em erros inexplicáveis.
- O servidor tem o banco de dados do RDS para Db2 catalogado localmente.

Se você quiser experimentar o processo com o código de exemplo a seguir, copie-o e salve-o em um arquivo chamado `MYJAVASP.java`.

```
import java.sql.*;
public class MYJAVASP
{
public static void my_JAVASP (String inparam) throws SQLException, Exception
{
try
{
// Obtain the calling context's connection details.
Connection myConn = DriverManager.getConnection("jdbc:default:connection");
String myQuery = "INSERT INTO TEST.TEST_TABLE VALUES (?, CURRENT DATE)";
PreparedStatement myStmt = myConn.prepareStatement(myQuery);
myStmt.setString(1, inparam);
myStmt.executeUpdate();
}
```

```
catch (SQLException sql_ex)
{
throw sql_ex;
}
catch (Exception ex)
{
throw ex;
}
}
```

O comando a seguir compila a rotina do Java.

```
~/sqllib/java/jdk64/bin/javac MYJAVASP.java
```

O comando a seguir cria o arquivo .jar.

```
~/sqllib/java/jdk64/bin/jar cvf MYJAVASP.jar MYJAVASP.class
```

Os comandos a seguir se conectam ao banco de dados chamado `MY_DB2_DATABASE` e instalam o arquivo .jar.

```
db2 "connect to MY_DB2_DATABASE user master_username using master_password"

db2 "call sqlj.install_jar('file:/tmp/MYJAVASP.jar','MYJAVASP')"
db2 "call sqlj.refresh_classes()"
```

Etapa 3: registrar o procedimento armazenado externo

Depois de instalar o arquivo .jar no banco de dados do RDS para Db2, registre-o como um procedimento armazenado executando o comando `db2 CREATE PROCEDURE` ou `db2 REPLACE PROCEDURE`.

O exemplo a seguir mostra como se conectar ao banco de dados e registrar a rotina do Java criada na etapa anterior como um procedimento armazenado.

```
db2 "connect to MY_DB2_DATABASE user master_username using master_password"

create procedure TESTSP.MYJAVASP (in input char(6))
specific myjavasp
dynamic result sets 0
```

```
deterministic
language java
parameter style java
no dbinfo
fenced
threadsafe
modifies sql data
program type sub
external name 'MYJAVASP!my_JAVASP';
```

Etapa 4: validar o procedimento armazenado externo

Use as etapas a seguir para testar o exemplo de procedimento armazenado externo que foi registrado na etapa anterior.

Como validar o procedimento armazenado externo

1. Crie uma tabela como `TEST.TEST_TABLE` no exemplo a seguir.

   ```
   db2 "create table TEST.TEST_TABLE(C1 char(6), C2 date)"
   ```

2. Chame o novo procedimento armazenado externo. A chamada retorna um status de `0`.

   ```
   db2 "call TESTSP.MYJAVASP('test')"
   Return Status = 0
   ```

3. Consulte a tabela que você criou na etapa 1 para verificar os resultados da chamada do procedimento armazenado.

   ```
   db2 "SELECT * from TEST.TEST_TABLE"
   ```

   A consulta produz um resultado semelhante ao seguinte exemplo:

   ```
   C1     C2
   ------ ----------
   test   02/05/2024
   ```

Limitações e problemas conhecidos do Amazon RDS para Db2

Os seguintes itens são limitações e problemas conhecidos no trabalho com o Amazon RDS para Db2:

Tópicos

- Limitação de autenticação
- Rotinas não protegidas
- Espaços de tabela de armazenamento não automáticos durante a migração
- Configuração do parâmetro db2_compatibility_vector
- Migrar bancos de dados que contêm pacotes INVALID

Limitação de autenticação

O Amazon RDS define `db2auth` como `JCC_ENFORCE_SECMEC` por padrão. No entanto, se você não quiser aplicar a criptografia de ID de usuário e senha pela rede, substitua essa configuração alterando o parâmetro `db2auth` para `CLEAR_TEXT` no grupo de parâmetros. Para ter mais informações, consulte Modificar parâmetros em um grupo de parâmetros de banco de dados no Amazon RDS.

Rotinas não protegidas

O RDS para Db2 não comporta a criação de rotinas não protegidas e a migração delas por meio do backup e da restauração de dados. Para conferir se o banco de dados contém alguma rotina não protegida, execute o seguinte comando SQL:

```
SELECT 'COUNT:' || count(*) FROM SYSCAT.ROUTINES where fenced='N' and routineschema not
 in ('SQLJ','SYSCAT','SYSFUN','SYSIBM','SYSIBMADM','SYSPROC','SYSTOOLS')
```

Espaços de tabela de armazenamento não automáticos durante a migração

O RDS para Db2 não permite a criação de espaços de tabela de armazenamento não automáticos. Quando você usa a restauração nativa para uma migração única do banco de dados, o RDS para Db2 converte automaticamente os espaços de tabela de armazenamento não automáticos em espaços automáticos e, depois, restaura o banco de dados para o RDS para Db2. Para ter

informações sobre migrações únicas, consulte Migrar do Linux para Linux para o Amazon RDS para Db2 e Migrar do AIX ou do Windows para Linux para o Amazon RDS para Db2.

Configuração do parâmetro db2_compatibility_vector

Com o Amazon RDS, você pode criar um banco de dados inicial ao criar a instância de banco de dados e, depois, modificar os parâmetros em um grupo de parâmetros associado. No entanto, para o Db2, se você quiser definir o parâmetro `db2_compatibility_vector` em um grupo de parâmetros, primeiro modifique o parâmetro em um grupo de parâmetros personalizado, crie a instância de banco de dados sem um banco de dados e, em seguida, crie um banco de dados usando o procedimento armazenado `rdsadmin.create_database`.

Como definir o parâmetro **`db2_compatibility_vector`**

1. Crie um grupo de parâmetros personalizado. (Não é possível modificar parâmetros em grupos padrão.)
2. Modifique o parâmetro.
3. Crie uma instância de banco de dados.
4. Crie um banco de dados usando o procedimento armazenado `rdsadmin.create_database`.
5. Associe o grupo de parâmetros à instância de banco de dados que contém o banco de dados.

Migrar bancos de dados que contêm pacotes INVALID

Se migrar bancos de dados do Db2 que contêm pacotes `INVALID` para o RDS para Db2 usando o comando `RESTORE`, você poderá encontrar problemas ao começar a usar os bancos de dados. Pacotes `INVALID` podem causar problemas devido à configuração de autorização para o usuário da instância de banco de dados `rdsdb` e à remoção da autorização de `PUBLIC`. Pacotes `INVALID` fazem com que os seguintes comandos falhem:

- `db2updv115`
- `db2 "call SYSPROC.ADMIN_REVALIDATE_DB_OBJECTS()"`

Antes de migrar o banco de dados com o comando `RESTORE`, certifique-se de que ele não contenha pacotes `INVALID` executando o seguinte comando:

```
db2 "select count(*) from sysibm.sysroutines where valid = 'N'"
```

Se o comando retornar uma contagem maior que zero, chame o seguinte comando:

```
db2 "call SYSPROC.ADMIN_REVALIDATE_DB_OBJECTS()"
```

Depois, chame o comando anterior para confirmar que o banco de dados não contém mais pacotes `INVALID`.

```
db2 "select count(*) from sysibm.sysroutines where valid = 'N'"
```

Agora está tudo pronto para fazer backup do banco de dados e restaurá-lo na instância de banco de dados do RDS para Db2.

Referência de procedimentos armazenados do Amazon RDS para Db2

É possível gerenciar instâncias de banco de dados do Amazon RDS para Db2 que executam o mecanismo do Db2 chamando procedimentos armazenados incorporados.

Procedimento armazenado	Categoria	Descrição
???	Bancos de dados	Use o procedimento armazenado `rdsadmin.activate_database` para ativar um banco de dados em uma instância de banco de dados autônoma do RDS para Db2.
the section called “rdsadmin.add_groups”	Conceder e revogar privilégios	Use o procedimento armazenado `rdsadmin.add_groups` para adicionar um ou mais grupos a um usuário para um banco de dados em uma instância de banco de dados do RDS para Db2.
the section called “rdsadmin.add_user”	Conceder e revogar privilégios	Use o procedimento armazenado `rdsadmin.add_user` para adicionar um usuário a uma lista de autorização de um banco de dados em uma instância de banco de dados do RDS para Db2.
the section called “rdsadmin.alter_bufferpool”	Grupos de buffer	Use o procedimento armazenado `rdsadmin.alter_bufferpool` para modificar um grupo de buffers de um banco de dados em uma instância de banco de dados do RDS para Db2.
the section called “rdsadmin.alter_tablespace”	Tablespaces	Use o procedimento armazenado `rdsadmin.alter_tablespace` para modificar um espaço de tabela de um banco de dados em uma instância de banco de dados do RDS para Db2.

Procedimento armazenado	Categoria	Descrição
the section called “rdsadmin.catalog_storage_access”	Acesso ao armazenamento	Use o procedimento armazenado `rdsadmin.catalog_storage_access` para catalogar um alias de armazenamento para acessar um bucket do Amazon S3 com arquivos de dados do Db2 para um banco de dados em uma instância de banco de dados do RDS para Db2.
the section called “rdsadmin.change_password”	Conceder e revogar privilégios	Use o procedimento armazenado `rdsadmin.change_password` para alterar a senha de um usuário de um banco de dados em uma instância de banco de dados do RDS para Db2.
the section called “rdsadmin.complete_rollforward”	Bancos de dados	Use o procedimento armazenado `rdsadmin.complete_rollforward` para colocar um banco de dados em uma instância de banco de dados do RDS para Db2 on-line a partir de um estado `ROLL-FORWARD PENDING`. Um estado `ROLL-FORWARD PENDING` ocorre quando você chama the section called “rdsadmin.rollforward_database”, mas define o parâmetro `complete_rollforward` como `FALSE`.
the section called “rdsadmin.configure_db_audit”	Políticas de auditoria	Use o procedimento armazenado `rdsadmin.configure_db_audit` para modificar uma política de auditoria de um banco de dados em uma instância de banco de dados do RDS para Db2. Se não existir uma política de auditoria, a execução desse procedimento armazenado cria uma política de auditoria.

Procedimento armazenado	Categoria	Descrição
the section called “rdsadmin.create_bufferpool”	Grupos de buffer	Use o procedimento armazenado `rdsadmin.create_bufferpool` para criar um grupo de buffers de um banco de dados em uma instância de banco de dados do RDS para Db2.
the section called “rdsadmin.create_database”	Bancos de dados	Use o procedimento armazenado `rdsadmin.create_database` para criar um banco de dados em uma instância de banco de dados do RDS para Db2.
the section called “rdsadmin.create_role”	Conceder e revogar privilégios	Use o procedimento armazenado `rdsadmin.create_role` para criar um perfil para anexar a um banco de dados em uma instância de banco de dados do RDS para Db2.
the section called “rdsadmin.create_tablespace”	Tablespaces	Use o procedimento armazenado `rdsadmin.create_tablespace` para criar um espaço de tabela de um banco de dados em uma instância de banco de dados do RDS para Db2.
the section called “rdsadmin.db2pd_command”	Bancos de dados	Use o procedimento armazenado `rdsadmin.db2pd_command` para coletar informações sobre um banco de dados em uma instância de banco de dados do RDS para Db2. Essas informações podem ajudar no monitoramento e na solução de problemas de bancos de dados do RDS para Db2.
the section called “rdsadmin.dbadm_grant”	Conceder e revogar privilégios	Use o procedimento armazenado `rdsadmin.dbadm_grant` para conceder um ou mais tipos de autorização (`DBADM`, `ACCESSCTRL` ou `DATAACCESS`) a um ou mais perfis, usuários ou grupos de um banco de dados em uma instância de banco de dados do RDS para Db2.

Procedimento armazenado	Categoria	Descrição
the section called “rdsadmin.dbadm_revoke”	Conceder e revogar privilégios	Use o procedimento armazenado `rdsadmin.dbadm_revoke` para revogar um ou mais tipos de autorização (`DBADM`, `ACCESSCTRL` ou `DATAACCESS`) de um ou mais perfis, usuários ou grupos de um banco de dados em uma instância de banco de dados do RDS para Db2.
???	Bancos de dados	Use o procedimento armazenado `rdsadmin.deactivate_database` para desativar um banco de dados em uma instância de banco de dados do RDS para Db2. Você pode desativar bancos de dados usados para preservar recursos de memória.
the section called “rdsadmin.disable_db_audit”	Políticas de auditoria	Use o procedimento armazenado `rdsadmin.disable_db_audit` para interromper o registro em log de auditoria e remover uma política de auditoria de um banco de dados em uma instância de banco de dados do RDS para Db2.
the section called “rdsadmin.drop_bufferpool”	Grupos de buffer	Use o procedimento armazenado `rdsadmin.drop_bufferpool` para descartar um grupo de buffers de um banco de dados em uma instância de banco de dados do RDS para Db2.
the section called “rdsadmin.drop_database”	Bancos de dados	Use o procedimento armazenado `rdsadmin.drop_database` para descartar um banco de dados de uma instância de banco de dados do RDS para Db2.

Procedimento armazenado	Categoria	Descrição
???	Conceder e revogar privilégios	Use o procedimento armazenado `rdsadmin.drop_role` para excluir um perfil de um banco de dados em uma instância de banco de dados do RDS para Db2.
the section called “rdsadmin.drop_tablespace”	Tablespaces	Use o procedimento armazenado `rdsadmin.drop_tablespace` para descartar um espaço de tabela de um banco de dados em uma instância de banco de dados do RDS para Db2.
the section called “rdsadmin.force_application”	Bancos de dados	Use o procedimento armazenado `rdsadmin.force_application` para forçar a saída de aplicações de um banco de dados em uma instância de banco de dados do RDS para Db2 para realizar manutenção.
the section called “rdsadmin.grant_role”	Conceder e revogar privilégios	Use o procedimento armazenado `rdsadmin.grant_role` para atribuir um perfil a um perfil, um usuário ou um grupo favorecido em um banco de dados em uma instância de banco de dados do RDS para Db2. Você também pode usar esse procedimento armazenado para conceder à função de favorecido a autorização `DBADM` para atribuir funções.

Procedimento armazenado	Categoria	Descrição
???	Bancos de dados	Use o procedimento armazenado `rdsadmin.list_archive_log_information` para exibir informações sobre logs de arquivamento para um banco de dados em uma instância de banco de dados do RDS para Db2. Essas informações incluem determinados detalhes, como tamanho e data de criação de arquivos de log individuais e armazenamento total usado pelos arquivos de log de arquivamento.
the section called “rdsadmin.list_users”	Conceder e revogar privilégios	Use o procedimento armazenado `rdsadmin.list_users` para exibir uma lista de usuários em uma lista de autorização para um banco de dados em uma instância de banco de dados do RDS para Db2.
???	Bancos de dados	Use o procedimento armazenado `rdsadmin.reactivate_database` para reativar um banco de dados em uma instância de banco de dados do RDS para Db2 depois de fazer alterações na configuração do banco de dados. Em relação a um banco de dados em uma instância de banco de dados autônoma, é possível usar esse procedimento armazenado ou o procedimento armazenado ???.
the section called “rdsadmin.remove_groups”	Conceder e revogar privilégios	Use o procedimento armazenado `rdsadmin.remove_groups` para remover um ou mais grupos de um usuário de um banco de dados em uma instância de banco de dados do RDS para Db2.

Procedimento armazenado	Categoria	Descrição
the section called “rdsadmin.remove_user”	Conceder e revogar privilégios	Use o procedimento armazenado `rdsadmin.remove_user` para remover um usuário de uma lista de autorização de um banco de dados em uma instância de banco de dados do RDS para Db2.
the section called “rdsadmin.rename_tablespace”	Tablespaces	Use o procedimento armazenado `rdsadmin.rename_tablespace` para renomear um espaço de tabela de um banco de dados em uma instância de banco de dados do RDS para Db2.
the section called “rdsadmin.restore_database”	Bancos de dados	Use o procedimento armazenado `rdsadmin.restore_database` para restaurar um banco de dados em uma instância de banco de dados do RDS para Db2 a partir de um bucket do Amazon S3.
the section called “rdsadmin.revoke_role”	Conceder e revogar privilégios	Use o procedimento armazenado `rdsadmin.revoke_role` para revogar um perfil de um perfil, um usuário ou um grupo favorecido em um banco de dados em uma instância de banco de dados do RDS para Db2.
the section called “rdsadmin.rollforward_database”	Bancos de dados	Use o procedimento armazenado `rdsadmin.rollforward_database` para colocar um banco de dados em uma instância de banco de dados do RDS para Db2 on-line e aplicar logs de transações depois de restaurar um banco de dados em uma instância de banco de dados do RDS para Db2 chamando the section called “rdsadmin.restore_database”.

Procedimento armazenado	Categoria	Descrição
the section called “rdsadmin.set_archive_log_retention”	Bancos de dados	Use o procedimento armazenado `rdsadmin.set_archive_log_retention` para configurar por quanto tempo reter arquivos de log de arquivamento de um banco de dados em uma instância de banco de dados do RDS para Db2. Você também pode usar esse procedimento armazenado para desabilitar a retenção de logs de arquivamento.
the section called “rdsadmin.set_configuration”	Bancos de dados	Use o procedimento armazenado `rdsadmin.set_configuration` para configurar determinadas configurações de um banco de dados em uma instância de banco de dados do RDS para Db2.
the section called “rdsadmin.show_archive_log_retention”	Bancos de dados	Use o procedimento armazenado `rdsadmin.show_archive_log_retention` para exibir a configuração atual de retenção do log de arquivamento de um banco de dados em uma instância de banco de dados do RDS para Db2.
the section called “rdsadmin.show_configuration”	Bancos de dados	Use o procedimento armazenado `rdsadmin.show_configuration` para exibir uma ou mais configurações que podem ser modificadas de um banco de dados em uma instância de banco de dados do RDS para Db2.
the section called “rdsadmin.uncatalog_storage_access”	Acesso ao armazenamento	Use o procedimento armazenado `rdsadmin.uncatalog_storage_access` para remover um alias de armazenamento para acessar um bucket do Amazon S3 com arquivos de dados do Db2.

Procedimento armazenado	Categoria	Descrição
the section called “rdsadmin.update_db_param”	Bancos de dados	Use o procedimento armazenado `rdsadmin.update_db_param` para atualizar parâmetros de banco de dados de um banco de dados em uma instância de banco de dados do RDS para Db2.

Tópicos

- Considerações para procedimentos armazenados do Amazon RDS para Db2
- Conceder e revogar privilégios
- Procedimentos armazenados para políticas de auditoria para o RDS para Db2
- Procedimentos armazenados para grupos de buffer do RDS para Db2
- Gerenciar bancos de dados
- Procedimentos armazenados para acesso ao armazenamento do RDS para Db2
- Procedimentos armazenados para espaços de tabela do RDS para Db2

Considerações para procedimentos armazenados do Amazon RDS para Db2

Antes de usar os procedimentos armazenados do sistema Amazon RDS para instâncias de banco de dados do RDS para Db2 que executam o mecanismo Db2, revise as seguintes informações:

- Antes de executar procedimentos armazenados, primeiro você deve se conectar ao banco de dados `rdsadmin` como usuário principal da instância de banco de dados do RDS para Db2. No exemplo a seguir, substitua *`master_username`* e *`master_password`* por suas próprias informações.

  ```
  db2 "connect to rdsadmin user master_username using master_password"
  ```

- Os procedimentos armazenados exibem o parâmetro `ERR_MESSAGE`, que indica se o procedimento armazenado foi executado com êxito ou não e por que não foi executado com êxito.

  Exemplos

  O exemplo a seguir indica que o procedimento armazenado foi executado com êxito.

  ```
  Parameter Name : ERR_MESSAGE
  Parameter Value : -
  Return Status = 0
  ```

  O exemplo a seguir indica que o procedimento armazenado não foi executado com êxito porque o nome do bucket do Amazon S3 usado no procedimento armazenado não era válido.

  ```
  Parameter Name : ERR_MESSAGE
  Parameter Value : Invalid S3 bucket name
  Return Status = -1006
  ```

Para ver as mensagens de erro exibidas ao chamar procedimentos armazenados, consulte the section called “Erros de procedimentos armazenados”.

Para obter informações sobre como conferir o status de um procedimento armazenado, consulte rdsadmin.get_task_status.

Conceder e revogar privilégios

Os procedimentos armazenados a seguir concedem e revogam privilégios para bancos de dados do Amazon RDS para Db2. Para executar esses procedimentos, o usuário principal deve primeiro se conectar ao banco de dados `rdsadmin`.

Tópicos

- rdsadmin.create_role
- rdsadmin.grant_role
- rdsadmin.revoke_role
- rdsadmin.add_user
- rdsadmin.change_password
- rdsadmin.list_users
- rdsadmin.remove_user
- rdsadmin.add_groups
- rdsadmin.remove_groups
- rdsadmin.dbadm_grant
- rdsadmin.dbadm_revoke

rdsadmin.create_role

Cria um perfil.

Sintaxe

```
db2 "call rdsadmin.create_role(
    'database_name',
    'role_name')"
```

Parâmetros

Os seguintes parâmetros são obrigatórios:

database_name

O nome do banco de dados no qual o comando será executado. O tipo de dados é `varchar`.

role_name

O nome do perfil que você deseja criar. O tipo de dados é `varchar`.

Observações de uso

Para ter informações sobre como conferir o status de criação de um perfil, consulte rdsadmin.get_task_status.

Exemplos

O exemplo a seguir cria um perfil denominado `MY_ROLE` para o banco de dados `DB2DB`.

```
db2 "call rdsadmin.create_role(
    'DB2DB',
    'MY_ROLE')"
```

rdsadmin.grant_role

Atribui um perfil a um perfil, um usuário ou um grupo.

Sintaxe

```
db2 "call rdsadmin.grant_role(
    ?,
    'database_name',
    'role_name',
    'grantee',
    'admin_option')"
```

Parâmetros

O seguinte parâmetro de saída é obrigatório:

?

Um marcador de parâmetro que gera o identificador exclusivo da tarefa. Esse parâmetro aceita apenas ?.

Os seguintes parâmetros de entrada são obrigatórios:

database_name

O nome do banco de dados no qual o comando será executado. O tipo de dados é `varchar`.

role_name

O nome do perfil que você deseja criar. O tipo de dados é `varchar`.

grantee

O perfil, o usuário ou o grupo para receber autorização. O tipo de dados é `varchar`. Valores válidos: `ROLE`, `USER`, `GROUP`, `PUBLIC`.

O formato deve ser um valor seguido pelo nome. Separe vários valores e nomes com vírgulas. Exemplo: "`USER` *`user1`*`, `*`user2`*`, GROUP` *`group1`*`, `*`group2`*" Substitua os nomes por suas próprias informações.

O seguinte parâmetro de entrada é opcional:

admin_option

Especifica se o beneficiário de `ROLE` tem autorização `DBADM` para atribuir perfis. O tipo de dados é `char`. O padrão é `N`.

Observações de uso

Para ter informações sobre como conferir o status de atribuição de um perfil, consulte rdsadmin.get_task_status.

Exemplos

O exemplo a seguir atribui um perfil chamado `ROLE_TEST` do banco de dados `TESTDB` ao perfil chamado `role1`, ao usuário chamado `user1` e ao grupo chamado `group1`. O `ROLE_TEST` recebe autorização de administrador para atribuir perfis.

```
db2 "call rdsadmin.grant_role(
    ?,
    'TESTDB',
    'ROLE_TEST',
    'ROLE role1, USER user1, GROUP group1',
    'Y')"
```

O exemplo a seguir atribui um perfil chamado `ROLE_TEST` do banco de dados `TESTDB` a `PUBLIC`. `ROLE_TEST` não recebe autorização de administrador para atribuir perfis.

```
db2 "call rdsadmin.grant_role(
    ?,
    'TESTDB',
    'ROLE_TEST',
    'PUBLIC')"
```

rdsadmin.revoke_role

Revoga um perfil de um perfil, um usuário ou um grupo.

Sintaxe

```
db2 "call rdsadmin.revoke_role(
    ?,
    'database_name',
    'role_name',
    'grantee')"
```

Parâmetros

O seguinte parâmetro de saída é obrigatório:

?

Um marcador de parâmetro que gera o identificador exclusivo da tarefa. Esse parâmetro aceita apenas ?.

Os seguintes parâmetros de entrada são obrigatórios:

database_name

O nome do banco de dados no qual o comando será executado. O tipo de dados é `varchar`.

role_name

O nome do perfil que você deseja revogar. O tipo de dados é `varchar`.

grantee

O perfil, o usuário ou o grupo que perderá a autorização. O tipo de dados é `varchar`. Valores válidos: `ROLE`, `USER`, `GROUP`, `PUBLIC`.

O formato deve ser um valor seguido pelo nome. Separe vários valores e nomes com vírgulas. Exemplo: "`USER` *`user1`*`, `*`user2`*`, GROUP` *`group1`*`, `*`group2`*" Substitua os nomes por suas próprias informações.

Observações de uso

Para ter informações sobre como conferir o status de atribuição de um perfil, consulte rdsadmin.get_task_status.

Exemplos

O exemplo a seguir revoga um perfil chamado `ROLE_TEST` para o banco de dados `TESTDB` do perfil chamado `role1`, do usuário chamado `user1` e do grupo chamado `group1`.

```
db2 "call rdsadmin.revoke_role(
    ?,
    'TESTDB',
    'ROLE_TEST',
    'ROLE role1, USER user1, GROUP group1')"
```

O exemplo a seguir revoga um perfil denominado `ROLE_TEST` para o banco de dados `TESTDB` de `PUBLIC`.

```
db2 "call rdsadmin.revoke_role(
    ?,
    'TESTDB',
    'ROLE_TEST',
    'PUBLIC')"
```

rdsadmin.add_user

Adiciona um usuário a uma lista de autorizações.

Sintaxe

```
db2 "call rdsadmin.add_user(
```

```
    'username',
    'password',
    'group_name,group_name')"
```

Parâmetros

Os seguintes parâmetros são obrigatórios:

nome de usuário

O nome de um usuário. O tipo de dados é `varchar`.

password

A senha de um usuário. O tipo de dados é `varchar`.

O seguinte parâmetro é opcional:

nome_grupo

O nome de um grupo ao qual você deseja adicionar o usuário. O tipo de dados é `varchar`. O padrão é uma string vazia ou nula.

Observações de uso

É possível adicionar um usuário a um ou mais grupos separando os nomes dos grupos com vírgulas.

É possível criar um grupo ao criar um usuário ou ao adicionar um grupo a um usuário existente. Não é possível criar um grupo sozinho.

> **Note**
>
> O número máximo de usuários que você pode adicionar chamando `rdsadmin.add_user` é 5 mil.

Para ter informações sobre como conferir o status de adição de um usuário, consulte rdsadmin.get_task_status.

Exemplos

O exemplo a seguir cria um usuário chamado `jorge_souza` e o atribui aos grupos chamados `sales` e `inside_sales`.

```
db2 "call rdsadmin.add_user(
    'jorge_souza',
    '*******',
    'sales,inside_sales')"
```

rdsadmin.change_password

Altera a senha de um usuário.

Sintaxe

```
db2 "call rdsadmin.change_password(
    'username',
    'new_password')"
```

Parâmetros

Os seguintes parâmetros são obrigatórios:

nome de usuário

O nome de um usuário. O tipo de dados é `varchar`.

new_password

Uma nova senha para o usuário. O tipo de dados é `varchar`.

Observações de uso

Para ter informações sobre como verificar o status da alteração de uma senha, consulte rdsadmin.get_task_status.

Exemplos

O exemplo a seguir altera a senha de `jorge_souza`.

```
db2 "call rdsadmin.change_password(
```

```
    'jorge_souza',
    '*******')"
```

rdsadmin.list_users

Lista os usuários em uma lista de autorizações.

Sintaxe

```
db2 "call rdsadmin.list_users()"
```

Observações de uso

Para ter informações sobre como conferir o status da lista de usuários, consulte rdsadmin.get_task_status.

rdsadmin.remove_user

Remove o usuário da lista de autorizações.

Sintaxe

```
db2 "call rdsadmin.remove_user('username')"
```

Parâmetros

O seguinte parâmetro é obrigatório:

nome de usuário

O nome de um usuário. O tipo de dados é `varchar`.

Observações de uso

Para ter informações sobre como conferir o status de remoção de um usuário, consulte rdsadmin.get_task_status.

Exemplos

O exemplo a seguir remove `jorge_souza` da possibilidade de acessar bancos de dados em instâncias de banco de dados do RDS para Db2.

```
db2 "call rdsadmin.remove_user('jorge_souza')"
```

rdsadmin.add_groups

Adiciona grupos a um usuário.

Sintaxe

```
db2 "call rdsadmin.add_groups(
    'username',
    'group_name,group_name')"
```

Parâmetros

Os seguintes parâmetros são obrigatórios:

nome de usuário

O nome de um usuário. O tipo de dados é `varchar`.

nome_grupo

O nome de um grupo ao qual você deseja adicionar o usuário. O tipo de dados é `varchar`. O padrão é uma string vazia.

Observações de uso

É possível adicionar um ou mais grupos a um usuário separando os nomes dos grupos com vírgulas. Para ter informações sobre como conferir o status de adição de grupos, consulte rdsadmin.get_task_status.

Exemplos

O exemplo a seguir adiciona os grupos `direct_sales` e `b2b_sales` ao usuário `jorge_souza`.

```
db2 "call rdsadmin.add_groups(
    'jorge_souza',
    'direct_sales,b2b_sales')"
```

rdsadmin.remove_groups

Remove grupos de um usuário.

Sintaxe

```
db2 "call rdsadmin.remove_groups(
    'username',
    'group_name,group_name')"
```

Parâmetros

Os seguintes parâmetros são obrigatórios:

nome de usuário

O nome de um usuário. O tipo de dados é `varchar`.

nome_grupo

O nome de um grupo do qual você deseja remover o usuário. O tipo de dados é `varchar`.

Observações de uso

É possível remover um ou mais grupos de um usuário separando os nomes dos grupos com vírgulas.

Para ter informações sobre como conferir o status de remoção de grupos, consulte rdsadmin.get_task_status.

Exemplos

O exemplo a seguir remove os grupos `direct_sales` e `b2b_sales` do usuário `jorge_souza`.

```
db2 "call rdsadmin.remove_groups(
    'jorge_souza',
    'direct_sales,b2b_sales')"
```

rdsadmin.dbadm_grant

Concede a autorização `DBADM`, `ACCESSCTRL` ou `DATAACCESS` a um perfil um usuário ou um grupo.

Sintaxe

```
db2 "call rdsadmin.dbadm_grant(
    ?,
```

```
    'database_name',
    'authorization',
    'grantee')"
```

Parâmetros

O seguinte parâmetro de saída é obrigatório:

?

Um marcador de parâmetro que gera o identificador exclusivo da tarefa. Esse parâmetro aceita apenas ?.

Os seguintes parâmetros de entrada são obrigatórios:

database_name

O nome do banco de dados no qual o comando será executado. O tipo de dados é `varchar`.

autorização

O tipo de autorização a ser concedida. O tipo de dados é `varchar`. Valores válidos: `DBADM`, `ACCESSCTRL`, `DATAACCESS`.

Separe vários tipos com vírgulas.

grantee

O perfil, o usuário ou o grupo para receber autorização. O tipo de dados é `varchar`. Valores válidos: `ROLE`, `USER`, `GROUP`.

O formato deve ser um valor seguido pelo nome. Separe vários valores e nomes com vírgulas. Exemplo: “`USER` *`user1`*`,` *`user2`*`,` `GROUP` *`group1`*`,` *`group2`*” Substitua os nomes por suas próprias informações.

Observações de uso

Deve existir o perfil para receber acesso.

Para ter informações sobre como conferir o status de concessão de acesso de administrador de banco de dados, consulte rdsadmin.get_task_status.

Exemplos

O exemplo a seguir concede acesso de administrador ao banco de dados denominado TESTDB para o perfil ROLE_DBA.

```
db2 "call rdsadmin.dbadm_grant(
    ?,
    'TESTDB',
    'DBADM',
    'ROLE ROLE_DBA')"
```

O exemplo a seguir concede acesso de administrador ao banco de dados denominado TESTDB para user1 e group1.

```
db2 "call rdsadmin.dbadm_grant(
    ?,
    'TESTDB',
    'DBADM',
    'USER user1, GROUP group1')"
```

O exemplo a seguir concede acesso de administrador ao banco de dados denominado TESTDB para user1, user2, group1 e group2.

```
db2 "call rdsadmin.dbadm_grant(
    ?,
    'TESTDB',
    'DBADM',
    'USER user1, user2, GROUP group1, group2')"
```

rdsadmin.dbadm_revoke

Revoga a autorização DBADM, ACCESSCTRL ou DATAACCESS de um perfil, um usuário ou um grupo.

Sintaxe

```
db2 "call rdsadmin.dbadm_revoke(
    ?,
    'database_name',
    'authorization',
    'grantee')"
```

Parâmetros

O seguinte parâmetro de saída é obrigatório:

?

O identificador exclusivo da tarefa. Esse parâmetro aceita apenas ?.

Os seguintes parâmetros de entrada são obrigatórios:

database_name

O nome do banco de dados no qual o comando será executado. O tipo de dados é varchar.

autorização

O tipo de autorização a ser revogada. O tipo de dados é varchar. Valores válidos: DBADM, ACCESSCTRL, DATAACCESS.

Separe vários tipos com vírgulas.

grantee

O perfil, o usuário ou o grupo do qual revogar a autorização. O tipo de dados é varchar. Valores válidos: ROLE, USER, GROUP.

O formato deve ser um valor seguido pelo nome. Separe vários valores e nomes com vírgulas. Exemplo: “USER *user1*, *user2*, GROUP *group1*, *group2*” Substitua os nomes por suas próprias informações.

Observações de uso

Para ter informações sobre como conferir o status de revogação de acesso de administrador de banco de dados, consulte rdsadmin.get_task_status.

Exemplos

O exemplo a seguir revoga o acesso de administrador ao banco de dados denominado TESTDB para o perfil ROLE_DBA.

```
db2 "call rdsadmin.dbadm_revoke(
    ?,
```

```
    'TESTDB',
    'DBADM',
    'ROLE ROLE_DBA')"
```

O exemplo a seguir revoga o acesso de administrador ao banco de dados denominado `TESTDB` para `user1` e `group1`.

```
db2 "call rdsadmin.dbadm_revoke(
    ?,
    'TESTDB',
    'DBADM',
    'USER user1, GROUP group1')"
```

O exemplo a seguir revoga o acesso de administrador ao banco de dados denominado `TESTDB` para `user1`, `user2`, `group1` e `group2`.

```
db2 "call rdsadmin.dbadm_revoke(
    ?,
    'TESTDB',
    'DBADM',
    'USER user1, user2, GROUP group1, group2')"
```

Procedimentos armazenados para políticas de auditoria para o RDS para Db2

Os procedimentos integrados armazenados descritos neste tópico gerenciam políticas de auditoria para bancos de dados do Amazon RDS para Db2 que usam o registro em log de auditoria. Para ter mais informações, consulte the section called “Registro em log de auditoria do Db2”. Para executar esses procedimentos, o usuário principal deve primeiro se conectar ao banco de dados `rdsadmin`.

Consulte os seguintes procedimentos integrados armazenados para obter informações sobre sintaxe, parâmetros, notas de uso e exemplos.

Procedimentos armazenados

- rdsadmin.configure_db_audit
- rdsadmin.disable_db_audit

rdsadmin.configure_db_audit

Configura a política de auditoria para o banco de dados do RDS para Db2 especificado por *db_name*. Se a política que você está configurando não existir, ela será criada ao chamar esse procedimento armazenado. Se essa política existir, ela será modificada ao chamar esse procedimento armazenado com os valores dos parâmetros fornecidos por você.

Sintaxe

```
db2 "call rdsadmin.configure_db_audit(
    'db_name',
    'category',
    'category_setting',
    '?')"
```

Parâmetros

Os parâmetros a seguir são obrigatórios.

db_name

O nome do banco de dados do RDS para Db2 para o qual configurar a política de auditoria. O tipo de dados é `varchar`.

category

O nome da categoria para a qual configurar essa política de auditoria. O tipo de dados é `varchar`. Os valores válidos para esse parâmetro são:

- `ALL`: com `ALL`, o Amazon RDS não inclui as categorias `CONTEXT`, `EXECUTE` ou `ERROR`.
- `AUDIT`
- `CHECKING`
- `CONTEXT`
- `ERROR`
- `EXECUTE`: você pode configurar essa categoria com dados ou sem dados. Com dados, significa também registrar em log os valores de dados de entrada fornecidos para quaisquer variáveis do host e marcadores de parâmetros. O padrão é sem dados. Consulte mais informações na descrição do parâmetro *category_setting* e os the section called “Exemplos”.
- `OBJMAINT`
- `SECMAINT`
- `SYSADMIN`
- `VALIDATE`

Consulte mais informações sobre essas categorias na Documentação do IBM Db2.

category_setting

A configuração da categoria de auditoria especificada. O tipo de dados é `varchar`.

A tabela a seguir mostra os valores válidos da configuração de cada categoria.

Categoria	Configurações válidas da categoria
`ALL`	`BOTH|FAILURE|SUCCESS|NONE`
`AUDIT`	
`CHECKING`	
`CONTEXT`	
`OBJMAINT`	

Categoria	Configurações válidas da categoria
SECMAINT SYSADMIN VALIDATE	
ERROR	AUDIT|NORMAL . O padrão é NORMAL.
EXECUTE	BOTH,WITH|BOTH,WITHOUT|FAILURE,WITH| FAILURE,WITHOUT|SUCCESS,WITH|SUCCESS ,WITHOUT|NONE

Observações de uso

Antes de chamar `rdsadmin.configure_db_audit`, verifique se a instância de banco de dados do RDS para Db2 com o banco de dados para o qual você está configurando a política de auditoria está associada a um grupo de opções que tenha a opção `DB2_AUDIT`. Para ter mais informações, consulte the section called “Configurar o registro em log de auditoria do Db2”.

Depois de configurar a política de auditoria, você pode verificar o status da configuração de auditoria do banco de dados seguindo as etapas em Verificar a configuração da auditoria.

Especificar `ALL` para o parâmetro `category` não inclui as categorias `CONTEXT`, `EXECUTE` ou `ERROR`. Para adicionar essas categorias à política de auditoria, chame `rdsadmin.configure_db_audit` separadamente com cada categoria que você deseja adicionar. Para ter mais informações, consulte the section called “Exemplos”.

Exemplos

Os exemplos a seguir criam ou modificam a política de auditoria para um banco de dados chamado `TESTDB`. Nos exemplos de 1 a 5, se a categoria `ERROR` não tiver sido configurada anteriormente, ela será definida como `NORMAL` (o padrão). Para alterar essa configuração para `AUDIT`, siga Example 6: Specifying the ERROR category.

Exemplo 1: especificar a categoria **`ALL`**

```
db2 "call rdsadmin.configure_db_audit('TESTDB', 'ALL', 'BOTH', ?)"
```

No exemplo, a chamada configura as categorias AUDIT, CHECKING, OBJMAINT, SECMAINT, SYSADMIN e VALIDATE na política de auditoria. Especificar BOTH significa que tanto os eventos bem-sucedidos quanto os com falha serão auditados para cada uma dessas categorias.

Exemplo 2: especificar a categoria **EXECUTE** com dados

```
db2 "call rdsadmin.configure_db_audit('TESTDB', 'EXECUTE', 'SUCCESS,WITH', ?)"
```

No exemplo, a chamada configura a categoria EXECUTE na política de auditoria. Especificar SUCCESS,WITH significa que os logs dessa categoria incluirão somente eventos bem-sucedidos, bem como os valores de dados de entrada fornecidos para variáveis do host e marcadores de parâmetros.

Exemplo 3: especificar a categoria **EXECUTE** sem dados

```
db2 "call rdsadmin.configure_db_audit('TESTDB', 'EXECUTE', 'FAILURE,WITHOUT', ?)"
```

No exemplo, a chamada configura a categoria EXECUTE na política de auditoria. Especificar FAILURE,WITHOUT significa que os logs dessa categoria incluirão somente eventos com falha e não incluirão valores de dados de entrada fornecidos para variáveis do host e marcadores de parâmetros.

Exemplo 4: especificar a categoria **EXECUTE** sem eventos de status

```
db2 "call rdsadmin.configure_db_audit('TESTDB', 'EXECUTE', 'NONE', ?)"
```

No exemplo, a chamada configura a categoria EXECUTE na política de auditoria. Especificar NONE significa que nenhum evento nessa categoria será auditado.

Exemplo 5: especificar a categoria **OBJMAINT**

```
db2 "call rdsadmin.configure_db_audit('TESTDB', 'OBJMAINT', 'NONE', ?)"
```

No exemplo, a chamada configura a categoria OBJMAINT na política de auditoria. Especificar NONE significa que nenhum evento nessa categoria será auditado.

Exemplo 6: especificar a categoria **ERROR**

```
db2 "call rdsadmin.configure_db_audit('TESTDB', 'ERROR', 'AUDIT', ?)"
```

No exemplo, a chamada configura a categoria `ERROR` na política de auditoria. Especificar `AUDIT` significa que todos os erros, incluindo erros que ocorrem no próprio registro em log de auditoria, são capturados nos logs. O tipo de erro padrão é `NORMAL`. Com `NORMAL`, os erros gerados pela auditoria são ignorados e somente os `SQLCODE` s dos erros associados à operação que está sendo executada são capturados.

rdsadmin.disable_db_audit

Interrompe o registro em log de auditoria do banco de dados do RDS para Db2 especificado por *db_name* e remove a política de auditoria configurada para ele.

> **Note**
>
> Esse procedimento armazenado remove somente as políticas de auditoria que foram configuradas usando a chamada de the section called “rdsadmin.configure_db_audit”.

Sintaxe

```
db2 "call rdsadmin.disable_db_audit('db_name', ?)"
```

Parâmetros

Os parâmetros a seguir são obrigatórios.

db_name

O nome do banco de dados do RDS para Db2 para o qual desabilitar o registro em log de auditoria. O tipo de dados é `varchar`.

Observações de uso

Chamar `rdsadmin.disable_db_audit` não desabilita o registro em log de auditoria da instância de banco de dados do RDS para Db2. Para desabilitar o registro em log de auditoria no nível da instância de banco de dados, remova o grupo de opções da instância de banco de dados. Para ter mais informações, consulte Desabilitar o registro em log de auditoria do Db2.

Exemplos

O exemplo a seguir desabilita o registro em log de auditoria para um banco de dados chamado `TESTDB`.

```
db2 "call rdsadmin.disable_db_audit('TESTDB', ?)"
```

Procedimentos armazenados para grupos de buffer do RDS para Db2

Os procedimentos armazenados integrados descritos neste tópico gerenciam pools de buffer para bancos de dados do Amazon RDS para Db2. Para executar esses procedimentos, o usuário principal deve primeiro se conectar ao banco de dados `rdsadmin`.

Esses procedimentos armazenados são usados em diversas tarefas. Essa lista não é exaustiva.

- Tarefas comuns para grupos de buffer
- Gerar relatórios de performance
- Cópia de metadados do banco de dados com db2look
- Criação de um banco de dados de repositório para IBM Db2 Data Management Console

Consulte os seguintes procedimentos integrados armazenados para obter informações sobre sintaxe, parâmetros, notas de uso e exemplos.

Procedimentos armazenados

- rdsadmin.create_bufferpool
- rdsadmin.alter_bufferpool
- rdsadmin.drop_bufferpool

rdsadmin.create_bufferpool

Cria um grupo de buffers.

Sintaxe

```
db2 "call rdsadmin.create_bufferpool(
    'database_name',
    'buffer_pool_name',
    buffer_pool_size,
    'immediate',
    'automatic',
    page_size,
    number_block_pages,
    block_size)"
```

Parâmetros

Os seguintes parâmetros são obrigatórios:

database_name

O nome do banco de dados no qual executar o comando. O tipo de dados é `varchar`.

buffer_pool_name

O nome do grupo de buffers a ser criado. O tipo de dados é `varchar`.

Os seguintes parâmetros são opcionais:

buffer_pool_size

O tamanho do grupo de buffers em número de páginas. O tipo de dados é `integer`. O padrão é `-1`.

immediate

Especifica se o comando é executado imediatamente. O tipo de dados é `char`. O padrão é `Y`.

automatic

Especifica se o grupo de buffers deve ser definido como automático. O tipo de dados é `char`. O padrão é `Y`.

page_size

O tamanho da página do grupo de buffers. O tipo de dados é `integer`. Valores válidos: `4096`, `8192`, `16384`, `32768`. O padrão é `8192`.

number_block_pages

O tamanho de páginas de blocos nos grupos de buffers. O tipo de dados é `integer`. O padrão é `0`.

block_size

O tamanho do bloco para as páginas de blocos. O tipo de dados é `integer`. Valores válidos: `2` a `256`. O padrão é `32`.

Observações de uso

Para ter informações sobre como conferir o status de criação de um grupo de buffers, consulte rdsadmin.get_task_status.

Exemplos

Exemplo 1: criação de um pool de buffer com parâmetros padrão

O exemplo a seguir cria um grupo de buffers chamado BP8 para um banco de dados chamado TESTDB com parâmetros padrão, de forma que o grupo de buffers use um tamanho de página de 8 KB.

```
db2 "call rdsadmin.create_bufferpool(
    'TESTDB',
    'BP8')"
```

Exemplo 2: criação de um pool de buffer para execução imediata com alocação automática

O exemplo a seguir cria um grupo de buffers chamado BP16 para um banco de dados chamado TESTDB que usa um tamanho de página de 16 KB com uma contagem inicial de páginas de 1.000 e é definido como automático. O Db2 executa o comando imediatamente. Se você usar uma contagem inicial de páginas de -1, o Db2 usará a alocação automática de páginas.

```
db2 "call rdsadmin.create_bufferpool(
    'TESTDB',
    'BP16',
    1000,
    'Y',
    'Y',
    16384)"
```

Exemplo 3: criação de um pool de buffer para execução imediata usando páginas em bloco

O exemplo a seguir cria um grupo de buffers chamado BP16 para um banco de dados chamado TESTDB. Esse grupo de buffer tem um tamanho de página de 16 KB com uma contagem inicial de páginas de 10.000. O Db2 executa o comando imediatamente usando 500 páginas de bloco com um tamanho de bloco de 512.

```
db2 "call rdsadmin.create_bufferpool(
    'TESTDB',
    'BP16',
```

```
    10000,
    'Y',
    'Y',
    16384,
    500,
    512)"
```

rdsadmin.alter_bufferpool

Altera um grupo de buffers.

Sintaxe

```
db2 "call rdsadmin.alter_bufferpool(
    'database_name',
    'buffer_pool_name',
    buffer_pool_size,
    'immediate',
    'automatic',
    change_number_blocks,
    number_block_pages,
    block_size)"
```

Parâmetros

Os seguintes parâmetros são obrigatórios:

database_name

O nome do banco de dados no qual executar o comando. O tipo de dados é `varchar`.

buffer_pool_name

O nome do grupo de buffers a ser alterado. O tipo de dados é `varchar`.

buffer_pool_size

O tamanho do grupo de buffers em número de páginas. O tipo de dados é `integer`.

Os seguintes parâmetros são opcionais:

immediate

Especifica se o comando é executado imediatamente. O tipo de dados é `char`. O padrão é `Y`.

automatic

Especifica se o grupo de buffers deve ser definido como automático. O tipo de dados é `char`. O padrão é `N`.

change_number_blocks

Especifica se há uma alteração no número de páginas de blocos no grupo de buffers. O tipo de dados é `char`. O padrão é `N`.

number_block_pages

O tamanho de páginas de blocos nos grupos de buffers. O tipo de dados é `integer`. O padrão é `0`.

block_size

O tamanho do bloco para as páginas de blocos. O tipo de dados é `integer`. Valores válidos: `2` a `256`. O padrão é `32`.

Observações de uso

Para ter informações sobre como conferir o status de alteração de um grupo de buffers, consulte rdsadmin.get_task_status.

Exemplos

O exemplo a seguir altera um grupo de buffers chamado `BP16` para um banco de dados chamado `TESTDB` para não automático e altera o tamanho para 10.000 páginas. O Db2 executa o comando imediatamente.

```
db2 "call rdsadmin.alter_bufferpool(
    'TESTDB',
    'BP16',
    10000,
    'Y',
    'N')"
```

rdsadmin.drop_bufferpool

Descarta um grupo de buffers.

Sintaxe

```
db2 "call rdsadmin.drop_bufferpool(
    'database_name',
    'buffer_pool_name'"
```

Parâmetros

Os seguintes parâmetros são obrigatórios:

database_name

O nome do banco de dados ao qual o grupo de buffers pertence. O tipo de dados é `varchar`.

buffer_pool_name

O nome do grupo de buffers a ser descartado. O tipo de dados é `varchar`.

Observações de uso

Para ter informações sobre como conferir o status de descarte de um grupo de buffers, consulte rdsadmin.get_task_status.

Exemplos

O exemplo a seguir descarta um grupo de buffers denominado `BP16` para um banco de dados chamado `TESTDB`.

```
db2 "call rdsadmin.drop_bufferpool(
    'TESTDB',
    'BP16')"
```

Gerenciar bancos de dados

Os procedimentos armazenados a seguir gerenciam bancos de dados do Amazon RDS para Db2. Para executar esses procedimentos, o usuário principal deve primeiro se conectar ao banco de dados `rdsadmin`.

Tópicos

- rdsadmin.create_database
- rdsadmin.drop_database
- rdsadmin.update_db_param
- rdsadmin.set_configuration
- rdsadmin.show_configuration
- rdsadmin.restore_database
- rdsadmin.rollforward_database
- rdsadmin.complete_rollforward
- rdsadmin.db2pd_command
- rdsadmin.force_application
- rdsadmin.set_archive_log_retention
- rdsadmin.show_archive_log_retention

rdsadmin.create_database

Cria um banco de dados.

Sintaxe

```
db2 "call rdsadmin.create_database('database_name')"
```

Parâmetros

> **Note**
>
> Esse procedimento armazenado não valida a combinação dos parâmetros obrigatórios. Ao chamar rdsadmin.get_task_status, a função definida pelo usuário pode exibir um erro devido a uma combinação inválida de `database_codeset`, `database_territory` e

`database_collation`. Para ter mais informações, consulte Choosing the code page, territory, and collation for your database na documentação do IBM Db2.

O seguinte parâmetro é obrigatório:

`database_name`

O nome do banco de dados a ser criado. O tipo de dados é `varchar`.

Os seguintes parâmetros são opcionais:

`database_page_size`

O tamanho padrão da página do banco de dados. Valores válidos: `4096`, `8192`, `16384`, `32768`. O tipo de dados é `integer`. O padrão é `8192`.

> **Important**
>
> O Amazon RDS é compatível com a atomicidade de gravação para páginas de 4 KiB, 8 KiB e 16 KiB. Por outro lado, páginas de 32 KiB correm o risco de interromper gravações ou de dados parciais serem gravados na mesa. Se estiver usando páginas de 32 KiB, recomendamos habilitar a recuperação para um ponto no tempo e os backups automatizados. Caso contrário, você corre o risco de não conseguir se recuperar de páginas interrompidas. Para ter mais informações, consulte the section called “Introdução aos backups” e the section called “Recuperação para um ponto no tempo”.

`database_code_set`

O conjunto de códigos do banco de dados. O tipo de dados é `varchar`. O padrão é `UTF-8`.

`database_territory`

O código de duas letras do país para o banco de dados. O tipo de dados é `varchar`. O padrão é `US`.

`database_collation`

A sequência de agrupamentos determina como as strings de caracteres armazenadas no banco de dados são classificadas e comparadas. O tipo de dados é `varchar`.

Valores válidos:

- `COMPATIBILITY`: uma sequência de agrupamento do IBM Db2 Versão 2.
- `EBCDIC_819_037`: página de código ISO Latin, agrupamento; CCSID 037 (inglês dos EUA EBCDIC).
- `EBCDIC_819_500`: página de código ISO Latin, agrupamento; CCSID 500 (EBCDIC internacional).
- `EBCDIC_850_037`: página de código ASCII Latin, agrupamento; CCSID 037 (inglês dos EUA EBCDIC).
- `EBCDIC_850_500`: página de código ASCII Latin, agrupamento; CCSID 500 (EBCDIC internacional).
- `EBCDIC_932_5026`: página de código ASCII Japanese, agrupamento; CCSID 037 (inglês dos EUA EBCDIC).
- `EBCDIC_932_5035`: página de código ASCII Japanese, agrupamento; CCSID 500 (EBCDIC internacional).
- `EBCDIC_1252_037`: página de código Windows Latin, agrupamento; CCSID 037 (inglês dos EUA EBCDIC).
- `EBCDIC_1252_500`: página de código Windows Latin, agrupamento; CCSID 500 (EBCDIC internacional).
- `IDENTITY`: agrupamento padrão. As strings são comparadas byte por byte.
- `IDENTITY_16BIT`: o esquema de codificação de compatibilidade para UTF-16: sequência de agrupamento de 8 bits (CESU-8). Para ter mais informações, consulte o Unicode Technical Report #26 do site do Unicode Consortium.
- `NLSCHAR`: somente para uso com a página de código Thai (CP874).
- `SYSTEM`: se você usar `SYSTEM`, o banco de dados usará a sequência de agrupamento automaticamente para `database_codeset` e `database_territory`.

O padrão é `IDENTITY`.

Além disso, o RDS para Db2 é compatível com os seguintes grupos de agrupamentos: `language-aware-collation` e `locale-sensitive-collation`. Para ter mais informações, consulte Choosing a collation for a Unicode database na documentação do IBM Db2.

database_autoconfigure_str

A sintaxe do comando AUTOCONFIGURE, por exemplo, 'AUTOCONFIGURE APPLY DB'. O tipo de dados é varchar. O padrão é uma string vazia ou nula.

Para ter mais informações, consulte AUTOCONFIGURE command na documentação do IBM Db2.

Observações de uso

Será possível criar um banco de dados chamando rdsadmin.create_database se não tiver especificado o nome do banco de dados ao criar a instância de banco de dados do RDS para Db2 usando o console do Amazon RDS ou a AWS CLI. Para ter mais informações, consulte Criar uma instância de banco de dados.

Considerações especiais:

- O comando CREATE DATABASE enviado à instância do Db2 usa a opção RESTRICTIVE.
- O RDS para Db2 usa somente AUTOMATIC STORAGE.
- O RDS para Db2 usa os valores padrão para NUMSEGS e DFT_EXTENT_SZ.
- O RDS para Db2 usa criptografia de armazenamento e não comporta a criptografia de banco de dados.

Para ter mais informações sobre essas considerações, consulte CREATE DATABASE command na documentação do IBM Db2.

Antes de chamar rdsadmin.create_database, é necessário se conectar ao banco de dados rdsadmin. No seguinte exemplo, substitua *master_username* e *master_password* pelas informações da instância de banco de dados do RDS para Db2:

```
db2 connect to rdsadmin user master_username using master_password
```

Para ter informações sobre como conferir o status de criação de um banco de dados, consulte rdsadmin.get_task_status.

Exemplos

O seguinte exemplo cria um banco de dados chamado `TESTJP` com uma combinação correta dos parâmetros *`database_code_set`*, *`database_territory`* e *`database_collation`* para o Japão:

```
db2 "call rdsadmin.create_database('TESTJP', 4096, 'IBM-437', 'JP', 'SYSTEM')"
```

rdsadmin.drop_database

Remove um banco de dados.

Sintaxe

```
db2 "call rdsadmin.drop_database('database_name')"
```

Parâmetros

O seguinte parâmetro é obrigatório:

`database_name`

O nome do banco de dados a ser descartado. O tipo de dados é `varchar`.

Observações de uso

Será possível descartar um banco de dados chamando `rdsadmin.drop_database` somente se as seguintes condições forem atendidas:

- Você não especificou o nome do banco de dados ao criar a instância de banco de dados do RDS para Db2 usando o console do Amazon RDS ou a AWS CLI. Para ter mais informações, consulte Criar uma instância de banco de dados.
- Você criou o banco de dados chamando o procedimento armazenado the section called “rdsadmin.create_database”.
- Você restaurou o banco de dados por meio de uma imagem off-line ou de backup chamando o procedimento armazenado the section called “rdsadmin.restore_database”.

Antes de chamar `rdsadmin.drop_database`, é necessário se conectar ao banco de dados `rdsadmin`. No seguinte exemplo, substitua *`master_username`* e *`master_password`* pelas informações da instância de banco de dados do RDS para Db2:

```
db2 connect to rdsadmin user master_username using master_password
```

Para ter informações sobre como conferir o status de descarte de um banco de dados, consulte rdsadmin.get_task_status.

Exemplos

O seguinte exemplo descarta um banco de dados chamado TESTDB:

```
db2 "call rdsadmin.drop_database('TESTDB')"
```

Exemplos de resposta

Se você transmitir um nome de banco de dados incorreto, o procedimento armazenado exibirá o seguinte exemplo de resposta:

```
SQL0438N Application raised error or warning with diagnostic text: "Cannot
drop database. Database with provided name does not exist". SQLSTATE=99993
```

Se você tiver criado o banco de dados usando o console do Amazon RDS ou a AWS CLI, o procedimento armazenado exibirá o seguinte exemplo de resposta:

```
Return Status = 0
```

Depois de receber `Return Status = 0`, chame o procedimento armazenado the section called "rdsadmin.get_task_status". Uma resposta semelhante ao seguinte exemplo explica o status:

```
1 ERROR DROP_DATABASE RDSDB 2023-10-10-16.33.03.744122 2023-10-10-16.33.30.143797 -
 2023-10-10-16.33.30.098857 Task execution has started.
2023-10-10-16.33.30.143797 Caught exception during executing task id 1, Aborting task.
Reason Dropping database created via rds CreateDBInstance api is not allowed.
Only database created using rdsadmin.create_database can be dropped
```

rdsadmin.update_db_param

Atualiza parâmetros do banco de dados.

Sintaxe

```
db2 "call rdsadmin.update_db_param(
    'database_name',
    'parameter_to_modify',
    'changed_value')"
```

Parâmetros

Os seguintes parâmetros são obrigatórios:

database_name

O nome do banco de dados para o qual executar a tarefa. O tipo de dados é `varchar`.

parameter_to_modify

O nome do parâmetro a ser modificado. O tipo de dados é `varchar`. Para ter mais informações, consulte Parâmetros do Amazon RDS para Db2.

changed_value

O valor para o qual o valor de parâmetro será alterado. O tipo de dados é `varchar`.

Observações de uso

Para ter informações sobre como conferir o status de atualização dos parâmetros do banco de dados, consulte rdsadmin.get_task_status.

Exemplos

O seguinte exemplo atualiza o parâmetro `archretrydelay` para `100` para um banco de dados chamado `TESTDB`:

```
db2 "call rdsadmin.update_db_param(
    'TESTDB',
    'archretrydelay',
    '100')"
```

O seguinte exemplo adia a validação de objetos criados em um banco de dados chamado `TESTDB` para evitar a verificação de dependências:

```
db2 "call rdsadmin.update_db_param(
    'TESTDB',
    'auto_reval',
    'deferred_force')"
```

Exemplos de resposta

Se você tentar modificar um parâmetro de configuração de banco de dados não compatível ou modificável, o procedimento armazenado exibirá o seguinte exemplo de resposta:

```
SQL0438N Application raised error or warning with diagnostic text: "Parameter
is either not supported or not modifiable to customers". SQLSTATE=99993
```

rdsadmin.set_configuration

Define configurações específicas do banco de dados.

Sintaxe

```
db2 "call rdsadmin.set_configuration(
    'name',
    'value')"
```

Parâmetros

Os seguintes parâmetros são obrigatórios:

name

O nome do ajuste de configurações. O tipo de dados é `varchar`.

value

O valor da configuração. O tipo de dados é `varchar`.

Observações de uso

A tabela a seguir mostra as configurações que podem ser controladas com `rdsadmin.set_configuration`.

Nome	Descrição
RESTORE_DATABASE_NUM_BUFFERS	O número de buffers a serem criados durante uma operação de restauração. Esse valor deve ser menor que o tamanho total de memória da classe de instância de banco de dados. Se essa configuração não estiver definida, o Db2 determinará o valor a ser usado durante a operação de restauração. Para obter mais informações, consulte a documentação do IBM Db2.
RESTORE_DATABASE_PARALLELISM	O número de manipuladores de buffer a serem criados durante uma operação de restauração. Esse valor deve ser menor que o dobro do número de vCPUs da instância de banco de dados. Se essa configuração não estiver definida, o Db2 determinará o valor a ser usado durante a operação de restauração. Para obter mais informações, consulte a documentação do IBM Db2.

Exemplos

O exemplo a seguir define a configuração RESTORE_DATABASE_PARALLELISM como 8.

```
db2 "call rdsadmin.set_configuration(
    'RESTORE_DATABASE_PARALLELISM',
    '8')"
```

O exemplo a seguir define a configuração RESTORE_DATABASE_NUM_BUFFERS como 150.

```
db2 "call rdsadmin.set_configuration(
    'RESTORE_DATABASE_NUM_BUFFERS',
    '150')"
```

rdsadmin.show_configuration

Exibe as configurações atuais que você pode definir usando o procedimento armazenado rdsadmin.set_configuration.

Sintaxe

```
db2 "call rdsadmin.show_configuration(
```

```
    'name')"
```

Parâmetros

O seguinte parâmetro é opcional:

name

O nome da configuração sobre a qual exibir informações. O tipo de dados é `varchar`.

Os seguintes nomes de configuração são válidos:

- RESTORE_DATABASE_NUM_BUFFERS: o número de buffers a serem criados durante uma operação de restauração.
- RESTORE_DATABASE_PARALLELISM: o número de manipuladores de buffer a serem criados durante uma operação de restauração.

Observações de uso

Se você não especificar o nome de uma configuração, `rdsadmin.show_configuration` exibirá as informações de todas as configurações que você pode definir usando o procedimento armazenado `rdsadmin.set_configuration`.

Exemplos

O exemplo a seguir exibe informações sobre a configuração `RESTORE_DATABASE_PARALLELISM` atual.

```
db2 "call rdsadmin.show_configuration(
    'RESTORE_DATABASE_PARALLELISM')"
```

rdsadmin.restore_database

Restaura um banco de dados.

Sintaxe

```
db2 "call rdsadmin.restore_database(
    ?,
    'database_name',
    's3_bucket_name',
```

```
    's3_prefix',
    restore_timestamp,
    'backup_type')"
```

Parâmetros

O seguinte parâmetro de saída é obrigatório:

?

Um marcador de parâmetro que gera uma mensagem de erro. Esse parâmetro aceita apenas ?.

Os seguintes parâmetros de entrada são obrigatórios:

database_name

O nome do banco de dados a ser restaurado. Esse nome deve corresponder ao nome do banco de dados na imagem de backup. O tipo de dados é `varchar`.

s3_bucket_name

O nome do bucket do Amazon S3 onde reside o backup. O tipo de dados é `varchar`.

s3_prefix

O prefixo a ser usado para correspondência de arquivos durante o download. O tipo de dados é `varchar`.

Se esse parâmetro estiver vazio, todos os arquivos no bucket do Amazon S3 serão baixados. Veja a seguir um exemplo de prefixo:

```
backupfolder/SAMPLE.0.rdsdb.DBPART000.20230615010101
```

restore_timestamp

O carimbo de data e hora da imagem de backup do banco de dados. O tipo de dados é `varchar`.

O carimbo de data e hora está incluído no nome do arquivo de backup. Por exemplo, `20230615010101` é o carimbo de data e hora do nome do arquivo. `SAMPLE.0.rdsdb.DBPART000.20230615010101.001`.

backup_type

O tipo de backup. O tipo de dados é `varchar`. Valores válidos: `OFFLINE`, `ONLINE`.

Use `ONLINE` para migrações com tempo de inatividade quase zero. Para ter mais informações, consulte Migrar do Linux para Linux com quase zero tempo de inatividade para o Amazon RDS para Db2.

Observações de uso

Será possível restaurar um banco de dados chamando `rdsadmin.restore_database` se não especificar o nome do banco de dados ao criar a instância de banco de dados do RDS para Db2 usando o console do Amazon RDS ou a AWS CLI. Para ter mais informações, consulte Criar uma instância de banco de dados.

Antes de restaurar um banco de dados, é necessário provisionar espaço de armazenamento para a instância de banco de dados do RDS para Db2 igual ou maior que a soma do tamanho do backup, e o banco de dados do Db2 original no disco. Ao restaurar o backup, o Amazon RDS extrai o arquivo de backup na instância de banco de dados do RDS para Db2.

Cada arquivo de backup deve ter 5 TB ou menos. Se um arquivo de backup exceder 5 TB, você deverá dividi o arquivo de backup em arquivos menores.

Para restaurar todos os arquivos usando o procedimento armazenado `rdsadmin.restore_database`, não inclua o sufixo do número do arquivo após o carimbo de data e hora nos nomes dos arquivos. Por exemplo, o *`s3_prefix`* `backupfolder/SAMPLE.0.rdsdb.DBPART000.20230615010101` restaura os seguintes arquivos:

```
SAMPLE.0.rdsdb.DBPART000.20230615010101.001
SAMPLE.0.rdsdb.DBPART000.20230615010101.002
SAMPLE.0.rdsdb.DBPART000.20230615010101.003
SAMPLE.0.rdsdb.DBPART000.20230615010101.004
SAMPLE.0.rdsdb.DBPART000.20230615010101.005
```

Para melhorar a performance das operações de restauração do banco de dados, é possível configurar o número de buffers e manipuladores de buffer a serem usados pelo RDS. Para conferir a configuração atual, use the section called “rdsadmin.show_configuration”. Para alterar a configuração, use the section called “rdsadmin.set_configuration”.

Para ter informações sobre como conferir o status de restauração de um banco de dados, consulte rdsadmin.get_task_status.

Para colocar o banco de dados on-line e aplicar logs de transações adicionais após a restauração do banco de dados, consulte rdsadmin.rollforward_database.

Exemplos

O seguinte exemplo restaura um backup off-line com um ou vários arquivos que têm o *s3_prefix* `backupfolder/SAMPLE.0.rdsdb.DBPART000.20230615010101`:

```
db2 "call rdsadmin.restore_database(
    ?,
    'SAMPLE',
    'amzn-s3-demo-bucket',
    'backupfolder/SAMPLE.0.rdsdb.DBPART000.20230615010101',
    20230615010101,
    'OFFLINE')"
```

rdsadmin.rollforward_database

Coloca o banco de dados on-line e aplica logs de transações adicionais depois de restaurar um banco de dados chamando rdsadmin.restore_database.

Sintaxe

```
db2 "call rdsadmin.rollforward_database(
    ?,
    'database_name',
    's3_bucket_name',
    s3_prefix,
    'rollfoward_to_option',
    'complete_rollforward')"
```

Parâmetros

O seguinte parâmetro de saída é obrigatório:

?

Um marcador de parâmetro que gera uma mensagem de erro. Esse parâmetro aceita apenas `?`.

Os seguintes parâmetros de entrada são obrigatórios:

database_name

O nome do banco de dados no qual a operação deve ser realizada. O tipo de dados é `varchar`.

s3_bucket_name

O nome do bucket do Amazon S3 onde reside o backup. O tipo de dados é `varchar`.

s3_prefix

O prefixo a ser usado para correspondência de arquivos durante o download. O tipo de dados é `varchar`.

Se esse parâmetro estiver vazio, todos os arquivos no bucket do S3 serão baixados. O seguinte exemplo é de um prefixo:

```
backupfolder/SAMPLE.0.rdsdb.DBPART000.20230615010101
```

Os seguintes parâmetros de entrada são opcionais:

rollforward_to_option

O ponto até o qual você pretende efetuar roll forward. O tipo de dados é `varchar`. Valores válidos: `END_OF_LOGS`, `END_OF_BACKUP`. O padrão é `END OF LOGS`.

complete_rollforward

Especifica se o processo de efetuar roll forward deve ser concluído. O tipo de dados é `varchar`. O padrão é `TRUE`.

Se `TRUE`, após a conclusão, o banco de dados estará on-line e acessível. Se `FALSE`, o banco de dados permanecerá em um estado `ROLL-FORWARD PENDING`.

Observações de uso

Depois de chamar rdsadmin.restore_database, será necessário chamar `rollforward_database` para aplicar os logs de arquivamento de um bucket do S3. Também é possível usar esse procedimento armazenado para restaurar logs de transações adicionais depois de chamar `rdsadmin.restore_database`.

Se você definir `complete_rollforward` como `FALSE`, o banco de dados estará em um estado `ROLL-FORWARD PENDING` e off-line. Para colocar o banco de dados on-line, é necessário chamar rdsadmin.complete_rollforward.

Para ter informações sobre como conferir o status da realização de roll forward do banco de dados, consulte rdsadmin.get_task_status.

Exemplos

O seguinte exemplo efetua roll forward em um backup on-line do banco de dados com logs de transações e, depois, coloca o banco de dados on-line:

```
db2 "call rdsadmin.rollforward_database(
    ?,
    null,
    null,
    'END_OF_LOGS',
    'TRUE')"
```

O seguinte exemplo avança para um backup on-line do banco de dados sem logs de transações e, depois, coloca o banco de dados on-line:

```
db2 "call rdsadmin.rollforward_database(
    ?,
    'TESTDB',
    'amzn-s3-demo-bucket',
    'logsfolder/,
    'END_OF_BACKUP',
    'TRUE')"
```

O seguinte exemplo efetua roll forward em um backup on-line do banco de dados com logs de transações e, depois, não coloca o banco de dados on-line:

```
db2 "call rdsadmin.rollforward_database(
    ?,
    'TESTDB',
    null,
    'onlinebackup/TESTDB',
    'END_OF_LOGS',
    'FALSE')"
```

O seguinte exemplo efetua roll forward em um backup on-line do banco de dados com logs de transações adicionais e, depois, não coloca o banco de dados on-line.

```
db2 "call rdsadmin.rollforward_database(
    ?,
    'TESTDB',
    'amzn-s3-demo-bucket',
```

```
    'logsfolder/S0000155.LOG',
    'END_OF_LOGS',
    'FALSE')"
```

rdsadmin.complete_rollforward

Coloca o banco de dados on-line de um estado `ROLL-FORWARD PENDING`.

Sintaxe

```
db2 "call rdsadmin.complete_rollforward(
    ?,
    'database_name')"
```

Parâmetros

O seguinte parâmetro de saída é obrigatório:

?

Um marcador de parâmetro que gera uma mensagem de erro. Esse parâmetro aceita apenas `?`.

O seguinte parâmetro de entrada é obrigatório:

`database_name`

O nome do banco de dados que você deseja colocar on-line. O tipo de dados é `varchar`.

Observações de uso

Se você chamou rdsadmin.rollforward_database com `complete_rollforward` definido como `FALSE`, o banco de dados está em um estado `ROLL-FORWARD PENDING` e offline. Para concluir o processo de roll forward e colocar o banco de dados on-line, chame `rdsadmin.complete_rollforward`.

Para ter informações sobre como conferir o status da realização de roll forward, consulte rdsadmin.get_task_status.

Exemplos

O seguinte exemplo coloca o banco de dados TESTDB on-line:

```
db2 "call rdsadmin.complete_rollfoward(
    ?,
    'TESTDB')"
```

rdsadmin.db2pd_command

Coleta informações sobre um banco de dados do RDS para Db2.

Sintaxe

```
db2 "call rdsadmin.db2pd_command('db2pd_cmd')"
```

Parâmetros

O seguinte parâmetro de entrada é obrigatório:

db2pd_cmd

O nome do comando db2pd que você deseja executar. O tipo de dados é `varchar`.

O parâmetro deve começar com um hífen. Para ver uma lista de parâmetros, consulte db2pd - Monitor and troubleshoot Db2 database command na documentação da IBM.

As seguintes opções não são compatíveis:

- `-addnode`
- `-alldatabases`
- `-alldbp`
- `-alldbs`
- `-allmembers`
- `-alm_in_memory`
- `-cfinfo`
- `-cfpool`
- `-command`
- `-dbpartitionnum`
- `-debug`

- `-dump`
- `-everything`
- `-file | -o`
- `-ha`
- `-interactive`
- `-member`
- `-pages`

  > **Note**
  >
  > O `-pages summary` é compatível.

- `-pdcollection`
- `-repeat`
- `-stack`
- `-totalmem`

A subopção `file` não é compatível; por exemplo, `db2pd -db testdb -tcbstats file=tcbstat.out`.

O uso de `stacks` não é compatível; por exemplo, `db2pd -edus interval=5 top=10 stacks`.

Observações de uso

Esse procedimento armazenado reúne informações que podem ajudar no monitoramento e na solução de problemas de bancos de dados do RDS para Db2.

O procedimento armazenado usa o utilitário `db2pd` da IBM para executar vários comandos. O utilitário `db2pd` exige a autorização `SYSADM`, que o usuário principal do RDS para Db2 não tem. No entanto, com o procedimento armazenado do Amazon RDS, o usuário principal pode usar o utilitário para executar vários comandos. Para ter mais informações sobre o utilitário, consulte db2pd - Monitor and troubleshoot Db2 database command na documentação da IBM.

A saída é restrita a no máximo 2 GB.

Para ter informações sobre como conferir o status da coleta de informações sobre o banco de dados, consulte rdsadmin.get_task_status.

Exemplos

O seguinte exemplo exibe o tempo de atividade de uma instância de banco de dados do RDS para Db2:

```
db2 "call rdsadmin.db2pd_command('-')"
```

O seguinte exemplo exibe o tempo de atividade de um banco de dados chamado TESTDB:

```
db2 "call rdsadmin.db2pd_command('-db TESTDB -')"
```

O seguinte exemplo exibe o uso de memória de uma instância de banco de dados do RDS para Db2:

```
db2 "call rdsadmin.db2pd_command('-dbptnmem')"
```

O seguinte exemplo exibe os conjuntos de memória de uma instância de banco de dados do RDS para Db2 e um banco de dados denominado TESTDB:

```
db2 "call rdsadmin.db2pd_command('-inst -db TESTDB -memsets')"
```

rdsadmin.force_application

Força a retirada das aplicações de um banco de dados do RDS para Db2.

Sintaxe

```
db2 "call rdsadmin.force_application(
    ?,
    'applications')"
```

Parâmetros

O seguinte parâmetro de saída é obrigatório:

?

Um marcador de parâmetro que gera uma mensagem de erro. Esse parâmetro aceita apenas ?.

O seguinte parâmetro de entrada é obrigatório:

applications

As aplicações cuja retirada de um banco de dados do RDS para Db2 você deseja forçar. O tipo de dados é `varchar`. Valores válidos: `ALL` ou *application_handle*.

Separe os nomes de várias aplicações com vírgulas. Exemplo: "*application_handle_1*, *application_handle_2*".

Observações de uso

Esse procedimento armazenado força a retirada de todas as aplicações de um banco de dados para que você possa realizar a manutenção.

O procedimento armazenado usa o comando `FORCE APPLICATION` da IBM. O comando `FORCE APPLICATION` exige autorização `SYSADM`, `SYSMAINT` ou `SYSCTRL`, que o usuário principal do RDS para Db2 não tem. No entanto, com o procedimento armazenado do Amazon RDS, o usuário principal pode usar o comando. Para ter mais informações, consulte FORCE APPLICATION command na documentação da IBM.

Para ter informações sobre como conferir o status da retirada forçada de aplicações de um banco de dados, consulte rdsadmin.get_task_status.

Exemplos

O seguinte exemplo força a retirada de todas as aplicações de um banco de dados do RDS para Db2:

```
db2 "call rdsadmin.force_application(
    ?,
    'ALL')"
```

O seguinte exemplo força a retirada de todos os manipuladores de aplicações 9991, 8891 e 1192 de um banco de dados do RDS para Db2:

```
db2 "call rdsadmin.force_application(
    ?,
    '9991, 8891, 1192')"
```

rdsadmin.set_archive_log_retention

Configura o intervalo de tempo (em horas) para retenção dos arquivos de log do banco de dados do RDS para Db2 especificado.

Sintaxe

```
db2 "call rdsadmin.set_archive_log_retention(
    ?,
    'database_name',
    'archive_log_retention_hours')"
```

Parâmetros

O seguinte parâmetro de saída é obrigatório:

?

Um marcador de parâmetro que gera uma mensagem de erro. Esse parâmetro aceita apenas `?`.

Os seguintes parâmetros de entrada são obrigatórios:

database_name

O nome do banco de dados cuja retenção de logs será configurada. O tipo de dados é `varchar`.

archive_log_retention_hours

O número de horas de retenção dos arquivos de log. O tipo de dados é `smallint`. O padrão é `0` e o máximo é 168 (sete dias).

Se o valor for `0`, o Amazon RDS não reterá os arquivos de log.

Observações de uso

Por padrão, o RDS para Db2 retém os logs por cinco minutos. Recomendamos que, se você usar ferramentas de replicação, como o AWS DMSpara captura de dados de alteração (CDC) ou IBM Q Replication, defina a retenção de logs nessas ferramentas como mais de cinco minutos.

É possível visualizar a configuração atual de retenção de logs chamando the section called “rdsadmin.show_archive_log_retention”.

Não é possível definir a configuração de retenção de logs no banco de dados `rdsadmin`.

Exemplos

O exemplo a seguir define o tempo de retenção de logs de um banco de dados chamado `TESTDB` como 24 horas.

```
db2 "call rdsadmin.set_archive_log_retention(
    ?,
    'TESTDB',
    '24')"
```

O exemplo a seguir desativa o tempo de retenção de logs de um banco de dados chamado `TESTDB`.

```
db2 "call rdsadmin.set_archive_log_retention(
    ?,
    'TESTDB',
    '0')"
```

rdsadmin.show_archive_log_retention

Exibe a configuração atual de retenção de logs do banco de dados especificado.

Sintaxe

```
db2 "call rdsadmin.show_archive_log_retention(
    ?,
    'database_name')"
```

Parâmetros

O seguinte parâmetro de saída é obrigatório:

?

Um marcador de parâmetro que gera uma mensagem de erro. Esse parâmetro aceita apenas `?`.

O seguinte parâmetro de entrada é obrigatório:

database_name

O nome do banco de dados cuja configuração de retenção de logs será mostrada. O tipo de dados é `varchar`.

Exemplos

O exemplo a seguir mostra a configuração de tempo de retenção de logs de um banco de dados chamado `TESTDB`.

```
db2 "call rdsadmin.show_archive_log_retention(
    ?
    'TESTDB')"
```

Procedimentos armazenados para acesso ao armazenamento do RDS para Db2

Os procedimentos integrados armazenados descritos neste tópico gerenciam o acesso ao armazenamento para bancos de dados do RDS para Db2 que usam o Amazon S3 para migrar dados. Para ter mais informações, consulte the section called “Migrar com o Amazon S3”.

Consulte os seguintes procedimentos integrados armazenados para obter informações sobre sintaxe, parâmetros, notas de uso e exemplos.

Procedimentos armazenados

- rdsadmin.catalog_storage_access
- rdsadmin.uncatalog_storage_access

rdsadmin.catalog_storage_access

Cataloga um alias de armazenamento para acessar um bucket do Amazon S3 com arquivos de dados do Db2.

Sintaxe

```
db2 "call rdsadmin.catalog_storage_access(
    ?,
    'alias',
    's3_bucket_name',
    'grantee_type',
    'grantee'
    )"
```

Parâmetros

O seguinte parâmetro de saída é obrigatório:

?

Um marcador de parâmetro que gera uma mensagem de erro. O tipo de dados é `varchar`.

Os seguintes parâmetros de entrada são obrigatórios:

alias

O alias para acessar o armazenamento remoto em um bucket do Amazon S3. O tipo de dados é `varchar`.

s3_bucket_name

O nome do bucket do Amazon S3 onde residem os dados. O tipo de dados é `varchar`.

grantee_type

O tipo do favorecido a receber autorização. O tipo de dados é `varchar`. Valores válidos: `USER`, `GROUP`.

grantee

O usuário ou o grupo para receber autorização. O tipo de dados é `varchar`.

Observações de uso

O Amazon RDS inclui o alias catalogado no perfil do IAM que você adicionou à instância de banco de dados do RDS para Db2. Se você remover o perfil do IAM da instância de banco de dados, o Amazon RDS excluirá o alias. Para ter mais informações, consulte the section called “Migrar com o Amazon S3”.

Para ter informações sobre como conferir o status da catalogação do alias, consulte rdsadmin.get_task_status.

Exemplos

O exemplo a seguir registra um alias chamado `SAMPLE`. O usuário `jorge_souza` recebe acesso ao bucket do Amazon S3 chamado `amzn-s3-demo-bucket`.

```
db2 "call rdsadmin.catalog_storage_access(
    ?,
    'SAMPLE',
    'amzn-s3-demo-bucket',
    'USER',
    'jorge_souza')"
```

rdsadmin.uncatalog_storage_access

Remove um alias de acesso ao armazenamento.

Sintaxe

```
db2 "call rdsadmin.uncatalog_storage_access(
    ?,
    'alias')"
```

Parâmetros

O seguinte parâmetro de saída é obrigatório:

?

Um marcador de parâmetro que gera uma mensagem de erro. O tipo de dados é `varchar`.

O seguinte parâmetro de entrada é obrigatório:

alias

O nome do alias de armazenamento a ser removido. O tipo de dados é `varchar`.

Observações de uso

Para ter informações sobre como conferir o status de remoção do alias, consulte rdsadmin.get_task_status.

Exemplos

O exemplo a seguir remove um alias chamado `SAMPLE`. Esse alias não concede mais acesso ao bucket do Amazon S3 ao qual estava associado.

```
db2 "call rdsadmin.uncatalog_storage_access(
    ?,
    'SAMPLE')"
```

Procedimentos armazenados para espaços de tabela do RDS para Db2

Os procedimentos integrados armazenados descritos neste tópico gerenciam tablespaces para bancos de dados do Amazon RDS para Db2. Para executar esses procedimentos, o usuário principal deve primeiro se conectar ao banco de dados `rdsadmin`.

Esses procedimentos armazenados são usados em diversas tarefas. Essa lista não é exaustiva.

- Tarefas comuns para espaços de tabela
- Gerar relatórios de performance
- Cópia de metadados do banco de dados com db2look
- Criação de um banco de dados de repositório para IBM Db2 Data Management Console

Consulte os seguintes procedimentos integrados armazenados para obter informações sobre sintaxe, parâmetros, notas de uso e exemplos.

Procedimentos armazenados

- rdsadmin.create_tablespace
- rdsadmin.alter_tablespace
- rdsadmin.rename_tablespace
- rdsadmin.drop_tablespace

rdsadmin.create_tablespace

Cria um espaço de tabela.

Sintaxe

```
db2 "call rdsadmin.create_tablespace(
    'database_name',
    'tablespace_name',
    'buffer_pool_name',
    tablespace_page_size,
    tablespace_initial_size,
    tablespace_increase_size,
    'tablespace_type',
    'tablespace_prefetch_size')"
```

Parâmetros

Os seguintes parâmetros são obrigatórios:

database_name

O nome do banco de dados no qual criar o espaço de tabela. O tipo de dados é `varchar`.

tablespace_name

O nome do espaço de tabela a ser criado. O tipo de dados é `varchar`.

O nome do espaço de tabela tem as seguintes restrições:

- Não pode ser igual ao nome de um espaço de tabela existente nesse banco de dados.
- Só pode conter os caracteres `_$#@a-zA-Z0-9`.
- Não pode começar com _ ou $.
- Não pode começar com SYS.

Os seguintes parâmetros são opcionais:

buffer_pool_name

O nome do grupo de buffer para atribuir ao espaço de tabela. O tipo de dados é `varchar`. O padrão é uma string vazia.

> **Important**
>
> É necessário já ter um grupo de buffers do mesmo tamanho de página para associar ao espaço de tabela.

tablespace_page_size

O tamanho da página do espaço de tabela em bytes. O tipo de dados é `integer`. Valores válidos: `4096`, `8192`, `16384`, `32768`. O padrão é o tamanho da página usado quando você criou o banco de dados chamando rdsadmin.create_database.

> **Important**
>
> O Amazon RDS é compatível com a atomicidade de gravação para páginas de 4 KiB, 8 KiB e 16 KiB. Por outro lado, páginas de 32 KiB correm o risco de interromper gravações

ou de dados parciais serem gravados na mesa. Se estiver usando páginas de 32 KiB, recomendamos habilitar a recuperação para um ponto no tempo e os backups automatizados. Caso contrário, você corre o risco de não conseguir se recuperar de páginas interrompidas. Para obter mais informações, consulte the section called "Introdução aos backups" e the section called "Recuperação para um ponto no tempo".

tablespace_initial_size

O tamanho inicial do espaço de tabela em kilobytes (KB). O tipo de dados é `integer`. Valores válidos: `48` ou superior. O padrão é nulo.

Se você não definir um valor, o Db2 definirá um valor apropriado para você.

> **Note**
>
> Esse parâmetro não é aplicável a espaços de tabela temporários, pois o sistema gerencia esses espaços.

tablespace_increase_size

A porcentagem pela qual aumentar o espaço de tabela quando ele ficar cheio. O tipo de dados é `integer`. Valores válidos: `1` a `100`. O padrão é nulo.

Se você não definir um valor, o Db2 definirá um valor apropriado para você.

> **Note**
>
> Esse parâmetro não é aplicável a espaços de tabela temporários, pois o sistema gerencia esses espaços.

tablespace_type

O tipo do espaço de tabela. O tipo de dados é `char`. Valores válidos: `U` (para dados do usuário), `T` (para dados temporários do usuário) ou `S` (para dados temporários do sistema). O padrão é `U`.

tablespace_prefetch_size

O tamanho da página de pré-busca do espaço de tabela. O tipo de dados é `char`. Valores válidos: `AUTOMATIC` (sem distinção entre maiúsculas e minúsculas) ou números inteiros positivos diferentes de zero que sejam menores ou iguais a 32767.

Observações de uso

O RDS para Db2 sempre cria um grande banco de dados para dados.

Para ter informações sobre como conferir o status de criação de um espaço de tabela, consulte rdsadmin.get_task_status.

Exemplos

Exemplo 1: criação de um tablespace e atribuição de um pool de buffers

O exemplo a seguir cria um espaço de tabela chamado `SP8` e atribui um grupo de buffers chamado `BP8` para um banco de dados chamado `TESTDB`. O espaço de tabela tem um tamanho de página de espaço de tabela inicial de 4.096 bytes, um espaço de tabela inicial de 1.000 KB e um aumento de tamanho de tabela definido como 50%.

```
db2 "call rdsadmin.create_tablespace(
    'TESTDB',
    'SP8',
    'BP8',
    4096,
    1000,
    50)"
```

Exemplo 2: criação de um tablespace temporário e atribuição de um pool de buffers

O exemplo a seguir cria um espaço de tabela temporário chamado `SP8`. Ele atribui um grupo de buffer chamado `BP8` que tem 8 KiB de tamanho para um banco de dados chamado `TESTDB`.

```
db2 "call rdsadmin.create_tablespace(
    'TESTDB',
    'SP8',
    'BP8',
    8192,
    NULL,
```

```
    NULL,
    'T')"
```

Exemplo 3: criar um espaço de tabela e atribuir um tamanho de página de pré-busca

O exemplo a seguir cria um espaço de tabela chamado SP8 para um banco de dados chamado TESTDB. O espaço de tabela tem um tamanho inicial de aumento de 50 e um tamanho de página de pré-busca de 800.

```
db2 "call rdsadmin.create_tablespace(
    'TESTDB',
    'SP8',
    NULL,
    NULL,
    NULL,
    50,
    NULL,
    '800')"
```

rdsadmin.alter_tablespace

Altera um espaço de tabela.

Sintaxe

```
db2 "call rdsadmin.alter_tablespace(
    'database_name',
    'tablespace_name',
    'buffer_pool_name',
    tablespace_increase_size,
    'max_size',
    'reduce_max',
    'reduce_stop',
    'reduce_value',
    'lower_high_water',
    'lower_high_water_stop',
    'switch_online',
    'tablespace_prefetch_size')"
```

Parâmetros

Os seguintes parâmetros são obrigatórios:

database_name

O nome do banco de dados que usa o espaço de tabela. O tipo de dados é `varchar`.

tablespace_name

O nome do espaço de tabela a ser alterado. O tipo de dados é `varchar`.

Os seguintes parâmetros são opcionais:

buffer_pool_name

O nome do grupo de buffer para atribuir ao espaço de tabela. O tipo de dados é `varchar`. O padrão é uma string vazia.

> **Important**
>
> É necessário já ter um grupo de buffers do mesmo tamanho de página para associar ao espaço de tabela.

tablespace_increase_size

A porcentagem pela qual aumentar o espaço de tabela quando ele ficar cheio. O tipo de dados é `integer`. Valores válidos: `1` a `100`. O padrão é `0`.

max_size

O tamanho máximo do espaço de tabela. O tipo de dados é `varchar`. Valores válidos: *inteiro* `K` | `M` | `G` ou `NONE`. O padrão é `NONE`.

reduce_max

Especifica se a marca d'água alta deve ser reduzida até o limite máximo. O tipo de dados é `char`. O padrão é `N`.

reduce_stop

Especifica se um comando anterior `reduce_max` ou `reduce_value` deve ser interrompido. O tipo de dados é `char`. O padrão é `N`.

reduce_value

O número ou a porcentagem para reduzir o limite máximo do espaço de tabela. O tipo de dados é `varchar`. Valores válidos: *inteiro* `K`| `M` | `G` ou `1` a `100`. O padrão é `N`.

lower_high_water

Especifica se o comando `ALTER TABLESPACE LOWER HIGH WATER MARK` deve ser executado. O tipo de dados é `char`. O padrão é `N`.

lower_high_water_stop

Especifica se o comando `ALTER TABLESPACE LOWER HIGH WATER MARK STOP` deve ser executado. O tipo de dados é `char`. O padrão é `N`.

switch_online

Especifica se o comando `ALTER TABLESPACE SWITCH ONLINE` deve ser executado. O tipo de dados é `char`. O padrão é `N`.

tablespace_prefetch_size

O tamanho da página de pré-busca do espaço de tabela. O tipo de dados é `char`. Valores válidos: `AUTOMATIC` (sem distinção entre maiúsculas e minúsculas) ou números inteiros positivos diferentes de zero que sejam menores ou iguais a 32767.

> **Note**
>
> Esse parâmetro só funciona com `buffer_pool_name`, `table_increase_size`, `max_size` e `switch_online`. Não funciona com `reduce_max`, `reduce_stop`, `reduce_value`, `lower_high_water` e `lower_high_water_stop`.

Observações de uso

Antes de chamar o procedimento armazenado, examine estas considerações:

- O procedimento armazenado `rdsadmin.alter_tablespace` não funcionará em um espaço de tabela com o `tablespace_type` definido como `T` para dados temporários do usuário.
- Os parâmetros opcionais `reduce_max`, `reduce_stop`, `reduce_value`, `lower_high_water`, `lower_high_water_stop` e `switch_online` são mutuamente excludentes. Não é possível combiná-los com nenhum outro parâmetro opcional, como `buffer_pool_name`, no comando `rdsadmin.alter_tablespace`. Para obter mais informações, consulte Statement not valid.

Para ter informações sobre como conferir o status de alteração de um espaço de tabela, consulte rdsadmin.get_task_status.

Para ver as mensagens de erro exibidas ao chamar procedimentos armazenados, consulte the section called “Erros de procedimentos armazenados”.

Exemplos

Exemplo 1: diminuição da marca d'água alta

O exemplo a seguir altera um espaço de tabela chamado SP8 e atribui um grupo de buffers denominado BP8 para um banco de dados denominado TESTDB para reduzir a marca d'água alta.

```
db2 "call rdsadmin.alter_tablespace(
    'TESTDB',
    'SP8',
    'BP8',
    NULL,
    NULL,
    'Y')"
```

Exemplo 2: redução da marca d'água alta

O exemplo a seguir executa o comando REDUCE MAX em um espaço de tabela chamado TBSP_TEST no banco de dados TESTDB.

```
db2 "call rdsadmin.alter_tablespace(
    'TESTDB',
    'TBSP_TEST',
    NULL,
    NULL,
    NULL,
    'Y')"
```

Exemplo 3: interrupção de comandos para reduzir a marca d'água alta

O exemplo a seguir executa o comando REDUCE STOP em um espaço de tabela chamado TBSP_TEST no banco de dados TESTDB.

```
db2 "call rdsadmin.alter_tablespace(
    'TESTDB',
    'TBSP_TEST',
    NULL,
    NULL,
```

```
    NULL,
    NULL,
    'Y')"
```

Exemplo 4: alterar o tamanho da página de pré-busca existente

O exemplo a seguir executa o comando `ALTER TABLESPACE SWITCH ONLINE` em um espaço de tabela chamado `TSBP_TEST` e altera o tamanho da página de pré-busca existente para 64.

```
db2 "call rdsadmin.alter_tablespace(
    'TESTDB',
    'TBSP_TEST',
    NULL,
    NULL,
    NULL,
    NULL,
    NULL,
    NULL,
    NULL,
    NULL,
    NULL,
    ‘Y’,
    ‘64’)"
```

rdsadmin.rename_tablespace

Renomeia um espaço de tabela.

Sintaxe

```
db2 "call rdsadmin.rename_tablespace(
    ?,
    'database_name',
    'source_tablespace_name',
    'target_tablespace_name')"
```

Parâmetros

Os seguintes parâmetros são obrigatórios:

?

Um marcador de parâmetro que gera uma mensagem de erro. Esse parâmetro só aceita ?.

database_name

O nome do banco de dados ao qual pertence o espaço de tabela. O tipo de dados é `varchar`.

source_tablespace_name

O nome do espaço de tabela a ser renomeado. O tipo de dados é `varchar`.

target_tablespace_name

O novo nome do espaço de tabela. O tipo de dados é `varchar`.

O nome tem as seguintes restrições:

- Não pode ser igual ao nome de um espaço de tabela existente.
- Só pode conter os caracteres `_$#@a-zA-Z0-9`.
- Não pode começar com _ ou $.
- Não pode começar com SYS.

Observações de uso

Para ter informações sobre como conferir o status da renomeação de um espaço de tabela, consulte rdsadmin.get_task_status.

Não é possível renomear espaços de tabela que pertençam ao banco de dados `rdsadmin`.

Exemplos

O exemplo a seguir renomeia um espaço de tabela denominado SP8 para SP9 em um banco de dados chamado TESTDB.

```
db2 "call rdsadmin.rename_tablespace(
    ?,
    'TESTDB',
    'SP8',
    'SP9')"
```

rdsadmin.drop_tablespace

Descarta um espaço de tabela.

Sintaxe

```
db2 "call rdsadmin.drop_tablespace(
    'database_name',
    'tablespace_name')"
```

Parâmetros

Os seguintes parâmetros são obrigatórios:

database_name

O nome do banco de dados ao qual pertence o espaço de tabela. O tipo de dados é `varchar`.

tablespace_name

O nome do espaço de tabela a ser descartado. O tipo de dados é `varchar`.

Observações de uso

Para ter informações sobre como conferir o status de descarte de um espaço de tabela, consulte rdsadmin.get_task_status.

Exemplos

O exemplo a seguir descarta um espaço de tabela denominado `SP8` de um banco de dados chamado `TESTDB`.

```
db2 "call rdsadmin.drop_tablespace(
    'TESTDB',
    'SP8')"
```

Referência de função definida pelo usuário do Amazon RDS para Db2

As seguintes funções definidas pelo usuário estão disponíveis para instâncias de banco de dados do Amazon RDS que executam o mecanismo do Db2.

Tópicos

- rdsadmin.get_task_status
- rdsadmin.list_databases

rdsadmin.get_task_status

Exibe o status de uma tarefa.

Sintaxe

```
db2 "select task_id, task_type, database_name, lifecycle,
    varchar(bson_to_json(task_input_params), 500) as task_params,
    cast(task_output as varchar(500)) as task_output
    from table(rdsadmin.get_task_status(task_id,'database_name','task_type'))"
```

Parâmetros

Os parâmetros a seguir são opcionais. Se você não fornecer parâmetros, a função definida pelo usuário exibirá o status de todas as tarefas de todos os bancos de dados. O Amazon RDS retém o histórico de tarefas por 35 dias.

`task_id`

O ID da tarefa que está sendo executada. Esse ID é exibido quando você executa uma tarefa. Padrão: `0`.

`database_name`

O nome do banco de dados no qual a tarefa está sendo executada.

`task_type`

O tipo da tarefa a ser consultada. Valores válidos: `ADD_GROUPS`, `ADD_USER`, `ALTER_BUFFERPOOL`, `ALTER_TABLESPACE`, `CHANGE_PASSWORD`, `COMPLETE_ROLLFORWARD`,

`CREATE_BUFFERPOOL`, `CREATE_DATABASE`, `CREATE_ROLE`, `CREATE_TABLESPACE`, `DROP_BUFFERPOOL`, `DROP_DATABASE`, `DROP_TABLESPACE`, `LIST_USERS`, `REMOVE_GROUPS`, `REMOVE_USER`, `RESTORE_DB`, `ROLLFORWARD_DB_LOG`, `ROLLFORWARD_STATUS`, `UPDATE_DB_PARAM`.

Observações de uso

É possível usar a função `rdsadmin.get_task_status` definida pelo usuário para conferir o status das tarefas a seguir referentes ao Amazon RDS para Db2. Essa lista não é exaustiva.

- Criar, alterar ou descartar um grupo de buffers
- Criar, alterar ou descartar um espaço de tabela
- Criar ou descartar um banco de dados
- Restaurar o backup de um banco de dados do Amazon S3
- Realizando roll forward dos logs de banco de dados do Amazon S3

Exemplos

No exemplo a seguir vemos as colunas exibidas quando `rdsadmin.get_task_status` é chamado.

```
db2 "describe select * from table(rdsadmin.get_task_status())"
```

O exemplo a seguir lista o status de todas as tarefas.

```
db2 "select task_id, task_type, database_name, lifecycle,
    varchar(bson_to_json(task_input_params), 500) as task_params,
    cast(task_output as varchar(500)) as task_output
    from table(rdsadmin.get_task_status(null,null,null))"
```

O exemplo a seguir lista o status de uma tarefa específica.

```
db2 "select task_id, task_type, database_name,
    varchar(bson_to_json(task_input_params), 500) as task_params
    from table(rdsadmin.get_task_status(1,null,null))"
```

O exemplo a seguir lista o status de uma tarefa e um banco de dados específicos.

```
db2 "select task_id, task_type, database_name,
    varchar(bson_to_json(task_input_params), 500) as task_params
    from table(rdsadmin.get_task_status(2,'SAMPLE',null))"
```

O exemplo a seguir lista o status de todas as tarefas ADD_GROUPS.

```
db2 "select task_id, task_type, database_name,
    varchar(bson_to_json(task_input_params), 500) as task_params
    from table(rdsadmin.get_task_status(null,null,'add_groups'))"
```

O exemplo a seguir lista o status de todas as tarefas de um banco de dados específico.

```
db2 "select task_id, task_type, database_name,
    varchar(bson_to_json(task_input_params), 500) as task_params
    from table(rdsadmin.get_task_status(null,'testdb', null))"
```

O exemplo a seguir mostra os valores JSON como colunas.

```
db2 "select varchar(r.task_type,25) as task_type, varchar(r.lifecycle,10) as lifecycle,
 r.created_at, u.* from
    table(rdsadmin.get_task_status(null,null,'restore_db')) as r,
 json_table(r.task_input_params, 'strict $' columns(s3_prefix varchar(500)
    null on empty, s3_bucket_name varchar(500) null on empty) error on error ) as U"
```

Resposta

A função `rdsadmin.get_task_status` definida pelo usuário exibe as seguintes colunas:

`TASK_ID`

O ID da tarefa.

`TASK_TYPE`

Depende dos parâmetros de entrada.

- `ADD_GROUPS`: adiciona grupos.
- `ADD_USER`: adiciona um usuário.
- `ALTER_BUFFERPOOL`: altera um grupo de buffers.
- `ALTER_TABLESPACE`: altera um espaço de tabela.

- CHANGE_PASSWORD : altera a senha de um usuário.
- COMPLETE_ROLLFORWARD: conclui uma tarefa `rdsadmin.rollforward_database` e ativa um banco de dados.
- CREATE_BUFFERPOOL: cria um grupo de buffers.
- CREATE_DATABASE: cria um banco de dados.
- CREATE_ROLE: cria um perfil do Db2 para um usuário.
- CREATE_TABLESPACE: cria um espaço de tabela.
- DROP_BUFFERPOOL: descarta um grupo de buffers.
- DROP_DATABASE: descarta um banco de dados.
- DROP_TABLESPACE: descarta um espaço de tabela.
- LIST_USERS: lista todos os usuários.
- REMOVE_GROUPS: remove os grupos.
- REMOVE_USER: remove um usuário.
- RESTORE_DB: restaura um banco de dados completo.
- ROLLFORWARD_DB_LOG: executa uma tarefa `rdsadmin.rollforward_database` nos logs do banco de dados.
- ROLLFORWARD_STATUS : retorna o status de uma tarefa `rdsadmin.rollforward_database`.
- UPDATE_DB_PARAM: atualiza os parâmetros de dados.

DATABASE_NAME

O nome do banco de dados ao qual a tarefa está associada.

COMPLETED_WORK_BYTES

O número de bytes restaurados pela tarefa.

DURATION_MINS

O tempo necessário para concluir a tarefa.

LIFECYCLE

O status da tarefa. Possíveis status:

- CREATED: depois que uma tarefa é enviada ao Amazon RDS, o Amazon RDS define o status como CREATED.

- IN_PROGRESS: depois de iniciar uma tarefa, o Amazon RDS define o status como IN_PROGRESS. Pode demorar até cinco minutos para que o status mude de CREATED para IN_PROGRESS.
- SUCCESS: após o término de uma tarefa, o Amazon RDS define o status como SUCCESS.
- ERROR: se uma tarefa de restauração falhar, o Amazon RDS definirá o status como ERROR. Para ter mais informações sobre o erro, consulte TASK_OUPUT.

CREATED_BY

O authid que criou o comando.

CREATED_AT

A data e hora em que a tarefa foi criada.

LAST_UPDATED_AT

A data e a hora em que a tarefa foi atualizada pela última vez.

TASK_INPUT_PARAMS

Os parâmetros diferem com base no tipo de tarefa. Todos os parâmetros de entrada são representados como um objeto JSON. Por exemplo, as chaves JSON para a tarefa RESTORE_DB são as seguintes:

- DBNAME
- RESTORE_TIMESTAMP
- S3_BUCKET_NAME
- S3_PREFIX

TASK_OUTPUT

Informações adicionais sobre a tarefa. No caso de erro durante a restauração nativa, essa coluna inclui as informações sobre o erro.

Exemplos de resposta

O exemplo de resposta a seguir mostra que um banco de dados chamado TESTJP foi criado com êxito. Para ter mais informações, consulte o procedimento armazenando the section called “rdsadmin.create_database”.

```
`1 SUCCESS CREATE_DATABASE RDSDB 2023-10-24-18.32.44.962689 2023-10-24-18.34.50.038523
 1 TESTJP { "CODESET" : "IBM-437", "TERRITORY" : "JP", "COLLATION" : "SYSTEM",
 "AUTOCONFIGURE_CMD" : "", "PAGESIZE" : 4096 }
2023-10-24-18.33.30.079048 Task execution has started.

2023-10-24-18.34.50.038523 Task execution has completed successfully`.
```

O exemplo de resposta a seguir explica por que o descarte de um banco de dados falhou. Para ter mais informações, consulte o procedimento armazenando the section called "rdsadmin.drop_database".

```
1 ERROR DROP_DATABASE RDSDB 2023-10-10-16.33.03.744122 2023-10-10-16.33.30.143797 -
 2023-10-10-16.33.30.098857 Task execution has started.
2023-10-10-16.33.30.143797 Caught exception during executing task id 1, Aborting task.
Reason Dropping database created via rds CreateDBInstance api is not allowed.
Only database created using rdsadmin.create_database can be dropped
```

O exemplo de resposta a seguir mostra a restauração bem-sucedida de um banco de dados. Para ter mais informações, consulte o procedimento armazenando the section called "rdsadmin.restore_database".

```
1 RESTORE_DB  SAMPLE  SUCCESS

{ "S3_BUCKET_NAME" : "amzn-s3-demo-bucket", "S3_PREFIX" :
 "SAMPLE.0.rdsdb3.DBPART000.20230413183211.001", "RESTORE_TIMESTAMP" :
 "20230413183211", "BACKUP_TYPE" : "offline" }




2023-11-06-18.31.03.115795 Task execution has started.
2023-11-06-18.31.04.300231 Preparing to download
2023-11-06-18.31.08.368827 Download complete. Starting Restore
2023-11-06-18.33.13.891356 Task Completed Successfully
```

rdsadmin.list_databases

Retorna uma lista de todos os bancos de dados em execução em uma instância de banco de dados do RDS para Db2.

Sintaxe

```
db2 "select * from table(rdsadmin.list_databases())"
```

Observações de uso

Essa função definida pelo usuário não especifica se os bancos de dados estão em um estado ativado ou desativado.

Se você não vir seus bancos de dados na lista, chame a função definida pelo usuário the section called “rdsadmin.get_task_status” e procure mensagens de erro.

Resposta

A função `rdsadmin.list_databases` definida pelo usuário exibe as seguintes colunas:

`DATABASE_NAME`

O nome de um banco de dados.

`CREATE_TIME`

A data e a hora em que o banco de dados foi criado.

Exemplos de resposta

O exemplo de resposta a seguir mostra uma lista de bancos de dados e os horários em que eles foram criados. `rdsadmin` é um banco de dados que o Amazon RDS gerencia e sempre aparece na saída.

```
DATABASE_NAME   CREATE_TIME
--------------- --------------------------
rdsadmin        2024-10-22-03.37.48.535671
TEST            2024-10-22-03.39.36.818679
TEST1           2024-10-22-03.57.15.218009
TEST2           2024-10-22-03.59.28.029556
```

Solução de problemas do Amazon RDS para Db2

O conteúdo a seguir pode ajudar a solucionar problemas que você encontrar com o RDS para Db2.

Para ter mais informações sobre como solucionar problemas gerais do Amazon RDS, consulte Solução de problemas para o Amazon RDS.

Tópicos

- Erro de conexão de banco de dados
- Erro de E/S de arquivo
- Erros de procedimentos armazenados

Erro de conexão de banco de dados

A mensagem de erro a seguir indica que um banco de dados falhou na conexão porque o servidor não tem memória suficiente.

```
SQL1643C The database manager failed to allocate shared memory because the
database manager instance memory limit has been reached.
```

Aumente a memória da instância de banco de dados e tente se conectar ao banco de dados novamente. Para obter informações sobre o uso da memória e recomendações para bancos de dados, consulte the section called “Vários bancos de dados do Db2”. Para obter informações sobre como atualizar a memória de um banco de dados do RDS para Db2, consulte the section called “rdsadmin.update_db_param”.

Erro de E/S de arquivo

Você pode encontrar um erro de E/S de arquivo por diferentes motivos, como quando você usa o comando LOAD ou chama o procedimento armazenado `rdsadmin.restore_database`.

Neste exemplo, você executa o comando LOAD a seguir.

```
db2 "call sysproc.admin_cmd('load from "DB2REMOTE://s3test//public/datapump/t6.del" of
 del lobs from "DB2REMOTE://s3test/public/datapump/" modified by lobsinfile MESSAGES ON
 SERVER insert INTO RDSDB.t6 nonrecoverable ')"
```

O comando LOAD exibe a seguinte mensagem:

```
 Result set 1
 --------------

 ROWS_READ            ROWS_SKIPPED         ROWS_LOADED          ROWS_REJECTED
ROWS_DELETED         ROWS_COMMITTED       ROWS_PARTITIONED     NUM_AGENTINFO_ENTRIES
MSG_RETRIEVAL




MSG_REMOVAL
 -------------------- -------------------- -------------------- --------------------
-------------------- -------------------- -------------------- --------------------
--------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------
                     -                    -                    -                    -
                    -                    -                    -                    -
SELECT SQLCODE, MSG FROM TABLE(SYSPROC.ADMIN_GET_MSGS('1594987316_285548770')) AS MSG




                                                                                    CALL
SYSPROC.ADMIN_REMOVE_MSGS('1594987316_285548770')

 1 record(s) selected.

 Return Status = 0

SQL20397W  Routine "SYSPROC.ADMIN_CMD" execution has completed, but at least
one error, "SQL1652", was encountered during the execution. More information
is available.  SQLSTATE=01H52
```

Para visualizar a mensagem de erro, execute o comando SQL conforme sugerido na resposta anterior. `SELECT SQLCODE, MSG FROM TABLE(SYSPROC.ADMIN_GET_MSGS('1594987316_285548770')) AS MSG` exibe a seguinte mensagem:

```
SQLCODE   MSG
```

```
---------
 ------------------------------------------------------------------------------------------
SQL2025N  An I/O error occurred. Error code “438”. Media on which this error occurred:
 “DB2REMOTE://s3test//public/datapump/t6.del”

SQL3500W The utility is beginning the LOAD phase at time “07/05/2024 21:21:48.082954”

SQL1652N File I/O error occurred
```

Os logs de diagnóstico do Db2 contêm um arquivo de log semelhante ao seguinte:

```
2024-07-05-21.20.09.440609+000 I1191321E864          LEVEL: Error
PID     : 2710                 TID : 139619509200640 PROC : db2sysc 0
INSTANCE: rdsdb                NODE : 000            DB   : NTP
APPHDL  : 0-12180              APPID: xxx.xx.x.xxx.xxxxx.xxxxxxxxxxxx
UOWID   : 5                    ACTID: 1
AUTHID  : ADMIN                HOSTNAME: ip-xx-xx-x-xx
EDUID   : 147                  EDUNAME: db2lmr 0
FUNCTION: DB2 UDB, oper system services, sqloS3Client_GetObjectInfo, probe:219
MESSAGE : ZRC=0x870F01B6=-2029059658=SQLO_FAILED
          "An unexpected error is encountered"
DATA #1 : String, 29 bytes
S3:HeadObject request failed.
DATA #2 : signed integer, 4 bytes
99
DATA #3 : String, 0 bytes
Object not dumped: Address: 0x00007EFC08A9AE38 Size: 0 Reason: Zero-length data
DATA #4 : String, 33 bytes
curlCode: 28, Timeout was reached
```

Esse erro de E/S do arquivo pode se originar de vários cenários diferentes. Por exemplo, a VPC associada ao grupo de segurança usado para criar a instância de banco de dados do RDS para Db2 pode não ter um endpoint do gateway do Amazon S3. Esse endpoint é essencial para permitir que o RDS para Db2 acesse o Amazon S3. Se a instância de banco de dados do RDS para Db2 estiver em sub-redes privadas, será necessário um endpoint do gateway do Amazon S3. É possível especificar se a instância de banco de dados usa sub-redes privadas ou públicas configurando grupos de sub-redes do Amazon RDS. Para obter mais informações, consulte Trabalhar com grupos de sub-redes de banco de dados.

Tópicos

- Etapa 1: Criar um endpoint do gateway da VPC para o Amazon S3

- Etapa 2: Confirmar se o endpoint do gateway da VPC para o Amazon S3 existe

Etapa 1: Criar um endpoint do gateway da VPC para o Amazon S3

Para que a instância de banco de dados do RDS para Db2 interaja com o Amazon S3, crie uma VPC e, em seguida, um endpoint do gateway do Amazon S3 para uso de sub-redes privadas.

Como criar um endpoint do gateway da VPC para o S3

1. Crie uma VPC. Para obter mais informações, consulte Crie uma VPC no Guia do usuário da Amazon Virtual Private Cloud.
2. Crie um endpoint do gateway do Amazon S3 para uso de sub-redes privadas. Para obter mais informações, consulte Gateway endpoints no Guia do AWS PrivateLink.

Etapa 2: Confirmar se o endpoint do gateway da VPC para o Amazon S3 existe

Confirme se você criou com sucesso um endpoint do gateway do Amazon S3 usando o AWS Management Console ou a AWS CLI.

Console

Como confirmar um endpoint do gateway do Amazon S3

1. Faça login no AWS Management Console e abra o console da Amazon VPC em https://console.aws.amazon.com/vpc.
2. No canto superior direito do console, escolha a Região da AWS da VPC.
3. Selecione a VPC que você criou.
4. Na guia Mapa de recursos, em Conexões de rede, confirme se um endpoint do gateway do Amazon S3 está listado.

AWS CLI

Para confirmar um endpoint do gateway do Amazon S3, execute o comando describe-vpc-endpoints. No exemplo a seguir, substitua *vpc_id* pelo ID da VPC, *region* pela sua Região da AWS, e *profile* pelo seu nome de perfil.

Para Linux, macOS ou Unix:

```
aws ec2 describe-vpc-endpoints \
    --filters "Name=vpc-id,Values=$vpc_id" \
    "Name=service-name,\
    Values=com.amazonaws.${region}.s3" \
    --region $region --profile=$profile \
    --query "VpcEndpoints[*].VpcEndpointId" --output text
```

Para Windows:

```
aws ec2 describe-vpc-endpoints ^
    --filters "Name=vpc-id,Values=$vpc_id" ^
    "Name=service-name,^
    Values=com.amazonaws.${region}.s3" ^
    --region $region --profile=$profile ^
    --query "VpcEndpoints[*].VpcEndpointId" --output text
```

Este comando produz uma saída semelhante ao exemplo a seguir se existir um endpoint do gateway do Amazon S3.

```
[
    "vpce-0ea810434ff0b97e4"
]
```

Este comando produz uma saída semelhante ao exemplo a seguir se não existir um endpoint do gateway do Amazon S3.

```
[]
```

Se você não vir um endpoint do gateway do Amazon S3 listado, então siga a Etapa 1: Criar um endpoint do gateway da VPC para o Amazon S3.

Erros de procedimentos armazenados

Esta sessão descreve vários erros exibidos ao chamar procedimentos armazenados e como resolvê-los.

Categoria	Erros de procedimentos armazenados
Bancos de dados	Erros rdsadmin.activate_database

Categoria	Erros de procedimentos armazenados
Bancos de dados	Erros rdsadmin.create_database
Bancos de dados	Erros rdsadmin.deactivate_database
Bancos de dados	Erros rdsadmin.drop_database
Bancos de dados	Erros rdsadmin.reactivate_database
Bancos de dados	erros rdsadmin.restore_database
Bancos de dados	Erros rdsadmin.update_db_param
Tablespaces	Erros rdsadmin.alter_tablespace

Erros rdsadmin.activate_database

Os erros a seguir podem ocorrer quando o procedimento armazenado ??? é chamado.

Erro	Mensagem de erro
Failed to allocate shared memory	`SQL1643C The database manager failed to allocate shared memory because the database manager instance memory limit has been reached.`
Unable to activate because of running processes	`The database can’t be activated because it's in the process of being created or restored.`

Falha ao alocar memória compartilhada

A mensagem de erro a seguir indica que o procedimento armazenado falhou ao ativar um banco de dados porque a instância de banco de dados não tem memória suficiente.

```
SQL1643C The database manager failed to allocate shared memory because the database
 manager instance memory limit has been reached.
```

Aumente a memória da instância de banco de dados e, em seguida, chame o procedimento armazenado `rdsadmin.activate_database` novamente. Para obter informações sobre o uso da memória e recomendações para bancos de dados, consulte the section called “Vários bancos de dados do Db2”.

Não é possível ativar devido a processos em execução

A mensagem de erro a seguir indica que o procedimento armazenado não pôde ativar um banco de dados porque o procedimento armazenado `rdsadmin.create_database` ou `rdsadmin.restore_database` está em execução.

```
The database can’t be activated because it's in the process of being created or
 restored.
```

Aguarde alguns minutos e, depois, chame o procedimento armazenado `rdsadmin.activate_database` novamente.

Erros rdsadmin.alter_tablespace

Os erros a seguir podem ocorrer quando o procedimento armazenado the section called “rdsadmin.alter_tablespace” é chamado.

Erro	Mensagem de erro

Statement not valid	`DB21034E The command was processed as an SQL statement because it was not a valid Command Line Processor command. During SQL processing it returned:`
	
	`SQL1763N Invalid ALTER TABLESPACE statement for table space "TBSP_TEST" due to reason "12"`

tablespace_prefetch_size value not valid	`Invalid tablespace_prefetch_size. Set value to AUTOMATIC or to a non-zero positive numerical value.`
tablespace_prefetch_size numerical value not valid	`Invalid tablespace_prefetch_size. The number of pages can't be greater than 32767.`

Erro	Mensagem de erro
Parameter can't be used with tablespace_prefetch_size	`You can't use tablespace_prefetch_size with {parameter }.`
Tablespace change failed	`The change to tablespace { tablespace_name}  failed because you can only alter LARGE or REGULAR tablespaces.`

Instrução não válida

A mensagem de erro a seguir indica que o procedimento armazenado combinou parâmetros opcionais mutuamente exclusivos com outros parâmetros opcionais. Os parâmetros opcionais `reduce_max, reduce_stop`, `reduce_value`, `lower_high_water`, `lower_high_water_stop` e `switch_online` para o procedimento armazenado `rdsadmin.alter_tablespace` são mutuamente exclusivos. Não é possível combiná-los com nenhum outro parâmetro opcional, como `buffer_pool_name`, no procedimento armazenado `rdsadmin.alter_tablespace`. Se você combiná-los, ao chamar a função `rdsadmin.get_task_status` definida pelo usuário, o Db2 exibirá essa mensagem de erro.

```
DB21034E The command was processed as an SQL statement because it was not a valid
 Command Line Processor command. During SQL processing it returned:
SQL1763N Invalid ALTER TABLESPACE statement for table space "TBSP_TEST" due to reason
 "12"
```

Chame o procedimento armazenado `rdsadmin.alter_tablespace` novamente sem combinar parâmetros opcionais mutuamente exclusivos com outros parâmetros opcionais. Depois, chame a função definida pelo usuário `rdsadmin.get_task_status`. Para ter mais informações, consulte rdsadmin.alter_tablespace e rdsadmin.get_task_status.

valor de tablespace_prefetch_size inválido

A mensagem de erro a seguir indica que você não definiu `tablespace_prefetch_size` como `AUTOMATIC` ou um valor numérico não positivo. Por exemplo, você tentou configurá-lo como `testinput`.

```
Invalid tablespace_prefetch_size. Set value to AUTOMATIC or to a non-zero positive
 numerical value.
```

Chame o procedimento armazenado `rdsadmin.alter_tablespace` novamente e defina `tablespace_prefetch_size` como `AUTOMATIC` ou um valor numérico não positivo.

valor numérico de tablespace_prefetch_size inválido

A mensagem de erro a seguir indica que você definiu `tablespace_prefetch_size` como um valor numérico maior que 32767.

```
Invalid tablespace_prefetch_size. The number of pages can't be greater than 32767.
```

Chame o procedimento armazenado `rdsadmin.alter_tablespace` novamente e defina `tablespace_prefetch_size` como um valor numérico positivo diferente de zero menor ou igual a 32767.

O parâmetro não pode ser usado com tablespace_prefetch_size

A mensagem de erro a seguir indica que você tentou usar `tablespace_prefetch_size` com um parâmetro incompatível.

```
You can't use tablespace_prefetch_size with {parameter}.
```

Chame o procedimento armazenado `rdsadmin.alter_tablespace` novamente e use somente `tablespace_prefetch_size` com parâmetros compatíveis. Consulte informações sobre parâmetros que podem ser usados com `tablespace_prefetch_size` em rdsadmin.alter_tablespace.

Falha na alteração do espaço de tabela

A mensagem de erro a seguir indica que você tentou alterar um espaço de tabela.

```
The change to tablespace {tablespace_name} failed because you can only alter LARGE or
 REGULAR tablespaces.
```

Erros rdsadmin.create_database

O erro a seguir pode ocorrer quando o procedimento armazenado the section called “rdsadmin.create_database” é chamado.

Erro	Mensagem de erro
Failed to allocate shared memory	`SQL1643C The database manager failed to allocate shared memory because the database manager instance memory limit has been reached.`

Falha ao alocar memória compartilhada

A mensagem de erro a seguir indica que o procedimento armazenado falhou ao criar um banco de dados porque a instância de banco de dados não tem memória suficiente.

```
SQL1643C The database manager failed to allocate shared memory because the database
 manager instance memory limit has been reached.
```

Aumente a memória da instância de banco de dados e, em seguida, chame o procedimento armazenado `rdsadmin.create_database` novamente. Para obter informações sobre o uso da memória e recomendações para bancos de dados, consulte the section called “Vários bancos de dados do Db2”.

Para confirmar que o banco de dados foi criado, chame a função the section called “rdsadmin.list_databases” definida pelo usuário e verifique se o novo banco de dados está listado.

Erros rdsadmin.deactivate_database

O erro a seguir pode ocorrer quando o procedimento armazenado ??? é chamado.

Erro	Mensagem de erro
Unable to deactivate because of running processes	`The database can’t be deactivated because it's in the process of being created or restored.`

Não é possível desativar devido a processos em execução

A mensagem de erro a seguir indica que o procedimento armazenado não pôde desativar um banco de dados porque o procedimento armazenado `rdsadmin.create_database` ou `rdsadmin.restore_database` está em execução.

```
The database can’t be deactivated because it's in the process of being created or
 restored.
```

Aguarde alguns minutos e, depois, chame o procedimento armazenado `rdsadmin.deactivate_database` novamente.

Erros rdsadmin.drop_database

Os erros a seguir podem ocorrer quando o procedimento armazenado the section called “rdsadmin.drop_database” é chamado.

Erro	Mensagem de erro
Database name doesn't exist	`SQL0438N Application raised error or warning with diagnostic text: "Cannot drop database. Database with provided name does not exist". SQLSTATE=99993`
Return status = 0	`Return Status = 0`
Dropping database not allowed	`1 ERROR DROP_DATABASE RDSDB 2023-10-10-16.33.0 3.744122 2023-10-10-16.33.30.143797 - 2023-10-1 0-16.33.30.098857 Task execution has started. 2023-10-10-16.33.30.143797 Caught exception during executing task id 1, Aborting task. Reason Dropping database created via rds CreateDBInstance api is not allowed. Only database created using rdsadmin. create_database can be dropped`

O nome do banco de dados não existe

A mensagem de erro a seguir indica que você transmitiu um nome de banco de dados incorreto no procedimento armazenado `rdsadmin.drop_database`.

```
SQL0438N Application raised error or warning with diagnostic text: "Cannot
drop database. Database with provided name does not exist". SQLSTATE=99993
```

Chame o procedimento armazenado `rdsadmin.drop_database` novamente com um nome de banco de dados correto. Para confirmar que o banco de dados foi descartado, chame a função

the section called “rdsadmin.list_databases” definida pelo usuário e verifique se o banco de dados descartado não está listado.

Status de retorno = 0

A mensagem de erro a seguir indica que o procedimento armazenado não pôde ser concluído.

```
Return Status = 0
```

Depois de receber `Return Status = 0`, chame a função definida pelo usuário rdsadmin.get_task_status.

Não é permitido descartar o banco de dados

A mensagem de erro a seguir indica que você criou o banco de dados usando o console do Amazon RDS ou a AWS CLI. Só é possível usar o procedimento armazenado `rdsadmin.drop_database` se você criou o banco de dados chamando o procedimento armazenado the section called “rdsadmin.create_database”.

```
1 ERROR DROP_DATABASE RDSDB 2023-10-10-16.33.03.744122 2023-10-10-16.33.30.143797 -
 2023-10-10-16.33.30.098857 Task execution has started.
2023-10-10-16.33.30.143797 Caught exception during executing task id 1, Aborting task.
Reason Dropping database created via rds CreateDBInstance api is not allowed.
Only database created using rdsadmin.create_database can be dropped
```

Para descartar um banco de dados que foi criado usando o console do Amazon RDS ou a AWS CLI, use um cliente para se conectar ao banco de dados e execute o comando apropriado.

Erros rdsadmin.reactivate_database

O erro a seguir pode ocorrer quando o procedimento armazenado ??? é chamado.

Erro	Mensagem de erro
Failed to allocate shared memory	`SQL1643C The database manager failed to allocate shared memory because the database manager instance memory limit has been reached.`

Erro	Mensagem de erro
Unable to reactivate because of running processes	`The database can’t be reactivated because it's in the process of being created or restored.`

Falha ao alocar memória compartilhada

A mensagem de erro a seguir indica que o procedimento armazenado falhou ao ativar um banco de dados porque a instância de banco de dados não tem memória suficiente.

```
SQL1643C The database manager failed to allocate shared memory because the database
 manager instance memory limit has been reached.
```

Aumente a memória da instância de banco de dados e, em seguida, chame o procedimento armazenado `rdsadmin.activate_database` novamente. Para obter informações sobre o uso da memória e recomendações para bancos de dados, consulte the section called “Vários bancos de dados do Db2”.

Não é possível reativar devido a processos em execução

A mensagem de erro a seguir indica que o procedimento armazenado não pôde reativar um banco de dados porque o procedimento armazenado `rdsadmin.create_database` ou `rdsadmin.restore_database` está em execução.

```
The database can’t be reactivated because it's in the process of being created or
 restored.
```

Aguarde alguns minutos e, depois, chame o procedimento armazenado `rdsadmin.reactivate_database` novamente.

erros rdsadmin.restore_database

Os seguintes erros podem ocorrer quando você chama o procedimento armazenado the section called “rdsadmin.restore_database”:

Erro	Mensagem de erro
Insufficient disk space	`Aborting task. Reason Restoring your database failed because of insufficient disk space. Increase the storage for your DB instance and rerun the rdsadmin.restore_database stored procedure.`
Internal error	`Caught exception during executing task id 104, Aborting task. Reason Internal Error`
Non-fenced routines not allowed	`Caught exception during executing task id 2, Aborting task. Reason Non fenced routines are not allowed. Please delete the routines and retry the restore.`
Tablespaces not restored	`Reason SQL0970N The system attempted to write to a read-only file. Reason SQL2563W The Restore process has completed successfully. However one or more table spaces from the backup were not restored.`

Espaço em disco insuficiente

A mensagem de erro a seguir indica que a instância de banco de dados não tem espaço em disco suficiente para restaurar o banco de dados:

```
Aborting task. Reason Restoring your database failed because of insufficient
 disk space. Increase the storage for your DB instance and rerun the
 rdsadmin.restore_database stored procedure.
```

O espaço livre na instância de banco de dados deve ser mais de duas vezes maior que o tamanho da imagem de backup. Se a imagem de backup estiver compactada, o espaço livre na instância de banco de dados deverá ser mais de três vezes maior que o tamanho da imagem de backup. Para obter mais informações, consulte the section called “Aumento da capacidade de armazenamento da instância de banco de dados”.

Aumente o espaço em disco e, em seguida, chame o procedimento armazenado `rdsadmin.restore_database` novamente. Para confirmar que o banco de dados foi restaurado, chame a função the section called “rdsadmin.list_databases” definida pelo usuário e verifique se o banco de dados restaurado está listado.

Erro interno

A mensagem de erro a seguir indica que o procedimento armazenado encontrou um erro interno:

```
Caught exception during executing task id 104, Aborting task. Reason Internal Error
```

Entrar em contato com o AWS Support.

Rotinas não protegidas não são permitidas

A seguinte mensagem de erro indica que o banco de dados contém rotinas não protegidas:

```
Caught exception during executing task id 2, Aborting task. Reason Non fenced routines
 are not allowed. Please delete the routines and retry the restore.
```

O RDS para Db2 não permite rotinas não protegidas. Remova as rotinas não protegidas do banco de dados de origem e chame `rdsadmin.restore_database` novamente. Para confirmar que o banco de dados foi restaurado, chame a função the section called “rdsadmin.list_databases” definida pelo usuário e verifique se o banco de dados restaurado está listado. Para obter mais informações, consulte the section called “Rotinas não protegidas”.

Espaços de tabela não restaurados

A seguinte mensagem de erro indica que o RDS para Db2 restaurou com sucesso o banco de dados, mas não conseguiu restaurar um ou mais espaços de tabela:

```
Reason SQL0970N The system attempted to write to a read-only file.
Reason SQL2563W The Restore process has completed successfully. However one or more
 table spaces from the backup were not restored.
```

O RDS para Db2 não permite armazenamento não automático. Converta o armazenamento não automático em automático e chame `rdsadmin.restore_database` novamente. Consulte mais informações em Converting a nonautomatic storage database to use automatic storage na documentação do IBM Db2.

Bancos de dados com armazenamento não automático de SMS exigem restauração manual. Se o seu banco de dados tiver armazenamento não automático de SMS, entre em contato com o AWS Support.

Consulte informações sobre o armazenamento não automático e migrações únicas em the section called “Espaços de tabela de armazenamento não automáticos durante a migração”.

Erros rdsadmin.update_db_param

O erro a seguir pode ocorrer quando o procedimento armazenado the section called “rdsadmin.update_db_param” é chamado.

Erro	Mensagem de erro
Parameter not supported or modifiable	`SQL0438N Application raised error or warning with diagnostic text: "Parameter is either not supported or not modifiable to customers". SQLSTATE=99993`

Parâmetro incompatível ou não modificável

A mensagem de erro a seguir indica que você tentou modificar um parâmetro de configuração de banco de dados que não é compatível ou não é modificável.

```
SQL0438N Application raised error or warning with diagnostic text: "Parameter
is either not supported or not modifiable to customers". SQLSTATE=99993
```

É possível ver quais parâmetros são modificáveis visualizando os grupos de parâmetros. Para obter mais informações, consulte the section called “Visualizar valores de parâmetros para um grupo de parâmetros de banco de dados”.

Amazon RDS para MariaDB

O Amazon RDS é compatível com várias versões do MariaDB para instâncias de banco de dados. Para ter informações completas sobre as versões compatíveis, consulte MariaDB em versões do Amazon RDS.

Para criar uma instância de banco de dados do MariaDB, use as ferramentas de gerenciamento ou interfaces do Amazon RDS. Depois, você pode usar as ferramentas do Amazon RDS para realizar ações de gerenciamento para a instância de banco de dados. Essas ações incluem o seguinte:

- Reconfigurar ou redimensionar a instância de banco de dados
- Autorizar conexões com a instância de banco de dados
- Criar e restaurar a partir de backups ou snapshots
- Criar secundários de multi-AZ
- Criar réplicas de leitura
- Monitorar a performance de sua instância de banco de dados

Para armazenar e acessar os dados em sua instância de banco de dados, use utilitários e aplicações padrão do MariaDB.

O MariaDB está disponível em todas as Regiões da AWS. Para obter mais informações sobre o Regiões da AWS, consulte Regiões, zonas de disponibilidade e Local Zones.

Você pode usar bancos de dados do Amazon RDS para MariaDB para criar aplicativos em conformidade com a norma HIPAA. É possível armazenar informações relacionadas à saúde, inclusive informações de saúde protegidas (PHI), sob um Acordo de associado comercial (BAA) com AWS. Para obter mais informações, consulte Compatibilidade com a HIPAA. AWS Os serviços no escopo foram totalmente avaliados por um auditor terceirizado e resultam em uma certificação, declaração de compatibilidade ou Authority to Operate (ATO). Para obter mais informações, consulte Serviços da AWS no escopo por programa de conformidade.

Antes de criar uma instância de banco de dados, conclua as etapas em Configuração do ambiente do Amazon RDS. Ao criar uma instância de banco de dados, o usuário principal do RDS obtém privilégios de DBA, com algumas limitações. Use essa conta para tarefas administrativas, como a criação de contas de banco de dados adicionais.

Você pode criar o seguinte:

- Instâncias de banco de dados
- DB snapshots
- Restaurações point-in-time
- Backups automatizados
- Backups manuais

Você pode usar instâncias de banco de dados executando o MariaDB em uma nuvem privada virtual (VPC) com base na Amazon VPC. Também é possível adicionar recursos à sua instância de banco de dados do MariaDB ativando várias opções. O Amazon RDS é compatível com implantações multi-AZ para MariaDB como uma solução de failover de alta disponibilidade.

> **Important**
>
> Para oferecer uma experiência de serviço gerenciada, o Amazon RDS não fornece acesso ao shell para as instâncias de banco de dados. Também restringe o acesso a determinados procedimentos e tabelas do sistema que necessitam de privilégios avançados. É possível acessar seu banco de dados usando clientes SQL padrão, como o cliente mysql. No entanto, não é possível acessar o host diretamente usando Telnet ou Secure Shell (SSH).

Tópicos

- Suporte ao recurso MariaDB no Amazon RDS
- MariaDB em versões do Amazon RDS
- Conectar-se à instância de banco de dados do MariaDB
- Proteger as conexões de instância de banco de dados do MariaDB
- Melhorar a performance das consultas para o RDS para MariaDB com leituras otimizadas do Amazon RDS
- Melhorar a performance das gravações com gravações otimizadas pelo Amazon RDS para MariaDB
- Atualizações do mecanismo de banco de dados do MariaDB
- Atualizar uma versão do mecanismo de snapshot de banco de dados do MariaDB
- Importar dados para uma instância de banco de dados do Amazon RDS para MariaDB
- Como trabalhar com a replicação do MariaDB no Amazon RDS

- Opções para o mecanismo de banco de dados do MariaDB
- Parâmetros para MariaDB
- Migrar dados de um snapshot de banco de dados do MySQL para uma instância de banco de dados do MariaDB
- Referência do MariaDB no SQL do Amazon RDS
- Fuso horário local para as instâncias de banco de dados do MariaDB
- Limitações e problemas conhecidos do RDS para MariaDB

Suporte ao recurso MariaDB no Amazon RDS

O RDS para MariaDB é compatível com os recursos do MariaDB. Alguns recursos podem ter suporte limitado ou privilégios restritos.

Você pode filtrar novos recursos do Amazon RDS na página What's New with Database? (Novidades sobre bancos de dados). Em Products (Produtos), escolha Amazon RDS. Em seguida, pesquise usando palavras-chave como **`MariaDB 2023`**.

> **Note**
>
> As listas a seguir não são completas.

Consulte mais informações sobre o suporte a recursos do MariaDB no Amazon RDS nos tópicos a seguir.

Tópicos

- Mecanismos de armazenamento compatíveis com MariaDB no Amazon RDS
- Aquecimento de cache do MariaDB no Amazon RDS
- Recursos do MariaDB incompatíveis com o Amazon RDS

Suporte a recursos do MariaDB nas versões principais do Amazon RDS para MariaDB

Nas seções a seguir, localize informações sobre compatibilidade de recursos do MariaDB nas versões principais do Amazon RDS para MariaDB:

Tópicos

- Suporte para MariaDB 11.4 no Amazon RDS
- Compatibilidade do MariaDB 10.11 com o Amazon RDS
- Compatibilidade do MariaDB 10.6 com o Amazon RDS
- Suporte para MariaDB 10.5 no Amazon RDS
- Suporte para MariaDB 10.4 no Amazon RDS

Para obter informações sobre versões secundárias compatíveis do Amazon RDS para MariaDB, consulte MariaDB em versões do Amazon RDS.

Suporte para MariaDB 11.4 no Amazon RDS

O Amazon RDS oferece suporte aos novos recursos descritos a seguir para instâncias de banco de dados que executam o MariaDB versão 11.4 ou posterior.

- Biblioteca criptográfica: o RDS para MariaDB substituiu o OpenSSL pelo AWS Libcrypto (AWS-LC), que tem certificação da norma FIPS 140-3.
- Simple Password Check Plugin: é possível usar o Simple Password Check Plugin do MariaDB para conferir se uma senha contém pelo menos um número específico de caracteres de determinado tipo. Para ter mais informações, consulte the section called “Plug-in de validação de senhas”.
- Cracklib Password Check Plugin: é possível usar o Cracklib Password Check Plugin do MariaDB para conferir a força das novas senhas. Para ter mais informações, consulte the section called “Plug-in de validação de senhas”.
- Aprimoramentos do InnoDB: os aprimoramentos abaixo incluem os seguintes itens:
  - O buffer de alteração foi removido. Para ter mais informações, consulte InnoDB Change Buffering.
  - A desfragmentação do InnoDB foi removida. Para ter mais informações, consulte InnoDB Defragmentation.
- Novo privilégio: o usuário administrador agora também tem o privilégio `SHOW CREATE ROUTINE`. Esse privilégio permite que o beneficiário visualize a instrução de definição `SHOW CREATE` de uma rotina pertencente a outro usuário. Para ter mais informações, consulte Database Privileges.
- Melhoria na replicação: as instâncias de banco de dados do MariaDB versão 11.4 oferecem suporte à indexação de logs binários. É possível criar um índice GTID para cada arquivo de log binário. Esses índices melhoram o desempenho da replicação ao reduzir o tempo necessário para localizar um GTID. Para ter mais informações, consulte Binlog Indexing.

- Parâmetros descontinuados ou removidos: os seguintes parâmetros foram descontinuados ou removidos para instâncias de banco de dados do MariaDB versão 11.4:
  - `engine_condition_pushdown` foi removido de optimizer_switch
  - innodb_change_buffer_max_size
  - innodb_defragment
  - `TLSv1.0` e `TLSv1.1` foram removidos de tls_version
- Novos valores padrão para um parâmetro: o valor padrão do parâmetro innodb_undo_tablespaces foi alterado de `0` para `3`.
- Novos valores padrão para parâmetros: os seguintes parâmetros têm novos valores padrão para instâncias de banco de dados do MariaDB versão 11.4:
  - Os valores válidos para o parâmetro binlog_row_image agora incluem `FULL_NODUP`.
  - Os valores válidos para o parâmetro OLD_MODE agora incluem `NO_NULL_COLLATION_IDS`.
- Novos parâmetros: os seguintes parâmetros são novos para instâncias de banco de dados do MariaDB versão 11.4:
  - O parâmetro transaction_isolation substitui o parâmetro tx_isolation.
  - O parâmetro transaction_read_only substitui o parâmetro tx_read_only.
  - O parâmetro block_encryption_mode define o modo de criptografia de bloco padrão para as funções AES_ENCRYPT() e AES_DECRYPT().
  - O parâmetro character_set_collations define substituições para agrupamentos padrão de conjuntos de caracteres.
  - Os parâmetros binlog_gtid_index, binlog_gtid_index_page_size e binlog_gtid_index_span_min definem as propriedades do índice GTID do log binário. Para ter mais informações, consulte Binlog Indexing.

Para conferir uma lista de todos os recursos do MariaDB 11.4 com a respectiva documentação, consulte Changes and improvements in MariaDB 11.4 e Release notes - MariaDB 11.4 series no site do MariaDB.

Para conferir uma lista de recursos não compatíveis, consulte Recursos do MariaDB incompatíveis com o Amazon RDS.

Compatibilidade do MariaDB 10.11 com o Amazon RDS

O Amazon RDS é compatível com estes novos recursos nas instâncias de banco de dados que executam o MariaDB versão 10.11 ou posterior.

- Plug-in de verificação de reutilização de senha: você pode usar o plug-in de verificação de reutilização de senha do MariaDB para impedir que os usuários reutilizem senhas e definir o período de retenção das senhas. Para obter mais informações, consulte Plug-in de verificação de reutilização de senha.
- Autorização GRANT TO PUBLIC: você pode conceder privilégios a todos os usuários que têm acesso ao seu servidor. Para obter mais informações, consulte GRANT TO PUBLIC.
- Separação dos privilégios SUPER e READ ONLY ADMIN: você pode remover os privilégios READ ONLY ADMIN de todos os usuários, mesmo aqueles que anteriormente tinham privilégios SUPER.
- Segurança: agora você pode definir a opção `--ssl` como padrão para seu cliente MariaDB. O MariaDB não desabilita mais silenciosamente o SSL se a configuração estiver incorreta.
- Comandos e funções SQL: agora você pode usar o comando `SHOW ANALYZE FORMAT=JSON` e as funções `ROW_NUMBER`, `SFORMAT` e `RANDOM_BYTES`. `SFORMAT` permite a formatação de strings e é habilitado por padrão. Você pode converter partição em tabela e tabela em partição usando um único comando. Há também várias melhorias em torno das funções `JSON_*()`. As funções `DES_ENCRYPT` e `DES_DECRYPT` foram descontinuadas na versão 10.10 e posterior. Para obter mais informações, consulte SFORMAT.
- Aprimoramentos do InnoDB: os aprimoramentos abaixo incluem os seguintes itens:
  - Melhorias na performance do redo log para reduzir a amplificação de gravação e melhorar a simultaneidade.
  - A possibilidade de alterar o espaço de tabela undo sem reinicializar o diretório de dados. Esse aprimoramento reduz as despesas operacionais indiretas do ambiente de gerenciamento. É necessário reiniciar, mas não é preciso reinicializar após a alteração do espaço de tabela undo.
  - Suporte para `CHECK TABLE … EXTENDED` e para índices descendentes internamente.
  - Melhorias na inserção em massa.
- Alterações no log binário: essas mudanças incluem os seguintes itens:
  - Registro em log de `ALTER` em duas fases para diminuir a latência da replicação. O parâmetro `binlog_alter_two_phase` é desabilitado por padrão, mas pode ser habilitado por meio de grupos de parâmetros.
  - Registro em log de `explicit_defaults_for_timestamp`.
  - Não há mais registro em log de `INCIDENT_EVENT` quando a transação pode ser revertida com segurança.
- Melhorias na replicação: as instâncias de banco de dados do MariaDB versão 10.11 usam a replicação GTID por padrão quando a principal a comporta. Além disso, `Seconds_Behind_Master` é mais preciso.

- Clientes: você pode usar novas opções de linha de comando para `mysqlbinlog` e `mariadb-dump`. Você pode usar `mariadb-dump` para despejar e restaurar dados históricos.
- Versionamento do sistema: você pode modificar o histórico. O MariaDB cria automaticamente novas partições.
- DDL atômico: `CREATE OR REPLACE` agora é atômico. Se a instrução não for bem-sucedida, ela será totalmente revertida.
- Gravação de redo log: o redo log grava de forma assíncrona.
- Funções armazenadas: as funções armazenadas agora comportam os mesmos parâmetros `IN`, `OUT` e `INOUT` dos procedimentos armazenados.
- Parâmetros obsoletos ou removidos: os seguintes parâmetros foram descontinuados ou removidos para instâncias de banco de dados do MariaDB versão 10.11:
  - innodb_change_buffering
  - innodb_disallow_writes
  - innodb_log_write_ahead_size
  - innodb_prefix_index_cluster_optimization
  - keep_files_on_create
  - old
- Parâmetros dinâmicos: os seguintes parâmetros agora são dinâmicos para instâncias de banco de dados do MariaDB versão 10.11:
  - innodb_log_file_size
  - innodb_write_io_threads
  - innodb_read_io_threads
- Novos valores padrão para parâmetros: os seguintes parâmetros têm novos valores padrão para instâncias de banco de dados MariaDB versão 10.11:
  - O valor padrão do parâmetro explicit_defaults_for_timestamp foi alterado de `OFF` para `ON`.
  - O valor padrão do parâmetro optimizer_prune_levelfoi alterado de `1` para `2`.
- Novos valores padrão para parâmetros: os seguintes parâmetros têm novos valores padrão para instâncias de banco de dados MariaDB versão 10.11:
  - Os valores válidos para o parâmetro old foram mesclados com aqueles para o parâmetro old-mode.
  - Os valores válidos para o parâmetro histogram_type agora incluem `JSON_HB`.

  - O intervalo de valores válido para o parâmetro innodb_log_buffer_size agora vai de `262144` a `4294967295` (de 256 KB a 4.096 MB).
  - O intervalo de valores válido para o parâmetro innodb_log_file_size agora vai de `4194304` a `512GB` (de 4 MB a 512 GB).
  - Os valores válidos para o parâmetro optimizer_prune_level agora incluem 2.
- Novos parâmetros: os seguintes parâmetros agora são dinâmicos para instâncias de banco de dados do MariaDB versão 10.11:
  - O parâmetro binlog_alter_two_phase pode melhorar a performance da replicação.
  - O parâmetro log_slow_min_examined_row_limit pode melhorar a performance.
  - Os parâmetros log_slow_query e arquivo_log_slow_query são aliases de `slow_query_log` e `slow_query_log_file`, respectivamente.
  - optimizer_extra_pruning_depth
  - system_versioning_insert_history

Para conferir uma lista de todos os recursos do MariaDB 10.11 com a respectiva documentação, consulte Changes and improvements in MariaDB 10.11 e Release notes - MariaDB 10.11 series no site do MariaDB.

Para conferir uma lista de recursos não compatíveis, consulte Recursos do MariaDB incompatíveis com o Amazon RDS.

Compatibilidade do MariaDB 10.6 com o Amazon RDS

O Amazon RDS é compatível com estes novos recursos nas instâncias de banco de dados que executam o MariaDB versão 10.6 ou posterior:

- Mecanismo de armazenamento do MyRocks: você pode usar o mecanismo de armazenamento MyRocks com o RDS para MariaDB para otimizar o consumo de armazenamento de suas aplicações Web de alta performance com uso intenso de gravação. Para obter mais informações, consulte Mecanismos de armazenamento compatíveis com MariaDB no Amazon RDS e MyRocks.
- Autenticação de banco de dados AWS Identity and Access Management (IAM): você pode usar a autenticação de banco de dados do IAM para melhorar a segurança e o gerenciamento central de conexões com suas instâncias de banco de dados do MariaDB. Para ter mais informações, consulte Autenticação do banco de dados do IAM para MariaDB, MySQL e PostgreSQL.

- Opções de atualização: agora você pode atualizar para o RDS para MariaDB versão 10.6 de qualquer versão principal anterior (10.3, 10.4, 10.5). Também é possível restaurar um snapshot de uma instância de banco de dados do MySQL 5.6 ou 5.7 existente para uma instância do MariaDB 10.6. Para ter mais informações, consulte Atualizações do mecanismo de banco de dados do MariaDB.
- Replicação atrasada: já é possível definir um período configurável pelo qual uma réplica de leitura fica atrasada em relação ao banco de dados de origem. Em uma configuração de replicação padrão do MariaDB, há um atraso mínimo de replicação entre a origem e a réplica. Com a replicação atrasada, é possível definir um atraso intencional como estratégia para a recuperação de desastres. Para obter mais informações, consulteConfigurar a replicação atrasada com o MariaDB
- Compatibilidade com o Oracle PL/SQL: usando o RDS para MariaDB versão 10.6, você pode migrar mais facilmente suas aplicações Oracle herdadas para o Amazon RDS. Para obter mais informações, consulte SQL_MODE=ORACLE.
- DDL atômico: suas instruções Dynamic Data Language (DDL – Linguagem de dados dinâmica) podem ser relativamente seguras contra falhas com o RDS para MariaDB versão 10.6. As instruções `CREATE TABLE`, `ALTER TABLE`, `RENAME TABLE`, `DROP TABLE`, `DROP DATABASE` e outras instruções DDL relacionadas agora são atômicas. Se a instrução não for bem-sucedida, ela será totalmente revertida. Para obter mais informações, consulte Atomic DDL (DDL atômica).
- Outros aprimoramentos: esses aprimoramentos incluem uma função `JSON_TABLE` para transformar dados JSON em formato relacional dentro do SQL e carregamento de dados de tabela vazia mais rápido com o InnoDB. Eles também incluem um novo `sys_schema` para análise e solução de problemas, aprimoramento do otimizador para ignorar índices não utilizados e melhorias de performance. Para obter mais informações, consulte JSON_TABLE.
- Novos valores padrão para parâmetros – os seguintes parâmetros têm novos valores padrão para instâncias de banco de dados MariaDB versão 10.6:
  - O valor padrão dos seguintes parâmetros foi alterado de `utf8` para `utf8mb3`:
    - character_set_client
    - character_set_connection
    - character_set_results
    - character_set_system

    Embora os valores padrões tenham sido alterados por esses parâmetros, não há alteração funcional. Para obter mais informações, consulte Supported Character Sets and Collations (Conjuntos e agrupamentos de caracteres compatíveis) na documentação do MariaDB.

- O valor padrão do parâmetro collation_connection foi alterado de `utf8_general_ci` para `utf8mb3_general_ci`. Embora o valor padrão tenha sido alterado para esse parâmetro, não há alteração funcional.
- O valor padrão do parâmetro old_mode mudou de não definido para `UTF8_IS_UTF8MB3`. Embora o valor padrão tenha sido alterado para esse parâmetro, não há alteração funcional.

Para obter uma lista de todos os recursos do MariaDB 10.6 e sua respectiva documentação, consulte Changes and improvements in MariaDB 10.6 (Alterações e melhorias no MariaDB 10.6) e Release notes - MariaDB 10.6 series (Notas de lançamento - série MariaDB 10.6) no site do MariaDB.

Para conferir uma lista de recursos não compatíveis, consulte Recursos do MariaDB incompatíveis com o Amazon RDS.

Suporte para MariaDB 10.5 no Amazon RDS

O Amazon RDS oferece suporte a estes novos recursos nas instâncias de banco de dados que executam o MariaDB versão 10.5 ou posterior:

- Melhorias do InnoDB– o MariaDB versão 10.5 inclui melhorias no InnoDB. Para obter mais informações, consulte InnoDB: Performance Improvements etc. (InnoDB: melhorias na performance, etc.) na documentação do MariaDB.
- Atualizações do esquema de performance – o MariaDB versão 10.5 inclui atualizações do esquema de performance. Para obter mais informações, consulte Performance Schema Updates to Match MySQL 5.7 Instrumentation and Tables (Atualizações do esquema de performance correspondentes a instrumentação e tabelas do MariaDB).
- Um arquivo no redo log do InnoDB – nas versões do MariaDB anteriores à versão 10.5, o valor do parâmetro `innodb_log_files_in_group` foi definido como 2. No MariaDB versão 10.5, o valor desse parâmetro é definido como 1.

  Se você estiver atualizando de uma versão anterior para o MariaDB versão 10.5 e não modificar os parâmetros, o valor do parâmetro `innodb_log_file_size` será inalterado. No entanto, ele se aplica a um arquivo de log em vez de dois. O resultado é que a instância de banco de dados MariaDB versão 10.5 atualizada usa metade do tamanho do redo log que estava usando antes da atualização. Essa mudança pode ter um impacto notável na performance. Para resolver esse problema, você pode dobrar o valor do parâmetro `innodb_log_file_size`. Para obter informações sobre como modificar parâmetros, consulte Modificar parâmetros em um grupo de parâmetros de banco de dados no Amazon RDS.

- Comando SHOW SLAVE STATUS sem suporte – em versões do MariaDB anteriores à versão 10.5, o comando `SHOW SLAVE STATUS` exigia o privilégio `REPLICATION SLAVE`. Na versão 10.5 do MariaDB, o comando `SHOW REPLICA STATUS` equivalente requer o privilégio `REPLICATION REPLICA ADMIN`. Esse novo privilégio não é concedido ao usuário mestre do RDS.

  Em vez de usar o comando `SHOW REPLICA STATUS`, execute o novo procedimento `mysql.rds_replica_status` armazenado para retornar informações semelhantes. Para obter mais informações, consulte mysql.rds_replica_status.
- Comando SHOW RELAYLOG EVENTS sem suporte – em versões do MariaDB anteriores à versão 10.5, o comando `SHOW RELAYLOG EVENTS` exigiu o privilégio `REPLICATION SLAVE`. Na versão 10.5 do MariaDB, esse comando requer o privilégio `REPLICATION REPLICA ADMIN`. Esse novo privilégio não é concedido ao usuário mestre do RDS.
- Novos valores padrão para parâmetros – os seguintes parâmetros têm novos valores padrão para instâncias de banco de dados MariaDB versão 10.5:
  - O valor padrão do parâmetro max_connections foi alterado para `LEAST({DBInstanceClassMemory/25165760},12000)`. Para obter informações sobre a função do parâmetro `LEAST`, consulte Funções de parâmetros de banco de dados.
  - O valor padrão do parâmetro innodb_adaptive_hash_index foi alterado para `OFF` (`0`).
  - O valor padrão do parâmetro innodb_checksum_algorithm foi alterado para `full_crc32`.
  - O valor padrão do parâmetro innodb_log_file_size foi alterado para 2 GB.

Para obter uma lista de todos os recursos do MariaDB 10.5 e suas respectivas documentações, consulte Changes and improvements in MariaDB 10.5 e Release notes - MariaDB 10.5 series no site do MariaDB.

Para ver uma lista de recursos não compatíveis, consulte Recursos do MariaDB incompatíveis com o Amazon RDS.

Suporte para MariaDB 10.4 no Amazon RDS

O Amazon RDS oferece suporte a estes novos recursos nas instâncias de banco de dados executando o MariaDB versão 10.4 ou posterior:

- Melhorias de segurança da conta do usuário – Expiração da senha e melhorias no bloqueio de conta
- Melhorias do otimizador – recurso Optimizer Trace

- Melhorias do InnoDB – Suporte instantâneo a DROP COLUMN e extensão VARCHAR instantânea para ROW_FORMAT=DYNAMIC e ROW_FORMAT=COMPACT
- Novos parâmetros – Incluindo tcp_nodedelay, tls_version e gtid_cleanup_batch_size

Para obter uma lista de todos os recursos do MariaDB 10.4 e suas respectivas documentações, consulte Changes and improvements in MariaDB 10.4 e Release notes - MariaDB 10.4 series no site do MariaDB.

Para ver uma lista de recursos não compatíveis, consulte Recursos do MariaDB incompatíveis com o Amazon RDS.

Mecanismos de armazenamento compatíveis com MariaDB no Amazon RDS

O RDS para MariaDB é compatível com os seguintes mecanismos de armazenamento.

Tópicos

- O mecanismo de armazenamento InnoDB
- O mecanismo de armazenamento MyRocks

Atualmente, não há suporte a outros mecanismos de armazenamento para o RDS para MariaDB.

O mecanismo de armazenamento InnoDB

Embora o MariaDB suporte vários mecanismos de armazenamento com recursos variados, nem todos eles são otimizados para durabilidade de dados e recuperação. O InnoDB é o mecanismo de armazenamento recomendado para instâncias de banco de dados MariaDB no Amazon RDS. Os recursos de restauração point-in-time e de snapshot do Amazon RDS exigem um mecanismo de armazenamento de recuperação de falha e são compatíveis somente com o mecanismo de armazenamento para a versão MariaDB.

Para obter mais informações, consulte InnoDB.

O mecanismo de armazenamento MyRocks

O mecanismo de armazenamento MyRocks está disponível no RDS para MariaDB versão 10.6 e posteriores. Antes de usar o mecanismo de armazenamento MyRocks em um banco de dados

de produção, recomendamos que você execute testes e benchmarking completos para verificar quaisquer benefícios potenciais em relação ao InnoDB para seu caso de uso.

O grupo de parâmetros padrão para o MariaDB versão 10.6 inclui parâmetros do MyRocks. Para ter mais informações, consulte Parâmetros para MariaDB e Grupos de parâmetros para Amazon RDS.

Para criar uma tabela que usa o mecanismo de armazenamento MyRocks, especifique `ENGINE=RocksDB` na instrução `CREATE TABLE`. O exemplo a seguir cria uma tabela que usa o mecanismo de armazenamento MyRocks.

```
CREATE TABLE test (a INT NOT NULL, b CHAR(10)) ENGINE=RocksDB;
```

É altamente recomendável não executar transações que envolvam as tabelas do InnoDB e do MyRocks. O MariaDB não garante ACID (atomicidade, consistência, isolamento, durabilidade) para transações em todos os mecanismos de armazenamento. Embora seja possível ter tabelas InnoDB e MyRocks em uma instância de banco de dados, não recomendamos essa abordagem, exceto durante uma migração de um mecanismo de armazenamento para o outro. Quando há tabelas InnoDB e MyRocks em uma instância de banco de dados, cada mecanismo de armazenamento tem seu próprio grupo de buffers, o que pode causar degradação da performance.

O MyRocks não é compatível com o isolamento de `SERIALIZABLE` ou bloqueios de lacunas. Portanto, geralmente você não pode usar o MyRocks com replicação baseada em instruções. Para obter mais informações, consulte MyRocks and Replication (MyRocks e replicação).

Atualmente, você só pode modificar os seguintes parâmetros do MyRocks:

- `rocksdb_block_cache_size`
- `rocksdb_bulk_load`
- `rocksdb_bulk_load_size`
- `rocksdb_deadlock_detect`
- `rocksdb_deadlock_detect_depth`
- `rocksdb_max_latest_deadlocks`

O mecanismo de armazenamento MyRocks e o mecanismo de armazenamento InnoDB podem competir pela memória com base nas configurações dos parâmetros `rocksdb_block_cache_size` e `innodb_buffer_pool_size`. Em alguns casos, você pode ter a intenção de usar apenas o mecanismo de armazenamento MyRocks em uma instância de banco de

dados específica. Se for o caso, recomendamos definir o parâmetro `innodb_buffer_pool_size` `minimal` para um valor mínimo e o `rocksdb_block_cache_size` para o valor mais alto possível.

Você pode acessar os arquivos de log do MyRocks usando as operações `DescribeDBLogFiles` e `DownloadDBLogFilePortion`.

Para obter mais informações sobre o MyRocks, consulte MyRocks no site do MariaDB.

Aquecimento de cache do MariaDB no Amazon RDS

O aquecimento de cache do InnoDB pode fornecer ganhos de performance para sua instância de banco de dados do MariaDB salvando o estado atual do grupo de buffer quando a instância de banco de dados é desativada e, em seguida, recarregando o grupo de buffer a partir das informações salvas quando a instância do banco de dados for iniciada. Essa abordagem ignora a necessidade de o grupo de buffer "aquecer" do uso normal de banco de dados e, em vez disso, pré-carrega o grupo de buffer com as páginas para consultas comuns conhecidas. Para obter mais informações sobre o aquecimento de cache, consulte Despejo e restauração do grupo de buffers na documentação do MariaDB.

O aquecimento de cache está habilitado por padrão para instâncias de banco de dados do MariaDB 10.3 e posteriores. Para habilitá-lo, defina os parâmetros `innodb_buffer_pool_dump_at_shutdown` e `innodb_buffer_pool_load_at_startup` para 1 no grupo de parâmetros da sua instância de banco de dados. Mudar esses valores de parâmetro em um grupo de parâmetros afeta todas as instâncias de banco de dados do MariaDB que usam esse grupo de parâmetros. Para habilitar o aquecimento de cache para instâncias específicas de banco de dados do MariaDB, talvez seja necessário criar um grupo de parâmetros nessas instâncias de banco de dados. Para obter informações sobre grupos de parâmetros, consulte Grupos de parâmetros para Amazon RDS.

O aquecimento de cache fornece principalmente um benefício de performance para instâncias de banco de dados que usam armazenamento padrão. Se você usar o armazenamento de PIOPS, não perceberá um benefício significativo de performance normalmente.

> **Important**
>
> Se sua instância de banco de dados do MariaDB não fechar normalmente, por exemplo durante um failover, o estado do grupo de buffer não é salvo no disco. Nesse caso, o MariaDB carrega qualquer arquivo de grupo de buffer disponível quando a instância de banco de dados for reiniciada. Não há problema, mas o grupo de buffer restaurado pode

não refletir o estado mais recente do grupo de buffer antes do reinício. Para garantir que você tenha um estado recente do grupo de buffer disponível para aquecer o cache do na inicialização, recomendamos que você descarte periodicamente o grupo de buffer "sob demanda". Você pode descartar ou carregar o grupo de buffer sob demanda.
Você pode criar um evento para descartar o grupo de buffer automaticamente e em um intervalo regular. Por exemplo, a seguinte declaração cria um evento chamado `periodic_buffer_pool_dump` que descarta o grupo de buffer de hora em hora.

```
CREATE EVENT periodic_buffer_pool_dump
   ON SCHEDULE EVERY 1 HOUR
   DO CALL mysql.rds_innodb_buffer_pool_dump_now();
```

Para obter mais informações, consulte Eventos na documentação do MariaDB.

Como descartar e carregar o grupo de buffer sob demanda

Você pode salvar e carregar o cache do sob demanda usando os seguintes procedimentos armazenados:

- Para descartar o estado atual do grupo de buffer para o disco, chame o procedimento armazenado mysql.rds_innodb_buffer_pool_dump_now.
- Para carregar o estado salvo do grupo de buffer a partir do disco, chame o procedimento armazenado mysql.rds_innodb_buffer_pool_load_now.
- Para cancelar uma operação de carregamento em andamento, chame o procedimento armazenado mysql.rds_innodb_buffer_pool_load_abort.

Recursos do MariaDB incompatíveis com o Amazon RDS

Os seguintes recursos do MariaDB não têm suporte no Amazon RDS:

- Mecanismo de armazenamento do S3
- Plug-in de autenticação – GSSAPI
- Plug-in de autenticação – soquete Unix
- AWSPlugin de criptografia do Key Management
- Replicação atrasada para versões do MariaDB anteriores à 10.6

- Criptografia nativa do MariaDB em repouso para InnoDB e Aria

  Você pode habilitar a criptografia em repouso para uma instância de banco de dados do MariaDB seguindo as instruções em Criptografar recursos do Amazon RDS.
- HandlerSocket
- Tipo de tabela JSON para versões do MariaDB anteriores à 10.6
- MariaDB ColumnStore
- Cluster Galera do MariaDB
- Replicação em várias origens
- Mecanismo de armazenamento MyRocks para versões do MariaDB anteriores à 10.6
- Plug-in de validação de senha, `simple_password_check` e `cracklib_password_check` para versões do MariaDB inferiores a 11.4
- Mecanismo de armazenamento do Spider
- Mecanismo de armazenamento do Sphinx
- Mecanismo de armazenamento do TokuDB
- Atributos de objeto específicos do mecanismo de armazenamento, conforme descrito em Atributos de nova tabela/campo/índice definidos pelo mecanismo na documentação do MariaDB
- Criptografia de tabela e do espaço de tabela
- Plug-in de gerenciamento de chaves Hashicorp
- Executar duas atualizações em paralelo

Para oferecer uma experiência de serviço gerenciado, o Amazon RDS não fornece acesso ao shell para instâncias de bancos de dados e restringe o acesso a determinados procedimentos e tabelas do sistema que exigem privilégios avançados. O Amazon RDS oferece suporte a acesso a bancos de dados em uma instância de banco de dados que usa qualquer aplicação de cliente SQL padrão. O Amazon RDS não permite o acesso direto do host a uma instância de banco de dados usando Telnet, Secure Shell (SSH) ou Windows Remote Desktop Connection.

MariaDB em versões do Amazon RDS

No MariaDB, os números de versão são organizados como versão X.Y.Z. Na terminologia do Amazon RDS, X.Y representa a versão principal, e Z é o número da versão secundária. Para implementações do Amazon RDS, uma alteração da versão é considerada primária se o número da versão primária for alterado, por exemplo, da versão 10.5 para a 10.6. Uma alteração da versão será considerada secundária se apenas o número da versão secundária for alterado; por exemplo, da versão 10.6.14 para a 10.6.16.

Tópicos

- Versões secundárias do MariaDB compatíveis com o Amazon RDS
- Versões principais do MariaDB compatíveis com o Amazon RDS
- Trabalhar com o ambiente de visualização de banco de dados
- MariaDB versão 11.4 no ambiente de versão prévia do banco de dados.
- Versões obsoletas do Amazon RDS para MariaDB

Versões secundárias do MariaDB compatíveis com o Amazon RDS

Atualmente, o Amazon RDS é compatível com as versões secundárias do MariaDB a seguir.

> **Note**
>
> As datas com apenas mês e ano são aproximadas e são atualizadas com uma data exata quando ela é conhecida.

No momento, a tabela a seguir mostra as versões secundárias do MariaDB 10.11 compatíveis com o Amazon RDS.

Versão do mecanismo do MariaDB	Data de lançamento para a comunidade	Data de lançamento do RDS	Data do fim do suporte padrão para o RDS
10.11.8	16 de maio de 2024	14 de junho de 2024	Setembro de 2025

Versão do mecanismo do MariaDB	Data de lançamento para a comunidade	Data de lançamento do RDS	Data do fim do suporte padrão para o RDS
10.11.7	7 de fevereiro de 2024	26 de fevereiro de 2024	Março de 2024
10.11.6	13 de novembro de 2023	12 de dezembro de 2023	Março de 2024
10.11.5	14 de agosto de 2023	7 de setembro de 2023	Março de 2024
10.11.4	7 de junho de 2023	21 de agosto de 2023	Março de 2024

No momento, a tabela a seguir mostra as versões secundárias do MariaDB 10.6 compatíveis com o Amazon RDS.

Versão do mecanismo do MariaDB	Data de lançamento para a comunidade	Data de lançamento do RDS	Data do fim do suporte padrão para o RDS
10.6.18	16 de maio de 2024	14 de junho de 2024	Setembro de 2025
10.6.17	7 de fevereiro de 2024	26 de fevereiro de 2024	Março de 2024
10.6.16	13 de novembro de 2023	12 de dezembro de 2023	Março de 2024
10.6.15	14 de agosto de 2023	7 de setembro de 2023	Março de 2024
10.6.14	7 de junho de 2023	22 de junho de 2023	Março de 2024
10.6.13	10 de maio de 2023	15 de junho de 2023	Março de 2024

No momento, a tabela a seguir mostra as versões secundárias do MariaDB 10.5 compatíveis com o Amazon RDS.

Versão do mecanismo do MariaDB	Data de lançamento para a comunidade	Data de lançamento do RDS	Data do fim do suporte padrão para o RDS
10.5.25	16 de maio de 2024	14 de junho de 2024	Setembro de 2025
10.5.24	7 de fevereiro de 2024	26 de fevereiro de 2024	Março de 2024
10.5.23	13 de novembro de 2023	12 de dezembro de 2023	Março de 2024
10.5.22	14 de agosto de 2023	7 de setembro de 2023	Março de 2024
10.5.21	7 de junho de 2023	22 de junho de 2023	Março de 2024
10.5.20	10 de maio de 2023	15 de junho de 2023	Março de 2024

No momento, a tabela a seguir mostra as versões secundárias do MariaDB 10.4 compatíveis com o Amazon RDS.

Versão do mecanismo do MariaDB	Data de lançamento para a comunidade	Data de lançamento do RDS	Data do fim do suporte padrão para o RDS
10.4.34	16 de maio de 2024	14 de junho de 2024	Fevereiro de 2025
10.4.33	7 de fevereiro de 2024	26 de fevereiro de 2024	Fevereiro de 2025
10.4.32	13 de novembro de 2023	12 de dezembro de 2023	Fevereiro de 2025

Versão do mecanismo do MariaDB	Data de lançamento para a comunidade	Data de lançamento do RDS	Data do fim do suporte padrão para o RDS
10.4.31	14 de agosto de 2023	7 de setembro de 2023	Fevereiro de 2025
10.4.30	7 de junho de 2023	22 de junho de 2023	Fevereiro de 2025
10.4.29	10 de maio de 2023	15 de junho de 2023	Fevereiro de 2025

Você pode especificar qualquer versão compatível atualmente do MariaDB ao criar uma nova instância de banco de dados. Você pode especificar a versão principal (como o MariaDB 10.5) e qualquer versão secundária com suporte para a versão principal especificada. Se nenhuma versão for especificada, o Amazon RDS usará uma versão compatível por padrão, geralmente a versão mais recente. Se uma versão principal for especificada, mas uma versão secundária não, o Amazon RDS usará por padrão um lançamento recente da versão principal que você especificou. Para ver uma lista de versões com suporte, bem como os padrões para instâncias de banco de dados recém-criadas, use o comando da `describe-db-engine-versions` AWS CLI.

Por exemplo, para listar as versões de mecanismo compatíveis com o RDS para MariaDB, execute o comando da CLI a seguir:

```
aws rds describe-db-engine-versions --engine mariadb --query "*[].
{Engine:Engine,EngineVersion:EngineVersion}" --output text
```

A versão padrão do MariaDB pode variar de acordo com a Região da AWS. Para criar uma instância de banco de dados com uma versão secundária específica, especifique a versão secundária durante a criação da instância de banco de dados. É possível determinar a versão secundária padrão para uma Região da AWS usando o seguinte comando da AWS CLI:

```
aws rds describe-db-engine-versions --default-only --engine mariadb
 --engine-version major-engine-version --region region --query "*[].
{Engine:Engine,EngineVersion:EngineVersion}" --output text
```

Substitua *`major-engine-version`* pela versão principal do mecanismo e substitua *`region`* pela Região da AWS. Por exemplo, o comando da AWS CLI a seguir retorna a versão secundária

padrão do mecanismo do MariaDB para a versão principal 10.5 e a Região da AWS Oeste dos EUA (Oregon) (us-west-2):

```
aws rds describe-db-engine-versions --default-only --engine mariadb --engine-version
 10.5 --region us-west-2 --query "*[].{Engine:Engine,EngineVersion:EngineVersion}" --
output text
```

Versões principais do MariaDB compatíveis com o Amazon RDS

As principais versões do RDS para MariaDB continuam disponíveis pelo menos até o fim da vida útil da comunidade para a versão da comunidade correspondente. É possível usar as seguintes datas para planejar seus ciclos de teste e atualização. Se a Amazon estender o suporte para uma versão do RDS para MariaDB por mais tempo do que o planejado originalmente, planejamos atualizar essa tabela para refletir a data posterior.

> **Note**
>
> As datas com apenas mês e ano são aproximadas e são atualizadas com uma data exata quando ela é conhecida.

Versão principal do MariaDB	Data de lançamento para a comunidade	Data de lançamento do RDS	Data do fim da vida útil para a comunidade	Data do fim do suporte padrão para o RDS
MariaDB 10.11	16 de fevereiro de 2023	21 de agosto de 2023	16 de fevereiro de 2028	Fevereiro de 2028
MariaDB 10.6	6 de julho de 2021	3 de fevereiro de 2022	6 de julho de 2026	Julho de 2026
MariaDB 10.5	24 de junho de 2020	21 de janeiro de 2021	24 de junho de 2025	Junho de 2025
MariaDB 10.4	18 de junho de 2019	6 de abril de 2020	18 de junho de 2024	Fevereiro de 2025

Trabalhar com o ambiente de visualização de banco de dados

As instâncias de banco de dados do RDS para MariaDB no ambiente de versão prévia do banco de dados são semelhantes às outras instâncias do RDS para MariaDB em termos de funcionalidade. No entanto, você não pode usar o ambiente Database Preview para cargas de trabalho de produção.

Os ambientes prévios têm as seguintes limitações:

- Todas as instâncias de banco de dados são excluídas 60 dias após serem criadas, juntamente com backups e snapshots.
- Você só pode usar o SSD de uso geral e o armazenamento em SSD IOPS provisionadas.
- Você não pode obter ajuda do Suporte para instâncias de banco de dados. Em vez disso, você pode postar suas perguntas na comunidade de perguntas e respostas gerenciadas do , AWSre:Post.
- Não é possível copiar um snapshot de uma instância de banco de dados para um ambiente de produção.

As opções a seguir são compatíveis com a visualização.

- Você pode criar instâncias de banco de dados usando as classes de instância de banco de dados db.m6i, db.r6i, db.m6g, db.m5, db.t3, db.r6g e db.r5. Para obter mais informações sobre as classes de instância do RDS, consulte Classes de instâncias de banco de dados .
- É possível usar implantações single-AZ e multi-AZ.
- É possível usar funções padrão de despejo e de carregamento do MariaDB para exportar ou importar bancos de dados para o ambiente de versão prévia do banco de dados.

Atributos não compatíveis com o ambiente de visualização de banco de dados

Os atributos a seguir não estão disponíveis no ambiente de visualização de banco de dados:

- Cópia de snapshots entre regiões
- Réplicas de leitura entre regiões
- RDS Proxy

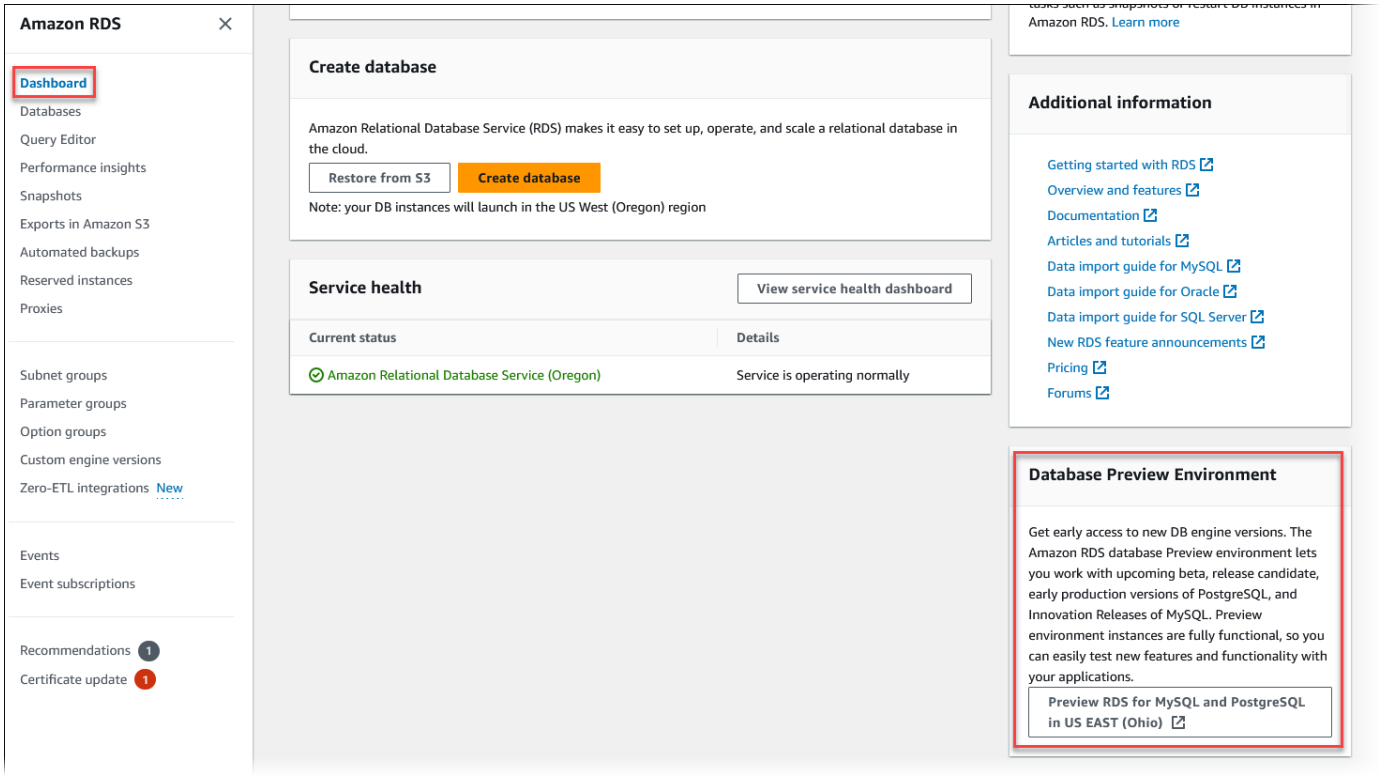
Criar uma nova instância de banco de dados no ambiente de visualização de banco de dados

Você pode criar uma instância de banco de dados no ambiente Database Preview usando o AWS Management ConsoleAWS CLI, a ou a API do RDS.

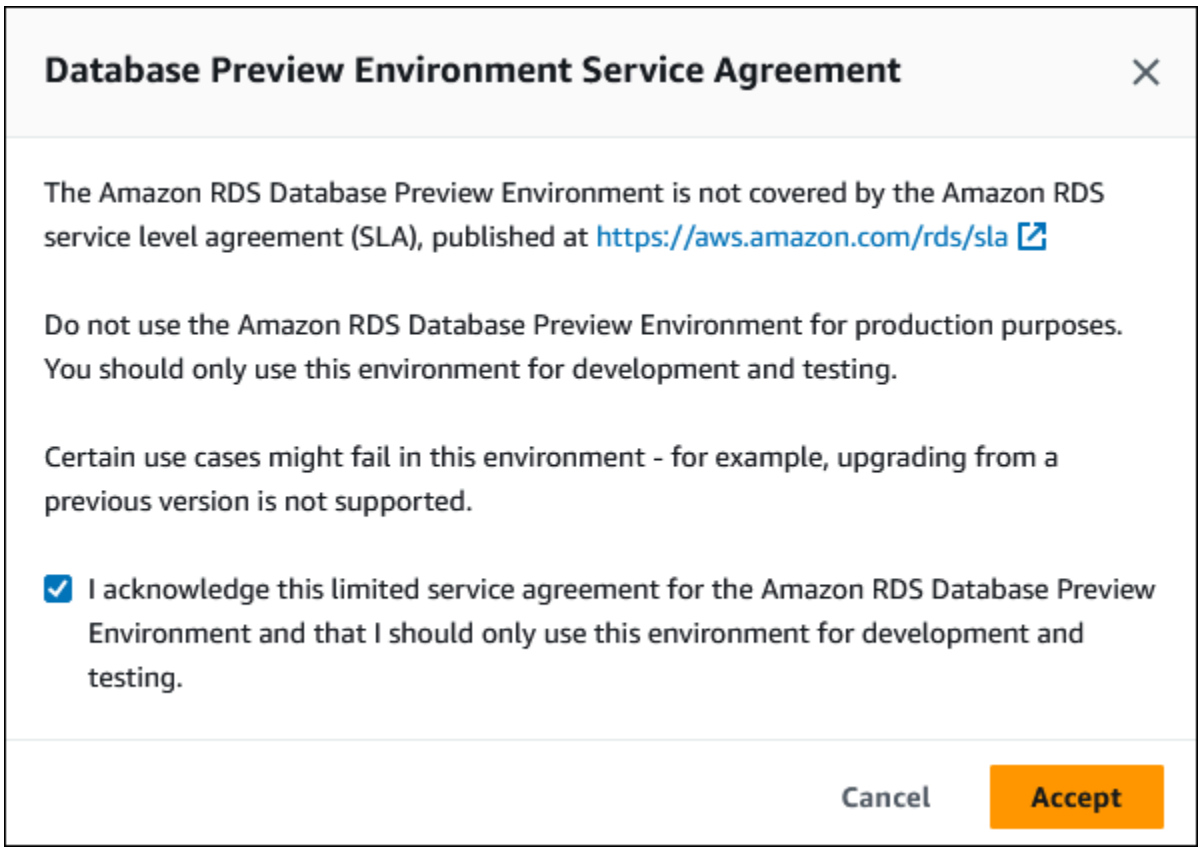
Console

Para criar uma instância de banco de dados no ambiente de visualização de banco de dados

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. Escolha Dashboard (Painel) no painel de navegação.
3. Na página Dashboard (Painel), localize a seção Database Preview Environment (Ambiente de visualização do banco de dados) na página Dashboard (Painel), conforme mostrado na imagem a seguir.



Você também pode navegar diretamente para o Ambiente de visualização do banco de dados. Antes de continuar, você deve reconhecer e aceitar as limitações.



4. Para criar a instância de banco de dados do RDS para MariaDB, siga o mesmo processo de criação de qualquer instância de banco de dados do Amazon RDS. Para obter mais informações, consulte o procedimento Console em Criar uma instância de banco de dados.

AWS CLI

Para criar uma instância no ambiente de visualização do banco de dados usando a API do RDS ou a AWS CLI, use o endpoint a seguir.

```
rds-preview.us-east-2.amazonaws.com
```

Para criar a instância de banco de dados do RDS para MariaDB, siga o mesmo processo de criação de qualquer instância de banco de dados do Amazon RDS. Para obter mais informações, consulte o procedimento AWS CLI em Criar uma instância de banco de dados.

API do RDS

Para criar uma instância no ambiente de visualização do banco de dados usando a API do RDS ou a , use o endpoint a seguir.

```
rds-preview.us-east-2.amazonaws.com
```

Para criar a instância de banco de dados do RDS para MariaDB, siga o mesmo processo de criação de qualquer instância de banco de dados do Amazon RDS. Para obter mais informações, consulte o procedimento API do RDS em Criar uma instância de banco de dados.

MariaDB versão 11.4 no ambiente de versão prévia do banco de dados.

O MariaDB versão 11.4 já está disponível no ambiente de versão prévia do banco de dados do Amazon RDS. O MariaDB versão 11.4 contém várias melhorias descritas em Changes and improvements in MariaDB 11.4. É possível usar o ambiente de versão prévia do banco de dados para testar as workloads em relação a essa versão antes que ela esteja disponível em todas as Regiões da AWS para workloads de produção.

Para obter informações sobre o ambiente de visualização de banco de dados, consulte the section called “Ambiente de visualização do banco de dados”. Para acessar o ambiente de visualização do console, selecione rds-preview/.

Versões obsoletas do Amazon RDS para MariaDB

As versões 10.0, 10.1, 10.2 e 10.3 do Amazon RDS para MariaDB estão obsoletas.

Para obter informações sobre a política de defasagem do Amazon RDS para o MariaDB, consulte Perguntas frequentes sobre o Amazon RDS.

Conectar-se à instância de banco de dados do MariaDB

Depois que o Amazon RDS provisionar a sua instância de banco de dados, será possível usar qualquer utilitário ou aplicativo cliente MariaDB padrão para se conectar à instância. Na string de conexão, especifique o endereço DNS (Domain Name System) do endpoint da instância de banco de dados como o parâmetro de host. Especifique também o número da porta do endpoint da instância de banco de dados como o parâmetro da porta.

É possível se conectar a uma instância de banco de dados do Amazon RDS para MariaDB usando ferramentas como o cliente MySQL de linha de comando. Para ter mais informações sobre como usar o cliente MySQL de linha de comando, consulte mysql command-line client, na documentação do MariaDB. O Heidi é uma aplicação baseada em GUI que você pode usar para se conectar. Para ter mais informações, consulte a página Download HeidiSQL. Para obter informações sobre como instalar o MySQL (incluindo o cliente MySQL de linha de comando), consulte Installing and upgrading MySQL (Instalar e atualizar o MySQL).

A maioria das distribuições do Linux inclui o cliente MariaDB em vez do cliente Oracle MySQL. Para instalar o cliente da linha de comando do MySQL no Amazon Linux 2023, execute o seguinte comando:

```
sudo dnf install mariadb105
```

Para instalar o cliente da linha de comando do MySQL no Amazon Linux 2, execute o seguinte comando:

```
sudo yum install mariadb
```

Para instalar o cliente de linha de comando do MySQL na maioria das distribuições do Linux baseadas em DEB, execute o comando a seguir.

```
apt-get install mariadb-client
```

Para conferir a versão do cliente de linha de comando do MySQL, execute o comando a seguir:

```
mysql --version
```

Para ler a documentação da versão atual do cliente do MySQL, execute o comando a seguir.

```
man mysql
```

Para se conectar a uma instância de banco de dados de fora de uma nuvem privada virtual (VPC) baseada em Amazon VPC, a instância de banco de dados deve estar acessível publicamente. Além disso, o acesso deve ser concedido usando as regras de entrada do grupo de segurança da instância de banco de dados, e os outros requisitos devem ser atendidos. Para ter mais informações, consulte Não é possível conectar-se à instância de banco de dados do Amazon RDS.

Você pode usar a criptografia SSL em conexões com uma instância de banco de dados MariaDB. Para ter mais informações, consulte Suporte de SSL/TLS para instâncias de banco de dados do MariaDB no Amazon RDS.

Para encontrar uma instância de banco de dados do RDS para MariaDB e conectar-se a ela, consulte os tópicos a seguir.

Tópicos

- Localizar as informações de conexão para uma instância de banco de dados MariaDB
- Conectar-se pelo cliente de linha de comandos do MySQL (descriptografado) para o RDS para MariaDB
- Conectar-se ao RDS para MariaDB com o driver JDBC da AWS e o driver Python da AWS
- Solução de problemas de conexões da instância de banco de dados MariaDB

Localizar as informações de conexão para uma instância de banco de dados MariaDB

As informações de conexão de uma instância de banco de dados incluem seu endpoint, porta e um usuário de banco de dados válido, como o usuário mestre. Por exemplo, suponha que um valor de endpoint seja `mydb.123456789012.us-east-1.rds.amazonaws.com`. Nesse caso, o valor da porta é `3306`, e o usuário do banco de dados é `admin`. Com essas informações, você especifica os seguintes valores em uma string de conexão:

- Para o host, nome de host ou nome DNS, especifique `mydb.123456789012.us-east-1.rds.amazonaws.com`.
- Para a porta, especifique `3306`.
- Para usuário, especifique `admin`.

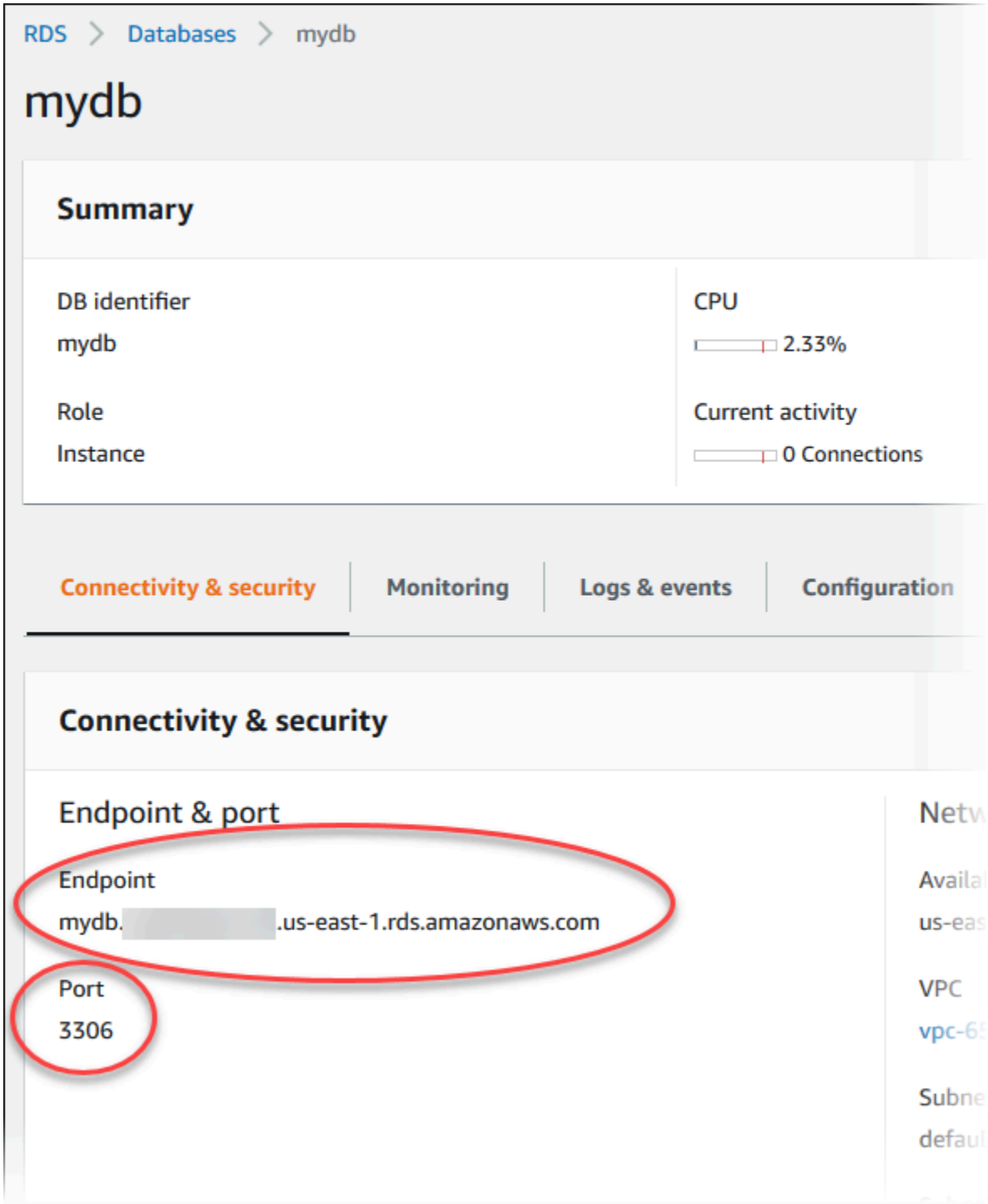
Para se conectar a uma instância de banco de dados, use qualquer cliente para o mecanismo de banco de dados do MariaDB. Por exemplo, você pode usar o cliente MySQL de linha de comando ou o MySQL Workbench.

Para localizar as informações de conexão de uma instância de banco de dados, você pode usar o AWS Management Console, comando da AWS Command Line Interface (AWS CLI) describe-db-instances ou a operação da API do Amazon RDS DescribeDBInstances para listar seus detalhes.

Console

Para encontrar as informações de conexão de uma instância de banco de dados no AWS Management Console

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Bancos de dados para exibir uma lista de suas instâncias de banco de dados.
3. Escolha o nome da instância de banco de dados MariaDB para exibir os detalhes.
4. Na guia Connectivity & security (Conectividade e segurança), copie o endpoint. Além disso, anote o número da porta. Você precisará do endpoint e do número da porta para conectar-se à instância de banco de dados.



5. Se você precisar encontrar o nome de usuário mestre, escolha a guia Configuração e visualize o valor do nome de usuário mestre .

AWS CLI

Para localizar as informações de conexão de uma instância de banco de dados MariaDB usando o AWS CLI, execute o comando describe-db-instances. Na chamada, consulte o ID da instância de banco de dados, o endpoint, a porta e o nome de usuário mestre.

Para Linux, macOS ou Unix:

```
aws rds describe-db-instances \
  --filters "Name=engine,Values=mariadb" \
  --query "*[].[DBInstanceIdentifier,Endpoint.Address,Endpoint.Port,MasterUsername]"
```

Para Windows:

```
aws rds describe-db-instances ^
  --filters "Name=engine,Values=mariadb" ^
  --query "*[].[DBInstanceIdentifier,Endpoint.Address,Endpoint.Port,MasterUsername]"
```

Sua saída deve ser similar à seguinte.

```
[
    [
        "mydb1",
        "mydb1.123456789012.us-east-1.rds.amazonaws.com",
        3306,
        "admin"
    ],
    [
        "mydb2",
        "mydb2.123456789012.us-east-1.rds.amazonaws.com",
        3306,
        "admin"
    ]
]
```

API do RDS

Para localizar as informações de conexão de uma instância de banco de dados usando a Amazon RDS API, chame a operação DescribeDBInstances . Na saída, encontre os valores para o endereço do endpoint, a porta do endpoint e o nome do usuário mestre.

Conectar-se pelo cliente de linha de comandos do MySQL (descriptografado) para o RDS para MariaDB

> **Important**
>
> Use uma conexão MySQL não criptografada apenas quando o cliente e o servidor estiverem na mesma VPC e a rede for confiável. Para obter informações sobre como usar conexões criptografadas, consulte Conectar-se à instância de banco de dados do MariaDB no Amazon RDS com SSL/TLS por meio do cliente de linha de comandos do MySQL (criptografado).

Para se conectar a uma instância de banco de dados usando o cliente MySQL de linha de comando, insira o comando a seguir em um prompt de comando em um computador cliente. Isso conecta você a um banco de dados em uma instância de banco de dados MariaDB. Substitua o nome DNS (endpoint) da instância de banco de dados *<endpoint>* e o nome de usuário mestre usado por *<mymasteruser>*. Forneça a senha mestra usada quando for solicitada uma senha.

```
mysql -h <endpoint> -P 3306 -u <mymasteruser> -p
```

Depois de digitar a senha do usuário, você verá uma saída semelhante à seguinte.

```
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 31
Server version: 10.6.10-MariaDB-log Source distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>
```

Conectar-se ao RDS para MariaDB com o driver JDBC da AWS e o driver Python da AWS

Conecte-se a instâncias de banco de dados do RDS para MariaDB com o driver JDBC da AWS e o driver Python da AWS. Para obter mais informações, consulte os tópicos a seguir.

Tópicos

- Conectar-se ao RDS para MariaDB com o driver JDBC da Amazon Web Services (AWS)
- Conectar-se ao RDS para MariaDB com o driver Python da Amazon Web Services (AWS)

Conectar-se ao RDS para MariaDB com o driver JDBC da Amazon Web Services (AWS)

O driver JDBC da Amazon Web Services (AWS) foi projetado como um wrapper JDBC avançado. Esse wrapper é complementar e amplia a funcionalidade do driver JDBC existente. O driver é compatível de forma intercambiável com os drivers MySQL Connector/J e MariaDB Connector/J da comunidade.

Para instalar o driver JDBC da AWS, anexe o arquivo .jar do driver JDBC da AWS (localizado na aplicação `CLASSPATH`) e mantenha referências ao respectivo driver da comunidade. Atualize o respectivo prefixo do URL de conexão da seguinte forma:

- `jdbc:mysql://` para `jdbc:aws-wrapper:mysql://`
- `jdbc:mariadb://` para `jdbc:aws-wrapper:mariadb://`

Consulte mais informações sobre o driver JDBC da AWS e siga as instruções para usá-lo em Amazon Web Services (AWS) JDBC Driver GitHub repository.

Conectar-se ao RDS para MariaDB com o driver Python da Amazon Web Services (AWS)

O driver Python da Amazon Web Services (AWS) foi projetado como um wrapper Python avançado. Esse wrapper é complementar e amplia a funcionalidade do driver Psycopg de código aberto. O driver Python da AWS é compatível com as versões 3.8 e posterior do Python. É possível instalar o pacote `aws-advanced-python-wrapper` usando o comando `pip`, bem como os pacotes de código aberto `psycopg`.

Para ter mais informações sobre o driver Python da AWS e instruções completas para usá-lo, consulte Amazon Web Services (AWS) Python Driver GitHub repository.

Solução de problemas de conexões da instância de banco de dados MariaDB

Duas causas comuns de falhas de conexão para uma nova instância de banco de dados são as seguintes:

- A instância de banco de dados foi criada usando um grupo de segurança que não autoriza conexões do dispositivo ou da instância do Amazon EC2 em que o aplicativo ou utilitário do MariaDB está sendo executado. A instância de banco de dados deve ter um grupo de segurança de VPC que autorize as conexões. Para ter mais informações, consulte Amazon VPC e Amazon RDS.

  É possível adicionar ou editar uma regra de entrada no grupo de segurança. Em Source (Origem), selecione My IP (Meu IP). Isso concede acesso à instância de banco de dados do endereço IP detectado no navegador.
- A instância de banco de dados foi criada usando a porta padrão 3306 e sua empresa emprega regras de firewall bloqueando conexões a essa porta a partir de dispositivos na rede de sua empresa. Para corrigir essa falha, recrie a instância com uma porta diferente.

Para ter mais informações sobre problemas de conexão, consulte Não é possível conectar-se à instância de banco de dados do Amazon RDS.

Proteger as conexões de instância de banco de dados do MariaDB

É possível gerenciar a segurança de suas instâncias de banco de dados do MariaDB.

Tópicos

- Segurança do MariaDB no Amazon RDS
- Usar o plug-in de validação de senhas do RDS para MariaDB
- Criptografar conexões de clientes com SSL/TLS para instâncias de banco de dados do MariaDB no Amazon RDS
- Atualizar aplicações para conexão com instâncias do MariaDB usando novos certificados SSL/TLS

Segurança do MariaDB no Amazon RDS

A segurança de instâncias de banco de dados do MariaDB é gerenciada em três níveis:

- O AWS Identity and Access Management controla quem pode executar ações de gerenciamento do Amazon RDS em instâncias de banco de dados. Ao se conectar à AWS usando credenciais do IAM, sua conta do IAM deve ter políticas do IAM que concedam as permissões necessárias para executar operações de gerenciamento do Amazon RDS. Para ter mais informações, consulte Gerenciamento de identidade e acesso no Amazon RDS.
- Quando você cria uma instância de banco de dados, é necessário usar um grupo de segurança da VPC para controlar quais dispositivos e instâncias do Amazon EC2 podem abrir conexões com o endpoint e a porta da instância de banco de dados. É possível estabelecer essas conexões usando o Transport Layer Security (TLS) e o Secure Sockets Layer (SSL). Além disso, as regras de firewall em sua empresa podem controlar se dispositivos sendo executados nela podem abrir conexões na instância de banco de dados.
- Uma vez que uma conexão foi aberta em uma instância de banco de dados do MariaDB, a autenticação do login e as permissões são aplicadas da mesma forma que em uma instância autônoma do MariaDB. Comandos, como `CREATE USER`, `RENAME USER`, `GRANT`, `REVOKE` e `SET PASSWORD` funcionam exatamente como em bancos de dados autônomos, assim como modificando diretamente tabelas de esquema de banco de dados.

Quando você cria uma instância de banco de dados do Amazon RDS, o usuário mestre apresenta os seguintes privilégios padrão:

- `alter`

- `alter routine`
- `create`
- `create routine`
- `create temporary tables`
- `create user`
- `create view`
- `delete`
- `drop`
- `event`
- `execute`
- `grant option`
- `index`
- `insert`
- `lock tables`
- `process`
- `references`
- `reload`

  Esse privilégio é limitado em instâncias de banco de dados MariaDB. Ele não concede acesso às operações `FLUSH LOGS` e `FLUSH TABLES WITH READ LOCK`.
- `replication client`
- `replication slave`
- `select`
- `show create routine`

  Esse privilégio é válido somente em instâncias de banco de dados do MariaDB que executam a versão 11.4 e posterior.
- `show databases`
- `show view`
- `trigger`
- `update`

Para obter mais informações sobre esses privilégios, consulte Gerenciamento da conta de usuário na documentação do MariaDB.

> **Note**
>
> Embora você possa excluir o usuário mestre em uma instância de banco de dados, não recomendamos fazer isso. Para recriar o usuário mestre, use a API `ModifyDBInstance` ou o `modify-db-instance` AWS CLI e especifique uma nova senha de usuário mestre com o parâmetro apropriado. Se o usuário mestre não existir na instância, ele será criado com a senha especificada.

Para fornecer serviços de gerenciamento para cada instância de banco de dados, o usuário `rdsadmin` é criado quando a instância de banco de dados é criada. Tentar descartar, renomear ou alterar a senha, ou alterar os privilégios, para a conta `rdsadmin` resulta em um erro.

Para permitir o gerenciamento da instância de banco de dados, os comandos `kill` e `kill_query` padrão foram restritos. Os comandos do Amazon RDS `mysql.rds_kill`, `mysql.rds_kill_query` e `mysql.rds_kill_query_id` são fornecidos para uso no MariaDB e também no MySQL para seja possível encerrar sessões ou consultas de usuários em instâncias de banco de dados.

Usar o plug-in de validação de senhas do RDS para MariaDB

A partir do RDS para MariaDB versão 11.4, é possível usar os seguintes plug-ins de validação de senhas para aprimorar a segurança das conexões a bancos de dados:

- simple_password_check: confere se uma senha contém pelo menos um número específico de caracteres de determinado tipo.
- cracklib_password_check: confere se uma senha aparece em um arquivo de dicionário da biblioteca CrackLib.

Para habilitar esses plug-ins, defina o valor do parâmetro `simple_password_check` ou `cracklib_password_check` como `FORCE_PLUS_PERMANENT` no grupo de parâmetros de banco de dados associado à instância de banco de dados. Quando esse valor é definido, o plug-in não pode ser desinstalado usando a instrução `UNINSTALL PLUGIN` em tempo de execução.

Para desabilitar esses plug-ins, defina o valor do parâmetro `simple_password_check` ou `cracklib_password_check` como `OFF` no grupo de parâmetros de banco de dados associado à instância de banco de dados. Quando esse valor é definido, as regras de validação do plug-in não se aplicam mais às novas senhas.

Para ter informações sobre como definir os valores dos parâmetros nos respectivos grupos, consulte the section called “Modificar parâmetros em um grupo de parâmetros de banco de dados”.

Criptografar conexões de clientes com SSL/TLS para instâncias de banco de dados do MariaDB no Amazon RDS

O Secure Sockets Layer (SSL) é um protocolo padrão do setor para proteger as conexões de rede entre o cliente e o servidor. Após o SSL versão 3.0, o nome mudou para Transport Layer Security (TLS). O Amazon RDS é compatível com criptografia SSL/TLS para instâncias de banco de dados do MariaDB. Usando o SSL/TLS, você pode criptografar uma conexão entre seu cliente de aplicação e sua instância de banco de dados do MariaDB. A compatibilidade com SSL/TLS está disponível em todas as Regiões da AWS.

Com o Amazon RDS, é possível proteger dados em trânsito criptografando conexões de clientes com instâncias de banco de dados do MariaDB por meio de SSL/TLS, exigindo SSL/TLS para todas as conexões com uma instância de banco de dados do MariaDB e conectando-se pelo cliente da linha de comando do MySQL com SSL/TLS (criptografado). As seções a seguir apresentam orientações sobre como configurar e utilizar a criptografia SSL/TLS para instâncias de banco de dados do MariaDB no Amazon RDS.

Tópicos

- Suporte de SSL/TLS para instâncias de banco de dados do MariaDB no Amazon RDS
- Exigir SSL/TLS para contas de usuário específicas em uma instância de banco de dados do MariaDB
- Exigir SSL/TLS para todas as conexões com uma instância de banco de dados do MariaDB no Amazon RDS
- Conectar-se à instância de banco de dados do MariaDB no Amazon RDS com SSL/TLS por meio do cliente de linha de comandos do MySQL (criptografado)

Suporte de SSL/TLS para instâncias de banco de dados do MariaDB no Amazon RDS

O Amazon RDS cria um certificado SSL/TLS e instala o certificado na instância de banco de dados quando o Amazon RDS provisionar a instância. Esses certificados são assinados por uma autoridade de certificado. O certificado SSL/TLS inclui o endpoint da instância de banco de dados como o nome comum (CN) do certificado SSL/TLS para se proteger contra ataques de falsificação.

Um certificado SSL/TLS criado pelo Amazon RDS é a entidade raiz confiável e deve funcionar na maioria dos casos, mas poderá falhar se a aplicação não aceitar cadeias de certificados. Se a aplicação não aceitar cadeias de certificados, tente usar um certificado intermediário para se conectar à Região da AWS. Por exemplo, é necessário usar um certificado intermediário para se conectar às regiões AWS GovCloud (US) com SSL/TLS.

Para obter informações sobre como baixar certificados, consulte Usar SSL/TLS para criptografar uma conexão com uma instância ou um cluster de banco de dados. Para obter mais informações sobre como usar o SSL/TLS com o MySQL, consulte Atualizar aplicações para conexão com instâncias do MariaDB usando novos certificados SSL/TLS.

O Amazon RDS para MariaDB é compatível com o Transport Layer Security (TLS) versões 1.3, 1.2, 1.1 e 1.0. A compatibilidade com o TLS depende da versão secundária do MariaDB. A tabela a seguir mostra a compatibilidade das versões secundárias do MariaDB com o TLS.

Versão do TLS	MariaDB 11.4	MariaDB 10.11	MariaDB 10.6	MariaDB 10.5	MariaDB 10.4
TLS 1.3	Todas as versões secundárias	Todas as versões secundárias	Todas as versões secundárias	Todas as versões secundárias	Todas as versões secundárias
TLS 1.2	Todas as versões secundárias	Todas as versões secundárias	Todas as versões secundárias	Todas as versões secundárias	Todas as versões secundárias
TLS 1.1	Sem compatibilidade	10.11.6 e anterior	10.6.16 e anterior	10.5.23 e anterior	10.4.32 e anterior

Versão do TLS	MariaDB 11.4	MariaDB 10.11	MariaDB 10.6	MariaDB 10.5	MariaDB 10.4
TLS 1.0	Sem compatibilidade	10.11.6 e anterior	10.6.16 e anterior	10.5.23 e anterior	10.4.32 e anterior

Exigir SSL/TLS para contas de usuário específicas em uma instância de banco de dados do MariaDB

É possível exigir criptografia SSL/TLS para conexões de contas de usuário especificadas com as instâncias de banco de dados do MariaDB no Amazon RDS. Proteger informações confidenciais contra acesso não autorizado ou interceptação é crucial para aplicar políticas de segurança nas quais a confidencialidade dos dados é uma questão importante.

Se deseja exigir conexões SSL/TLS para contas de usuários específicos, siga uma das instruções abaixo (dependendo da versão do MySQL) para exigir conexões SSL/TLS na conta de usuário `encrypted_user`.

Para fazer isso, use a instrução a seguir.

```
ALTER USER 'encrypted_user'@'%' REQUIRE SSL;
```

Para obter mais informações sobre as conexões SSL/TLS com o MariaDB, consulte Securing Connections for Client and Server (Como proteger conexões para cliente e servidor) na documentação do MariaDB.

Exigir SSL/TLS para todas as conexões com uma instância de banco de dados do MariaDB no Amazon RDS

Use o parâmetro `require_secure_transport` para exigir que todas as conexões de usuários à sua instância de banco de dados do MariaDB usem SSL/TLS. Por padrão, o parâmetro `require_secure_transport` é definido como `OFF`. Você pode definir o parâmetro `require_secure_transport` como `ON` para exigir SSL/TLS para conexões com a sua instância de banco de dados.

> **Note**
>
> O parâmetro `require_secure_transport` só é compatível com o MariaDB versão 10.5 e posterior.

Você pode definir o valor de parâmetro `require_secure_transport` atualizando o grupo de parâmetros do banco de dados de sua instância de banco de dados. Você não precisa reinicializar sua instância de banco de dados para que a alteração entre em vigor.

Quando o parâmetro `require_secure_transport` é definido como `ON` para uma instância de banco de dados, um cliente de banco de dados poderá se conectar a ela se puder estabelecer uma conexão criptografada. Caso contrário, uma mensagem de erro semelhante à seguinte é retornada para o cliente:

```
ERROR 1045 (28000): Access denied for user 'USER'@'localhost' (using password: YES |
 NO)
```

Para obter informações sobre como configurar parâmetros, consulte Modificar parâmetros em um grupo de parâmetros de banco de dados no Amazon RDS.

Para obter mais informações sobre o parâmetro `require_secure_transport`, consulte a documentação do MariaDB.

Conectar-se à instância de banco de dados do MariaDB no Amazon RDS com SSL/TLS por meio do cliente de linha de comandos do MySQL (criptografado)

Os parâmetros `mysql` do programa cliente serão ligeiramente diferentes, se você estiver usando a versão do MySQL 5.7, a versão do MySQL 8.0 ou a versão do MariaDB.

Para descobrir qual versão você tem, execute o comando `mysql` com a opção `--version`. No exemplo a seguir, a saída mostra que o programa cliente é do MariaDB.

```
$ mysql --version
mysql  Ver 15.1 Distrib 10.5.15-MariaDB, for osx10.15 (x86_64) using readline 5.1
```

A maioria das distribuições Linux, como o Amazon Linux, CentOS, SUSE e Debian, substituiu o MySQL pelo MariaDB, e a versão do `mysql` neles é de MariaDB.

Para conectar-se à sua instância de banco de dados usando o SSL/TLS, siga estas etapas:

Como conectar-se a uma instância de banco de dados com o SSL/TLS usando o cliente de linha de comando do MySQL

1. Você pode baixar um certificado raiz que funciona para todas as Regiões da AWS.

   Para obter informações sobre como baixar certificados, consulte Usar SSL/TLS para criptografar uma conexão com uma instância ou um cluster de banco de dados.
2. Use um cliente da linha de comando do MySQL para conectar-se a uma instância de banco de dados com criptografia SSL/TLS. Para o parâmetro `-h`, substitua o nome do DNS (endpoint) da instância de banco de dados. Para o parâmetro `--ssl-ca`, substitua o nome do arquivo de certificado SSL/TLS. Para o parâmetro `-P`, substitua a porta da sua instância de banco de dados. Para o parâmetro `-u`, substitua o nome de usuário de um usuário de banco de dados válido, como o usuário principal. Digite a senha do usuário mestre quando lhe for solicitado.

   O exemplo a seguir mostra como iniciar o cliente usando o parâmetro `--ssl-ca` usando o cliente MariaDB:

   ```
   mysql -h mysql-instance1.123456789012.us-east-1.rds.amazonaws.com --ssl-ca=global-bundle.pem --ssl -P 3306 -u myadmin -p
   ```

   Para exigir que a conexão SSL/TLS verifique o endpoint da instância de banco de dados em relação ao endpoint no certificado SSL/TLS, insira o seguinte comando:

   ```
   mysql -h mysql-instance1.123456789012.us-east-1.rds.amazonaws.com --ssl-ca=global-bundle.pem --ssl-verify-server-cert -P 3306 -u myadmin -p
   ```

   O exemplo a seguir mostra como iniciar o cliente usando o parâmetro `--ssl-ca` com o MySQL 5.7 e versões posteriores.

   ```
   mysql -h mysql-instance1.123456789012.us-east-1.rds.amazonaws.com --ssl-ca=global-bundle.pem --ssl-mode=REQUIRED -P 3306 -u myadmin -p
   ```
3. Digite a senha do usuário mestre quando lhe for solicitado.

Você deve ver saída semelhante ao seguinte:

```
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 31
Server version: 10.6.10-MariaDB-log Source distribution
```

```
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>
```

Atualizar aplicações para conexão com instâncias do MariaDB usando novos certificados SSL/TLS

Em 13 de janeiro de 2023, o Amazon RDS publicou novos certificados de autoridade de certificação (CA) para se conectar às suas instâncias de banco de dados do RDS via Secure Socket Layer ou Transport Layer Security (SSL/TLS). A seguir, você pode encontrar informações sobre como atualizar seus aplicativos para usar os novos certificados.

Este tópico pode ajudá-lo a determinar se seus aplicativos exigem verificação do certificado para se conectar às suas instâncias de bancos de dados.

> **Note**
>
> Alguns aplicativos são configurados para se conectar ao MariaDB apenas quando podem verificar com sucesso o certificado no servidor. Para esses aplicativos, você deve atualizar os repositórios confiáveis de aplicativos cliente para incluir os novos certificados de CA.
> É possível especificar os seguintes modos SSL: `disabled`, `preferred` e `required`.
> Quando você usa o modo `preferred` do SSL e o certificado de CA não existe ou não está atualizado, a conexão volta a não usar SSL e ainda se conecta com êxito.
> Recomendamos evitar o modo `preferred` No modo `preferred`, se a conexão encontrar um certificado inválido, ele para de usar criptografia e continua sem criptografia.

Depois de atualizar seus certificados de CA nos armazenamentos confiáveis do aplicativo cliente, você pode fazer o rodízio dos certificados nas suas instâncias de banco de dados. É altamente recomendável testar esses procedimentos em um ambiente de desenvolvimento ou teste antes de implementá-los em seus ambientes de produção.

Para obter mais informações sobre a mudança de certificados, consulte Alternar o certificado SSL/TLS. Para obter mais informações sobre como fazer download de certificados, consulte Usar SSL/TLS para criptografar uma conexão com uma instância ou um cluster de banco de dados. Para obter

informações sobre o uso de SSL/TLS com instâncias de bancos de dados MariaDB, consulte Suporte de SSL/TLS para instâncias de banco de dados do MariaDB no Amazon RDS.

Tópicos

- Determinar se um cliente requer verificação de certificado para se conectar
- Atualizar o armazenamento confiável de aplicações
- Exemplo de código Java para estabelecer conexões SSL

Determinar se um cliente requer verificação de certificado para se conectar

É possível verificar se os clientes JDBC e MySQL exigem verificação de certificado para se conectarem.

JDBC

O exemplo a seguir com o MySQL Connector/J 8.0 mostra uma maneira de verificar as propriedades da conexão JDBC de um aplicativo para determinar se conexões bem-sucedidas exigem um certificado válido. Para obter mais informações sobre todas as opções de conexão JDBC para MySQL, consulte Propriedades da configuração na documentação do MySQL.

Ao usar o MySQL Connector/J 8.0, uma conexão SSL exigirá verificação com base no certificado de CA de servidor se as propriedades da sua conexão tiverem `sslMode` definido como `VERIFY_CA` ou `VERIFY_IDENTITY`, como no exemplo a seguir.

```
Properties properties = new Properties();
properties.setProperty("sslMode", "VERIFY_IDENTITY");
properties.put("user", DB_USER);
properties.put("password", DB_PASSWORD);
```

> **Note**
>
> Se você usar o MySQL Java Connector v5.1.38 ou posterior, ou o MySQL Java Connector v8.0.9 ou posterior para se conectar a seus bancos de dados, mesmo que você não tenha configurado explicitamente seus aplicativos para usar SSL/TLS ao se conectar a seus bancos de dados, esses drivers cliente usam SSL/TLS como padrão. Além disso, ao usar SSL/TLS, eles executam a verificação parcial do certificado e haverá falha na conexão se o certificado do servidor de banco de dados tiver expirado.

Especifique uma senha diferente do prompt mostrado aqui como prática recomendada de segurança.

MySQL

Os exemplos a seguir com o cliente MySQL mostram duas maneiras de verificar a conexão MySQL de um script para determinar se conexões bem-sucedidas exigem um certificado válido. Para obter mais informações sobre todas as opções de conexão com o cliente MySQL, consulte Configuração no lado do cliente para conexões criptografadas na documentação do MySQL.

Ao usar o Cliente MySQL 5.7 ou MySQL 8.0, uma conexão SSL requer verificação com base no certificado de CA de servidor se, para a opção `--ssl-mode`, você especificar `VERIFY_CA` ou `VERIFY_IDENTITY`, como no exemplo a seguir.

```
mysql -h mysql-database.rds.amazonaws.com -uadmin -ppassword --ssl-ca=/tmp/ssl-cert.pem
 --ssl-mode=VERIFY_CA
```

Ao usar o Cliente MySQL 5.6, uma conexão SSL exigirá verificação com base no certificado de CA de servidor se você especificar a opção `--ssl-verify-server-cert`, como no exemplo a seguir.

```
mysql -h mysql-database.rds.amazonaws.com -uadmin -ppassword --ssl-ca=/tmp/ssl-cert.pem
 --ssl-verify-server-cert
```

Atualizar o armazenamento confiável de aplicações

Para obter informações sobre como atualizar o repositório confiável para aplicativos MySQL, consulte o tópico sobre como Usar o TLS/SSL com MariaDB Connector/J, na documentação do MariaDB.

Para obter informações sobre como baixar o certificado raiz, consulte Usar SSL/TLS para criptografar uma conexão com uma instância ou um cluster de banco de dados.

Para obter exemplos de scripts que importam certificados, consulte Script de exemplo para importar certificados para o seu armazenamento confiável.

> **Note**
>
> Ao atualizar o armazenamento confiável, é possível reter certificados mais antigos, além de adicionar os novos certificados.

Se você estiver usando o driver JDBC MariaDB Connector/J em um aplicativo, defina as seguintes propriedades nesse aplicativo.

```
System.setProperty("javax.net.ssl.trustStore", certs);
System.setProperty("javax.net.ssl.trustStorePassword", "password");
```

Ao iniciar o aplicativo, defina as seguintes propriedades.

```
java -Djavax.net.ssl.trustStore=/path_to_truststore/MyTruststore.jks -
Djavax.net.ssl.trustStorePassword=my_truststore_password com.companyName.MyApplication
```

> **Note**
>
> Especifique senhas diferentes dos prompts mostrados aqui como prática recomendada de segurança.

Exemplo de código Java para estabelecer conexões SSL

O exemplo de código a seguir mostra como configurar a conexão SSL usando o JDBC.

```
private static final String DB_USER = "admin";

        private static final String DB_USER = "user name";
        private static final String DB_PASSWORD = "password";
        // This key store has only the prod root ca.
        private static final String KEY_STORE_FILE_PATH = "file-path-to-keystore";
        private static final String KEY_STORE_PASS = "keystore-password";
```

```
    public static void main(String[] args) throws Exception {
        Class.forName("org.mariadb.jdbc.Driver");

        System.setProperty("javax.net.ssl.trustStore", KEY_STORE_FILE_PATH);
        System.setProperty("javax.net.ssl.trustStorePassword", KEY_STORE_PASS);

        Properties properties = new Properties();
        properties.put("user", DB_USER);
        properties.put("password", DB_PASSWORD);


        Connection connection = DriverManager.getConnection("jdbc:mysql://ssl-mariadb-
public.cni62e2e7kwh.us-east-1.rds.amazonaws.com:3306?useSSL=true",properties);
        Statement stmt=connection.createStatement();

        ResultSet rs=stmt.executeQuery("SELECT 1 from dual");

        return;
    }
```

Important

Depois de determinar que suas conexões com o banco de dados usam SSL/TLS e ter atualizado o armazenamento confiável de aplicações, você poderá atualizar o banco de dados para usar os certificados rds-ca-rsa2048-g1. Para obter instruções, consulte a etapa 3 em Atualizar o certificado CA modificando a instância ou o cluster de banco de dados. Especifique uma senha diferente do prompt mostrado aqui como prática recomendada de segurança.

Melhorar a performance das consultas para o RDS para MariaDB com leituras otimizadas do Amazon RDS

É possível acelerar o processamento de consultas para o RDS para MariaDB com leituras otimizadas do Amazon RDS. Uma instância de banco de dados do RDS para MariaDB que utiliza leituras otimizadas do RDS pode obter um processamento de consultas até duas vezes mais rápido em comparação com uma instância de banco de dados que não utiliza o recurso.

Tópicos

- Visão geral das leituras otimizadas do RDS
- Casos de uso para leituras otimizadas do RDS
- Práticas recomendadas para leituras otimizadas do RDS
- Utilizar leituras otimizadas do RDS
- Monitorar instâncias de banco de dados que utilizam leituras otimizadas do RDS
- Limitações das leituras otimizadas do RDS

Visão geral das leituras otimizadas do RDS

Quando você usa uma instância de banco de dados do RDS para MariaDB com leituras otimizadas do RDS ativadas, sua instância de banco de dados obtém uma performance de consulta mais rápida por meio do uso de um armazenamento de instâncias. Um armazenamento de instâncias fornece armazenamento temporário em nível de bloco para a instância de banco de dados. O armazenamento está localizado em unidades de estado sólido (SSDs) Non-Volatile Memory Express (NVMe) anexadas fisicamente ao servidor host. Esse armazenamento é otimizado para baixa latência, performance de E/S aleatória muito alta e alto throughput de leitura de sequencial.

As leituras otimizadas do RDS são ativadas por padrão quando uma instância de banco de dados utiliza uma classe de instância de banco de dados com um armazenamento de instâncias, como db.m5d ou db.m6gd. Com leituras otimizadas do RDS, alguns objetos temporários são armazenados no armazenamento de instâncias. Esses objetos temporários incluem arquivos temporários internos, tabelas temporárias internas em disco, arquivos de mapa de memória e arquivos de cache de log binário. Para ter mais informações sobre o armazenamento de instância, consulte Armazenamento de instância do Amazon EC2 no Amazon Elastic Compute Cloud Manual do usuário para instâncias do Linux.

As workloads que geram objetos temporários no MariaDB para processamento de consultas podem aproveitar o armazenamento de instâncias para acelerar o processamento das consultas. Esse tipo de workload inclui consultas que envolvem classificações, agregações de hash, uniões de alta carga, expressões de tabela comuns (CTEs) e consultas em colunas não indexadas. Esses volumes de armazenamento de instâncias oferecem um nível superior de IOPS e performance, independentemente das configurações de armazenamento utilizadas para armazenamento persistente do Amazon EBS. Como as leituras otimizadas do RDS transferem operações em objetos temporários para o armazenamento de instâncias, as operações de entrada e saída por segundo (IOPS) ou o throughput do armazenamento persistente (Amazon EBS) agora podem ser utilizadas para operações em objetos persistentes. Essas operações incluem leituras e gravações regulares de arquivos de dados e operações do mecanismo em segundo plano, como limpar e inserir mesclagens de buffer.

> **Note**
>
> Os snapshots manuais e automatizados do RDS contêm apenas arquivos de mecanismo para objetos persistentes. Os objetos temporários criados no armazenamento de instâncias não estão incluídos nos snapshots do RDS.

Casos de uso para leituras otimizadas do RDS

Se você tem workloads que dependem muito de objetos temporários, como tabelas ou arquivos internos, para a execução de consultas, você pode se beneficiar da ativação de leituras otimizadas do RDS. Os seguintes casos de uso são candidatos para leituras otimizadas do RDS:

- Aplicações que executam consultas analíticas com expressões de tabela comuns (CTEs) complexas, tabelas derivadas e operações de agrupamento
- Réplicas de leitura que oferecem tráfego de leitura intenso com consultas não otimizadas
- Aplicações que executam consultas de relatórios dinâmicos ou sob demanda que envolvem operações complexas, como consultas com as cláusulas `GROUP BY` e `ORDER BY`
- Workloads que utilizam tabelas temporárias internas para processamento de consultas

  Você pode monitorar a variável de status do mecanismo `created_tmp_disk_tables` para determinar o número de tabelas temporárias baseadas em disco criadas em sua instância de banco de dados.

- Aplicações que criam grandes tabelas temporárias, diretamente ou em procedimentos, para armazenar resultados intermediários
- Consultas de banco de dados que realizam agrupamento ou classificação em colunas não indexadas

Práticas recomendadas para leituras otimizadas do RDS

Use as práticas recomendadas a seguir para leituras otimizadas do RDS:

- Adicione uma lógica de repetição para consultas somente leitura caso elas falhem devido ao armazenamento de instância estar cheio durante a execução.
- Monitore o espaço de armazenamento disponível no armazenamento de instâncias com a métrica do CloudWatch `FreeLocalStorage`. Se o armazenamento de instâncias estiver atingindo seu limite devido à workload na instância de banco de dados, modifique a instância de banco de dados para utilizar uma classe de instância de banco de dados maior.
- Quando sua instância de banco de dados tiver memória suficiente, mas ainda estiver atingindo o limite de armazenamento no armazenamento de instâncias, aumente o valor `binlog_cache_size` para manter as entradas de log binário específicas da sessão na memória. Essa configuração impede a gravação das entradas de log binário em arquivos temporários de cache de log binário no disco.

  O parâmetro `binlog_cache_size` é específico da sessão. É possível alterar o valor de cada nova sessão. A configuração desse parâmetro pode aumentar a utilização da memória na instância de banco de dados durante a workload de pico. Portanto, considere aumentar o valor do parâmetro com base no padrão de workload de sua aplicação e na memória disponível na instância de banco de dados.
- Utilize o valor padrão de `MIXED` para o `binlog_format`. Dependendo do tamanho das transações, definir `binlog_format` como `ROW` pode ocasionar arquivos de cache de log binário grandes no armazenamento de instâncias.
- Evite realizar alterações em massa em uma única transação. Esses tipos de transações podem gerar arquivos de cache de log binário grandes no armazenamento de instâncias e causar problemas quando o armazenamento de instâncias está cheio. Considere dividir as gravações em várias transações pequenas para minimizar o uso de armazenamento para arquivos de cache de log binário.

Utilizar leituras otimizadas do RDS

Quando você provisiona uma instância de banco de dados do RDS para MariaDB com uma das seguintes classes de instância de banco de dados em uma implantação de instância de banco de dados single-AZ ou multi-AZ, a instância de banco de dados utiliza automaticamente o recurso Leituras otimizadas pelo RDS.

Para ativar as leituras otimizadas do RDS, execute um destes procedimentos:

- Crie uma instância de banco de dados do RDS para MariaDB utilizando uma dessas classes de instância de banco de dados. Para ter mais informações, consulte Criar uma instância de banco de dados do Amazon RDS.
- Modifique uma instância de banco de dados do RDS para MariaDB para utilizar uma dessas classes de instância de banco de dados. Para ter mais informações, consulte Modificar uma instância de banco de dados do Amazon RDS.

O recurso Leituras otimizadas pelo RDS está disponível em todas as Regiões da AWS onde há suporte para uma ou mais dessas classes de instância de banco de dados com SSD NVMe local. Para obter informações sobre classes de instância de banco de dados, consulte the section called “Classes da instância de banco de dados”.

A disponibilidade da classe de instância de banco de dados difere em Regiões da AWS. Para determinar se uma classe de instância de banco de dados é compatível com uma Região da AWS específica, consulte the section called “Determinar o suporte para classes de instância de bancos de dados nas Regiões da AWS”.

Se você não quiser utilizar leituras otimizadas do RDS, modifique sua instância de banco de dados para que ela não use uma classe de instância de banco de dados compatível com o recurso.

Monitorar instâncias de banco de dados que utilizam leituras otimizadas do RDS

Você pode monitorar instâncias de banco de dados que utilizam leituras otimizadas do RDS com as seguintes métricas do CloudWatch:

- `FreeLocalStorage`
- `ReadIOPSLocalStorage`
- `ReadLatencyLocalStorage`

- `ReadThroughputLocalStorage`
- `WriteIOPSLocalStorage`
- `WriteLatencyLocalStorage`
- `WriteThroughputLocalStorage`

Essas métricas fornecem dados sobre armazenamento de instâncias, IOPS e throughput. Para ter mais informações sobre essas métricas, consulte Métricas específicas da instância do Amazon CloudWatch para Amazon RDS.

Limitações das leituras otimizadas do RDS

As seguintes limitações se aplicam às leituras otimizadas do RDS:

- O recurso de leituras otimizadas do RDS é compatível com as seguintes versões do RDS para MariaDB:
  - 10.11.4 e versões 10.11 posteriores
  - 10.6.7 e versões 10.6 posteriores
  - 10.5.16 e versões 10.5 posteriores
  - 10.4.25 e versões 10.4 posteriores

  Para obter informações sobre as versões do RDS para MariaDB, consulte MariaDB em versões do Amazon RDS.
- Você não pode alterar a localização de objetos temporários para armazenamento persistente (Amazon EBS) nas classes de instância de banco de dados compatíveis com leituras otimizadas do RDS.
- Quando o registro em log binário está ativado em uma instância de banco de dados, o tamanho máximo da transação é limitado pelo tamanho do armazenamento de instância. No MariaDB, qualquer sessão que exija mais armazenamento do que o valor de `binlog_cache_size` grava alterações de transações em arquivos temporários de cache de log binário, que são criados no armazenamento de instância.
- As transações podem falhar quando o armazenamento de instância está cheio.

Melhorar a performance das gravações com gravações otimizadas pelo Amazon RDS para MariaDB

Você pode melhorar a performance das transações de gravação com as gravações otimizadas pelo Amazon RDS para MariaDB. Quando seu banco de dados do RDS para MariaDB usa gravações otimizadas pelo RDS, ele pode atingir um throughput de transações de gravação até duas vezes maior.

Tópicos

- Visão geral das gravações otimizadas pelo RDS
- Utilizar gravações otimizadas pelo RDS
- Habilitando gravações otimizadas para RDS em um banco de dados existente
- Limitações das gravações otimizadas pelo RDS

Visão geral das gravações otimizadas pelo RDS

Quando você ativa as gravações otimizadas pelo Amazon RDS, seus bancos de dados do RDS para MariaDB gravam apenas uma vez ao liberar dados em um armazenamento durável sem a necessidade do buffer de gravação dupla. Os bancos de dados continuam fornecendo proteções de propriedade ACID para transações de banco de dados confiáveis, além de desempenho aprimorado.

Bancos de dados relacionais, como o MariaDB, fornecem as propriedades ACID de atomicidade, consistência, isolamento e durabilidade para transações de banco de dados confiáveis. Para ajudar a fornecer essas propriedades, o MariaDB usa uma área de armazenamento de dados denominada buffer de gravação dupla que evita erros parciais de gravação de página. Esses erros ocorrem quando há uma falha de hardware enquanto o banco de dados está atualizando uma página, como no caso de uma queda de energia. Um banco de dados MariaDB pode detectar gravações parciais de páginas e recuperá-las com uma cópia da página no buffer de gravação dupla. Embora essa técnica ofereça proteção, ela também ocasiona operações extras de gravação. Para ter mais informações sobre o buffer de gravação dupla do MariaDB, consulte Buffer de gravação dupla do InnoDB na documentação do MariaDB.

Com as gravações otimizadas pelo RDS ativadas, os bancos de dados do RDS para MariaDB gravam apenas uma vez ao liberar dados em um armazenamento durável sem a necessidade do buffer de gravação dupla. As gravações otimizadas pelo RDS são úteis se você executa workloads de uso intenso de gravação em seus bancos de dados do RDS para MariaDB. Exemplos de bancos

de dados com workloads de uso intenso de gravação incluem os compatíveis com pagamentos digitais, negociação financeira e aplicações de jogos.

Esses bancos de dados são executados em classes de instância de banco de dados que utilizam o AWS Nitro System. Devido à configuração de hardware nesses sistemas, o banco de dados pode gravar páginas de 16 KiB diretamente nos arquivos de dados de forma confiável e durável em uma única etapa. O sistema AWS Nitro possibilita gravações otimizadas pelo RDS.

Você pode definir o novo parâmetro de banco de dados `rds.optimized_writes` para controlar o recurso de gravações otimizadas pelo RDS para bancos de dados do RDS para MariaDB. Acesse esse parâmetro nos grupos de parâmetros de banco de dados do RDS para MariaDB para as seguintes versões:

- 11.4.3 e versões 11.4 posteriores
- 10.11.4 e versões 10.11 posteriores
- 10.6.10 e versões 10.6 posteriores

Defina o parâmetro com os seguintes valores:

- `AUTO`: ative as gravações otimizadas pelo RDS se o banco de dados for compatível com o recurso. Desative as gravações otimizadas pelo RDS se o banco de dados não for compatível com o recurso. Essa é a configuração padrão.
- `OFF`: desative as gravações otimizadas pelo RDS mesmo se o banco de dados for compatível com o recurso.

Se você migrar um banco de dados do RDS para MariaDB configurado para usar gravações otimizadas pelo RDS para uma classe de instância de banco de dados que não seja compatível com o recurso, o RDS desativará automaticamente as gravações otimizadas pelo RDS para o banco de dados.

Quando as gravações otimizadas pelo RDS estão desativadas, o banco de dados usa o buffer de gravação dupla do MariaDB.

Para determinar se um banco de dados do RDS para MariaDB está usando gravações otimizadas pelo RDS, visualize o valor atual do parâmetro `innodb_doublewrite` para o banco de dados. Se o banco de dados estiver usando gravações otimizadas pelo RDS, esse parâmetro será definido como `FALSE` (`0`).

Utilizar gravações otimizadas pelo RDS

Você pode ativar as gravações otimizadas pelo RDS ao criar um banco de dados do RDS para MariaDB com o console do RDS, a AWS CLI ou a API do RDS. As gravações otimizadas pelo RDS são ativadas automaticamente quando as duas condições a seguir se aplicam durante a criação do banco de dados:

- Você especifica uma versão de mecanismo de banco de dados e uma classe de instância de banco de dados compatíveis com as gravações otimizadas pelo RDS.
  - O recurso Leituras otimizadas pelo RDS é compatível com as seguintes versões do RDS para MariaDB:
    - 11.4.3 e versões 11.4 posteriores
    - 10.11.4 e versões 10.11 posteriores
    - 10.6.10 e versões 10.6 posteriores

    Para obter informações sobre as versões do RDS para MariaDB, consulte MariaDB em versões do Amazon RDS.
  - As gravações otimizadas pelo RDS são compatíveis com bancos de dados do RDS para MariaDB que usam as seguintes classes de instância de banco de dados:
    - db.m7i
    - db.m7g
    - db.m6g
    - db.m6gd
    - db.m6i
    - db.m5
    - db.m5d
    - db.r7i
    - db.r7g
    - db.r6g
    - db.r6gd
    - db.r6i
    - db.r5
    - db.r5b

- db.r5d
- db.x2idn
- db.x2iedn

  Para obter informações sobre classes de instância de banco de dados, consulte the section called “Classes da instância de banco de dados”.

  A disponibilidade da classe de instância de banco de dados difere em Regiões da AWS. Para determinar se uma classe de instância de banco de dados é compatível com uma Região da AWS específica, consulte the section called “Determinar o suporte para classes de instância de bancos de dados nas Regiões da AWS”.

- No grupo de parâmetros associado ao banco de dados, o parâmetro `rds.optimized_writes` é definido como `AUTO`. Nos grupos de parâmetros padrão, esse parâmetro é sempre definido como `AUTO`.

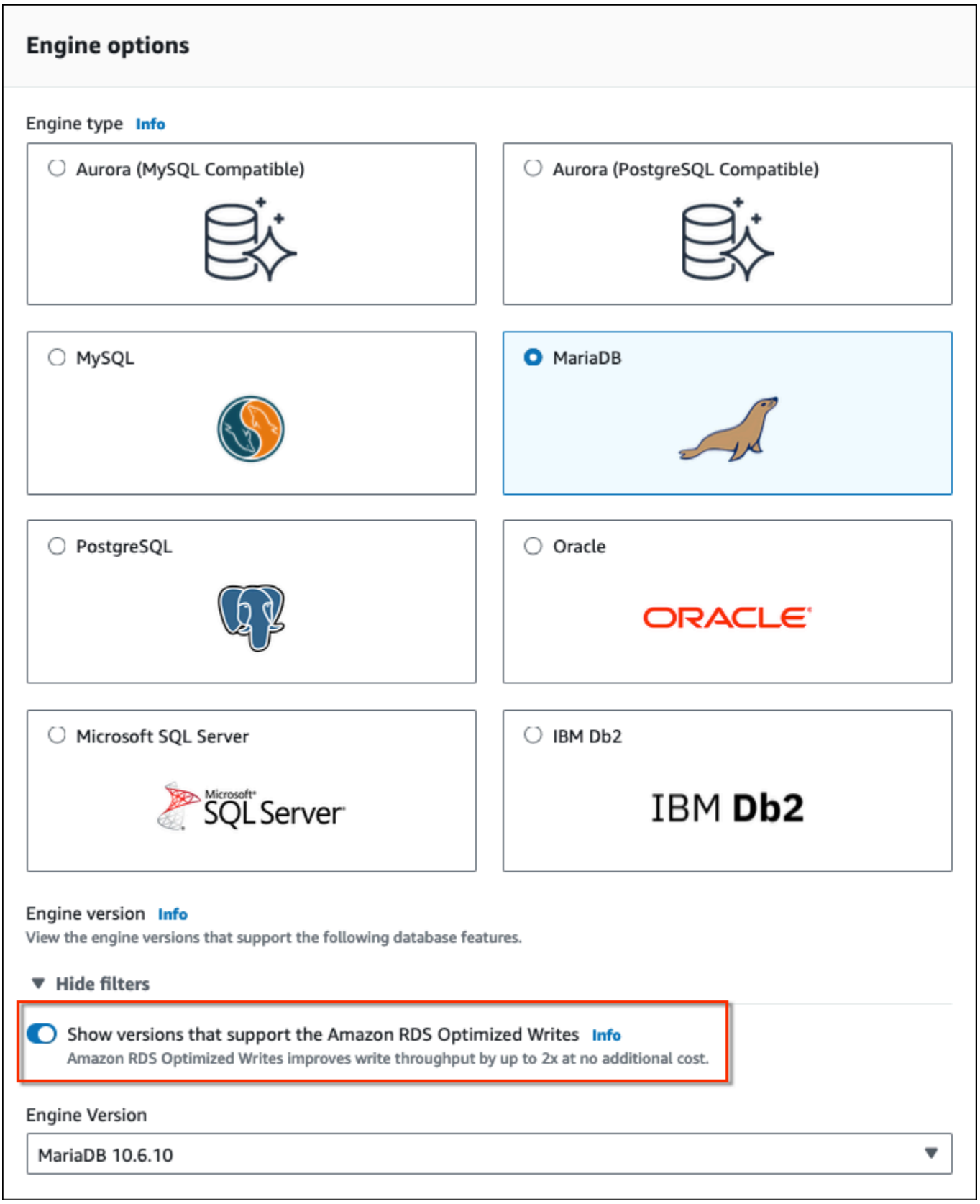
Se você quiser usar uma versão do mecanismo de banco de dados e uma classe de instância de banco de dados que sejam compatíveis com gravações otimizadas pelo RDS, mas não quer usar esse recurso, especifique um grupo de parâmetros personalizado ao criar o banco de dados. Nesse grupo de parâmetros, defina o parâmetro `rds.optimized_writes` como `OFF`. Se você quiser que o banco de dados use gravações otimizadas pelo RDS posteriormente, você pode definir o parâmetro `AUTO` para ativá-lo. Para obter informações sobre como criar grupos de parâmetros personalizados e definir parâmetros, consulte Grupos de parâmetros para Amazon RDS.

Para obter informações sobre como criar uma instância de banco de dados, consulte Criar uma instância de banco de dados do Amazon RDS.

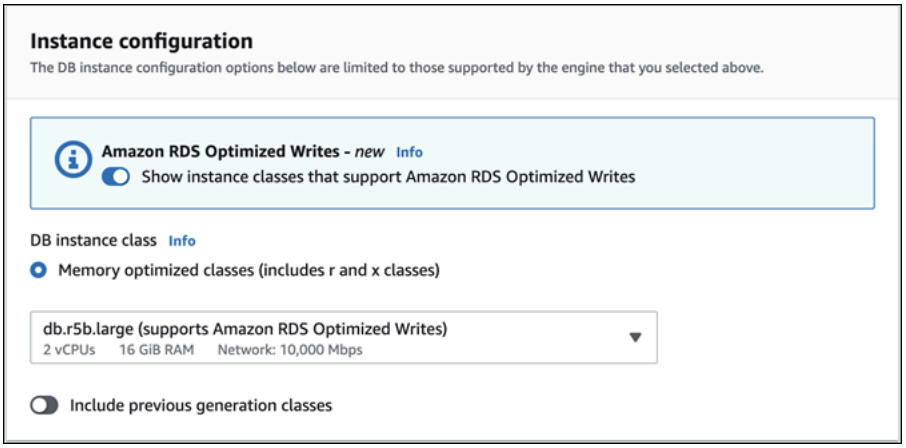
Console

Ao usar o console do RDS para criar um banco de dados do RDS para MariaDB, você pode filtrar as versões do mecanismo de banco de dados e as classes de instância de banco de dados compatíveis com gravações otimizadas pelo RDS. Depois de ativar os filtros, é possível selecionar entre as versões disponíveis do mecanismo de banco de dados e classes de instância de banco de dados.

Para selecionar uma versão do mecanismo de banco de dados que seja compatível com gravações otimizadas pelo RDS, filtre as versões do mecanismo de banco de dados do RDS para MariaDB compatíveis com o recurso em Versão do mecanismo e, depois, selecione uma versão.


Engine options
Engine type Info
Aurora (MySQL Compatible)
Aurora (PostgreSQL Compatible)
MySQL
MariaDB
PostgreSQL
Oracle
ORACLE
Microsoft SQL Server
Microsoft SQL Server
IBM Db2
IBM Db2
Engine version Info
View the engine versions that support the following database features.
Hide filters
Show versions that support the Amazon RDS Optimized Writes Info
Amazon RDS Optimized Writes improves write throughput by up to 2x at no additional cost.
Engine Version
MariaDB 10.6.10

Na seção Instance configuration (Configuração da instância), filtre as classes de instância de banco de dados que são compatíveis com gravações otimizadas pelo RDS e selecione uma classe de instância de banco de dados.



Depois de fazer essas seleções, você pode selecionar outras configurações que atendam aos seus requisitos e concluir a criação do banco de dados do RDS para MariaDB com o console.

AWS CLI

Para criar uma instância de banco de dados utilizando o comando da AWS CLI, execute o comando create-db-instance. Os valores `--engine-version` e `--db-instance-class` devem ser compatíveis com gravações otimizadas pelo RDS. Além disso, verifique se o grupo de parâmetros associado a uma instância de banco de dados tem o parâmetro `rds.optimized_writes` definido como AUT0. O exemplo associa o grupo de parâmetros padrão à instância de banco de dados.

Example Criar uma instância de banco de dados que utilize gravações otimizadas pelo RDS

Para Linux, macOS ou Unix:

```
aws rds create-db-instance \
    --db-instance-identifier mydbinstance \
    --engine mariadb \
    --engine-version 10.6.10 \
    --db-instance-class db.r5b.large \
    --manage-master-user-password \
    --master-username admin \
    --allocated-storage 200
```

Para Windows:

```
aws rds create-db-instance ^
```

```
    --db-instance-identifier mydbinstance ^
    --engine mariadb ^
    --engine-version 10.6.10 ^
    --db-instance-class db.r5b.large ^
    --manage-master-user-password ^
    --master-username admin ^
    --allocated-storage 200
```

API do RDS

Você pode criar uma instância de banco de dados utilizando a operação CreateDBInstance. Ao utilizar essa operação, os valores `EngineVersion` e `DBInstanceClass` devem ser compatíveis com gravações otimizadas pelo RDS. Além disso, verifique se o grupo de parâmetros associado a uma instância de banco de dados tem o parâmetro `rds.optimized_writes` definido como `AUTO`.

Habilitando gravações otimizadas para RDS em um banco de dados existente

Para modificar um banco de dados do RDS for MariaDB existente para ativar as gravações otimizadas do RDS, o banco de dados deve ter sido criado com uma versão do mecanismo de banco de dados e uma classe de instância de banco de dados compatíveis. Além disso, o banco de dados deve ter sido criado após o lançamento do RDS Optimized Writes em 7 de março de 2023, pois a configuração necessária do sistema de arquivos subjacente é incompatível com a dos bancos de dados criados antes do lançamento. Se essas condições forem atendidas, você poderá ativar as gravações otimizadas para RDS definindo o `rds.optimized_writes` parâmetro como. `AUTO`

Se seu banco de dados não foi criado com uma versão de mecanismo, classe de instância ou configuração de sistema de arquivos compatível, você pode usar o RDS Blue/Green Deployments para migrar para uma configuração compatível. Ao criar a implantação azul/verde, faça o seguinte:

- Selecione Habilitação de gravações otimizadas no banco de dados verde e, em seguida, especifique uma versão do mecanismo e uma classe de instância de banco de dados compatíveis com as gravações otimizadas do RDS. Para obter uma lista das versões dos mecanismos compatíveis, consulte the section called “Usando com um banco de dados”.
- Em Armazenamento, escolha Atualizar a configuração do sistema de arquivos de armazenamento. Essa opção atualiza o banco de dados para uma configuração de sistema de arquivos subjacente compatível.

Ao criar a implantação azul/verde, se o `rds.optimized_writes` parâmetro estiver definido para`AUTO`, as gravações otimizadas do RDS serão ativadas automaticamente no ambiente verde. Você pode então fazer a transição azul/verde, o que promove o ambiente verde para o novo ambiente de produção.

Para ter mais informações, consulte the section called “Criar uma implantação azul/verde”.

Limitações das gravações otimizadas pelo RDS

Ao restaurar um banco de dados do RDS para MariaDB por meio de um snapshot, você só poderá ativar as gravações otimizadas pelo RDS no banco de dados se todas as condições a seguir se aplicarem:

- O snapshot foi criado a partir de um banco de dados compatível com gravações otimizadas pelo RDS.
- O snapshot foi criado por meio de um banco de dados criado após o lançamento de gravações otimizadas pelo RDS.
- O snapshot foi restaurado a partir de um banco de dados compatível com gravações otimizadas pelo RDS.
- O banco de dados restaurado é associado a um grupo de parâmetros que tenha o parâmetro `rds.optimized_writes` definido como `AUTO`.

Atualizações do mecanismo de banco de dados do MariaDB

Quando o Amazon RDS oferece suporte a uma nova versão de um mecanismo de banco de dados, você pode atualizar suas instâncias de banco de dados para essa nova versão. Há dois tipos de atualizações para instâncias de banco de dados MariaDB: atualizações de versão principal e atualizações de versão secundária.

As atualizações da versão principal podem conter as alterações de banco de dados incompatíveis com os aplicativos existentes. Como resultado, você deve realizar manualmente as atualizações de versões principais das suas instâncias de banco de dados. Você pode iniciar uma atualização de versão principal modificando manualmente sua instância de banco de dados. No entanto, antes de realizar uma atualização de versão principal, recomendamos seguir as instruções descritas em Atualização da versão principal do RDS para MariaDB.

Por outro lado, atualizações de versões secundárias incluem apenas alterações compatíveis com versões anteriores dos aplicativos existentes. Você pode iniciar uma atualização de versão secundária manualmente modificando sua instância de banco de dados. Ou é possível habilitar a opção Auto minor version upgrade (Atualização automática da versões secundárias) ao criar ou modificar uma instância de banco de dados. Isso significa que sua instância de banco de dados será atualizada automaticamente depois que o Amazon RDS testar e aprovar a nova versão. Para obter informações sobre como realizar uma atualização, consulte Atualizar a versão de mecanismo de uma instância de banco de dados.

Se a instância de banco de dados MariaDB estiver usando réplicas de leitura, atualize todas as réplicas de leitura antes de atualizar a instância de origem. Se a instância de banco de dados estiver em uma implantação multi-AZ, as réplicas do gravador e em espera serão atualizadas. A instância de banco de dados pode não estar disponível até que a atualização seja concluída.

Para obter mais informações sobre as versões do MariaDB com suporte e o gerenciamento de versões, consulte MariaDB em versões do Amazon RDS.

As atualizações do mecanismo de banco de dados necessitam de tempo de inatividade. A duração do tempo de inatividade varia de acordo com o tamanho de sua instância de banco de dados.

> **Tip**
>
> Você pode minimizar o tempo de inatividade necessário para a atualização da instância de banco de dados utilizando uma implantação azul/verde. Para ter mais informações, consulte Usar implantações azul/verde do Amazon RDS para atualizações de banco de dados.

Tópicos

- Considerações sobre as atualizações do MariaDB
- Encontrar destinos de atualização válidos
- Números de versão do MariaDB
- Números de versão do RDS no RDS para MariaDB
- Atualização da versão principal do RDS para MariaDB
- Atualizar uma instância de banco de dados MariaDB
- Atualizações automáticas da versão secundária do RDS para MariaDB
- Usar uma réplica de leitura para reduzir o tempo de inatividade ao fazer a atualização de um banco de dados do RDS para MariaDB

Considerações sobre as atualizações do MariaDB

O Amazon RDS usa dois ou mais snapshots de banco de dados durante o processo de upgrade. O Amazon RDS gera até dois snapshots da instância de banco de dados antes de fazer qualquer alteração de upgrade. Se o upgrade não funcionar para seus bancos de dados, você poderá restaurar um desses snapshots para criar uma instância de banco de dados executando a versão antiga. O Amazon RDS gera outro snapshot da instância de banco de dados quando o upgrade é concluído. O Amazon RDS obtém esses snapshots independentemente de o AWS Backup gerenciar ou não os backups da instância de banco de dados.

> **Note**
>
> O Amazon RDS só tirará snapshot de banco de dadoss se você tiver definido o período de retenção de backup para sua instância de banco de dados como um número maior que 0. Para alterar o período de retenção de backup, consulte Modificar uma instância de banco de dados do Amazon RDS.

Após a conclusão da atualização, você não pode reverter para a versão anterior do mecanismo de banco de dados. Se quiser retornar à versão anterior, restaure o primeiro DB snapshot tirado para criar outra instância de banco de dados.

Você controla quando atualizar sua instância do banco de dados para uma nova versão compatível com o Amazon RDS. Esse nível de controle ajuda você a manter a compatibilidade com versões

específicas de banco de dados e testar novas versões com seu aplicativo antes de implantá-lo na produção. Quando você estiver pronto, poderá executar as atualizações de versão quando achar melhor.

Se sua instância de banco de dados estiver usando uma replicação de leitura, atualize todas as réplicas de leitura antes de atualizar a instância de origem.

Se a sua instância de banco de dados estiver em uma implantação multi-AZ, as instâncias de banco de dados principal e em espera serão atualizadas. As instâncias de bancos de dados primária e de espera são atualizadas ao mesmo tempo, e há uma interrupção no serviço até a atualização ser concluída. O tempo para a interrupção varia de acordo com o mecanismo de banco de dados, a versão do mecanismo e o tamanho da instância de banco de dados.

Encontrar destinos de atualização válidos

Ao usar o AWS Management Console para atualizar uma instância de banco de dados, ele exibe os destinos de atualização válidos para a instância de banco de dados. Também é possível executar o seguinte comando da AWS CLI para identificar os destinos de atualizações válidos para uma instância de banco de dados:

Para Linux, macOS ou Unix:

```
aws rds describe-db-engine-versions \
  --engine mariadb \
  --engine-version version_number \
  --query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" --
output text
```

Para Windows:

```
aws rds describe-db-engine-versions ^
  --engine mariadb ^
  --engine-version version_number ^
  --query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" --
output text
```

Por exemplo, para identificar os destinos de atualização válidos de uma instância de banco de dados do MariaDB versão 10.5.17, execute o seguinte comando da AWS CLI:

Para Linux, macOS ou Unix:

```
aws rds describe-db-engine-versions \
  --engine mariadb \
  --engine-version 10.5.17 \
  --query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" --
output text
```

Para Windows:

```
aws rds describe-db-engine-versions ^
  --engine mariadb ^
  --engine-version 10.5.17 ^
  --query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" --
output text
```

Números de versão do MariaDB

A sequência de numeração de versões do mecanismo de banco de dados do RDS para MariaDB está no formato major.minor.patch.YYYYMMDD or major.minor.patch, por exemplo, 10.11.5.R2.20231201 ou 10.4.30. O formato usado depende da versão do mecanismo do MariaDB.

principal

O número da versão principal é o número inteiro e a primeira parte fracionária do número da versão, por exemplo, 10.11. Uma atualização de versão principal aumenta a parte principal do número da versão. Por exemplo, uma atualização de 10.5.20 para 10.6.12 é uma atualização de versão principal, em que 10.5 e 10.6 são os números da versão principal.

secundária

O número da versão secundária é a terceira parte do número da versão, por exemplo, 5 em 10.11.5.

patch

O patch é a quarta parte do número da versão, por exemplo, o R2 em 10.11.5.R2. Uma versão do patch do RDS inclui correções de bugs importantes adicionadas a uma versão secundária após o lançamento.

YYYYMMDD

O patch é a quinta parte do número da versão, por exemplo, 20231201 em 10.11.5.R2.20231201. Uma versão da data do RDS é um patch de segurança que inclui correções de segurança

importantes adicionadas a uma versão secundária após o lançamento. Ela não inclui nenhuma correção que possa mudar o comportamento de um mecanismo.

A tabela a seguir explica o esquema de nomenclatura do RDS para MariaDB versão 10.11.

10.11 versão secundária	Esquema da nomenclatura
≥5	Novas instâncias de banco de dados usam major.minor.patch. YYMMDD, por exemplo, 10.11.5.R2.20231201. As instâncias de banco de dados existentes podem usar major.minor.patch, por exemplo, 10.11.5.R2, até a próxima atualização de versão principal ou secundária.
< 5	As instâncias de banco de dados existentes usam major.min or.patch, por exemplo, 10.11.4.R2.

A tabela a seguir explica o esquema de nomenclatura do RDS para MariaDB versão 10.6.

10.6 versão secundária	Esquema da nomenclatura
≥ 14	Novas instâncias de banco de dados usam major.minor.patch. YYMMDD, por exemplo, 10.6.14.R2.20231201. As instâncias de banco de dados existentes podem usar major.minor.patch, por exemplo, 10.6.14.R2, até a próxima atualização de versão principal ou secundária.
< 14	As instâncias de banco de dados existentes usam major.min or.patch, por exemplo, 10.6.13.R2.

A tabela a seguir explica o esquema de nomenclatura do RDS para MariaDB versão 10.5.

10.5 versão secundária	Esquema da nomenclatura
≥ 21	Novas instâncias de banco de dados usam major.minor.patch. YYMMDD, por exemplo, 10.5.21.R2.20231201.

10.5 versão secundária	Esquema da nomenclatura
	As instâncias de banco de dados existentes podem usar major.minor.patch, por exemplo, 10.5.21.R2, até a próxima atualização de versão principal ou secundária.
< 21	As instâncias de banco de dados existentes usam major.minor.patch, por exemplo, 10.5.20.R2.

A tabela a seguir explica o esquema de nomenclatura do RDS para MariaDB versão 10.4.

10.4 versão secundária	Esquema da nomenclatura
≥ 30	Novas instâncias de banco de dados usam major.minor.patch.YYMMDD, por exemplo, 10.4.30.R2.20231201. As instâncias de banco de dados existentes podem usar major.minor.patch, por exemplo, 10.4.30.R2, até a próxima atualização de versão principal ou secundária.
< 30	As instâncias de banco de dados existentes usam major.minor.patch, por exemplo, 10.4.29.R2.

Números de versão do RDS no RDS para MariaDB

Os números de versão do RDS usam o esquema de nomenclatura *major*.*minor*.*patch* ou *major*.*minor*.*patch*.*YYYYMMDD*. Uma versão do patch do RDS inclui correções de bugs importantes adicionadas a uma versão secundária após o lançamento. Uma versão de data do RDS (*YYMMDD*) é um patch de segurança. Um patch de segurança não inclui nenhuma correção que possa mudar o comportamento do mecanismo.

Para identificar o número da versão do Amazon RDS do banco de dados, você deve primeiro criar a extensão `rds_tools` usando o seguinte comando:

```
CREATE EXTENSION rds_tools;
```

É possível descobrir o número da versão do banco de dados do RDS para MariaDB com a seguinte consulta SQL:

```
mysql> select mysql.rds_version();
```

Por exemplo, consultar um banco de dados do RDS para MariaDB 10.6.14 exibe a seguinte saída:

```
+---------------------+
| mysql.rds_version() |
+---------------------+
| 10.6.14.R2.20231201  |
+---------------------+
1 row in set (0.01 sec)
```

Atualização da versão principal do RDS para MariaDB

As atualizações da versão principal podem conter as alterações de banco de dados incompatíveis com as aplicações existentes. Como resultado, o Amazon RDS não aplica atualizações da versão principal automaticamente. É necessário modificar manualmente sua instância de banco de dados. Convém testar completamente qualquer atualização antes de aplicá-la às suas instâncias de produção.

O Amazon RDS oferece suporte para as seguintes atualizações in-loco para versões principais do mecanismo de banco de dados MariaDB:

- Qualquer versão do MariaDB para o MariaDB 11.4
- Qualquer versão do MariaDB para MariaDB 10.11
- Qualquer versão do MariaDB para MariaDB 10.6
- MariaDB 10.4 para MariaDB 10.5

Se você estiver usando um parameter group personalizado e realizar uma atualização para uma versão principal, deverá especificar um parameter group padrão para a nova versão do mecanismo de banco de dados ou criar seu próprio parameter group personalizado para essa nova versão. Associar o novo parameter group à instância de banco de dados requer que o banco de dados seja reinicializado pelo cliente depois que a atualização terminar. O status do parameter group da instância será `pending-reboot` se a instância precisar ser reinicializada para aplicar as alterações do parameter group. O status do grupo de parâmetros de uma instância pode ser visualizado

no AWS Management Console ou executando uma chamada “describe”, como `describe-db-instances`.

Atualizar uma instância de banco de dados MariaDB

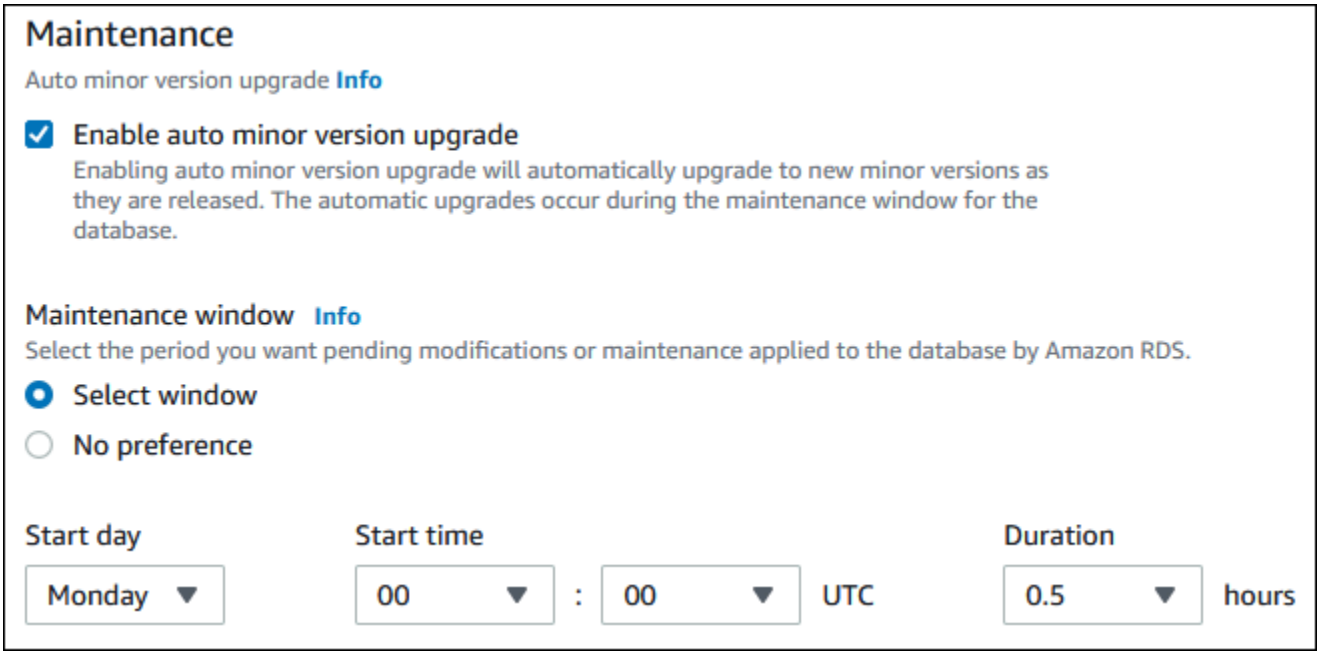
Para obter informações sobre como atualizar manual ou automaticamente uma instância de banco de dados MariaDB, consulte Atualizar a versão de mecanismo de uma instância de banco de dados.

Atualizações automáticas da versão secundária do RDS para MariaDB

Se você especificar as seguintes configurações ao criar ou modificar uma instância de banco de dados, poderá fazer com que sua instância de banco de dados seja atualizada automaticamente.

- A configuração Auto minor version upgrade (Atualização automática de versão secundária) está habilitada.
- A configuração Backup retention period (Período de retenção de backup) é maior que 0.

No AWS Management Console, essas configurações estão em Additional configuration (Configuração adicional). A imagem a seguir mostra a configuração Auto minor version upgrade (Atualização automática para versão secundária).



Para ter mais informações sobre essas configurações, consulte Configurações para instâncias de banco de dados.

Para algumas versões principais do RDS para MariaDB em algumas Regiões da AWS, uma versão secundária é designada pelo RDS como a versão de atualização automática. Depois que uma versão secundária foi testada e aprovada pelo Amazon RDS, a atualização de versão secundária ocorre automaticamente durante a janela de manutenção. O RDS não define automaticamente versões secundárias lançadas mais recentemente como a versão de atualização automática. Antes de o RDS designar uma versão de atualização automática, diversos critérios são considerados, como estes:

- Problemas de segurança conhecidos
- Bugs no MariaDB edição da comunidade
- Estabilidade geral de frota desde que a versão secundária foi lançada.

> **Note**
>
> O suporte para o uso do TLS nas versões 1.0 e 1.1 foi removido a partir de versões secundárias específicas do MariaDB. Consulte informações sobre as versões secundárias compatíveis do MariaDB em the section called “Suporte de SSL/TLS para o MariaDB”.

É possível executar o comando da AWS CLI a seguir para determinar a versão atual secundária de destino de atualização automática para uma versão secundária do MariaDB especificada em uma Região da AWS específica.

Para Linux, macOS ou Unix:

```
aws rds describe-db-engine-versions \
--engine mariadb \
--engine-version minor_version \
--region region \
--query "DBEngineVersions[*].ValidUpgradeTarget[*].
{AutoUpgrade:AutoUpgrade,EngineVersion:EngineVersion}" \
--output text
```

Para Windows:

```
aws rds describe-db-engine-versions ^
--engine mariadb ^
--engine-version minor_version ^
```

```
--region region ^
--query "DBEngineVersions[*].ValidUpgradeTarget[*].
{AutoUpgrade:AutoUpgrade,EngineVersion:EngineVersion}" ^
--output text
```

Por exemplo, o seguinte comando da AWS CLI determina o destino da atualização secundária automática para a versão secundária 10.5.16 do MariaDB na Região da AWS Leste dos EUA (Ohio) (us-east-2).

Para Linux, macOS ou Unix:

```
aws rds describe-db-engine-versions \
--engine mariadb \
--engine-version 10.5.16 \
--region us-east-2 \
--query "DBEngineVersions[*].ValidUpgradeTarget[*].
{AutoUpgrade:AutoUpgrade,EngineVersion:EngineVersion}" \
--output table
```

Para Windows:

```
aws rds describe-db-engine-versions ^
--engine mariadb ^
--engine-version 10.5.16 ^
--region us-east-2 ^
--query "DBEngineVersions[*].ValidUpgradeTarget[*].
{AutoUpgrade:AutoUpgrade,EngineVersion:EngineVersion}" ^
--output table
```

A saída é semelhante à seguinte.

```
---------------------------------
|    DescribeDBEngineVersions    |
+--------------+-----------------+
|  AutoUpgrade |  EngineVersion  |
+--------------+-----------------+
|  True        |  10.5.17        |
|  False       |  10.5.18        |
|  False       |  10.5.19        |
|  False       |  10.6.5         |
|  False       |  10.6.7         |
|  False       |  10.6.8         |
```

```
|  False       |  10.6.10        |
|  False       |  10.6.11        |
|  False       |  10.6.12        |
+--------------+-----------------+
```

Neste exemplo, o valor de `AutoUpgrade` é `True` para MariaDB versão 10.5.17. Então, o destino da atualização secundária automática é MariaDB versão 10.5.17, que está destacado na saída.

Uma instância de banco de dados MariaDB é atualizada automaticamente durante a janela de manutenção, caso os seguintes critérios sejam atendidos:

- A configuração Auto minor version upgrade (Atualização automática de versão secundária) está habilitada.
- A configuração Backup retention period (Período de retenção de backup) é maior que 0.
- A instância de banco de dados está executando uma versão de mecanismo de banco de dados secundária que é menor que a versão secundária de atualização automática.

Para obter mais informações, consulte Atualizar automaticamente a versão do mecanismo espelho.

Usar uma réplica de leitura para reduzir o tempo de inatividade ao fazer a atualização de um banco de dados do RDS para MariaDB

Na maioria dos casos, uma implantação azul/verde é a melhor opção para reduzir o tempo de inatividade ao atualizar uma instância de banco de dados MariaDB. Para ter mais informações, consulte Usar implantações azul/verde do Amazon RDS para atualizações de banco de dados.

Se não for possível usar uma implantação azul/verde e se sua instância de banco de dados MariaDB estiver em uso com uma aplicação de produção, você poderá usar o procedimento a seguir para atualizar a versão de banco de dados da sua instância de banco de dados. Esse procedimento pode reduzir o tempo de inatividade do seu aplicativo.

Usando uma réplica de leitura, você pode executar a maioria das etapas de manutenção antes do tempo e minimizar as alterações necessárias durante a interrupção real. Com essa técnica, você pode testar e preparar a nova instância de banco de dados sem fazer alterações em sua instância de banco de dados existente.

O procedimento a seguir mostra um exemplo de atualização do MariaDB versão 10.5 para o MariaDB versão 10.6. Você pode usar as mesmas etapas gerais nas atualizações para outras versões principais.

Como atualizar um banco de dados MariaDB enquanto uma instância de banco de dados está em uso

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. Crie uma réplica de leitura de sua instância de banco de dados do MariaDB 10.5. Esse processo cria uma cópia atualizável do seu banco de dados. Outras réplicas de leitura da instância de banco de dados também podem existir.

   a. No console, escolha Databases (Bancos de dados) e depois escolha a instância de banco de dados que você deseja atualizar.
   b. Em Actions (Ações), escolha Create read replica (Criar réplica de leitura).
   c. Forneça o valor do DB instance identifier (Identificador de instância de banco de dados) para a réplica de leitura e verifique se a DB instance class (Classe da instância de banco de dados) e as outras configurações correspondem à instância de banco de dados MariaDB 10.5.
   d. Escolha Create read replica (Criar réplica de leitura).
3. (Opcional) Quando a réplica de leitura for criada e Status for exibida Available (Disponível), converta a réplica de leitura em uma implantação multi-AZ e ative backups.

   Por padrão, uma réplica de leitura é criada como uma implantação single-AZ com backups desativados. Como a réplica de leitura se torna a instância de banco de dados de produção, uma prática recomendada é configurar uma implantação multi-AZ e habilitar backups agora.

   a. No console, selecione Databases (Bancos de dados) e a réplica de leitura que você acabou de criar.
   b. Selecione Modify.
   c. Para Multi-AZ deployment (Implantação multi-AZ), escolha Create a standby instance (Criar uma instância em espera).
   d. Em Backup Retention Period (Período de retenção de backup), escolha um valor positivo diferente de zero, como 3 dias, depois selecione Continue (Continuar).
   e. Em Scheduling of modifications (Programação de modificações), selecione Apply immediately (Aplicar imediatamente).
   f. Selecione Modify DB instance (Modificar instância de banco de dados).
4. Quando o Status da réplica de leitura mostrar Available (Disponível), atualize a réplica de leitura para o MariaDB 10.6.

a. No console, selecione Databases (Bancos de dados) e a réplica de leitura que você acabou de criar.

b. Selecione Modify.

c. Em DB engine version (Versão do mecanismo de banco de dados), selecione a versão do MariaDB 10.6 para a qual deseja atualizar, depois escolha Continue (Continuar).

d. Em Scheduling of modifications (Programação de modificações), selecione Apply immediately (Aplicar imediatamente).

e. Selecione Modify DB instance (Modificar instância de banco de dados) para iniciar a atualização.

5. Quando a atualização for concluída e o Status mostrar Available (Disponível), verifique se a réplica de leitura está atualizada em relação à instância de banco de dados MariaDB 10.5 de origem. Para verificar, conecte-se à réplica de leitura e execute o comando `SHOW REPLICA STATUS`. Se o campo `Seconds_Behind_Master` for `0`, a replicação está atualizada.

> **Note**
>
> Versões anteriores do MariaDB usavam `SHOW SLAVE STATUS` em vez de `SHOW REPLICA STATUS`. Se você estiver usando uma versão do MariaDB anterior à 10.6, use `SHOW SLAVE STATUS`.

6. (Opcional) Crie uma réplica de leitura da réplica de leitura.

   Se você quiser que a instância de banco de dados tenha uma réplica de leitura depois que ela for promovida para uma instância de banco de dados autônoma, você pode criar a réplica de leitura agora.

   a. No console, selecione Databases (Bancos de dados) e selecione a réplica de leitura que você acabou de atualizar.

   b. Em Actions (Ações), escolha Create read replica (Criar réplica de leitura).

   c. Forneça o valor do DB instance identifier (Identificador de instância de banco de dados) para a réplica de leitura e verifique se a DB instance class (Classe da instância de banco de dados) e as outras configurações correspondem à instância de banco de dados MariaDB 10.5.

   d. Escolha Create read replica (Criar réplica de leitura).

7. (Opcional) Configure um grupo de parâmetros de banco de dados personalizado para a réplica de leitura.

   Se você quiser que a instância de banco de dados use um grupo de parâmetros personalizado depois de ser promovida para uma instância de banco de dados autônoma, poderá criar o grupo de parâmetros de banco de dados agora pode associá-lo à réplica de leitura.

   a. Crie um grupo de parâmetros de banco de dados personalizado para o MariaDB 10.6. Para obter instruções, consulte Criar um grupo de parâmetros de banco de dados no Amazon RDS.

   b. Modifique os parâmetros que você deseja alterar no grupo de parâmetros de banco de dados que você acabou de criar. Para obter instruções, consulte Modificar parâmetros em um grupo de parâmetros de banco de dados no Amazon RDS.

   c. No console, selecione Databases (Bancos de dados) e a réplica de leitura.

   d. Selecione Modify.

   e. Em DB parameter group (Grupo de parâmetros do banco de dados), escolha o grupo de parâmetros do banco de dados MariaDB 10.6 que você acabou de criar, depois selecione Continue (Continuar).

   f. Em Scheduling of modifications (Programação de modificações), selecione Apply immediately (Aplicar imediatamente).

   g. Selecione Modify DB instance (Modificar instância de banco de dados) para iniciar a atualização.

8. Torne sua réplica de leitura do MariaDB 10.6 uma instância de banco de dados autônoma.

   Important

   Quando você promove a réplica de leitura do MariaDB 10.6 para uma instância de banco de dados autônoma, ela deixa de ser uma réplica da instância de banco de dados MariaDB 10.5. Recomendamos que você promova a réplica de leitura do MariaDB 10.6 durante uma janela de manutenção, quando a instância de banco de dados MariaDB 10.5 de origem estiver no modo somente leitura e todas as operações de gravação estiverem suspensas. Quando a promoção terminar, você poderá direcionar suas operações de gravação para a instância de banco de dados MariaDB 10.6 atualizada, para garantir que nenhuma operação de gravação seja perdida.
   Além disso, recomendamos que, antes de promover a réplica de leitura do MariaDB 10.6, você realize todas as operações de DDL (linguagem de definição de dados)

necessárias na réplica de leitura do MariaDB 10.6. Um exemplo é a criação de índices. Essa abordagem evita efeitos negativos sobre a performance da réplica de leitura do MariaDB 10.6 depois da promoção. Para promover uma réplica de leitura, use o procedimento a seguir.

a. No console, selecione Databases (Bancos de dados) e selecione a réplica de leitura que você acabou de atualizar.
b. Em Actions (Ações), selecione Promote (Promover).
c. Selecione Yes (Sim) para habilitar backups automatizados da instância da réplica de leitura. Para obter mais informações, consulte Introdução aos backups.
d. Escolha Continue.
e. Escolha Promote Read Replica (Promover réplica de leitura).

9. Agora, você possui uma versão atualizada do seu banco de dados MariaDB. Neste ponto, você pode direcionar as aplicações para a nova instância de banco de dados MariaDB 10.6.

Atualizar uma versão do mecanismo de snapshot de banco de dados do MariaDB

Com o Amazon RDS, você pode criar um snapshot de banco de dados do volume de armazenamento da instância de banco de dados do MariaDB. Ao criar um snapshot de banco de dados, ele é baseado na versão do mecanismo usada pela instância de banco de dados. É possível fazer a atualização da versão do mecanismo para os snapshots de banco de dados.

No caso do RDS para MariaDB, você pode atualizar para todas as versões de mecanismo disponíveis. É possível atualizar snapshots de banco de dados criptografados ou não criptografados.

Para visualizar as versões de mecanismo disponíveis para o snapshot de banco de dados do RDS para MariaDB, use o exemplo da AWS CLI a seguir.

```
aws rds describe-db-engine-versions --engine mariadb --include-all --engine-
version example-engine-version --query "DBEngineVersions[*].ValidUpgradeTarget[*].
{EngineVersion:EngineVersion}" --output text
```

Se você não vir resultados para o snapshot, a versão do seu mecanismo pode estar obsoleta. Se a versão do mecanismo tiver sido descontinuada, recomendamos que você atualize para o destino de atualização da versão principal mais recente ou para um dos outros destinos de atualização disponíveis para essa versão. Para ter mais informações, consulte Opções de atualização para snapshots de banco de dados com versões de mecanismo incompatíveis para o RDS para MariaDB.

Após restaurar um snapshot de banco de dados atualizado para uma nova versão do mecanismo, certifique-se de testar se a atualização foi bem-sucedida. Para obter mais informações sobre uma atualização de versão principal, consulte the section called “Atualizações do mecanismo de banco de dados do MariaDB”. Para saber como restaurar um snapshot de banco de dados, consulte the section called “Restaurar uma instância de banco de dados”.

> **Note**
>
> Não é possível atualizar snapshots de banco de dados automatizados criados durante o processo de backup automatizado.

É possível atualizar um snapshot de banco de dados usando o AWS Management Console, a AWS CLI ou a API do RDS.

Console

Para atualizar uma versão do mecanismo de snapshot de banco de dados usando o AWS Management Console, use o procedimento a seguir.

Para atualizar um snapshot de banco de dados

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, selecione Snapshots.
3. Escolha o snapshot que você deseja atualizar.
4. Em Actions (Ações), selecione Upgrade snapshot (Atualizar snapshot). A página Upgrade snapshot (Atualizar snapshot) é exibida.
5. Escolha a opção New engine version (Nova versão de mecanismo) para a qual atualizar.
6. Escolha Save changes (Salvar alterações) para atualizar o snapshot.

   Durante o processo de atualização, todas as ações do snapshot serão desabilitadas para esse snapshot de banco de dados. Além disso, o status do snapshot de banco de dados muda de Disponível para Fazendo upgrade e, depois, muda para Ativo após a conclusão. Se não for possível atualizar o snapshot de banco de dados devido a problemas de corrupção do snapshot, o status mudará para Indisponível. Não é possível recuperar o snapshot desse estado.

   > **Note**
   >
   > Se a atualização do snapshot de banco de dados falhar, o snapshot será revertido para o estado original com a versão original.

AWS CLI

Para atualizar um snapshot de banco de dados para uma nova versão do mecanismo de banco de dados, execute o comando modify-db-snapshot da AWS CLI.

Opções

- `--db-snapshot-identifier` – o identificador do snapshot de banco de dados a ser atualizado. O identificador deve ser um nome de recurso da Amazon (ARN) exclusivo. Para obter mais informações, consulte Nomes de recurso da Amazon (ARNs) no Amazon RDS.

- `--engine-version` – a versão do mecanismo para a qual será feita a atualização do snapshot de banco de dados.

Example

Para Linux, macOS ou Unix:

```
aws rds modify-db-snapshot \

    --db-snapshot-identifier my_db_snapshot \
    --engine-version new_version
```

Para Windows:

```
aws rds modify-db-snapshot ^
    --db-snapshot-identifier my_db_snapshot ^
    --engine-version new_version
```

Amazon RDS API

Para atualizar um snapshot de banco de dados para uma nova versão do mecanismo de banco de dados, chame a operação ModifyDBSnapshot da API do RDS.

Parâmetros

- `DBSnapshotIdentifier` – o identificador do snapshot de banco de dados a ser atualizado. O identificador deve ser um nome de recurso da Amazon (ARN) exclusivo. Para obter mais informações, consulte Nomes de recurso da Amazon (ARNs) no Amazon RDS.
- `EngineVersion` – a versão do mecanismo para a qual será feita a atualização do snapshot de banco de dados.

Opções de atualização para snapshots de banco de dados com versões de mecanismo incompatíveis para o RDS para MariaDB

A tabela a seguir mostra para quais versões do mecanismo é possível fazer a atualização de uma versão de mecanismo incompatível com snapshots de banco de dados do RDS para MariaDB.

> **Note**
>
> Talvez você precise atualizar o snapshot de banco de dados mais de uma vez para fazer a atualização para a versão do mecanismo escolhida.

Versão do mecanismo do snapshot de banco de dados	Versões do mecanismo disponíveis para atualização
10.0.17, 10.0.24, 10.0.28, 10.0.31, 10.0.32, 10.0.34, 10.0.35	10.0.38, 10.1.48, 10.6.19, 10.6.20, 10.11.9, 10.11.10, 11.4.3, 11.4.4
10.1.14, 10.1.16, 10.1.19, 10.1.23, 10.1.26, 10.1.31, 10.1.34	10.1.48, 10.2.44, 10.6.19, 10.6.20, 10.11.9, 10.11.10, 11.4.3, 11.4.4
10.2.11, 10.2.12, 10.2.15, 10.2.21, 10.2.32, 10.2.37, 10.2.39, 10.2.40, 10.2.41, 10.2.43	10.2.44, 10.3.39, 10.6.19, 10.6.20, 10.6.21, 10.11.9, 10.11.10, 10.11.11, 11.4.3, 11.4.4
10.3.8, 10.3.13, 10.3.20, 10.3.23, 10.3.28, 10.3.31, 10.3.32	10.3.39, 10.4.34, 10.6.19, 10.6.20, 10.6.21, 10.11.9, 10.11.10, 10.11.11, 11.4.3, 11.4.4
10.4.8	10.4.34, 10.5.16, 10.5.17, 10.5.18, 10.5.20, 10.5.21, 10.5.22, 10.5.23, 10.5.24, 10.5.25, 10.5.26, 10.5.27, 10.5.28, 10.6.8, 10.6.10, 10.6.11, 10.6.13, 10.6.14, 10.6.15, 10.6.16, 10.6.18, 10.6.19, 10.6.20, 10.6.21, 10.11.4, 10.6.17, 10.11.5, 10.11.6, 10.11.7, 10.11.8, 10.11.9, 10.11.10, 10.11.11

Importar dados para uma instância de banco de dados do Amazon RDS para MariaDB

Você pode usar várias técnicas diferentes para importar os dados para uma instância de banco de dados do RDS para MariaDB. A melhor abordagem depende de inúmeros fatores:

- origem dos dados;
- Quantidade de dados
- importação única ou contínua;
- quantidade de tempo de inatividade.

Se você também estiver migrando uma aplicação com os dados, é importante considerar a quantidade de tempo de inatividade.

A tabela a seguir lista técnicas para importar dados em uma instância de banco de dados do RDS para MariaDB.

> **Note**
>
> O Amazon RDS é compatível somente com a importação do Amazon S3 para instâncias de banco de dados do RDS para MySQL. Atualmente, não há suporte para a importação de backups criados com `mariadb-backup` no RDS para MariaDB.

Origem	Quantidade de dados	Uma vez ou contínuo	Tempo de inatividade da aplicação	Técnica	Mais informações
Banco de dados do MariaDB existente	Any	Contínuo	Mínimo	Configure a replicação com um banco de dados MariaDB existente como origem da replicação. É possível configurar a replicação em uma instância de banco de dados do	Configurar a replicação da posição do

Origem	Quantidade de dados	Uma vez ou contínuo	Tempo de inatividade da aplicação	Técnica	Mais informações
on-premises ou no Amazon EC2				MariaDB usando identificadores de transações globais (GTIDs) do MariaDB quando a instância externa for o MariaDB versão 10.0.24 ou posterior, ou usando coordenadas de log binário para instâncias do MariaDB em versões anteriores à 10.0.24. Os GTIDs do MariaDB são implantados de forma diferente dos GTIDs do MySQL, os quais não são compatíveis com o Amazon RDS.	arquivo de log binário com uma instância de origem externa Importar dados para uma instância de banco de dados do Amazon RDS para MariaDB com tempo de inatividade reduzido

Origem	Quantidade de dados	Uma vez ou contínuo	Tempo de inatividade da aplicação	Técnica	Mais informações
Qualquer banco de dados existente	Quaisquer	Uma vez ou contínuo	Mínimo	Use o AWS Database Migration Service para migrar o banco de dados com o mínimo de tempo de inatividade e, para diversos mecanismos de banco de dados (DB), continuar a replicação contínua.	What is AWS Database Migration Service (O que é o) e Using a MySQL-compatible database as a target for AWS DMS (Usar um banco de dados compatível com MySQL como destino para o) no Guia do

Origem	Quantidade de dados	Uma vez ou contínuo	Tempo de inatividade da aplicação	Técnica	Mais informações
					usuário do AWS Database Migration Service
Instância de banco de dados MariaDB existente	Any	Uma vez ou contínuo	Mínimo	Crie uma réplica de leitura para a replicação contínua. Promova a réplica de leitura para a criação única de uma instância de banco de dados.	Trabalhar com réplicas de leitura de instância de banco de dados

Origem	Quantidade de dados	Uma vez ou contínuo	Tempo de inatividade da aplicação	Técnica	Mais informações
Banco de dados existente do MariaDB	Pequeno	Uma vez	Alguns	Copie os dados diretamente para a instância de banco de dados do MariaDB usando um utilitário da linha de comando.	Importar dados de um banco de dados externo do MariaDB para uma instância de banco de dados do Amazon RDS para MariaDB

Origem	Quantidade de dados	Uma vez ou contínuo	Tempo de inatividade da aplicação	Técnica	Mais informações
Dados não armazenados em um banco de dados existente	Médio	Uma vez	Alguns	Crie arquivos simples e importe-os usando instruções `LOAD DATA LOCAL INFILE` do MariaDB.	Importar dados de qualquer fonte para uma instância de banco de dados do Amazon RDS para MariaDB

> **Note**
>
> O banco de dados do sistema `mysql` contém informações de autenticação e autorização necessárias para fazer login na instância de banco de dados e acessar os dados. Descartar, alterar, renomear ou truncar tabelas, dados ou outros conteúdos do banco de dados `mysql` na instância de banco de dados pode resultar em erros e pode tornar a instância de banco de dados e os dados inacessíveis. Se isso ocorrer, você pode restaurar a instância de banco de dados a partir de um snapshot usando o comando AWS CLI da restore-db-instance-from-db-snapshot. É possível recuperar a instância de banco de dados usando o comando restore-db-instance-to-point-in-time.

Considerações sobre a importação de dados para o MariaDB

O conteúdo a seguir apresenta informações técnicas relacionadas ao carregamento de dados no MariaDB. Este conteúdo é destinado a usuários que estão familiarizados com a arquitetura do servidor MariaDB.

registro em log binário

A habilitação do registro em log binário reduz a performance do carregamento de dados e requer até quatro vezes mais espaço em disco em comparação com o registro em log desabilitado. O tamanho da transação usada para carregar os dados afeta diretamente a performance do sistema e as necessidades de espaço em disco. As transações maiores exigem mais recursos.

Tamanho da transação

O tamanho da transação influencia os seguintes aspectos dos carregamentos de dados do MariaDB:

- Consumo de recursos
- Utilização de espaço em disco
- Processo de retomada
- Tempo de recuperação
- Formato de entrada (arquivos simples ou SQL)

Esta seção descreve como o tamanho da transação afeta o registro em log binário e justifica a desabilitação do registro em log binário durante grandes cargas de dados. Você pode habilitar e desabilitar o registro em log binário configurando o período de retenção de backup automatizado do Amazon RDS. Valores diferentes de zero habilitam o registro em log binário, enquanto um valor de zero o desabilita. Para ter mais informações, consulte Backup retention period (Período de retenção de backup).

Esta seção descreve também o impacto de grandes transações sobre o InnoDB e por que é importante que o tamanho das transações seja pequeno.

Transações pequenas

Para pequenas transações, o registro em log binário duplica o número de gravações em disco necessárias para carregar os dados. Esse efeito pode degradar severamente a performance de outras sessões de banco de dados e aumentar o tempo necessário para carregar os dados. A degradação experimentada depende em parte dos seguintes fatores:

- Taxa de upload
- Outra atividade do banco de dados que ocorre durante o carregamento
- Capacidade da instância de banco de dados do Amazon RDS

Os logs binários também consomem um espaço em disco aproximadamente igual à quantidade de dados carregados enquanto os logs são copiados e removidos. O Amazon RDS minimiza isso fazendo backup e removendo os logs binários com frequência.

Transações grandes

Quanto a transações grandes, o registro em log binário triplica as IOPS e o uso do disco pelos seguintes motivos:

- O cache de log binário armazena os dados da transação temporariamente no disco.
- Esse cache se amplia de acordo com o tamanho da transação, o que consome espaço em disco.
- Quando a transação (de confirmação ou reversão) é concluída, o sistema copia o cache para o log binário.

Esse processo cria três cópias dos dados:

- os dados originais;
- o cache no disco;
- a entrada final do log binário.

Cada operação de gravação incorre em E/S adicional, afetando ainda mais a performance.

Por isso, o registro em log binário requer o triplo de espaço em disco em comparação com o registro desabilitado. Por exemplo, carregar 10 GiB de dados como uma única transação cria três cópias:

- 10 GiB para os dados da tabela;
- 10 GiB para o cache de log binário;
- 10 GiB para o arquivo de log binário.

O total de espaço em disco temporário necessário é 30 GiB.

Considerações importantes sobre o espaço em disco:

- O arquivo de cache persiste até que a sessão termine ou uma nova transação crie outro cache.
- O log binário permanece até o momento em que ele é copiado, possivelmente mantendo 20 GiB (cache e log) por um período prolongado.

Se você usar `LOAD DATA LOCAL INFILE` para carregar os dados, a recuperação de dados criará uma quarta cópia caso o banco de dados tenha que ser recuperado de um backup feito antes do carregamento. Durante a recuperação, o MariaDB extrai dados de log binário em um arquivo simples. Em seguida, o MariaDB executa `LOAD DATA LOCAL INFILE`. Com base no exemplo anterior, essa recuperação requer um total de espaço em disco temporário de 40 GiB ou 10 GiB cada para tabela, cache, log e arquivo local. Sem pelo menos 40 GiB de espaço livre em disco, a recuperação falhará.

Otimizar grandes carregamentos de dados

No caso de grandes carregamentos de dados, desabilite o registro em log binário para reduzir os custos indiretos e os requisitos de espaço em disco. Você pode desabilitar o registro em log binário definindo o período de retenção de backup como 0. Após o término do carregamento, substitua o período de retenção de backup por um valor apropriado diferente de zero. Para ter mais informações, consulte Modificar uma instância de banco de dados do Amazon RDS e Período de retenção de backup na tabela de configurações.

> **Note**
>
> Se a instância de banco de dados for uma instância de banco de dados de origem para réplicas de leitura, não será possível definir o período de retenção de backup como 0.

Antes de carregar os dados, recomendamos criar um snapshot do banco de dados. Para ter mais informações, consulte Gerenciar backups manuais.

InnoDB

As informações a seguir sobre o registro em log de desfazer e as opções de recuperação ajudam a manter as transações do InnoDB pequenas para otimizar a performance do banco de dados.

Conceitos básicos sobre o registro em log de desfazer no InnoDB

Desfazer é um mecanismo de registro em log que permite a reversão de transações e comporta o controle de simultaneidade de várias versões (MVCC).

Para o MariaDB 10.11 e versões anteriores, os logs de desfazer são armazenados no espaço de tabela do sistema InnoDB (geralmente ibdata1) e são retidos até que o thread de limpeza os remova. Por esse motivo, grandes transações de carregamento de dados podem fazer com que o espaço de tabela do sistema se torne muito grande e consuma espaço em disco que não pode ser recuperado sem recriar o banco de dados.

Para todas as versões do MariaDB, o thread de limpeza só deve remover quaisquer logs de desfazer depois que a transação ativa mais antiga for confirmada ou revertida. Se o banco de dados estiver processando outras transações durante o carregamento, os logs de desfazer também se acumularão e não poderão ser removidos, mesmo que as transações sejam confirmadas e nenhuma outra precise dos logs de desfazer para o MVCC. Nessa situação, todas as transações, incluindo as transações somente leitura, ficam mais lentas. Essa desaceleração ocorre porque todas as transações acessam todas as linhas que qualquer transação, não apenas a transação de carregamento, altera. Na prática, as transações devem examinar os logs de desfazer cuja eliminação foi impedida pelas transações de carregamento de longa duração durante uma limpeza de logs de desfazer. Isso afeta a performance de qualquer operação que acesse linhas modificadas.

Opções de recuperação de transações do InnoDB

Embora o InnoDB otimize as operações de confirmação, a reversão de grandes transações é lenta. Para uma recuperação mais rápida, execute uma recuperação para um ponto no tempo ou restaure um snapshot do banco de dados. Para obter mais informações, consulte Recuperação para um ponto no tempo e Restaurar uma instância de banco de dados.

Formatos de importação de dados

O MariaDB aceita dois formatos de importação de dados: arquivos simples e SQL. Analise as informações sobre cada formato para determinar a melhor opção para suas necessidades.

Arquivos simples

Para transações pequenas, carregue arquivos simples com `LOAD DATA LOCAL INFILE`. Esse formato de importação de dados pode oferecer os seguintes benefícios em relação ao uso do SQL:

- regras de tráfego de rede;
- custos de transmissão de dados mais baixos;
- redução dos custos indireto de processamento do banco de dados;
- processamento de fluxos.

`LOAD DATA LOCAL INFILE` carrega todo o arquivo simples como uma única transação. Mantenha os arquivos individuais com um tamanho pequeno para obter as seguintes vantagens:

- Capacidade de retomar: é fácil manter o controle de quais arquivos foram carregados. Se surgir um problema durante o carregamento, você poderá continuar de onde parou. Você pode precisar retransmitir alguns dados ao Amazon RDS, mas, com arquivos pequenos, a quantidade retransmitida é mínima.
- Carregamento de dados em paralelo: se você tiver IOPS e largura de banda da rede suficientes para carregamento de um único arquivo, o carregamento em paralelo pode economizar tempo.
- Controle da taxa de carregamento: se o carregamento de dados tiver um impacto negativo em outros processos, você poderá controlar a taxa de carregamento aumentando o intervalo entre os arquivos.

Grandes transações reduzem os benefícios de usar `LOAD DATA LOCAL INFILE` para importar dados. Quando você não conseguir dividir uma grande quantidade de dados em arquivos menores, considere a possibilidade de usar o SQL.

SQL

O SQL tem uma vantagem fundamental em relação aos arquivos simples: é fácil manter as transações com um tamanho pequeno. Contudo, o tempo de carregamento do SQL pode ser significativamente mais longo do que o dos arquivos simples. Além disso, após uma falha, pode ser difícil determinar onde retomar. Não é possível reiniciar os arquivos mariadb-dump. Se ocorrer uma falha ao carregar um arquivo mariadb-dump, você deve modificá-lo ou substituí-lo para que o carregamento possa continuar. Ou, alternativamente, depois de corrigir a causa da falha, você pode restaurar para o ponto no tempo anterior ao carregamento e reenviar o arquivo. Para ter mais informações, consulte Recuperação para um ponto no tempo.

Usar snapshots de banco de dados do Amazon RDS para pontos de verificação de banco de dados

Se você carrega dados por longos períodos, como horas ou dias, sem registro em log binário, use snapshots de banco de dados para oferecer pontos de verificação periódicos em prol da segurança dos dados. Cada snapshot de banco de dados cria uma cópia consistente da instância de banco de dados, que serve como ponto de recuperação durante falhas do sistema ou eventos de corrupção de dados. Como os snapshots de banco de dados são rápidos, os pontos de verificação frequentes têm um impacto mínimo na performance do carregamento. Você pode excluir snapshots de banco de

dados anteriores sem afetar a durabilidade ou os recursos de recuperação do banco de dados. Para ter mais informações sobre snapshots de banco de dados, consulte Gerenciar backups manuais.

Reduzir os tempos de carregamento do banco de dados

Os seguintes itens são dicas adicionais para reduzir os tempos de carregamento:

- Crie todos os índices secundários antes de carregar dados em bancos de dados do MariaDB. Ao contrário de outros sistemas de banco de dados, o MariaDB reconstrói a tabela inteira ao adicionar ou modificar índices secundários. Esse processo cria outra tabela com alterações no índice, copia todos os dados e descarta a tabela original.
- Carregue os dados na ordem da chave primária. Quanto a tabelas do InnoDB, isso pode reduzir os tempos de carregamento em 75% a 80% e diminuir o tamanho do arquivo de dados em 50%.
- Desabilite as restrições de chave estrangeira definindo `foreign_key_checks` como `0`. Isso com frequência é necessário para arquivos simples carregados com `LOAD DATA LOCAL INFILE`. Com relação a qualquer carregamento, desabilitar as verificações de chave estrangeira acelera o carregamento de dados. Após a conclusão do carregamento, habilite novamente as restrições configurando `foreign_key_checks` como 1 e verifique os dados.
- Carregue os dados em paralelo, a menos que esteja se aproximando de um limite de recursos. Para habilitar o carregamento simultâneo em vários segmentos da tabela, use tabelas particionadas quando apropriado.
- Para reduzir os custos indiretos de execução do SQL, combine várias instruções `INSERT` em operações `INSERT` únicas de vários valores. `mariadb-dump` implementa essa otimização automaticamente.
- Reduza as operações de E/S de log do InnoDB configurando `innodb_flush_log_at_trx_commit` como `0`. Após a conclusão do carregamento, restaure `innodb_flush_log_at_trx_commit` para 1.

> **Warning**
>
> Ao definir `innodb_flush_log_at_trx_commit` como `0`, o InnoDB descarrega os respectivos logs a cada segundo em vez de a cada confirmação. Essa configuração aumenta a performance, mas pode gerar risco de perda de transações durante falhas no sistema.

- Se você estiver carregando dados em uma instância de banco de dados que não tem réplicas de leitura, defina `sync_binlog` como `0`. Após a conclusão do carregamento, restaure `sync_binlog parameter` para `1`.
- Carregue os dados antes de converter a instância de banco de dados em uma implantação multi-AZ. Se a instância de banco de dados já utiliza uma implantação multi-AZ, não recomendamos mudar para uma implantação single-AZ para carregamento de dados. Se fizer isso, obterá apenas pequenas melhorias.

Importar dados de um banco de dados externo do MariaDB para uma instância de banco de dados do Amazon RDS para MariaDB

Também é possível importar dados de um banco de dados MySQL ou MariaDB existente para uma instância de banco de dados do RDS para MariaDB ou do RDS para MySQL. Para isso, copie o banco de dados com mysqldump e conecte-o diretamente à instância de banco de dados do RDS para MariaDB ou do RDS para MySQL. O utilitário de linha de comando `mysqldump` é comumente usado para fazer backups e transferir dados de um servidor MariaDB ou MySQL para outro. Ele está incluído no software cliente MySQL e MariaDB.

No MariaDB 10.5, o cliente é chamado de mariadb-dump. A partir do MariaDB 11.0.1, você deve usar `mariadb-dump` em vez de `mysqldump`.

> **Note**
>
> Se você estiver importando ou exportando grandes volumes de dados com uma instância de banco de dados do MySQL, será mais confiável e mais rápido mover dados para dentro e para fora do Amazon RDS usando arquivos de backup `xtrabackup` e o Amazon S3. Para ter mais informações, consulte Restauração de um backup em uma instância de banco de dados MySQL.
>
> O Amazon RDS não é compatível com `xtrabackup` para MariaDB ou `mariabackup`. O Amazon RDS é compatível somente com a importação do Amazon S3 para MySQL.

Um comando `mysqldump` típico para mover dados de um banco de dados externo para uma instância de banco de dados do Amazon RDS é semelhante ao exemplo a seguir. Substitua os valores por suas próprias informações. Para o MariaDB 11.0.1 e versões posteriores, substitua `mysqldump` por `mariadb-dump` e `mysql` por `mariadb`.

```
mysqldump -u local_user \
    --databases database_name \
    --single-transaction \
    --compress \
    --order-by-primary  \
    --routines=0 \
    --triggers=0 \
    --events=0 \
    -plocal_password | mysql -u RDS_user \
        --port=port_number \
        --host=host_name \
        -pRDS_password
```

> **Important**
>
> Não deixe um espaço entre a opção `-p` e a senha inserida.
> Faz parte das práticas recomendadas de segurança especificar credenciais diferentes dos prompts mostrados neste exemplo.

Analise as seguintes recomendações e considerações:

- Exclua os seguintes esquemas do arquivo de despejo:
  - `sys`
  - `performance_schema`
  - `information_schema`

  O utilitário `mysqldump` e `mariadb-dump` exclui esses esquemas por padrão.
- Para migrar usuários e privilégios, considere o uso de uma ferramenta que gera a linguagem de controle de dados (DCL) para recriá-los, como o utilitário pt-show-grants.
- Para realizar a importação, verifique se o usuário que está fazendo isso tem acesso à instância de banco de dados. Para obter mais informações, consulteControlar acesso com grupos de segurança

Os parâmetros utilizados são os seguintes:

- `-u local_user`: use para especificar um nome do usuário. Ao usar esse parâmetro pela primeira vez, especifique o nome de uma conta de usuário no banco de dados local MariaDB ou MySQL que você identifica com o parâmetro `--databases`.
- `--databases database_name`: use para especificar o nome do banco de dados na instância do MySQL ou MariaDB local que você deseja importar para o Amazon RDS.
- `--single-transaction` – use para garantir que todos os dados carregados do banco de dados local sejam consistentes com um único ponto no tempo. Se houver outros processos alterando os dados enquanto eles são lidos por `mysqldump`, usar esse parâmetro ajudará a manter a integridade dos dados.
- `--compress` – use para reduzir o consumo de largura de banda da rede, compactando os dados do banco de dados local antes de os enviar para o Amazon RDS.
- `--order-by-primary` – use para reduzir o tempo de carregamento, ordenando os dados de cada tabela de acordo com a chave primária.
- `--routines`: use se houver determinadas rotinas, como funções ou procedimentos armazenados, no banco de dados que você está copiando. Defina o parâmetro como `0`, o que exclui as rotinas durante o processo de importação. Depois, recrie manualmente as rotinas no banco de dados do Amazon RDS.
- `--triggers`: use se houver gatilhos no banco de dados que você está copiando. Defina o parâmetro como `0`, o que exclui os gatilhos durante o processo de importação. Depois, recrie manualmente os gatilhos no banco de dados do Amazon RDS.
- `--events`: use se houver eventos no banco de dados que você está copiando. Defina o parâmetro como `0`, o que exclui os eventos durante o processo de importação. Em seguida, recrie manualmente os eventos no banco de dados do Amazon RDS.
- `-plocal_password`: use para especificar uma senha. Ao usar esse parâmetro pela primeira vez, especifique a senha da conta de usuário que você identifica com o primeiro parâmetro `-u`.
- `-u RDS_user`: use para especificar um nome do usuário. Ao usar esse parâmetro pela segunda vez, especifique o nome de uma conta de usuário no banco de dados padrão para a instância de banco de dados MariaDB ou MySQL que você identifica com o parâmetro `--host`.
- `--port port_number`: use para especificar a porta da sua instância de banco de dados MariaDB ou MySQL. Por padrão, essa porta é 3306, a menos que você tenha alterado o valor ao criar a instância de banco de dados.
- `--host host_name`: use para especificar o nome do Sistema de Nome de Domínio (DNS) do endpoint da instância de banco de dados Amazon RDS, por exemplo,

`myinstance.123456789012.us-east-1.rds.amazonaws.com`. Você pode encontrar o valor do endpoint nos detalhes da instância do banco de dados no console do Amazon RDS.

- `-p`*`RDS_password`*: use para especificar uma senha. Na segunda utilização desse parâmetro, você especifica a senha da conta de usuário identificada pelo segundo parâmetro `-u`.

Certifique-se de criar procedimentos armazenados, acionadores, funções ou eventos manualmente no seu banco de dados Amazon RDS. Se algum desses objetos estiver no banco de dados que você está copiando, exclua-os quando executar `mysqldump` ou `mariadb-dump`. Para fazer isso, inclua os seguintes parâmetros com o comando `mysqldump` ou `mariadb-dump`:

- `--routines=0`
- `--triggers=0`
- `--events=0`

Exemplo

O exemplo a seguir copia o banco de dados de exemplo `world` no host local para uma instância de banco de dados do RDS para MySQL. Substitua os valores por suas próprias informações. Para copiar o banco de dados de exemplo para uma instância de banco de dados do RDS para MariaDB, substitua `mysqldump` por `mariadb-dump` e `mysql` por `mariadb`.

Para Linux, macOS ou Unix:

```
sudo mysqldump -u local_user \
    --databases world \
    --single-transaction \
    --compress \
    --order-by-primary  \
    --routines=0 \
    --triggers=0 \
    --events=0 \
    -plocal_password | mysql -u rds_user \
        --port=3306 \
        --host=my_instance.123456789012.us-east-1.rds.amazonaws.com \
        -pRDS_password
```

Para Windows:

Execute o comando a seguir em um prompt de comando que foi aberto clicando com o botão direito do mouse em Prompt de comando no menu de programas do Windows e escolhendo Executar como administrador. Substitua os valores por suas próprias informações. Para copiar o banco de dados de exemplo para uma instância de banco de dados do RDS para MariaDB, substitua `mysqldump` por `mariadb-dump` e `mysql` por `mariadb`.

```
mysqldump -u local_user ^
    --databases world ^
    --single-transaction ^
    --compress ^
    --order-by-primary  ^
    --routines=0 ^
    --triggers=0 ^
    --events=0 ^
    -plocal_password | mysql -u RDS_user ^
        --port=3306 ^
        --host=my_instance.123456789012.us-east-1.rds.amazonaws.com ^
        -pRDS_password
```

> **Note**
>
> Faz parte das práticas recomendadas de segurança especificar credenciais diferentes dos prompts mostrados no exemplo.

Importar dados para uma instância de banco de dados do Amazon RDS para MariaDB com tempo de inatividade reduzido

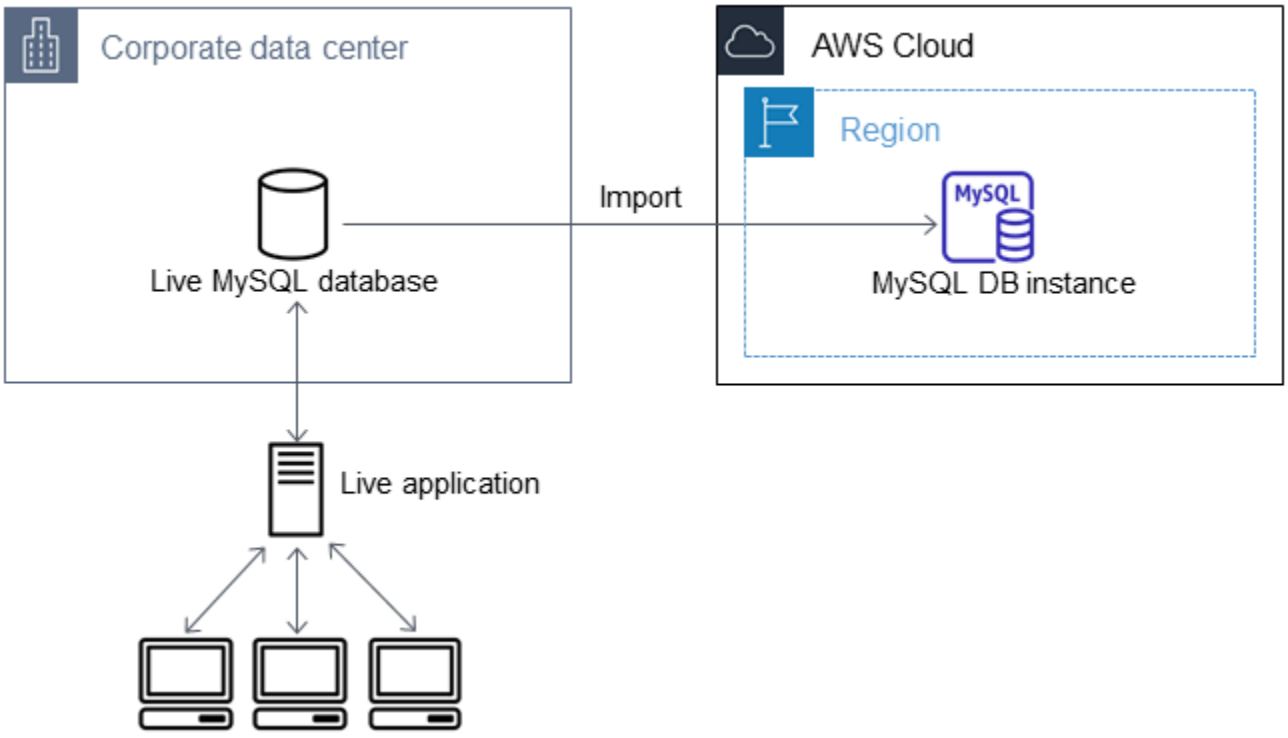
Em alguns casos, pode ser preciso importar dados de um banco de dados MariaDB ou MySQL compatível com uma aplicação ativa para uma instância de banco de dados do RDS para MariaDB, uma instância de banco de dados do RDS para MySQL ou um cluster de banco de dados multi-AZ do RDS para MySQL. Use o procedimento a seguir para minimizar o impacto sobre a disponibilidade das aplicações. Este procedimento também pode ser útil se você está trabalhando com um banco de dados muito grande. Usando esse procedimento, você pode reduzir o custo da importação ao diminuir a quantidade de dados transmitidos pela rede para a AWS.

Neste procedimento, você transfere uma cópia dos dados do banco de dados para uma instância do Amazon EC2 e importa os dados em um novo banco de dados do Amazon RDS. Depois, use a replicação para atualizar o banco de dados do Amazon RDS em relação à instância externa

dinâmica, antes de redirecionar sua aplicação ao banco de dados do Amazon RDS. Se a instância externa for do MariaDB 10.0.24 ou posterior e a instância de destino for do RDS para MariaDB, configure a replicação do MariaDB com base em identificadores de transação global (GTIDs). Caso contrário, configure a replicação com base nas coordenadas do log binário. Recomendamos a replicação baseada em GTID caso o banco de dados externo comporte esse tipo de replicação, já que é um método mais confiável. Para obter mais informações, consulte Global Transaction ID na documentação do MariaDB.

> **Note**
>
> Se você quiser importar dados para uma instância de banco de dados RDS para MySQL e o seu ambiente for compatível, recomendamos mover dados para dentro e para fora do Amazon RDS usando arquivos de backup e o Amazon S3. Para ter mais informações, consulte Restauração de um backup em uma instância de banco de dados MySQL.

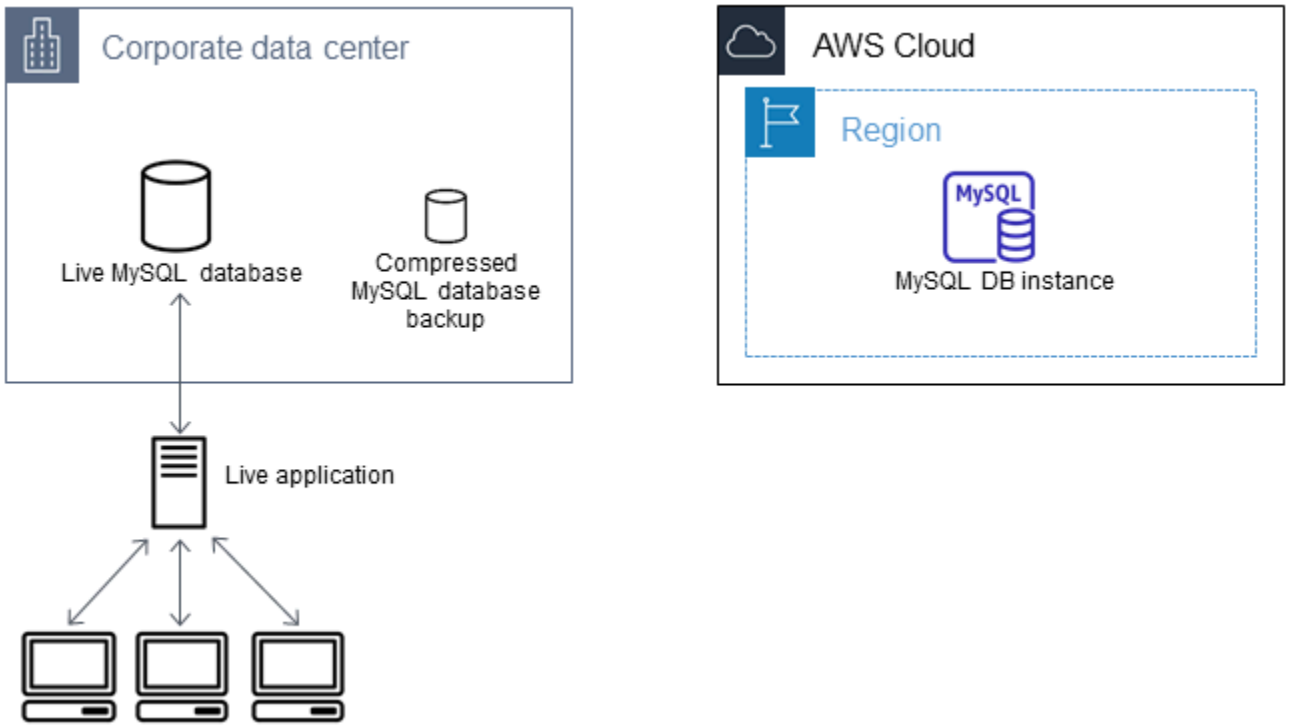
O diagrama a seguir mostra como importar um banco de dados MySQL externo para um banco de dados MySQL no Amazon RDS.



Tarefa 1: criar uma cópia do banco de dados existente

A primeira etapa no processo de migração de uma grande quantidade de dados para um banco de dados do RDS para MariaDB ou RDS para MySQL com o mínimo de tempo de inatividade é criar uma cópia dos dados de origem.

O diagrama a seguir mostra como criar um backup do banco de dados MySQL.



Você pode usar o utilitário `mysqldump` para criar um backup de banco de dados no formato SQL ou de texto delimitado. No MariaDB 10.5, o cliente é chamado de mariadb-dump. A partir do MariaDB 11.0.1, você deve usar `mariadb-dump` em vez de `mysqldump`. Recomendamos que você faça uma execução de teste com cada formato em um ambiente que não seja de produção para ver qual método minimiza o tempo de execução de mysqldump.

Recomendamos também que você pondere a performance de `mysqldump` em relação ao benefício oferecido usando o formato de texto delimitado para o carregamento. Um backup usando o formato de texto delimitado cria um arquivo de texto separado por tabulação para cada tabela que está sendo despejada. Você pode carregar esses arquivos em paralelo usando o comando `LOAD DATA LOCAL INFILE` para reduzir a quantidade de tempo necessária para importar seu banco de dados. Para ter mais informações, consulte Carregar os dados para o MariaDB ou Carregar os dados para o MySQL.

Antes de iniciar a operação de backup, certifique-se de definir as opções de replicação no banco de dados MariaDB ou MySQL que está copiando para o Amazon RDS. As opções de replicação incluem

ativar o registro em log binário e definir um ID de servidor exclusivo. Configurar essas opções faz com que seu servidor comece a registrar em log transações de banco de dados e o prepara para ser um uma instância de replicação de origem mais adiante nesse processo.

Analise as seguintes recomendações e considerações:

- Use a opção `--single-transaction` com `mysqldump` porque ela despeja um estado consistente do banco de dados. Para garantir um arquivo de despejo válido, não execute instruções de linguagem de definição de dados (DDL) enquanto `mysqldump` estiver em execução. É possível programar uma janela de manutenção para essas operações.
- Exclua os seguintes esquemas do arquivo de despejo:
  - `sys`
  - `performance_schema`
  - `information_schema`

  O utilitário `mysqldump` e `mariadb-dump` exclui esses esquemas por padrão.
- Para migrar usuários e privilégios, considere o uso de uma ferramenta que gera a linguagem de controle de dados (DCL) para recriá-los, como o utilitário pt-show-grants.

Como definir opções de replicação

1. Edite o arquivo `my.cnf`. Esse arquivo geralmente está localizado em `/etc`.

   ```
   sudo vi /etc/my.cnf
   ```

   Adicione as opções `log_bin` e `server_id` à seção `[mysqld]`. A opção `log_bin` fornece um identificador de nome de arquivo para arquivos de log binário. A opção `server_id` fornece um identificador exclusivo para o servidor em relações entre origem e réplica.

   O seguinte exemplo mostra a seção `[mysqld]` atualizada de um arquivo `my.cnf`:

   ```
   [mysqld]
   log-bin=mysql-bin
   server-id=1
   ```

   Para ter mais informações, consulte Setting the Replication Source Configuration na documentação do MySQL.

2. Para replicação com um cluster de banco de dados multi-AZ, defina os parâmetros `ENFORCE_GTID_CONSISTENCY` e `GTID_MODE` como `ON`.

```
mysql> SET @@GLOBAL.ENFORCE_GTID_CONSISTENCY = ON;
```

```
mysql> SET @@GLOBAL.GTID_MODE = ON;
```

   Essas configurações não são necessárias para a replicação com uma instância de banco de dados.

3. Reinicie o serviço `mysql`.

```
sudo service mysqld restart
```

Como criar uma cópia de backup do banco de dados existente

1. Crie um backup dos dados por meio do utilitário `mysqldump` ou `mariadb-dump`, especificando o formato SQL ou de texto delimitado.

   Para o MySQL 8.0.25 e versões anteriores, especifique `--master-data=2` para criar um arquivo de backup que possa ser usado para iniciar a replicação entre servidores. Para o MySQL 8.0.26 e versões posteriores, especifique `--source-data=2` para criar um arquivo de backup que possa ser usado para iniciar a replicação entre servidores. Para ter mais informações, consulte mysqldump — A Database Backup Program na documentação do MySQL.

   Para melhorar a performance e garantir a integridade dos dados, use as opções `--order-by-primary` e `--single-transaction` para `mysqldump`.

   Para evitar a inclusão do banco de dados do sistema MySQL no backup, não use a opção `--all-databases` com `mysqldump`. Para ter mais informações, consulte Creating a Data Snapshot Using mysqldump na documentação do MySQL.

   Use `chmod`, se necessário, para garantir que o diretório no qual o arquivo de backup está sendo criado seja gravável.

> **Important**
>
> No Windows, execute a janela de comando como um administrador.

- Para produzir a saída SQL, use o seguinte comando:

  Para Linux, macOS ou Unix:

```
sudo mysqldump \
    --databases database_name \
    --master-data=2  \
    --single-transaction \
    --order-by-primary \
    -r backup.sql \
    -u local_user \
    -ppassword
```

> **Note**
>
> Faz parte das práticas recomendadas de segurança especificar credenciais diferentes dos prompts mostrados no exemplo.

  Para Windows:

```
mysqldump ^
    --databases database_name ^
    --master-data=2  ^
    --single-transaction ^
    --order-by-primary ^
    -r backup.sql ^
    -u local_user ^
    -ppassword
```

> **Note**
>
> Faz parte das práticas recomendadas de segurança especificar credenciais diferentes dos prompts mostrados no exemplo.

- Para produzir uma saída de texto delimitado, use o seguinte comando:

  Para Linux, macOS ou Unix:

```
sudo mysqldump \
    --tab=target_directory \
    --fields-terminated-by ',' \
    --fields-enclosed-by '"' \
    --lines-terminated-by 0x0d0a \
    database_name \
    --master-data=2 \
    --single-transaction \
    --order-by-primary \
    -ppassword
```

Para Windows:

```
mysqldump ^
    --tab=target_directory ^
    --fields-terminated-by "," ^
    --fields-enclosed-by """ ^
    --lines-terminated-by 0x0d0a ^
    database_name ^
    --master-data=2 ^
    --single-transaction ^
    --order-by-primary ^
    -ppassword
```

> **Note**
>
> Faz parte das práticas recomendadas de segurança especificar credenciais diferentes dos prompts mostrados no exemplo.
> Certifique-se de criar procedimentos armazenados, acionadores, funções ou eventos manualmente no seu banco de dados Amazon RDS. Se algum desses objetos estiver no banco de dados que você está copiando, exclua-os quando executar `mysqldump`. Para fazer isso, inclua estes argumentos com o comando`mysqldump`:
>
> - `--routines=0`
> - `--triggers=0`
> - `--events=0`

Para o MySQL 8.0.22 e versões anteriores e versões do MariaDB, quando você executa `mysqldump` e especifica o formato de texto delimitado, um comentário `CHANGE MASTER TO` é exibido. Esse comentário contém o nome e a posição do arquivo de log mestre. Para o MySQL 8.0.23 e versões posteriores, quando você executa `mysqldump` usando o formato de texto delimitado, um comentário `CHANGE REPLICATION SOURCE TO` é exibido. Esse comentário contém o nome e a posição do arquivo de log de origem. Se a versão da instância externa não for MariaDB 10.0.24 e versões posteriores ou MySQL 8.0.23 e versões posteriores, anote os valores referentes a `MASTER_LOG_FILE` e `MASTER_LOG_POS`. Esses valores são necessários ao configurar a replicação.

A seguinte saída é retornada para versões do MariaDB e para o MySQL 8.0.22 e versões anteriores:

```
-- Position to start replication or point-in-time recovery from
--
-- CHANGE MASTER TO MASTER_LOG_FILE='mysql-bin-changelog.000031',
 MASTER_LOG_POS=107;
```

A seguinte saída é retornada para o MySQL 8.0.23 e versões posteriores:

```
-- Position to start replication or point-in-time recovery from
--
-- CHANGE SOURCE TO SOURCE_LOG_FILE='mysql-bin-changelog.000031',
 SOURCE_LOG_POS=107;
```

Para o MySQL 8.0.22 e versões anteriores, se você estiver usando o formato SQL, poderá obter o nome e a posição do arquivo de log mestre no comentário `CHANGE MASTER TO` no arquivo de backup. Para o MySQL 8.0.23 e versões posteriores, se você estiver usando o formato SQL, poderá obter o nome e a posição do arquivo de log de origem no comentário `CHANGE REPLICATION SOURCE TO` no arquivo de backup. Se a instância externa for o MariaDB versão 10.0.24 ou posteriores, você poderá obter o GTID na próxima etapa.

2. Se a versão da instância externa que você está usando for MariaDB 10.0.24 ou posterior, use a replicação baseada em GTID. Execute `SHOW MASTER STATUS` na instância externa do MariaDB para obter o nome e a posição do arquivo de log binário. Em seguida, converta-o em um GTID executando `BINLOG_GTID_POS` na instância externa do MariaDB.

```
SELECT BINLOG_GTID_POS('binary_log_file_name', binary_log_file_position);
```

Anote o GTID retornado. Você precisará dele para configurar a replicação.

3. Compacte os dados copiados para reduzir a quantidade de recursos de rede necessários para copiar seus dados para o banco de dados do Amazon RDS. Anote o tamanho do arquivo de backup. Você precisará dessas informações ao determinar o tamanho de uma instância do Amazon EC2 a ser criada. Quando terminar, compacte o arquivo de backup usando o GZIP ou seu utilitário de compactação preferido.

   - Para compactar a saída SQL, use o seguinte comando:

     ```
     gzip backup.sql
     ```

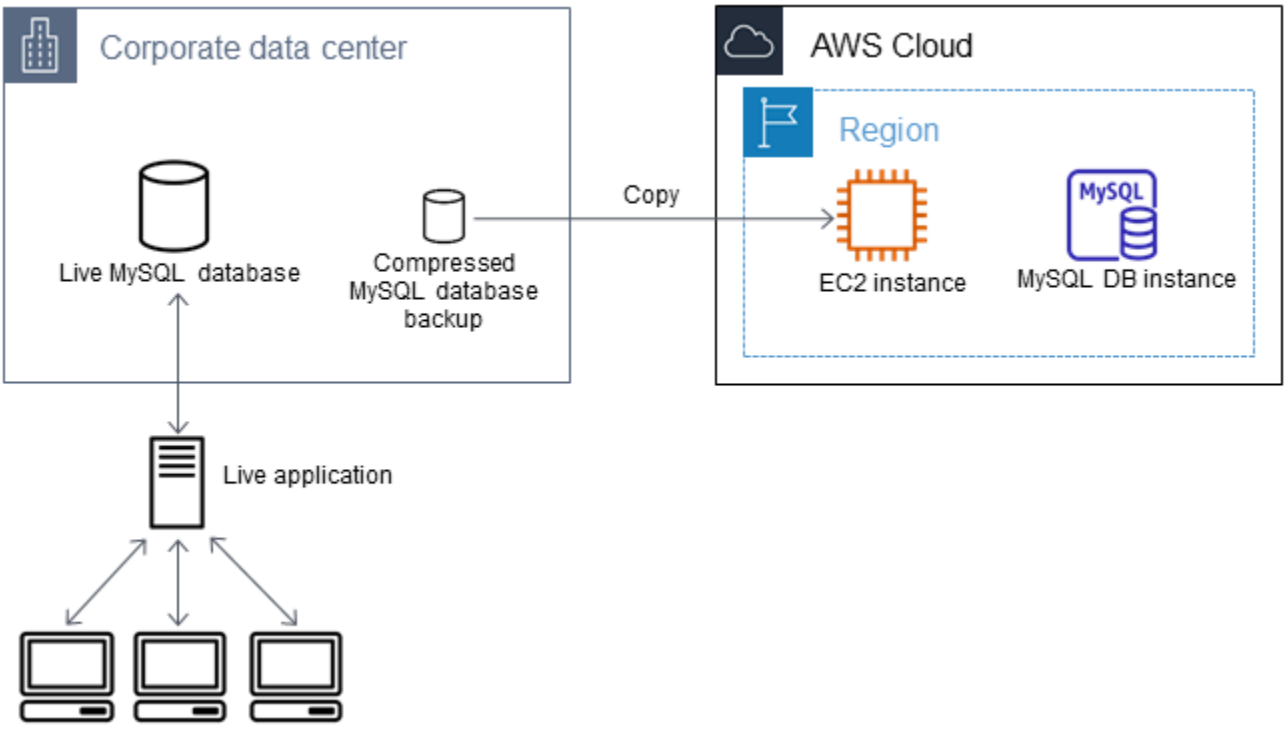
   - Para compactar a saída de texto delimitado, use o seguinte comando:

     ```
     tar -zcvf backup.tar.gz target_directory
     ```

Tarefa 2: criar uma instância do Amazon EC2 e copiar o banco de dados compactado

Copiar seu arquivo de backup de banco de dados compactado para uma instância do Amazon EC2 requer menos recursos de rede do que fazer uma cópia direta de dados não compactados entre instâncias de bancos de dados. Depois que seus dados estiverem no Amazon EC2, você poderá copiá-los de lá diretamente para o banco de dados MariaDB ou MySQL. Para reduzir o custo dos recursos de rede, a instância do Amazon EC2 deve estar na mesma Região da AWS que a instância de banco de dados do Amazon RDS. Ter a instância do Amazon EC2 na mesma Região da AWS que o banco de dados do Amazon RDS também reduz a latência da rede durante a importação.

O diagrama a seguir mostra como copiar o backup do banco de dados para uma instância do Amazon EC2.



Como criar uma instância do Amazon EC2 e copiar seus dados

1. Na Região da AWS em que você planeja criar o banco de dados do Amazon RDS, crie uma nuvem privada virtual (VPC), um grupo de segurança de VPC e uma sub-rede de VPC. Certifique-se de que as regras de entrada para o grupo de segurança da VPC permitam os endereços IP necessários para a aplicação se conectar ao AWS. Você pode especificar uma variedade de endereços IP (por exemplo, `203.0.113.0/24`) ou outro grupo de segurança de VPC. É possível usar o console da Amazon VPC para criar e gerenciar VPCs, sub-redes e grupos de segurança. Para ter mais informações, consulte Conceitos básicos da Amazon VPC no Manual do usuário da Amazon Virtual Private Cloud.
2. Abra o console do Amazon EC2 e escolha a Região da AWS para conter tanto a instância do Amazon EC2 como o banco de dados do Amazon RDS. Inicie uma instância do Amazon EC2 usando a VPC, a sub-rede e o grupo de segurança que você criou na Etapa 1. Certifique-se de selecionar um tipo de instância com armazenamento suficiente para o arquivo de backup de banco de dados quando ele não estiver compactado. Para ter detalhes sobre instâncias do Amazon EC2, consulte Comece a usar o Amazon EC2 no Guia do usuário do Amazon Elastic Compute Cloud.
3. Para se conectar ao banco de dados do Amazon RDS pela sua instância do Amazon EC2, edite seu grupo de segurança da VPC. Adicione uma regra de entrada especificando o endereço IP privado da sua instância do EC2. Você pode encontrar o endereço IP privado na guia Details

(Detalhes) do painel Instance (Instância) na janela do console do EC2. Para editar o grupo de segurança de VPC e adicionar uma regra de entrada, escolha Security Groups (Grupos de segurança) no painel de navegação do console do EC2, escolha o grupo de segurança e, em seguida, adicione uma regra de entrada para MySQL ouAurora especificando o endereço IP privado da sua instância do EC2. Para saber como adicionar uma regra de entrada a um grupo de segurança de VPC, consulte Regras de grupos de segurança no Manual do usuário da Amazon Virtual Private Cloud.

4. Copie o arquivo de backup de banco de dados compactado do seu sistema local para a sua instância do Amazon EC2. Use `chmod`, se necessário, para garantir que você tenha permissão de gravação para o diretório de destino da instância do Amazon EC2. Você pode usar `scp` ou um cliente Secure Shell (SSH) para copiar o arquivo. O seguinte comando é um exemplo de comando `scp`:

```
scp -r -i key pair.pem backup.sql.gz ec2-user@EC2 DNS:/target_directory/backup.sql.gz
```

> **Important**
>
> Ao copiar dados sensíveis, use um protocolo de transferência de rede segura.

5. Conecte-se à sua instância do Amazon EC2 e instale as atualizações e as ferramentas do cliente MySQL mais recentes usando os seguintes comandos:

```
sudo yum update -y
sudo yum install mysql -y
```

Para ter mais informações, consulte Conectar-se à instância no Guia do usuário do Amazon Elastic Compute Cloud para Linux.

> **Important**
>
> Este exemplo instala o cliente MySQL em uma distribuição de imagem de máquina da Amazon (AMI) do Amazon Linux. Este exemplo não instala o cliente MySQL em uma distribuição diferente, como Ubuntu ou Red Hat Enterprise Linux. Para ter informações sobre como instalar o MySQL, consulte Installing MySQL na documentação do MySQL.

6. Enquanto estiver conectado à sua instância do Amazon EC2, descompacte o arquivo de backup do banco de dados. Os comandos a seguir são exemplos.

- Para descompactar a saída SQL, use o seguinte comando:

```
gzip backup.sql.gz -d
```

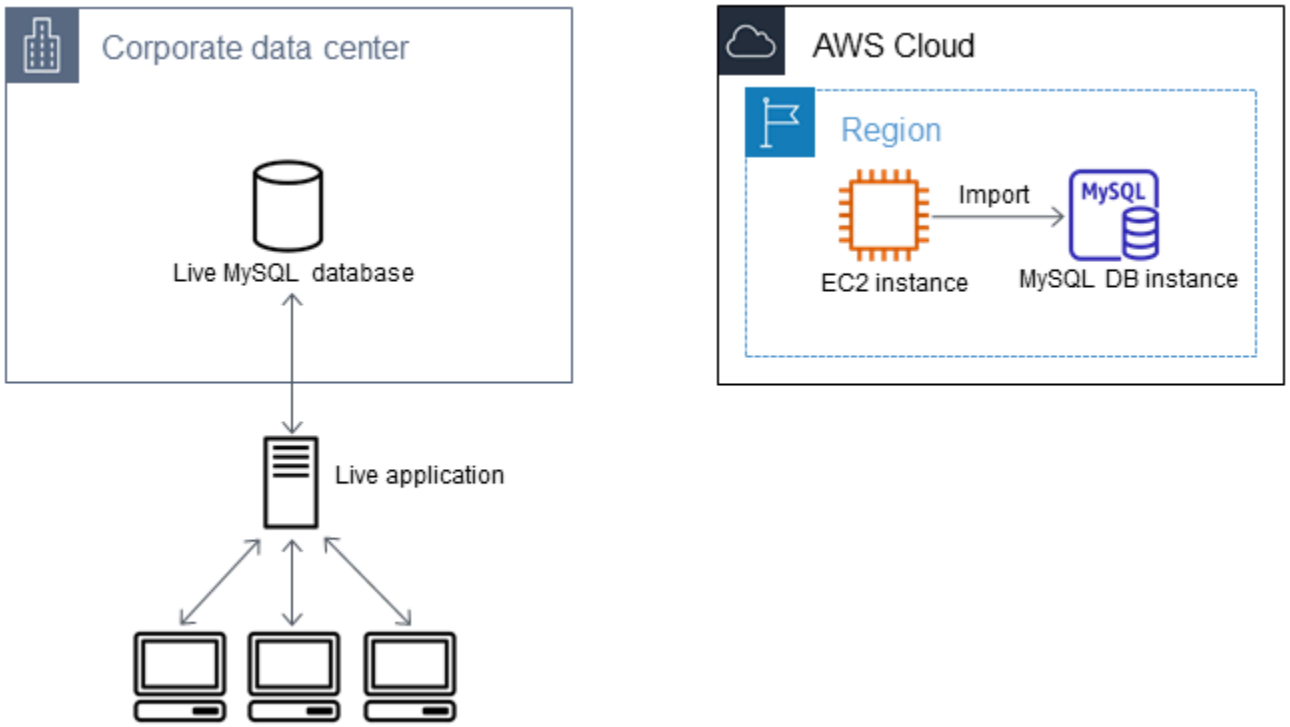
- Para descomprimir a saída de texto delimitada, use o seguinte comando:

```
tar xzvf backup.tar.gz
```

Tarefa 3: criar um banco de dados MariaDB ou MySQL e importar os dados da instância do Amazon EC2

Ao criar uma instância de banco de dados do RDS para MariaDB, uma instância de banco de dados do RDS para MySQL ou um cluster de banco de dados multi-AZ do RDS para MySQL na mesma Região da AWS que a instância do Amazon EC2, é possível importar o arquivo de backup do banco de dados do Amazon EC2 mais rapidamente do que pela internet.

O diagrama a seguir mostra como importar o backup de uma instância do Amazon EC2 para um banco de dados MySQL.



Como criar um banco de dados MariaDB ou MySQL e importar seus dados

1. Determine qual classe de instância de banco de dados e que quantidade de espaço de armazenamento são necessários para atender à workload esperada para esse banco de dados do Amazon RDS. Como parte desse processo, decida sobre o espaço suficiente e a capacidade de processamento para os seus procedimentos de carregamento de dados. Além disso, decida o que é necessário para lidar com a workload de produção. Você pode estimar isso com base no tamanho e nos recursos do banco de dados MySQL ou MariaDB de origem. Para ter mais informações, consulte Classes de instâncias de banco de dados .
2. Crie uma instância de banco de dados ou um cluster de banco de dados multi-AZ na Região da AWS que contém a instância do Amazon EC2.

   Para criar um cluster de banco de dados multi-AZ do RDS para MySQL, siga as instruções em Criar um cluster de banco de dados multi-AZ para o Amazon RDS.

   Para criar uma instância de banco de dados do RDS para MariaDB ou do RDS para MySQL, siga as instruções em Criar uma instância de banco de dados do Amazon RDS e use as seguintes diretrizes:

   - Especifique uma versão do mecanismo de banco de dados que seja compatível com a sua instância de banco de dados de origem.
     - Se a instância de origem for do MySQL 5.5.x, a instância de banco de dados do Amazon RDS deverá ser do MySQL.
     - Se a instância de origem for do MySQL 5.6.x ou 5.7.x, a instância de banco de dados do Amazon RDS deverá ser do MySQL ou do MariaDB.
     - Se a sua instância de origem for o MySQL 8.0.x, a instância de banco de dados do Amazon RDS deverá ser o MySQL 8.0.x.
     - Se a sua instância de origem for o MySQL 8.4.x, a instância de banco de dados do Amazon RDS deverá ser o MySQL 8.4.x.
     - Se a instância de origem for do MariaDB 5.5 ou superior, a instância de banco de dados do Amazon RDS deverá ser do MariaDB.
   - Especifique a mesma nuvem privada virtual (VPC) e o grupo de segurança da VPC para sua instância do Amazon EC2. Essa abordagem garante que sua instância do Amazon EC2 e sua instância do Amazon RDS sejam visíveis uma para a outra na rede. Verifique se sua instância de banco de dados pode ser acessada publicamente. Para configurar a replicação com o

banco de dados de origem, conforme descrito em uma seção mais adiante, a instância de banco de dados deve ser acessível ao público.

- Não configure várias zonas de disponibilidade, retenção de backup nem réplicas de leitura até depois de ter importado o backup do banco de dados. Quando a importação estiver concluída, você poderá configurar o recurso multi-AZ e a retenção de backup para a instância de produção.

3. Reveja as opções de configuração padrão para o banco de dados do Amazon RDS. Se o grupo de parâmetros padrão para o banco de dados não tiver as opções de configuração desejadas, encontre outro que as inclua ou crie um grupo de parâmetros. Para ter mais informações sobre como criar um grupo de parâmetros, consulte Grupos de parâmetros para Amazon RDS.
4. Conecte-se ao novo banco de dados do Amazon RDS como usuário principal. Crie os usuários necessários para dar suporte aos administradores, aplicações e serviços que precisam acessar a instância de banco de dados. O nome do host para o banco de dados do Amazon RDS é o valor do endpoint dessa instância, de banco de dados sem o número da porta; por exemplo, `mysampledb.123456789012.us-west-2.rds.amazonaws.com`. Você pode encontrar o valor do endpoint nos detalhes do banco de dados no console do Amazon RDS.
5. Conecte-se à sua instância Amazon EC2. Para ter mais informações, consulte Conectar-se à instância no Guia do usuário do Amazon Elastic Compute Cloud para Linux.
6. Conecte-se ao seu banco de dados do Amazon RDS como um host remoto pela sua instância do Amazon EC2 usando o comando `mysql`. O seguinte comando é um exemplo:

```
mysql -h host_name -P 3306 -u db_master_user -p
```

O *host_name* é o endpoint do banco de dados do Amazon RDS.

7. No prompt `mysql`, execute o comando `source` e passe a ele o nome do arquivo de despejo do banco de dados. Esse comando carrega os dados na instância de banco de dados do Amazon RDS.

- Para o formato SQL, use o seguinte comando:

```
mysql> source backup.sql;
```

- Para o formato de texto delimitado, primeiro crie o banco de dados, se ele não for o banco de dados padrão que você criou ao configurar o banco de dados do Amazon RDS.

```
mysql> create database database_name;
```

```
mysql> use database_name;
```

Em seguida, crie as tabelas.

```
mysql> source table1.sql
mysql> source table2.sql
etc...
```

Em seguida, importe os dados.

```
mysql> LOAD DATA LOCAL INFILE 'table1.txt' INTO TABLE table1 FIELDS TERMINATED BY
 ',' ENCLOSED BY '"' LINES TERMINATED BY '0x0d0a';
mysql> LOAD DATA LOCAL INFILE 'table2.txt' INTO TABLE table2 FIELDS TERMINATED BY
 ',' ENCLOSED BY '"' LINES TERMINATED BY '0x0d0a';
etc...
```

Para melhorar a performance, você pode realizar essas operações em paralelo de várias conexões, para que todas as tabelas sejam criadas e carregadas ao mesmo tempo.

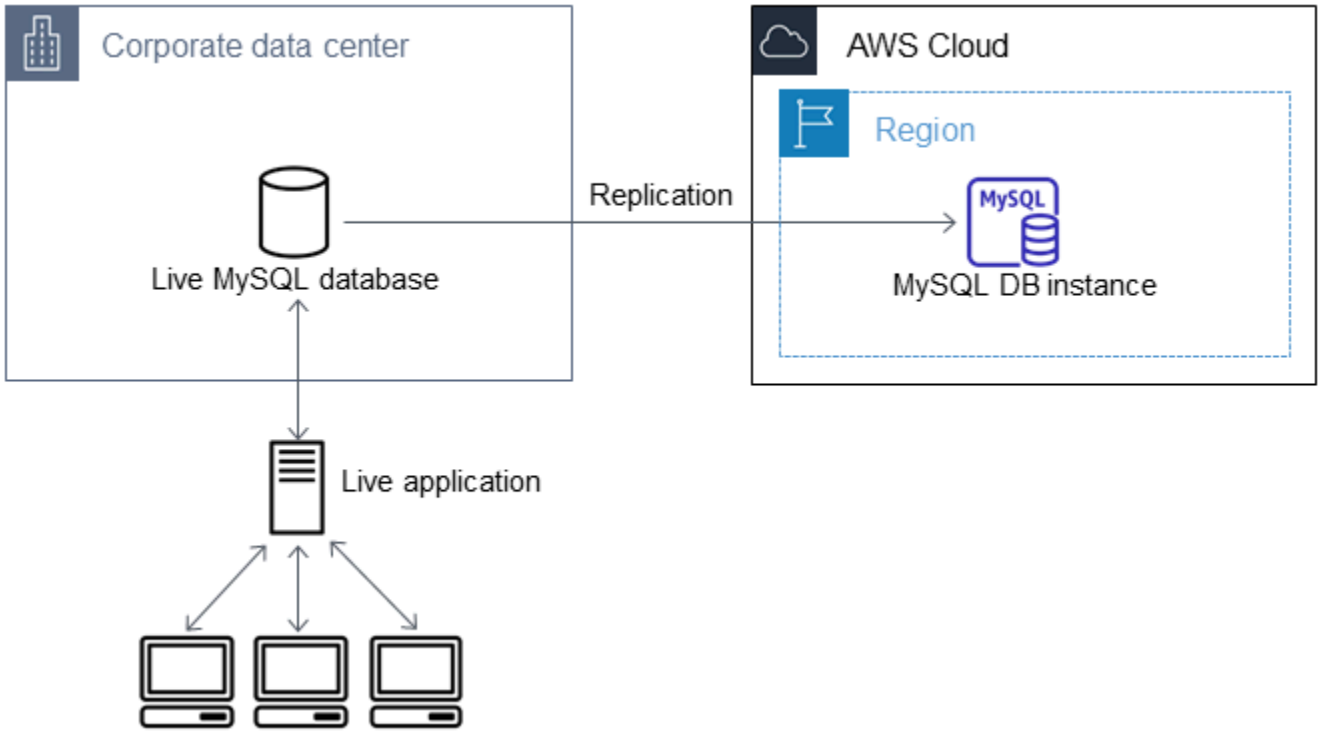
> **Note**
>
> Se você usou qualquer opção de formatação de dados com `mysqldump` quando despejou inicialmente a tabela, use as mesmas opções com `LOAD DATA LOCAL INFILE` para assegurar uma interpretação adequada do conteúdo do arquivo de dados.

8. Execute uma consulta `SELECT` simples em uma ou duas das tabelas no banco de dados importado para verificar se a importação foi bem-sucedida.

Se você não precisa mais da instância do Amazon EC2 usada neste procedimento, encerre-a para reduzir o uso de recursos de uso da AWS. Para encerrar uma instância do EC2, consulte Como encerrar uma instância no Guia do usuário do Amazon Elastic Compute Cloud.

Tarefa 4: replicar dados do banco de dados externo para o novo banco de dados do Amazon RDS

Seu banco de dados de origem provavelmente foi atualizado durante o tempo decorrido entre a cópia e a transferência dos dados para o banco de dados MariaDB ou MySQL. Assim, você pode usar a replicação para atualizar o banco de dados copiado com o banco de dados de origem.



As permissões necessárias para iniciar a replicação em um banco de dados do Amazon RDS são restritas e não estão disponíveis ao seu usuário principal do Amazon RDS. Por esse motivo, use o procedimento armazenado apropriado do Amazon RDS para a versão principal do mecanismo:

- mysql.rds_set_external_master (RDS para MariaDB e RDS para MySQL versões principais 8.0 e anteriores)
- mysql.rds_set_external_source (RDS para MySQL versões principais 8.4 e posteriores)
- mysql.rds_set_external_master_gtid para configurar a replicação e mysql.rds_start_replication para iniciar a replicação

Como iniciar a replicação

Na Tarefa 1, ao definir as opções de replicação, você ativou o registro em log binário e definiu um ID de servidor exclusivo para o banco de dados de origem. Agora, é possível configurar seu banco de

dados do Amazon RDS como uma réplica com seu banco de dados dinâmico como a instância de replicação de origem.

1. No console do Amazon RDS, adicione o endereço IP do servidor que hospeda o banco de dados de origem ao grupo de segurança da VPC para o banco de dados do Amazon RDS. Para ter mais informações sobre grupos de segurança da VPC, consulte Configurar regras de grupo de segurança no Manual do usuário da Amazon Virtual Private Cloud.

   Você também pode precisar configurar a rede local para permitir conexões com o endereço IP do banco de dados do Amazon RDS, para que ele possa se comunicar com a instância de origem. Para localizar o endereço IP do banco de dados do Amazon RDS, use o comando `host`:

   ```
   host host_name
   ```

   O *`host_name`* é o nome DNS do endpoint do banco de dados do Amazon RDS; por exemplo, `myinstance.123456789012.us-east-1.rds.amazonaws.com`. Você pode encontrar o valor do endpoint nos detalhes da instância do banco de dados no console do Amazon RDS.
2. Usando o cliente de sua preferência, conecte-se à instância de origem e crie um usuário a ser usado para a replicação. Esta conta é usada unicamente para replicação e deve estar restrita ao seu domínio para melhorar a segurança. O seguinte comando é um exemplo:

   ```
   CREATE USER 'repl_user'@'mydomain.com' IDENTIFIED BY 'password';
   ```

   > **Note**
   >
   > Especifique credenciais diferentes dos prompts mostrados aqui como prática recomendada de segurança.

3. Para a instância de origem, conceda os privilégios `REPLICATION CLIENT` e `REPLICATION SLAVE` para seu usuário de replicação. Por exemplo, para conceder os privilégios de `REPLICATION CLIENT` e `REPLICATION SLAVE` em todos os bancos de dados para o usuário '`repl_user`' de seu domínio, emita o seguinte comando:

   ```
   GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'mydomain.com';
   ```

4. Se você tiver usado o formato SQL para criar o arquivo de backup e a versão da instância externa não for MariaDB 10.0.24 ou posterior, veja o conteúdo desse arquivo executando o seguinte comando:

```
cat backup.sql
```

O arquivo inclui um comentário `CHANGE MASTER TO` que contém o nome e a posição do arquivo de log mestre. Esse comentário está incluído no arquivo de backup quando você usa a opção `--master-data` com `mysqldump`. Observe os valores para `MASTER_LOG_FILE` e `MASTER_LOG_POS`.

```
--
-- Position to start replication or point-in-time recovery from
--

-- CHANGE MASTER TO MASTER_LOG_FILE='mysql-bin-changelog.000031', MASTER_LOG_POS=107;
```

Se você usou o formato de texto delimitado para criar o arquivo de backup e a versão da instância externa não for MariaDB 10.0.24 ou posterior, você já deve ter coordenadas de logs binários da Etapa 1 do procedimento “Como criar uma cópia de backup do banco de dados existente” na Tarefa 1.

Se a versão da instância externa for MariaDB 10.0.24 ou posterior, você já deve ter o GTID com base no qual iniciar a replicação na Etapa 2 do procedimento “Como criar uma cópia de backup do banco de dados existente” na Tarefa 1.

5. Torne o banco de dados do Amazon RDS a réplica. Se a versão da instância externa não for MariaDB 10.0.24 ou posterior, conecte-se ao banco de dados do Amazon RDS como usuário principal e identifique o banco de dados de origem como a instância de replicação de origem usando o comando o procedimento armazenado apropriado do Amazon RDS.

   - mysql.rds_set_external_master (RDS para MariaDB e RDS para MySQL versões principais 8.0 e anteriores)
   - mysql.rds_set_external_source (RDS para MySQL versões principais 8.4 e posteriores)

   Se você tiver um arquivo de backup no formato SQL, use o nome do arquivo de log mestre e a posição do log mestre que você determinou na Etapa 4. Se você usou o formato de texto delimitado, use o nome e a posição que você determinou ao criar os arquivos de backup. Os seguintes comandos são exemplos:

   MySQL 8.4 e versões posteriores

   ```
   CALL mysql.rds_set_external_source ('myserver.mydomain.com', 3306,
   ```

```
    'repl_user', 'password', 'mysql-bin-changelog.000031', 107, 1);
```

MariaDB e MySQL 8.0 e versões anteriores

```
CALL mysql.rds_set_external_master ('myserver.mydomain.com', 3306,
    'repl_user', 'password', 'mysql-bin-changelog.000031', 107, 1);
```

> **Note**
>
> Especifique credenciais diferentes dos prompts mostrados aqui como prática recomendada de segurança.

Se a instância externa for o MariaDB 10.0.24 ou posterior, conecte-se ao banco de dados do Amazon RDS como o usuário principal e identifique o banco de dados de origem como a instância de replicação de origem usando o comando mysql.rds_set_external_master_gtid. Use o GTID que você determinou na Etapa 2 do procedimento “Como criar uma cópia de backup do banco de dados existente” na Tarefa 1. O seguinte comando é um exemplo:

```
CALL mysql.rds_set_external_master_gtid ('source_server_ip_address', 3306,
 'ReplicationUser', 'password', 'GTID', 1);
```

O *source_server_ip_address* é o endereço IP da instância de replicação de origem. No momento, não é possível usar um endereço DNS privado do EC2.

> **Note**
>
> Especifique credenciais diferentes dos prompts mostrados aqui como prática recomendada de segurança.

6. No banco de dados do Amazon RDS, para iniciar a replicação, execute o comando mysql.rds_start_replication abaixo:

```
CALL mysql.rds_start_replication;
```

7. No banco de dados do Amazon RDS, para determinar quando a réplica está atualizada com a instância de replicação de origem, execute o comando SHOW REPLICA STATUS. Os resultados

do comando `SHOW REPLICA STATUS` incluem o campo `Seconds_Behind_Master`. Quando o campo `Seconds_Behind_Master` retorna 0, a réplica é atualizada com a instância de replicação de origem.

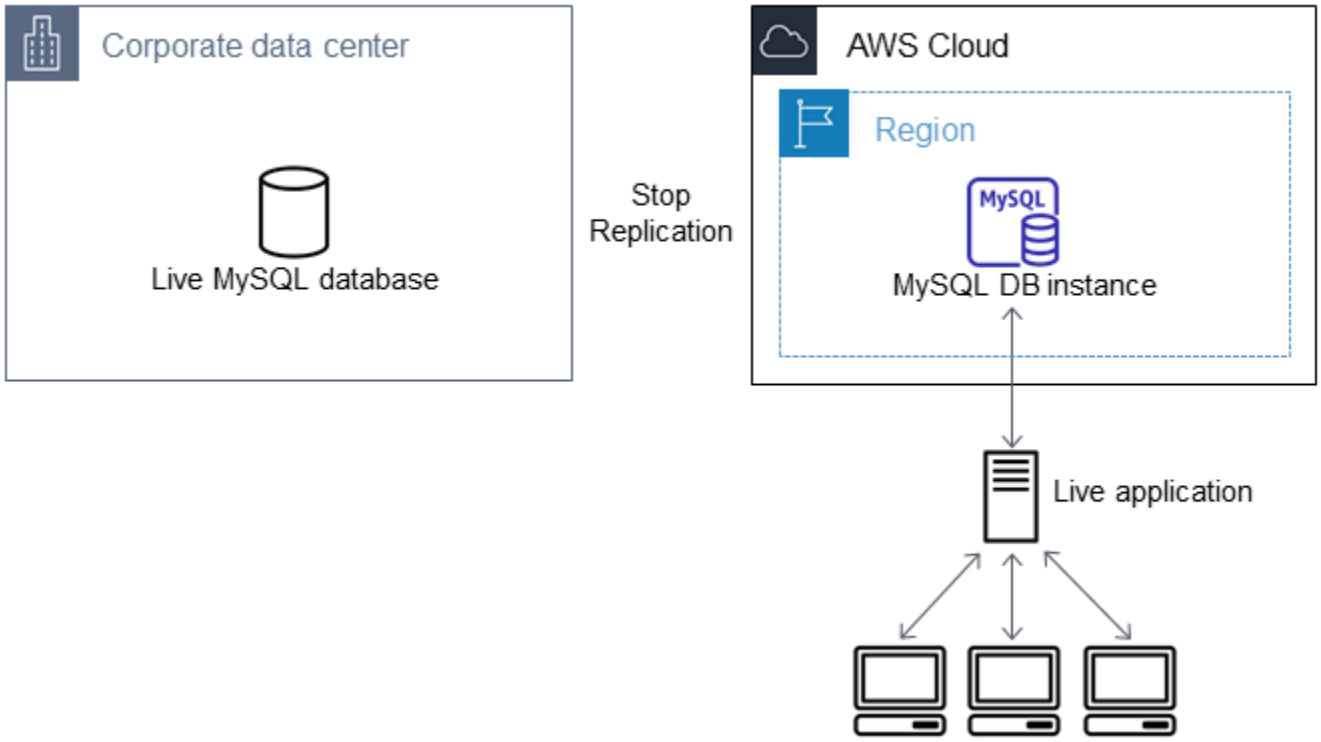
> **Note**
>
> As versões anteriores do MySQL usavam `SHOW SLAVE STATUS` em vez de `SHOW REPLICA STATUS`. Se você estiver usando uma versão do MySQL anterior à 8.0.23, use `SHOW SLAVE STATUS`.

Para uma instância de banco de dados MariaDB 10.5, 10.6 ou 10.11, execute o procedimento mysql.rds_replica_status em vez do comando MySQL.

8. Depois que o banco de dados do Amazon RDS estiver atualizado, ative backups automatizados para poder restaurar esse banco de dados, se necessário. Você pode ativar ou modificar backups automatizados para um banco de dados do Amazon RDS usando o console do Amazon RDS. Para ter mais informações, consulte Introdução aos backups.

Tarefa 5: redirecionar a aplicação ativa para a instância do Amazon RDS

Depois que o banco de dados MariaDB ou MySQL estiver atualizado com a instância de replicação de origem, será possível atualizar a aplicação ativa para usar a instância do Amazon RDS.



Como redirecionar a aplicação dinâmica para o banco de dados MariaDB ou MySQL e interromper a replicação

1. Para adicionar o grupo de segurança de VPC para o banco de dados do Amazon RDS, adicione o endereço IP do servidor que hospeda a aplicação. Para ter mais informações sobre como modificar um grupo de segurança de VPC, consulte Configurar regras de grupo de segurança no Manual do usuário da Amazon Virtual Private Cloud.
2. Verifique se o campo `Seconds_Behind_Master` nos resultados do comando SHOW REPLICA STATUS é 0, o que indica que a réplica está atualizada com a instância de replicação de origem.

```
SHOW REPLICA STATUS;
```

> **Note**
>
> As versões anteriores do MySQL usavam `SHOW SLAVE STATUS` em vez de `SHOW REPLICA STATUS`. Se você estiver usando uma versão do MySQL anterior à 8.0.23, use `SHOW SLAVE STATUS`.

Para uma instância de banco de dados MariaDB 10.5, 10.6 ou 10.11, execute o procedimento mysql.rds_replica_status em vez do comando MySQL.

3. Feche todas as conexões com a origem quando suas transações terminarem.
4. Atualize sua aplicação para usar o banco de dados do Amazon RDS. Normalmente, essa atualização envolve a alteração das configurações de conexão para identificar o nome do host e a porta do banco de dados do Amazon RDS, a conta de usuário e a senha para conexão e o banco de dados para uso.
5. Conecte-se à instância de banco de dados.

   No caso de um cluster de banco de dados multi-AZ, conecte-se a uma instância de banco de dados gravadora.
6. Interrompa a replicação da instância do Amazon RDS usando o comando mysql.rds_stop_replication abaixo:

   ```
   CALL mysql.rds_stop_replication;
   ```
7. Execute o procedimento armazenado apropriado do Amazon RDS no banco de dados do Amazon RDS para redefinir a configuração de replicação e fazer com que a instância não seja mais identificada como uma réplica.

   - mysql.rds_reset_external_master (RDS para MariaDB e RDS para MySQL versões principais 8.0 e anteriores)
   - mysql.rds_reset_external_source (RDS para MySQL versões principais 8.4 e posteriores)

   MySQL 8.4 e versões posteriores

   ```
   CALL mysql.rds_reset_external_source;
   ```

   MariaDB e MySQL 8.0 e versões anteriores

   ```
   CALL mysql.rds_reset_external_master;
   ```
8. Ative recursos adicionais do Amazon RDS, como o suporte a multi-AZ e réplicas de leitura. Para obter mais informações, consulte Configurar e gerenciar uma implantação multi-AZ para o Amazon RDS e Trabalhar com réplicas de leitura de instância de banco de dados.

Importar dados de qualquer fonte para uma instância de banco de dados do Amazon RDS para MariaDB

As seguintes etapas explicam como importar dados para uma instância de banco de dados do Amazon RDS:

Etapa 1: Criar arquivos simples contendo os dados a serem carregados

Use um formato comum, como valores separados por vírgula (CSV), para armazenar os dados a serem carregados. Cada tabela deve ter um arquivo próprio. Não é possível combinar os dados de várias tabelas no mesmo arquivo. Dê a cada arquivo o mesmo nome que a tabela correspondente. A extensão do arquivo pode ser qualquer opção desejada. Por exemplo, se o nome da tabela for `sales`, o nome do arquivo poderá ser `sales.csv` ou `sales.txt`.

Se possível, ordene os dados pela chave primária da tabela que está sendo carregada. Fazer isso melhora drasticamente os tempos de carregamento e minimiza os requisitos de armazenamento em disco.

A velocidade e a eficiência desse procedimento dependem de manter o tamanho dos arquivos pequenos. Se o tamanho descompactado de qualquer arquivo individual for maior que 1 GiB, divida-o em vários arquivos e carregue cada um separadamente.

Em sistemas semelhantes ao Unix (incluindo o Linux), use o comando `split`. Por exemplo, o seguinte comando divide o arquivo `sales.csv` em vários arquivos com menos de 1 GiB, dividindo apenas em intervalos de linha (-C 1024m). Os nomes dos novos arquivos incluem sufixos numéricos ascendentes. O comando a seguir produz arquivos com nomes, como `sales.part_00` e `sales.part_01`.

```
split -C 1024m -d sales.csv sales.part_
```

Utilitários semelhantes estão disponíveis para outros sistemas operacionais.

Você pode armazenar os arquivos simples em qualquer lugar. No entanto, ao carregar os dados na Etapa 5, você deve invocar o shell `mysql` do mesmo local em que os arquivos estão ou usar o caminho absoluto para os arquivos ao executar `LOAD DATA LOCAL INFILE`.

Etapa 2: impedir que quaisquer aplicações acessem a instância de banco de dados de destino

Antes de iniciar um grande carregamento, impeça que qualquer atividade da aplicação acesse a instância de banco de dados de destino para a qual planeja fazer o carregamento. Recomendamos isso especialmente se outras sessões modificarem as tabelas que estão sendo carregadas ou as tabelas às quais elas fazem referência. Isso reduz o risco de violações de restrição durante o carregamento e melhora a performance do carregamento. Também possibilita restaurar a instância de banco de dados até o ponto antes do carregamento, sem perder as alterações feitas pelos processos não envolvidos no carregamento.

É que isso pode não ser possível ou prático. Se você não conseguir impedir que as aplicações acessem a instância de banco de dados antes do carregamento, tome medidas para garantir a disponibilidade e a integridade dos seus dados. As etapas específicas necessárias variam muito dependendo dos casos específicos de uso e dos requisitos do site.

Etapa 3: Criar um snapshot de banco de dados

Se você planeja carregar dados em uma nova instância de banco de dados que não contém dados, pode ignorar essa etapa. Do contrário, recomendamos criar snapshots de banco de dados da instância de banco de dados de destino do Amazon RDS antes e depois do carregamento dos dados. Os snapshots de banco de dados do Amazon RDS são backups completos da instância de banco de dados que podem ser usados para restaurá-la em um estado conhecido. Quando você inicia um snapshot de banco de dados, as operações de E/S na sua instância de banco de dados são momentaneamente suspensas enquanto é feito o backup do banco de dados.

Criar um snapshot de banco de dados imediatamente antes do carregamento possibilita a restauração do banco de dados para o estado anterior ao carregamento, se necessário. Um snapshot de banco de dados feito imediatamente após a carga evita que você tenha que carregar os dados novamente caso ocorra um problema. Você também pode usar snapshots de banco de dados após o carregamento para importar dados para novas instâncias de banco de dados.

O exemplo a seguir executa o comando create-db-snapshot da AWS CLI para criar um snapshot de banco de dados da instância `AcmeRDS` e fornece ao snapshot do banco de dados o identificador `"preload"`.

Para Linux, macOS ou Unix:

```
aws rds create-db-snapshot \
    --db-instance-identifier AcmeRDS \
```

```
    --db-snapshot-identifier preload
```

Para Windows:

```
aws rds create-db-snapshot ^
    --db-instance-identifier AcmeRDS ^
    --db-snapshot-identifier preload
```

Você também pode usar a restauração da funcionalidade de snapshot do banco de dados para criar instâncias de banco de dados de teste para execuções simuladas ou para desfazer alterações feitas durante o carregamento.

Tenha em mente que restaurar um banco de dados de um snapshot de banco de dados cria uma nova instância de banco de dados que, como todas as instâncias de banco de dados, possui um identificador exclusivo e um endpoint. Para restaurar a instância de banco de dados sem alterar o endpoint, primeiro exclua a instância de banco de dados para poder reutilizar o endpoint.

Por exemplo, para criar uma instância de banco de dados para execuções simuladas ou outros testes, forneça à instância de banco de dados um identificador próprio. Na exemplo, o identificador é `AcmeRDS-2`. O exemplo se conecta à instância de banco de dados usando o endpoint associado ao `AcmeRDS-2`. Para ter mais informações, consulte restore-db-instance-from-db-snapshot.

Para Linux, macOS ou Unix:

```
aws rds restore-db-instance-from-db-snapshot \
    --db-instance-identifier AcmeRDS-2 \
    --db-snapshot-identifier preload
```

Para Windows:

```
aws rds restore-db-instance-from-db-snapshot ^
    --db-instance-identifier AcmeRDS-2 ^
    --db-snapshot-identifier preload
```

Para reutilizar o endpoint existente, primeiro exclua a instância de banco de dados e forneça ao banco de dados restaurado o mesmo identificador. Para ter mais informações, consulte delete-db-instance.

O exemplo a seguir também cria um snapshot de banco de dados final da instância de banco de dados antes de excluí-la. Isso é opcional, porém é recomendado.

Para Linux, macOS ou Unix:

```
aws rds delete-db-instance \
    --db-instance-identifier AcmeRDS \
    --final-db-snapshot-identifier AcmeRDS-Final

aws rds restore-db-instance-from-db-snapshot \
    --db-instance-identifier AcmeRDS \
    --db-snapshot-identifier preload
```

Para Windows:

```
aws rds delete-db-instance ^
    --db-instance-identifier AcmeRDS ^
    --final-db-snapshot-identifier AcmeRDS-Final

aws rds restore-db-instance-from-db-snapshot ^
    --db-instance-identifier AcmeRDS ^
    --db-snapshot-identifier preload
```

Etapa 4 (opcional): desativar os backups automatizados do Amazon RDS

> **Warning**
>
> Não desative os backups automatizados se você precisa realizar uma recuperação para um ponto no tempo.

A desativação de backups automatizados ajuda a otimizar a performance e não é necessária para carregamentos de dados. Ao desativar os backups automatizados, todos os backups existentes são apagados. Consequentemente, após a desativação dos backups automatizados, não é possível realizar uma recuperação para um ponto no tempo. Os snapshots do banco de dados manuais não são afetados com a desativação dos backups automatizados. Todos os snapshots de banco de dados manuais existentes ainda estão disponíveis para restauração.

Desativar os backups automatizados reduz o tempo de carregamento em cerca de 25% e reduz a quantidade de espaço de armazenamento necessária durante o carregamento. Se você planeja carregar dados em uma nova instância de banco de dados que não contém dados, desativar os backups é uma maneira fácil de acelerar o carregamento e evitar o uso do armazenamento adicional

necessário para backups. No entanto, em alguns casos, você pode planejar fazer o carregamento em uma instância de banco de dados que já contenha dados. Se for o caso, avalie os benefícios da desativação de backups em relação ao impacto da perda da capacidade de realizar a recuperação em um ponto anterior no tempo.

As instâncias de bancos de dados têm backups automatizados desativados por padrão (com um período de retenção de um dia). Para desatiar os backups automáticos, defina o período de retenção de backup para zero. Após o carregamento, você pode reativar os backups definindo o período de retenção de backup como um valor diferente de zero. Para ativar ou desativar os backups, o Amazon RDS desliga a instância de banco de dados e a reinicia para habilitar ou desabilitar o registro em log do MariaDB ou do MySQL.

Execute o comando AWS CLI da `modify-db-instance` para definir a retenção do backup como zero e aplicar a alteração imediatamente. Definir o período de retenção como zero exige uma reinicialização da instância de banco de dados. Por isso, aguarde até que a reinicialização tenha sido concluída antes de prosseguir. Para obter mais informações, consulte modify-db-instance.

Para Linux, macOS ou Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier AcmeRDS \
    --apply-immediately \
    --backup-retention-period 0
```

Para Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier AcmeRDS ^
    --apply-immediately ^
    --backup-retention-period 0
```

Você pode verificar o status sua instância de banco de dados com o comando describe-db-instances. O seguinte exemplo mostra o status da instância de banco de dados `AcmeRDS`:

```
aws rds describe-db-instances --db-instance-identifier AcmeRDS --query "*[].
{DBInstanceStatus:DBInstanceStatus}"
```

Quando o status da instância de banco de dados for `available`, você poderá prosseguir para a próxima etapa.

Etapa 5: Carregar os dados

Para ler as linhas dos arquivos simples nas tabelas do banco de dados, use a instrução MySQL `LOAD DATA LOCAL INFILE`.

> **Note**
>
> Você deve invocar o shell `mysql` do mesmo local em que seus arquivos simples estão ou usar o caminho absoluto para os arquivos ao executar `LOAD DATA LOCAL INFILE`.

O seguinte exemplo mostra como carregar dados de um arquivo chamado `sales.txt` em uma tabela denominada `Sales` no banco de dados:

```
mysql> LOAD DATA LOCAL INFILE 'sales.txt' INTO TABLE Sales FIELDS TERMINATED BY ' '
 ENCLOSED BY '' ESCAPED BY '\\';
Query OK, 1 row affected (0.01 sec)
Records: 1  Deleted: 0  Skipped: 0  Warnings: 0
```

Para ter mais informações sobre a instrução `LOAD DATA`, consulte LOAD DATA Statement na documentação do MySQL.

Etapa 6: reativar os backups automatizados do Amazon RDS

Se você desativou os backups automatizados do Amazon RDS na Etapa 4, depois que o carregamento for concluído, ative os backups automatizados definindo o período de retenção de backup de volta para o valor anterior ao carregamento. Conforme mencionado na Etapa 4, como o Amazon RDS reinicia a instância de banco de dados, prepare-se para uma breve interrupção.

O seguinte exemplo executa o comando modify-db-instance da AWS CLI para ativar os backups automatizados da instância de banco de dados `AcmeRDS` e define o período de retenção para um dia:

Para Linux, macOS ou Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier AcmeRDS \
    --backup-retention-period 1 \
    --apply-immediately
```

Para Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier AcmeRDS ^
    --backup-retention-period 1 ^
    --apply-immediately
```

Como trabalhar com a replicação do MariaDB no Amazon RDS

Normalmente, você usa réplicas de leitura para configurar a replicação entre as instâncias de banco de dados do Amazon RDS. Para obter informações gerais sobre réplicas de leitura, consulte Trabalhar com réplicas de leitura de instância de banco de dados. Para obter informações específicas sobre como trabalhar com réplicas de leitura no Amazon RDS para MariaDB, consulte Trabalhar com réplicas de leitura do MariaDB.

Você também pode configurar a replicação com base nas coordenadas do log binário para as instâncias de banco de dados MariaDB. Para as instâncias do MariaDB, você também pode configurar a replicação com base nos IDs de transações globais (GTIDs), o que oferece uma melhor segurança contra falhas. Para obter mais informações, consulte Configurar a replicação baseada em GTID com uma instância de origem externa.

Estas são as outras opções de replicação disponíveis com o RDS para MariaDB:

- É possível configurar a replicação entre uma instância de banco de dados do RDS para MariaDB e uma instância do MySQL ou do MariaDB externa ao Amazon RDS. Para obter informações sobre como configurar a replicação com uma origem externa, consulte Configurar a replicação da posição do arquivo de log binário com uma instância de origem externa.
- Você pode configurar a replicação para importar bancos de dados de uma instância MySQL ou do MariaDB externos ao Amazon RDS, ou exportar bancos de dados para essas instâncias. Para ter mais informações, consulte Importar dados para uma instância de banco de dados do Amazon RDS para MariaDB com tempo de inatividade reduzido e Exportar dados de uma instância de banco de dados MySQL usando replicação.

Para todas essas opções de replicação, use replicação baseada em linha, em instrução ou mista. A replicação baseada em linha só replica as linhas alteradas resultantes de uma instrução SQL. A replicação baseada em instrução copia toda a instrução SQL. A replicação mista usa uma replicação baseada em instrução quando possível, mas alterna para replicação baseada em linha quando as instruções SQL não seguras para replicação baseada em instrução são executadas. Na maioria dos casos, a replicação mista é recomendada. O formato de log binário da instância de banco de dados determina se a replicação se baseia na linha, na instrução ou mista. Para obter informações sobre como definir o formato de log binário, consulte Configurar o registro em log binário do MariaDB.

Para obter informações sobre a compatibilidade de replicação entre as versões do MariaDB, consulte Compatibilidade de replicação na documentação do MariaDB.

Tópicos

- Trabalhar com réplicas de leitura do MariaDB
- Configurar a replicação baseada em GTID com uma instância de origem externa
- Configurar a replicação da posição do arquivo de log binário com uma instância de origem externa

Trabalhar com réplicas de leitura do MariaDB

Encontre a seguir informações específicas sobre como trabalhar com réplicas de leitura no Amazon RDS para MariaDB. Para obter informações gerais sobre as réplicas de leitura e as instruções de como usá-las, consulte Trabalhar com réplicas de leitura de instância de banco de dados.

- Configurar filtros de replicação com o MariaDB
- Configurar a replicação atrasada com o MariaDB
- Atualizar réplicas de leitura com o MariaDB
- Trabalhar com implantações de réplicas de leitura multi-AZ com o MariaDB
- Usar réplicas de leitura em cascata com o RDS para MariaDB
- Monitoramento de réplicas de leitura do MariaDB
- Início e interrupção de replicação com réplicas de leitura do MariaDB
- Solução de problemas da réplica de leitura do MariaDB

Configurar réplicas de leitura com o MariaDB

Para que uma instância de banco de dados do MariaDB possa servir como uma fonte de replicação, você deve ativar os backups automáticos na instância de banco de dados de origem definindo o período de retenção do backup como um valor diferente de 0. Esse requisito também se aplica a uma réplica de leitura que seja a instância de banco de dados de origem de outra réplica de leitura.

Você pode criar até quinze réplicas de leitura de uma instância de banco de dados na mesma região. Para que a replicação funcione efetivamente, cada réplica de leitura deve ter a mesma quantidade de recursos de computação e armazenamento que a instância de banco de dados de origem. Se você dimensionar a instância de banco de dados de origem, dimensione as réplicas de leitura também.

O RDS para MariaDB é compatível com réplicas de leitura em cascata. Para saber mais sobre como configurar réplicas de leitura em cascata, consulte Usar réplicas de leitura em cascata com o RDS para MariaDB.

É possível executar várias ações simultâneas de criação ou exclusão de réplicas de leitura que fazem referência à mesma instância de banco de dados de origem. Ao realizar essas ações, permaneça dentro do limite de quinze réplicas de leitura para cada instância de origem.

Configurar filtros de replicação com o MariaDB

Você pode usar filtros de replicação para especificar quais bancos de dados e tabelas são replicados com uma réplica de leitura. Os filtros de replicação podem incluir bancos de dados e tabelas na replicação ou excluí-los da replicação.

Veja a seguir alguns casos de uso para filtros de replicação:

- Para reduzir o tamanho de uma réplica de leitura. Com a filtragem de replicação, você pode excluir os bancos de dados e tabelas que não são necessários na réplica de leitura.
- Para excluir bancos de dados e tabelas de réplicas de leitura por motivos de segurança.
- Para replicar diferentes bancos de dados e tabelas para casos de uso específicos em diferentes réplicas de leitura. Por exemplo, você pode usar réplicas de leitura específicas para análise ou fragmentação.
- Para uma instância de banco de dados que tenha réplicas de leitura em diferentes Regiões da AWS, para replicar diferentes bancos de dados ou tabelas em diferentes Regiões da AWS.

> **Note**
>
> É possível usar filtros de replicação para especificar quais bancos de dados e tabelas serão replicados com uma instância de banco de dados primária do MariaDB configurada como uma réplica em uma topologia de replicação de entrada. Para obter mais informações sobre essa configuração, consulte Configurar a replicação da posição do arquivo de log binário com uma instância de origem externa.

Tópicos

- Definir parâmetros de filtragem de replicação do RDS para MariaDB
- banco de dados primário filtragem de replicação do RDS para MariaDB
- Exemplos de filtragem de replicação no RDS para MariaDB
- Visualizar os filtros de replicação para uma réplica de leitura

Definir parâmetros de filtragem de replicação do RDS para MariaDB

Para configurar filtros de replicação, defina os seguintes parâmetros de filtragem de replicação na réplica de leitura:

- `replicate-do-db` – Replicar alterações nos bancos de dados especificados. Quando você define esse parâmetro para uma réplica de leitura, somente os bancos de dados especificados no parâmetro são replicados.
- `replicate-ignore-db` – Não replique as alterações nos bancos de dados especificados. Quando o parâmetro `replicate-do-db` é definido para uma réplica de leitura, esse parâmetro não é avaliado.
- `replicate-do-table` – Replicar alterações nas tabelas especificadas. Quando você define esse parâmetro para uma réplica de leitura, somente as tabelas especificadas no parâmetro são replicadas. Além disso, quando o parâmetro `replicate-do-db` ou `replicate-ignore-db` é definido, o banco de dados que inclui as tabelas especificadas deve ser incluído na replicação com a réplica de leitura.
- `replicate-ignore-table` – Não replique as alterações nas tabelas especificadas. Quando o parâmetro `replicate-do-table` é definido para uma réplica de leitura, esse parâmetro não é avaliado.
- `replicate-wild-do-table` – Replicar tabelas com base nos padrões de nome de banco de dados e tabela especificados. Os caracteres curinga `%` e `_` são compatíveis. Quando o parâmetro `replicate-do-db` ou `replicate-ignore-db` estiver definido, certifique-se de incluir o banco de dados que inclui as tabelas especificadas na replicação com a réplica de leitura.
- `replicate-wild-ignore-table` – Não replique tabelas com base nos padrões de nome de banco de dados e tabela especificados. Os caracteres curinga `%` e `_` são compatíveis. Quando o parâmetro `replicate-do-table` ou `replicate-wild-do-table` é definido para uma réplica de leitura, esse parâmetro não é avaliado.

Os parâmetros são avaliados na ordem em que estão listados. Para obter mais informações sobre como esses parâmetros funcionam, consulte a documentação do MariaDB.

Por padrão, cada um desses parâmetros tem um valor vazio. Em cada réplica de leitura, você pode usar esses parâmetros para definir, alterar e excluir filtros de replicação. Quando você define um desses parâmetros, separe cada filtro dos outros com uma vírgula.

Você pode usar % os caracteres curinga _ e nos parâmetros `replicate-wild-do-table` e `replicate-wild-ignore-table`. O curinga % corresponde a qualquer número de caracteres e o caractere curinga _ corresponde apenas a um caractere.

O formato de log binário da instância de banco de dados de origem é importante para replicação porque determina o registro de alterações de dados. A configuração do parâmetro `binlog_format` determina se a replicação é baseada em linha ou baseada em declaração. Para obter mais informações, consulte Configurar o registro em log binário do MariaDB.

> **Note**
>
> Todas as instruções DDL (Data Definition Language, linguagem de definição de dados) são replicadas como instruções, independentemente da `binlog_format` configuração na instância de banco de dados de origem.

banco de dados primário filtragem de replicação do RDS para MariaDB

As seguintes limitações se aplicam à filtragem de replicação do RDS para MariaDB:

- Cada parâmetro de filtragem de replicação tem um limite de 2.000 caracteres.
- As vírgulas não são compatíveis em filtros de replicação.
- As opções `binlog_do_db` e `binlog_ignore_db` do MariaDB para filtragem de log binário não são compatíveis.
- A filtragem de replicação não suporta transações XA.

  Para obter mais informações, consulte Restrictions on XA Transactions na documentação do MySQL.
- A filtragem de replicação não é compatível com o RDS para MariaDB versão 10.2.

Exemplos de filtragem de replicação no RDS para MariaDB

Para configurar a filtragem de replicação para uma réplica de leitura, modifique os parâmetros de filtragem de replicação no grupo de parâmetros associado à réplica de leitura.

> **Note**
>
> Não é possível modificar um grupo de parâmetros padrão. Se a réplica de leitura estiver usando um grupo de parâmetros padrão, crie um novo grupo de parâmetros e o associe à instância de banco de dados. Para obter mais informações sobre grupos de parâmetros de banco de dados, consulte Grupos de parâmetros para Amazon RDS.

Você pode definir parâmetros em um grupo de parâmetros usando a AWS Management Console, a AWS CLI ou a API do RDS. Para obter informações sobre como configurar parâmetros, consulte Modificar parâmetros em um grupo de parâmetros de banco de dados no Amazon RDS. Quando você define parâmetros em um grupo de parâmetros, todas as instâncias de banco de dados associadas ao grupo de parâmetros usam as configurações de parâmetro. Se você definir os parâmetros de filtragem de replicação em um grupo de parâmetros, verifique se o grupo de parâmetros está associado apenas a réplicas de leitura. Deixe os parâmetros de filtragem de replicação vazios para instâncias de banco de dados de origem.

Os exemplos a seguir definem os parâmetros usando o AWS CLI. Estes exemplos definem `ApplyMethod` para `immediate` de modo que as mudanças do parâmetro ocorram imediatamente depois que o comando CLI termina. Se você quiser que uma alteração pendente seja aplicada depois que a réplica de leitura for reinicializada, defina como `ApplyMethod` `pending-reboot`.

Os exemplos a seguir definem filtros de replicação:

- Including databases in replication
- Including tables in replication
- Including tables in replication with wildcard characters
- Escaping wildcard characters in names
- Excluding databases from replication
- Excluding tables from replication
- Excluding tables from replication using wildcard characters

Example Incluir bancos de dados em replicação

O exemplo a seguir inclui os bancos de dados `mydb1` e `mydb2` na replicação. Quando você define `replicate-do-db` para uma réplica de leitura, somente os bancos de dados especificados no parâmetro são replicados.

Para Linux, macOS ou Unix:

```
aws rds modify-db-parameter-group \
  --db-parameter-group-name myparametergroup \
  --parameters "[{"ParameterName": "replicate-do-db", "ParameterValue": "mydb1,mydb2",
 "ApplyMethod":"immediate"}]"
```

Para Windows:

```
aws rds modify-db-parameter-group ^
  --db-parameter-group-name myparametergroup ^
  --parameters "[{"ParameterName": "replicate-do-db", "ParameterValue": "mydb1,mydb2",
 "ApplyMethod":"immediate"}]"
```

Example Incluir tabelas na replicação

O exemplo a seguir inclui as tabelas `table1` e `table2` no banco de dados `mydb1` na replicação.

Para Linux, macOS ou Unix:

```
aws rds modify-db-parameter-group \
  --db-parameter-group-name myparametergroup \
  --parameters "[{"ParameterName": "replicate-do-table", "ParameterValue":
 "mydb1.table1,mydb1.table2", "ApplyMethod":"immediate"}]"
```

Para Windows:

```
aws rds modify-db-parameter-group ^
  --db-parameter-group-name myparametergroup ^
  --parameters "[{"ParameterName": "replicate-do-table", "ParameterValue":
 "mydb1.table1,mydb1.table2", "ApplyMethod":"immediate"}]"
```

Example Incluir tabelas na replicação usando caracteres curinga

O exemplo a seguir inclui tabelas com nomes que começam com `orders` e `returns` no banco de dados `mydb` na replicação.

Para Linux, macOS ou Unix:

```
aws rds modify-db-parameter-group \
  --db-parameter-group-name myparametergroup \
  --parameters "[{"ParameterName": "replicate-wild-do-table", "ParameterValue":
 "mydb.orders%,mydb.returns%", "ApplyMethod":"immediate"}]"
```

Para Windows:

```
aws rds modify-db-parameter-group ^
  --db-parameter-group-name myparametergroup ^
  --parameters "[{"ParameterName": "replicate-wild-do-table", "ParameterValue":
 "mydb.orders%,mydb.returns%", "ApplyMethod":"immediate"}]"
```

Example Caracteres de escape curinga em nomes

O exemplo a seguir mostra como usar o caractere de escape \ para liberar um caractere curinga que faz parte de um nome.

Suponha que você tenha vários nomes de tabela no banco de dados `mydb1` que começam com `my_table`, e você deseja incluir essas tabelas na replicação. Os nomes das tabelas incluem um sublinhado, que também é um caractere curinga, portanto, o exemplo escapa ao sublinhado nos nomes das tabelas.

Para Linux, macOS ou Unix:

```
aws rds modify-db-parameter-group \
  --db-parameter-group-name myparametergroup \
  --parameters "[{"ParameterName": "replicate-wild-do-table", "ParameterValue": "my
\_table%", "ApplyMethod":"immediate"}]"
```

Para Windows:

```
aws rds modify-db-parameter-group ^
```

```
  --db-parameter-group-name myparametergroup ^
  --parameters "[{"ParameterName": "replicate-wild-do-table", "ParameterValue": "my
\_table%", "ApplyMethod":"immediate"}]"
```

Example Excluir bancos de dados da replicação

O exemplo a seguir exclui os bancos de dados `mydb1` e `mydb2` da replicação.

Para Linux, macOS ou Unix:

```
aws rds modify-db-parameter-group \
  --db-parameter-group-name myparametergroup \
  --parameters "[{"ParameterName": "replicate-ignore-db", "ParameterValue":
 "mydb1,mydb2", "ApplyMethod":"immediate"}]"
```

Para Windows:

```
aws rds modify-db-parameter-group ^
  --db-parameter-group-name myparametergroup ^
  --parameters "[{"ParameterName": "replicate-ignore-db", "ParameterValue":
 "mydb1,mydb2", "ApplyMethod":"immediate"}]"
```

Example Excluir tabelas da replicação

O exemplo a seguir exclui tabelas `table1` e `table2` no banco de dados `mydb1` da replicação.

Para Linux, macOS ou Unix:

```
aws rds modify-db-parameter-group \
  --db-parameter-group-name myparametergroup \
  --parameters "[{"ParameterName": "replicate-ignore-table", "ParameterValue":
 "mydb1.table1,mydb1.table2", "ApplyMethod":"immediate"}]"
```

Para Windows:

```
aws rds modify-db-parameter-group ^
  --db-parameter-group-name myparametergroup ^
  --parameters "[{"ParameterName": "replicate-ignore-table", "ParameterValue":
 "mydb1.table1,mydb1.table2", "ApplyMethod":"immediate"}]"
```

Example Excluir tabelas da replicação usando caracteres curinga

O exemplo a seguir exclui tabelas com nomes que começam com `orders` e `returns` no banco de dados `mydb` da replicação.

Para Linux, macOS ou Unix:

```
aws rds modify-db-parameter-group \
  --db-parameter-group-name myparametergroup \
  --parameters "[{"ParameterName": "replicate-wild-ignore-table", "ParameterValue":
 "mydb.orders%,mydb.returns%", "ApplyMethod":"immediate"}]"
```

Para Windows:

```
aws rds modify-db-parameter-group ^
  --db-parameter-group-name myparametergroup ^
  --parameters "[{"ParameterName": "replicate-wild-ignore-table", "ParameterValue":
 "mydb.orders%,mydb.returns%", "ApplyMethod":"immediate"}]"
```

Visualizar os filtros de replicação para uma réplica de leitura

Você pode visualizar os filtros de replicação de uma réplica de leitura das seguintes maneiras:

- Verifique as configurações dos parâmetros de filtragem de replicação no grupo de parâmetros associado à réplica de leitura.

  Para obter instruções, consulte Visualizar valores de parâmetros para um grupo de parâmetros de banco de dados no Amazon RDS.
- Em um cliente MariaDB, conecte-se à réplica de leitura e execute a instrução `SHOW REPLICA STATUS`.

  Na saída, os campos a seguir mostram os filtros de replicação para a réplica de leitura:
  - `Replicate_Do_DB`
  - `Replicate_Ignore_DB`
  - `Replicate_Do_Table`
  - `Replicate_Ignore_Table`
  - `Replicate_Wild_Do_Table`
  - `Replicate_Wild_Ignore_Table`

Para obter mais informações sobre esses campos, consulte Verificar o status da replicação na documentação do MySQL.

> **Note**
>
> Versões anteriores do MariaDB usavam `SHOW SLAVE STATUS` em vez de `SHOW REPLICA STATUS`. Se você estiver usando uma versão do MariaDB anterior à 10.5, use `SHOW SLAVE STATUS`.

Configurar a replicação atrasada com o MariaDB

Você pode usar a replicação atrasada como uma estratégia para a recuperação de desastres. Com a replicação atrasada, você especifica o tempo mínimo, em segundos, para atrasar a replicação da origem para a réplica de leitura. Em caso de um desastre, como uma tabela excluída acidentalmente, você executa as seguintes etapas para recuperar-se rapidamente do desastre:

- Interrompa a replicação da réplica de leitura antes que a alteração que causou o desastre seja enviada para ela.

  Use o procedimento armazenado mysql.rds_stop_replication para interromper a replicação.
- Promova a réplica de leitura para ser a nova instância de banco de dados de origem usando as instruções em Promoção de uma réplica de leitura a uma instância de banco de dados autônoma.

> **Note**
>
> - A replicação atrasada é compatível com o MariaDB 10.6 e posteriores.
> - Use procedimentos armazenados para configurar a replicação atrasada. Você não pode configurar a replicação atrasada com o AWS Management Console, a AWS CLI ou a API do Amazon RDS.
> - Você pode usar a replicação com base em identificadores de transação global (GTIDs) em uma configuração de replicação atrasada.

Tópicos

- Configurar replicação atrasada durante a criação da réplica de leitura

- Modificar replicação atrasada de uma réplica de leitura existente
- Promover uma réplica de leitura

Configurar replicação atrasada durante a criação da réplica de leitura

Para configurar a replicação atrasada para qualquer réplica de leitura futura criada a partir de uma instância de banco de dados, execute o procedimento armazenado mysql.rds_set_configuration com o parâmetro `target delay`.

Para configurar a replicação atrasada durante a criação da réplica de leitura

1. Usando um cliente MariaDB, conecte-se à instância de banco de dados MariaDB que será a origem para réplicas de leitura como o usuário primário.
2. Execute o procedimento armazenado mysql.rds_set_configuration com o parâmetro `target delay`.

   Por exemplo, execute o procedimento armazenado a seguir para especificar que a replicação é atrasada em pelo menos uma hora (3,600 segundos) para qualquer réplica de leitura criada a partir da instância de banco de dados atual.

   ```
   call mysql.rds_set_configuration('target delay', 3600);
   ```

   > **Note**
   >
   > Após executar esse procedimento armazenado, qualquer réplica de leitura que você criar usando a AWS CLI ou a API do Amazon RDS será configurada com a replicação atrasada pelo número de segundos especificado.

Modificar replicação atrasada de uma réplica de leitura existente

Para modificar a replicação atrasada para uma réplica de leitura existente, execute o procedimento armazenado mysql.rds_set_source_delay.

Para modificar a replicação atrasada para uma réplica de leitura existente

1. Usando um cliente do MariaDB, conecte-se à réplica de leitura como o usuário primário.
2. Use o procedimento armazenado mysql.rds_stop_replication para interromper a replicação.

3. Execute o procedimento armazenado mysql.rds_set_source_delay.

   Por exemplo, execute o procedimento armazenado a seguir para especificar que a replicação para a réplica de leitura é atrasada em pelo menos uma hora (3.600 segundos) para qualquer réplica de leitura criada a partir da instância de banco de dados atual.

   ```
   call mysql.rds_set_source_delay(3600);
   ```

4. Use o procedimento armazenado mysql.rds_start_replication para iniciar a replicação.

Promover uma réplica de leitura

Após a replicação ser interrompida, em um cenário de recuperação de desastres, você pode promover uma réplica de leitura para ser a nova instância de banco de dados de origem. Para obter informações sobre como promover uma réplica de leitura, consulte Promoção de uma réplica de leitura a uma instância de banco de dados autônoma.

Atualizar réplicas de leitura com o MariaDB

As réplicas de leitura foram projetadas para oferecer suporte a consultas de leitura, mas você pode precisar fazer atualizações ocasionais. Por exemplo, talvez seja necessário adicionar um índice para acelerar tipos específicos de consultas que acessam a réplica. Você pode habilitar as atualizações configurando o parâmetro `read_only` como 0 no grupo de parâmetros de banco de dados da réplica de leitura.

Trabalhar com implantações de réplicas de leitura multi-AZ com o MariaDB

É possível criar uma réplica de leitura a partir de implantações de instâncias de banco de dados single-AZ ou multi-AZ. Você pode usar implantações multi-AZ para melhorar a durabilidade e a disponibilidade de dados essenciais. No entanto, não é possível usar o multi-AZ secundário para atender a consultas somente leitura. Em vez disso, crie réplicas de leitura de instâncias de banco de dados multi-AZ de alto tráfego para descarregar consultas somente leitura. Se a instância de origem de uma implantação multi-AZ falhar na secundária, todas as réplicas de leitura associadas serão automaticamente alteradas para usar a secundária (não a primária) como a origem de replicação. Para obter mais informações, consulte Configurar e gerenciar uma implantação multi-AZ para o Amazon RDS.

É possível criar uma réplica de leitura como uma instância de banco de dados multi-AZ. O Amazon RDS cria um em modo de espera de sua réplica em outra zona de disponibilidade para suporte a

failover da réplica. Você pode criar a réplica de leitura como uma instância de banco de dados multi-AZ independentemente de o banco de dados de origem ser ou não uma instância de banco de dados multi-AZ.

Usar réplicas de leitura em cascata com o RDS para MariaDB

O RDS para MariaDB é compatível com réplicas de leitura em cascata. Com réplicas de leitura em cascata, é possível escalar leituras sem adicionar sobrecarga à instância de banco de dados do RDS para MariaDB de origem.

Com réplicas de leitura em cascata, sua instância de banco de dados do RDS para MariaDB envia dados para a primeira réplica de leitura da cadeia. Essa réplica de leitura envia dados para a segunda réplica na cadeia e assim por diante. O resultado final é que todas as réplicas de leitura na cadeia têm as alterações da instância de banco de dados do RDS para MariaDB, mas sem a sobrecarga apenas na instância de banco de dados de origem.

É possível criar uma série de até três réplicas de leitura em uma cadeia de uma instância de banco de dados de origem do RDS para MariaDB. Por exemplo, suponha que você tenha uma instância de banco de dados do RDS para MariaDB, `mariadb-main`. Você pode fazer o seguinte:

- Começando com `mariadb-main`, crie a primeira réplica de leitura na cadeia, `read-replica-1`.
- Na `read-replica-1`, crie a próxima réplica de leitura na cadeia, `read-replica-2`.
- Finalmente, na `read-replica-2`, crie a terceira réplica de leitura na cadeia, `read-replica-3`.

Não é possível criar outra réplica de leitura além dessa terceira réplica de leitura em cascata na série de `mariadb-main`. Uma série completa de instâncias de uma instância de banco de dados de origem do RDS para MariaDB até o final de uma série de réplicas de leitura em cascata pode consistir em, no máximo, quatro instâncias de banco de dados.

Para que as réplicas de leitura em cascata funcionem, cada instância de banco de dados do RDS para MariaDB de origem deve ter os backups automatizados ativados. Para ativar backups automáticos em uma réplica de leitura, primeiro crie a réplica de leitura e a modifique para ativar backups automáticos. Para ter mais informações, consulte Como criar uma réplica de leitura.

Como em qualquer réplica de leitura, é possível promover uma réplica de leitura que faz parte de uma cascata. A promoção de uma réplica de leitura de uma cadeia de réplicas de leitura remove essa réplica da cadeia. Por exemplo, suponha que você queira mover parte da workload da instância de banco de dados `mariadb-main` para uma nova instância para uso somente pelo departamento

de contabilidade. Pressupondo a cadeia com três réplicas de leitura do exemplo, você decide promover `read-replica-2`. A cadeia é afetada da seguinte forma:

- A promoção de `read-replica-2` a remove da cadeia de replicação.
  - Ela agora é uma instância de banco de dados de leitura/gravação completa.
  - Ela continua replicando para `read-replica-3`, da mesma forma como estava fazendo antes da promoção.
- A `mariadb-main` continua a replicar para a `read-replica-1`.

Para obter mais informações sobre como promover réplicas de leitura, consulte Promoção de uma réplica de leitura a uma instância de banco de dados autônoma.

Monitoramento de réplicas de leitura do MariaDB

Para as réplicas de leitura do MariaDB, você pode monitorar o atraso da replicação no Amazon CloudWatch visualizando a métrica `ReplicaLag` do Amazon RDS. A métrica `ReplicaLag` relata o valor do campo `Seconds_Behind_Master` do comando `SHOW REPLICA STATUS`.

> **Note**
>
> Versões anteriores do MariaDB usavam `SHOW SLAVE STATUS` em vez de `SHOW REPLICA STATUS`. Se você estiver usando uma versão do MariaDB anterior à 10.5, use `SHOW SLAVE STATUS`.

As causas comuns para o atraso da replicação do MariaDB são as seguintes:

- Uma queda de rede.
- Gravar em tabelas com índices em uma réplica de leitura. Se o parâmetro `read_only` não estiver definido como 0 na réplica de leitura, isso poderá interromper a replicação.
- Uso de um mecanismo de armazenamento não transacional, como o MyISAM. A replicação só é compatível com o mecanismo de armazenamento InnoDB no MariaDB.

Quando a métrica `ReplicaLag` chega a 0, isso mostra que a réplica alcançou a instância do banco de dados de origem. Se a métrica `ReplicaLag` retornar -1, então a replicação não está ativa no momento. `ReplicaLag` = -1 é equivalente a `Seconds_Behind_Master` = `NULL`.

Início e interrupção de replicação com réplicas de leitura do MariaDB

Você pode interromper e reiniciar o processo de replicação em uma instância de banco de dados do Amazon RDS ao chamar os procedimentos armazenados do sistema mysql.rds_stop_replication e mysql.rds_start_replication. Você pode fazer isso ao replicar entre duas instâncias do Amazon RDS para operações de longa duração, como a criação de índices grandes. Você também precisa interromper e iniciar a replicação ao importar ou exportar bancos de dados. Para obter mais informações, consulte Importar dados para um banco de dados do Amazon RDS para MySQL com tempo de inatividade reduzido e Exportar dados de uma instância de banco de dados MySQL usando replicação.

Se a replicação for interrompida por mais de 30 dias consecutivos, seja manualmente ou devido a um erro de replicação, o Amazon RDS a encerrará entre a instância de banco de dados de origem e todas as réplicas de leitura. Isso acontece para evitar um aumento nos requisitos de armazenamento da instância de banco de dados de origem e nos tempos de failover prolongado. A instância de banco de dados da réplica de leitura ainda está disponível. No entanto, a replicação não pode ser retomada porque os logs binários exigidos pela réplica de leitura são excluídos da instância de banco de dados de origem após o encerramento da replicação. Você pode criar uma nova réplica de leitura para a instância de banco de dados de origem a fim de restabelecer a replicação.

Solução de problemas da réplica de leitura do MariaDB

As tecnologias de replicação do MariaDB são assíncronas. Como são assíncronas, são esperados ocasionais aumentos de `BinLogDiskUsage` na instância de banco de dados de origem e `ReplicaLag` na réplica de leitura. Por exemplo, um volume elevado de operações de gravação para a instância de banco de dados de origem pode ocorrer em paralelo. Por outro lado, as operações de gravação na réplica de leitura são serializadas usando um único thread de E/S, o que pode ocasionar um atraso entre a instância de origem e a réplica de leitura. Para obter mais informações sobre réplicas somente leitura na documentação do MariaDB, acesse Visão geral da replicação.

Você pode fazer várias coisas para reduzir o atraso entre as atualizações de uma instância de banco de dados de origem e as atualizações subsequentes da réplica de leitura, como o seguinte:

- Dimensionar uma réplica de leitura para ter um tamanho de armazenamento e uma categoria de instância de banco de dados comparáveis à da instância de banco de dados de origem.
- Assegurar-se de que as configurações de parâmetros nos grupos de parâmetros de banco de dados utilizados pela instância de banco de dados de origem e pela réplica de leitura são compatíveis. Para mais informações e um exemplo, consulte a discussão sobre o parâmetro `max_allowed_packet` posteriormente nesta seção.

O Amazon RDS monitora o status de replicação de suas réplicas de leitura e atualiza o campo `Replication State` da instância da réplica de leitura para `Error` caso a replicação seja interrompida por qualquer motivo. Um exemplo pode ser se as consultas DML forem executadas no seu conflito de réplica de leitura com as atualizações feitas na instância de banco de dados de origem.

Você pode analisar os detalhes do erro associado gerado pelo mecanismo do MariaDB visualizando o campo `Replication Error`. Os eventos que indicam o status da réplica de leitura também são gerados, incluindo RDS-EVENT-0045, RDS-EVENT-0046 e RDS-EVENT-0047. Para mais informações sobre eventos e como se inscrever neles, consulte Trabalhar com a notificação de eventos do Amazon RDS. Se for retornada uma mensagem de erro do MariaDB, analise o erro na documentação de mensagens de erro do MariaDB.

Um problema comum que pode causar erros de replicação é quando o valor do parâmetro `max_allowed_packet` para uma réplica de leitura é menor que o do parâmetro `max_allowed_packet` para a instância de banco de dados de origem. O parâmetro `max_allowed_packet` é um parâmetro personalizado que pode ser definido em um grupo de parâmetros de banco de dados usado para especificar o código de DML que pode ser executado no banco de dados. Em alguns casos, o valor do parâmetro `max_allowed_packet` no grupo de parâmetros de banco de dados associado a uma instância de banco de dados de origem é menor do que o valor do parâmetro `max_allowed_packet` no grupo de parâmetros de banco de dados associado à réplica de leitura da origem. Nesses casos, o processo de replicação pode exibir um erro (Packet bigger than 'max_allowed_packet' bytes [Pacote maior que a quantidade máxima de bytes]) e interromper a replicação. É possível corrigir o erro fazendo com que a origem e a réplica de leitura usem grupos de parâmetros de banco de dados com os mesmos valores do parâmetro `max_allowed_packet`.

Outras situações comuns que podem causar erros de replicação incluem o seguinte:

- A gravação em tabelas em uma réplica de leitura. Se estiver criando índices em uma réplica de leitura, você precisará ter o parâmetro `read_only` definido como 0 para criar os índices. Se você estiver gravando em tabelas na réplica de leitura, isso poderá interromper a replicação.
- O uso de um mecanismo de armazenamento não transacional, como MyISAM. As réplicas de leitura exigem um mecanismo de armazenamento transacional. A replicação só é compatível com o mecanismo de armazenamento InnoDB no MariaDB.
- Usando consultas não deterministas inseguras, como `SYSDATE()`. Para obter mais informações, consulte Determinação de instruções seguras e inseguras no registro de logs binários.

Se você acreditar que pode ignorar um erro com segurança, siga as etapas descritas em Ignorar o erro de replicação atual para o RDS para MySQL. Caso contrário, você pode excluir a réplica de leitura e criar uma instância usando o mesmo identificador de instância de banco de dados para que o endpoint permaneça o mesmo que o da sua antiga réplica de leitura. Se um erro de replicação for corrigido, o `Replication State` mudará para replicating.

Para as instâncias de banco de dados do MariaDB, em alguns casos, as réplicas de leitura não poderão ser alternadas para a secundária se alguns eventos de log binário (binlog) não forem liberados durante a falha. Nesses casos, exclua e recrie manualmente as réplicas de leitura. Você pode reduzir a chance disso acontecer definindo os seguintes valores de parâmetro: `sync_binlog=1` e `innodb_flush_log_at_trx_commit=1`. Essas configurações podem reduzir a performance, portanto, teste o impacto delas antes de implantar as alterações em um ambiente de produção.

Configurar a replicação baseada em GTID com uma instância de origem externa

Você pode configurar a replicação com base em identificadores de transação global (GTIDs) de uma instância externa do MariaDB da versão 10.0.24 ou posteriores em uma instância de banco de dados do MariaDB. Siga estas diretrizes ao configurar uma instância de origem externa e uma réplica no Amazon RDS:

- Monitore eventos de failover da instância do banco de dados do RDS para MariaDB que é sua réplica. Se ocorrer um failover, a instância de banco de dados que for a sua réplica poderá ser recriada em um novo host com um endereço de rede diferente. Para obter informações sobre como monitorar eventos de failover, consulte Trabalhar com a notificação de eventos do Amazon RDS.
- Mantenha os logs binários (binlogs) em sua instância de origem até confirmar que foram aplicados à réplica. Essa manutenção garante que seja possível restaurar a instância de origem em caso de falha.
- Ative backups automáticos em sua instância do banco de dados MariaDB no Amazon RDS. Ativar backups automáticos garante que seja possível restaurar sua réplica para um determinado ponto no tempo, caso precise ressincronizar sua instância de origem e sua réplica. Para obter informações sobre backups e restauração Point-In-Time, consulte Backup, restauração e exportação de dados.

> **Note**
>
> As permissões necessárias para iniciar a replicação em uma instância de banco de dados MariaDB são restritas e não estão disponíveis ao seu usuário mestre do Amazon RDS. Devido a isso, você deve usar os comandos mysql.rds_set_external_master_gtid e mysql.rds_start_replication do Amazon RDS para configurar a replicação entre seu banco de dados em e o seu banco de dados do RDS para MariaDB.

Para iniciar a replicação entre uma instância de origem externa e uma instância de banco de dados MariaDB no Amazon RDS, use o seguinte procedimento.

Como iniciar a replicação

1. Confirme que a instância de origem MariaDB é somente leitura:

   ```
   mysql> FLUSH TABLES WITH READ LOCK;
   mysql> SET GLOBAL read_only = ON;
   ```

2. Obtenha o GTID atual da instância externa do MariaDB. Você pode fazer isso usando `mysql` ou o editor de consulta de sua preferência para executar `SELECT @@gtid_current_pos;`.

   O GTID é formatado como `<domain-id>-<server-id>-<sequence-id>`. Um GTID típico é parecido com **`0-1234510749-1728`**. Para obter mais informações sobre GTIDs e suas partes de componentes, consulte ID de transação global na documentação do MariaDB.

3. Copie o banco de dados da instância do MariaDB externa para a instância do banco de dados MariaDB usando `mysqldump`. Para bancos de dados muito grandes, convém usar o procedimento em Importar dados para um banco de dados do Amazon RDS para MySQL com tempo de inatividade reduzido.

   Para Linux, macOS ou Unix:

   ```
   mysqldump \
       --databases database_name \
       --single-transaction \
       --compress \
       --order-by-primary \
       -u local_user \
       -plocal_password | mysql \
           --host=hostname \
   ```

```
        --port=3306 \
        -u RDS_user_name \
        -pRDS_password
```

Para Windows:

```
mysqldump ^
    --databases database_name ^
    --single-transaction ^
    --compress ^
    --order-by-primary \
    -u local_user \
    -plocal_password | mysql ^
        --host=hostname ^
        --port=3306 ^
        -u RDS_user_name ^
        -pRDS_password
```

> **Note**
>
> Confirme que não há um espaço entre a opção `-p` e a senha inserida. Especifique uma senha diferente do prompt mostrado aqui como prática recomendada de segurança.

Use as opções `--host`, `--user (-u)`, `--port` e `-p` no comando `mysql` para especificar o nome do host, o nome do usuário, a porta e a senha para se conectar a sua instância de banco de dados MariaDB. O nome do host é o nome de DNS do endpoint da instância de banco de dados MariaDB, por exemplo, `myinstance.123456789012.us-east-1.rds.amazonaws.com`. Você pode encontrar o valor do endpoint nos detalhes da instância no Console de gerenciamento do Amazon RDS.

4. Confirme que é possível gravar na instância do MariaDB novamente.

```
mysql> SET GLOBAL read_only = OFF;
mysql> UNLOCK TABLES;
```

5. No console de gerenciamento do Amazon RDS, adicione o endereço IP do servidor, que hospeda o banco de dados MariaDB externo, ao grupo de segurança da VPC para a instância de banco de dados MariaDB. Para obter mais informações sobre como modificar um grupo de

segurança da VPC, acesse Grupos de segurança para a VPC no Guia do usuário da Amazon Virtual Private Cloud.

O endereço IP poderá ser alterado se as seguintes condições forem atendidas:

- Se você estiver usando um endereço IP público para a comunicação entre a instância de origem externa e a instância de banco de dados.
- A instância de origem externa foi interrompida e reiniciada.

Se essas condições forem atendidas, verifique o endereço IP antes de adicioná-lo.

Talvez também seja necessário configurar sua rede local para permitir conexões com o endereço IP de sua instância de banco de dados MariaDB, para que ela possa se comunicar com sua instância externa do MariaDB. Para encontrar o endereço IP da instância de banco de dados MariaDB, use o comando `host`.

```
host db_instance_endpoint
```

O nome do host é o nome de DNS do endpoint da instância de banco de dados MariaDB.

6. Usando o cliente de sua preferência, conecte-se à instância MariaDB externa e crie um usuário do MariaDB a ser usado para a replicação. Esta conta é usada unicamente para replicação e deve estar restrita ao seu domínio para melhorar a segurança. Veja um exemplo a seguir.

```
CREATE USER 'repl_user'@'mydomain.com' IDENTIFIED BY 'password';
```

> **Note**
>
> Especifique uma senha diferente do prompt mostrado aqui como prática recomendada de segurança.

7. Para a instância externa do MariaDB, conceda privilégios de `REPLICATION CLIENT` e `REPLICATION SLAVE` para seu usuário de replicação. Por exemplo, para conceder os privilégios de `REPLICATION CLIENT` e `REPLICATION SLAVE` em todos os bancos de dados para o usuário '`repl_user`' de seu domínio, emita o seguinte comando.

```
GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'mydomain.com';
```

8. Torne uma réplica a instância de banco de dados MariaDB. Conecte-se à instância de banco de dados MariaDB como o usuário mestre e identifique o banco de dados MariaDB externo como a instância de origem de replicação usando o comando mysql.rds_set_external_master_gtid. Use o GTID que você determinou na etapa 2. Veja um exemplo a seguir.

```
CALL mysql.rds_set_external_master_gtid ('mymasterserver.mydomain.com', 3306,
 'repl_user', 'password', 'GTID', 1);
```

> **Note**
>
> Especifique uma senha diferente do prompt mostrado aqui como prática recomendada de segurança.

9. Na instância de banco de dados MariaDB, emita o comando mysql.rds_start_replication, para iniciar a replicação:

```
CALL mysql.rds_start_replication;
```

Configurar a replicação da posição do arquivo de log binário com uma instância de origem externa

É possível configurar a replicação entre uma instância de banco de dados do RDS para MySQL ou do MariaDB e uma instância do MySQL ou do MariaDB externa ao Amazon RDS usando a replicação de arquivos de log binários.

Tópicos

- Antes de começar
- Configurar a replicação da posição do arquivo de log binário com uma instância de origem externa

Antes de começar

Você pode configurar a replicação usando a posição do arquivo de log binário de transações replicadas.

As permissões necessárias para iniciar a replicação em uma instância de banco de dados do Amazon RDS são restritas e não estão disponíveis ao seu usuário mestre do Amazon RDS. Por isso,

use os comandos mysql.rds_set_external_master (RDS para MariaDB e RDS para MySQL versões principais 8.0 e anteriores) ou mysql.rds_set_external_source (RDS para MySQL versões principais 8.4 e posteriores) e mysql.rds_start_replication do Amazon RDS para configurar a replicação entre seu banco de dados on-line e seu banco de dados do Amazon RDS.

Para definir o formato de registro em log binário para um banco de dados MySQL e MariaDB, atualize o parâmetro `binlog_format`. Se a instância de banco de dados usar o grupo de parâmetros da instância de banco de dados padrão, crie um novo grupo de parâmetros de banco de dados para modificar o parâmetro `binlog_format`. No MariaDB e no MySQL 8.0 e versões anteriores, `binlog_format` tem como padrão `MIXED`. No entanto, você também pode definir o `binlog_format` como `ROW` ou `STATEMENT` se precisar de um formato específico de log binário (binlog). Reinicie a instância de banco de dados para que a alteração entre em vigor. No MySQL 8.4 e versões superiores, `binlog_format` tem como padrão `ROW`.

Para obter informações sobre como definir o parâmetro `binlog_format`, consulte Configurar o registro em log binário do RDS para MySQL para bancos de dados single-AZ. Para obter informações sobre as implicações de tipos de replicação do MySQL diferentes, consulte Vantagens e desvantagens da replicação baseada em instrução e baseada em linha na documentação do MySQL.

Configurar a replicação da posição do arquivo de log binário com uma instância de origem externa

Siga estas diretrizes ao configurar uma instância de origem externa e uma réplica no Amazon RDS:

- Monitore eventos de failover para a instância do banco de dados do Amazon RDS, que é sua réplica. Se ocorrer um failover, a instância de banco de dados que for a sua réplica poderá ser recriada em um novo host com um endereço de rede diferente. Para obter informações sobre como monitorar eventos de failover, consulte Trabalhar com a notificação de eventos do Amazon RDS.
- Retenha os logs binários na instância de origem até confirmar que eles foram aplicados à réplica. Esta manutenção garante que seja possível restaurar sua instância de origem em caso de falha.
- Ative backups automatizados na sua instância de banco de dados do Amazon RDS. Ativar backups automáticos garante que seja possível restaurar sua réplica para um determinado ponto no tempo, caso seja necessário ressincronizar sua instância de origem e sua réplica. Para obter informações sobre backups e restauração Point-In-Time, consulte Backup, restauração e exportação de dados.

Como configurar a replicação do arquivo de log binário com uma instância de origem externa

1. Torne a instância de origem MySQL ou MariaDB somente leitura.

```
mysql> FLUSH TABLES WITH READ LOCK;
mysql> SET GLOBAL read_only = ON;
```

2. Execute o comando `SHOW MASTER STATUS` na instância do MySQL ou MariaDB de origem para determinar a localização do log binário.

   Você recebe um resultado semelhante ao seguinte exemplo.

```
File                          Position
-----------------------------------
 mysql-bin-changelog.000031      107
-----------------------------------
```

3. Copie o banco de dados da instância externa para a instância de banco de dados do Amazon RDS usando `mysqldump`. Para bancos de dados muito grandes, convém usar o procedimento em Importar dados para um banco de dados do Amazon RDS para MySQL com tempo de inatividade reduzido.

   Para Linux, macOS ou Unix:

```
mysqldump --databases database_name \
    --single-transaction \
    --compress \
    --order-by-primary \
    -u local_user \
    -plocal_password | mysql \
        --host=hostname \
        --port=3306 \
        -u RDS_user_name \
        -pRDS_password
```

   Para Windows:

```
mysqldump --databases database_name ^
    --single-transaction ^
    --compress ^
    --order-by-primary ^
```

```
    -u local_user ^
    -plocal_password | mysql ^
        --host=hostname ^
        --port=3306 ^
        -u RDS_user_name ^
        -pRDS_password
```

> **Note**
>
> Confirme que não há um espaço entre a opção `-p` e a senha inserida.

Para especificar o nome do host, o nome do usuário, a porta e a senha para se conectar à instância de banco de dados Amazon RDS, use as opções `--host`, `--user (-u)`, `--port` e `-p` no comando `mysql`. O nome do host é o nome do Serviço de Nome de Domínio (DNS) do endpoint da instância de banco de dados do Amazon RDS, por exemplo, `myinstance.123456789012.us-east-1.rds.amazonaws.com`. Você pode encontrar o valor do endpoint nos detalhes da instância no AWS Management Console.

4. Torne a instância MySQL ou MariaDB de origem novamente gravável.

```
mysql> SET GLOBAL read_only = OFF;
mysql> UNLOCK TABLES;
```

Para obter mais informações sobre como fazer backups para usar com a replicação, consulte a documentação do MySQL.

5. No AWS Management Console, adicione o endereço IP do servidor que hospeda o banco de dados externo ao grupo de segurança da nuvem privada virtual (VPC) para a instância de banco de dados Amazon RDS. Para obter mais informações sobre como modificar um grupo de segurança da VPC, consulte Grupos de segurança para a VPC no Guia do usuário da Amazon Virtual Private Cloud.

   O endereço IP poderá ser alterado se as seguintes condições forem atendidas:

   - Se você estiver usando um endereço IP público para a comunicação entre a instância de origem externa e a instância de banco de dados.
   - A instância de origem externa foi interrompida e reiniciada.

Se essas condições forem atendidas, verifique o endereço IP antes de adicioná-lo.

Você também pode precisar configurar a rede local para permitir conexões com o endereço IP da instância de banco de dados do Amazon RDS. Isso é realizado para que a rede local consiga se comunicar com a instância do MySQL ou do MariaDB externa. Para localizar o endereço IP da instância de banco de dados do Amazon RDS, use o comando `host`.

```
host db_instance_endpoint
```

O nome do host é o nome de DNS do endpoint da instância de banco de dados do Amazon RDS.

6. Usando o cliente de sua preferência, conecte-se à instância externa e crie um usuário a ser usado para a replicação. Use essa conta unicamente para replicação e restrinja-a ao seu domínio para melhorar a segurança. Veja um exemplo a seguir.

```
CREATE USER 'repl_user'@'mydomain.com' IDENTIFIED BY 'password';
```

> **Note**
>
> Especifique uma senha diferente do prompt mostrado aqui como prática recomendada de segurança.

7. Para a instância externa, conceda os privilégios `REPLICATION CLIENT` e `REPLICATION SLAVE` para seu usuário de replicação. Por exemplo, para conceder os privilégios de `REPLICATION CLIENT` e `REPLICATION SLAVE` em todos os bancos de dados para o usuário '`repl_user`' de seu domínio, emita o seguinte comando.

```
GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'mydomain.com';
```

8. Transforma a instância de banco de dados do Amazon RDS na réplica. Para fazer isso, primeiro se conecte à instância de banco de dados do Amazon RDS como usuário mestre. Depois disso, identifique o banco de dados MySQL ou MariaDB externo como a instância de origem usando o comando mysql.rds_set_external_source (RDS para MySQL versões principais 8.4 e posteriores) ou mysql.rds_set_external_master (RDS para MariaDB e RDS para MySQL versões principais 8.0 e anteriores). Use o nome do arquivo de log mestre e a posição do log mestre que você determinou na etapa 2. Os comandos a seguir são exemplos.

MySQL 8.4

```
CALL mysql.rds_set_external_source ('mysourceserver.mydomain.com', 3306,
 'repl_user', 'password', 'mysql-bin-changelog.000031', 107, 1);
```

MariaDB e MySQL 8.0 e 5.7

```
CALL mysql.rds_set_external_master ('mymasterserver.mydomain.com', 3306,
 'repl_user', 'password', 'mysql-bin-changelog.000031', 107, 1);
```

> **Note**
>
> No RDS para MySQL, é possível optar por usar a replicação atrasada executando o procedimento armazenado mysql.rds_set_external_source_with_delay (RDS para MySQL versões principais 8.4 e superiores) ou mysql.rds_set_external_master_with_delay (RDS para MariaDB e RDS para MySQL versões principais 8.0 e anteriores). No RDS para MySQL, um motivo para usar a replicação atrasada é ativar a recuperação de desastres com o procedimento armazenado mysql.rds_start_replication_until. Atualmente, o RDS para MariaDB é compatível com a replicação atrasada, mas não com o procedimento `mysql.rds_start_replication_until`.

9. Na instância de banco de dados do Amazon RDS, emita o comando mysql.rds_start_replication para iniciar a replicação.

```
CALL mysql.rds_start_replication;
```

Opções para o mecanismo de banco de dados do MariaDB

A seguir, você encontrará uma descrição das opções, ou recursos adicionais, que estão disponíveis para instâncias do Amazon RDS que executam o mecanismo de banco de dados do MariaDB. Para ativar essas opções, adicione-as a um grupo de opções personalizado e, depois, associe o grupo de opções à sua instância de banco de dados. Para ter mais informações sobre como trabalhar com grupos de opções, consulte Trabalhar com grupos de opções.

O Amazon RDS oferece suporte às seguintes opções do MariaDB:

ID da opção	Versões do mecanismo
`MARIADB_AUDIT_PLUGIN`	MariaDB 10.3 e posterior

Suporte ao plugin de auditoria do MariaDB

O Amazon RDS oferece suporte ao MariaDB Audit Plugin nas instâncias do banco de dados do MariaDB. O MariaDB Audit Plugin registra a atividade do banco de dados, como usuários que fazem logon no banco de dados, as consultas são executadas contra o banco de dados e muito mais. O registro da atividade do banco de dados é armazenado em um arquivo de log.

Definições de opções do plugin de auditoria

O Amazon RDS oferece suporte para as seguintes configurações para a opção MariaDB Audit Plugin.

> **Note**
>
> Se você não definir uma configuração de opção no console do RDS, o RDS usará a configuração padrão.

Configuração da opção	Valores válidos	Valor padrão	Descrição
`SERVER_AUDIT_FILE_PATH`	`/rdsdbdata/log/audit/`	`/rdsdbdata/log/audit/`	A localização dos arquivos de log. O arquivo de log contém o registro de atividades especificadas em `SERVER_AUDIT_EVENTS` . Para

Configuração da opção	Valores válidos	Valor padrão	Descrição
			ter mais informações, consulte Como visualizar e listar arquivos de log do banco de dados e Arquivos de log do banco de dados MariaDB.
SERVER_AUDIT_FILE_ROTATE_SIZE	1–1000000000	1000000	O tamanho em bytes que, quando alcançado, faz com que o arquivo rotacione. Para ter mais informações, consulte Alternância e retenção de logs para o MariaDB.
SERVER_AUDIT_FILE_ROTATIONS	0–100	9	O número de rotações de log para salvar quando `server_audit_output_type=file` . Se definido como 0, o arquivo de log nunca é alternado. Para ter mais informações, consulte Alternância e retenção de logs para o MariaDB e Como baixar um arquivo de log de banco de dados.

Configuração da opção	Valores válidos	Valor padrão	Descrição
SERVER_AUDIT_EVENTS	CONNECT, QUERY, TABLE, QUERY_DDL , QUERY_DML , QUERY_DML_NO_SELECT , QUERY_DCL	CONNECT, QUERY	Os tipos de atividades a serem gravados no log. A instalação do MariaDB Audit Plugin é registrada em log. • CONNECT: registrar conexões bem-sucedidas e sem êxito com o banco de dados e desconexões do banco de dados. • QUERY: registrar o texto de todas as consultas executadas no banco de dados. • TABLE: tabelas de log afetadas por consultas quando as consultas são executadas no banco de dados. • QUERY_DDL : semelhante ao evento de QUERY, mas retorna somente consultas de linguagem de definição de dados (DDL) (CREATE, ALTER, etc.) • QUERY_DML : semelhante ao evento de QUERY, mas retorna somente consultas de linguagem de manipulação de dados (DML) (INSERT, UPDATE, etc., bem como SELECT). • QUERY_DML_NO_SELECT : semelhante ao evento QUERY_DML , mas não registra consultas de log SELECT. • QUERY_DCL : semelhante ao evento de QUERY, mas retorna somente consultas de linguagem de controle de dados (DCL) (GRANT, REVOKE, etc.)

Configuração da opção	Valores válidos	Valor padrão	Descrição
SERVER_AUDIT_INCL_USERS	Vários valores separados por vírgulas	Nenhum	Inclua apenas atividades dos usuários especificados. Por padrão, a atividade é registrada para todos os usuários. SERVER_AUDIT_INCL_USERS e SERVER_AUDIT_EXCL_USERS são mutuamente exclusivos. Se você adicionar valores ao SERVER_AUDIT_INCL_USERS , certifique-se de que nenhum valor seja adicionado ao SERVER_AUDIT_EXCL_USERS .

Configuração da opção	Valores válidos	Valor padrão	Descrição
SERVER_AUDIT_EXCL_USERS	Vários valores separados por vírgulas	Nenhum	Exclua a atividade dos usuários especificados. Por padrão, a atividade é registrada para todos os usuários. SERVER_AUDIT_INCL_USERS e SERVER_AUDIT_EXCL_USERS são mutuamente exclusivos. Se você adicionar valores ao SERVER_AUDIT_EXCL_USERS , certifique-se de que nenhum valor seja adicionado ao SERVER_AUDIT_INCL_USERS . O usuário rdsadmin consulta o banco de dados a cada segundo para verificar a integridade do banco de dados. Dependendo das suas outras configurações, essa atividade pode fazer com que o tamanho do seu arquivo de log cresça muito rapidamente. Se você não precisa registrar essa atividade, adicione o usuário rdsadmin à lista SERVER_AUDIT_EXCL_USERS . **Note** CONNECTA atividade é sempre registrada para todos os usuários, mesmo se o usuário é especificado para essa configuração de opção.
SERVER_AUDIT_LOGGING	ON	ON	O registro em log está ativo. O único valor válido é ON. O Amazon RDS não oferece suporte à desativação do registro em log. Se quiser desativar o registro log, remova o MariaDB Audit Plugin. Para ter mais informações, consulte Remover o MariaDB Audit Plugin.

Configuração da opção	Valores válidos	Valor padrão	Descrição
SERVER_AUDIT_QUERY_LOG_LIMIT	0–2147483647	1024	O limite do tamanho da string de consulta em um registro.

Adicionar o MariaDB Audit Plugin

O processo geral para adicionar o MariaDB Audit Plugin a uma instância de banco de dados é o seguinte:

1. Crie um novo grupo de opções, ou copie ou modifique um existente.
2. Adicione a opção ao grupo de opções.
3. Associe o grupo de opções à instância de banco de dados.

Depois de adicionar o MariaDB Audit Plugin, você não precisará reiniciar sua instância de banco de dados. Assim que o grupo de opções estiver ativo, a auditoria começará imediatamente.

Para adicionar o MariaDB Audit Plugin

1. Determine o grupo de opções que você deseja usar. Você pode criar um novo grupo de opções ou usar um existente. Se você quiser usar um grupo de opções existente, vá para a próxima etapa. Caso contrário, crie um grupo de opções de banco de dados personalizado. Escolha mariadb em Engine (Mecanismo) e escolha 10.3 ou posteriores para Major engine version (Versão principal do mecanismo). Para ter mais informações, consulte Criar um grupo de opções.
2. Adicione a opção MARIADB_AUDIT_PLUGIN ao grupo de opções e defina as configurações da opção. Para ter mais informações sobre a adição de opções, consulte Adicionar uma opção a um grupo de opções. Para ter mais informações sobre cada configuração, consulte Definições de opções do plugin de auditoria.
3. Aplique o grupo de opções a uma instância de banco de dados nova ou existente.

- Para uma nova instância de banco de dados, você aplica o grupo de opções ao executar a instância. Para ter mais informações, consulte Criar uma instância de banco de dados do Amazon RDS.
- Para uma instância de banco de dados existente, aplique o grupo de opções modificando a instância de banco de dados e anexando o novo grupo de opções. Para ter mais informações, consulte Modificar uma instância de banco de dados do Amazon RDS.

Visualizar e baixar o log do plugin de auditoria do MariaDB

Depois de habilitar o MariaDB Audit Plugin, você acessará os resultados nos arquivos de log da mesma forma que você acessa outros arquivos de log baseados em texto. Os arquivos de log de auditoria estão localizados em `/rdsdbdata/log/audit/`. Para obter informações sobre como visualizar o arquivo de log no console, consulte Como visualizar e listar arquivos de log do banco de dados. Para obter informações sobre como baixar o arquivo de log, consulte Como baixar um arquivo de log de banco de dados.

Modificar as configurações do plugin de auditoria do MariaDB

Após habilitar o MariaDB Audit Plugin, você pode modificar as configurações do plugin. Para ter mais informações sobre como modificar as configurações da opção, consulte Modificar uma configuração de opção. Para ter mais informações sobre cada configuração, consulte Definições de opções do plugin de auditoria.

Remover o MariaDB Audit Plugin

O Amazon RDS não oferece suporte à desativação do registro em log no MariaDB Audit Plugin. No entanto, você pode remover o plugin de uma instância de banco de dados. Quando você remove o MariaDB Audit Plugin, a instância de banco de dados é reiniciada automaticamente para interromper a auditoria.

Para remover o MariaDB Audit Plugin de uma instância de banco de dados, siga um destes procedimentos:

- Remova a opção MariaDB Audit Plugin do grupo de opções ao qual ela pertence. Essa alteração afeta todas as instâncias de bancos de dados que usam o grupo de opções. Para ter mais informações, consulte Remover uma opção de um grupo de opções
- Modifique a instância de banco de dados e especifique um grupo de opções diferente que não inclua o plugin. Essa alteração afeta uma única instância de banco de dados. Você pode

especificar um grupo de opções padrão (vazio) ou criar um grupo de opções personalizado diferente. Para ter mais informações, consulte Modificar uma instância de banco de dados do Amazon RDS.

Parâmetros para MariaDB

Por padrão, uma instância de banco de dados MariaDB usa um grupo de parâmetros de banco de dados específico para um banco de dados MariaDB. Esse grupo de parâmetros contém alguns mas não todos os parâmetros contidos nos grupos de parâmetros de banco de dados do Amazon RDS para o mecanismo de banco de dados MySQL. Ele também contém alguns parâmetros novos específicos do MariaDB. Para obter informações sobre como trabalhar com grupos de parâmetros e definir parâmetros, consulte Grupos de parâmetros para Amazon RDS.

Visualizar parâmetros do MariaDB

Os parâmetros do RDS for MariaDB são definidos com os valores padrão do mecanismo de armazenamento que você selecionou. Para obter mais informações sobre os parâmetros do MariaDB, consulte a documentação do MariaDB. Para obter mais informações sobre os mecanismos de armazenamento do MariaDB, consulte Mecanismos de armazenamento compatíveis com MariaDB no Amazon RDS.

Você pode exibir os parâmetros disponíveis para uma versão específica do RDS for MariaDB usando o console do RDS ou a AWS CLI. Para obter informações sobre como visualizar os parâmetros em um grupo de parâmetros do MariaDB no console do RDS, consulte Visualizar valores de parâmetros para um grupo de parâmetros de banco de dados no Amazon RDS.

Usando a AWS CLI, você pode visualizar os parâmetros de uma versão do RDS for MariaDB executando o comando `describe-engine-default-parameters`. Especifique um destes valores a seguir para a opção `--db-parameter-group-family`:

- `mariadb10.11`
- `mariadb10.6`
- `mariadb10.5`
- `mariadb10.4`
- `mariadb10.3`

Por exemplo, para visualizar parâmetros do RDS for MariaDB, versão 10.6, execute o comando a seguir.

```
aws rds describe-engine-default-parameters --db-parameter-group-family mariadb10.6
```

O resultado será semelhante ao mostrado a seguir.

```
{
    "EngineDefaults": {
        "Parameters": [
            {
                "ParameterName": "alter_algorithm",
                "Description": "Specify the alter table algorithm.",
                "Source": "engine-default",
                "ApplyType": "dynamic",
                "DataType": "string",
                "AllowedValues": "DEFAULT,COPY,INPLACE,NOCOPY,INSTANT",
                "IsModifiable": true
            },
            {
                "ParameterName": "analyze_sample_percentage",
                "Description": "Percentage of rows from the table ANALYZE TABLE will
 sample to collect table statistics.",
                "Source": "engine-default",
                "ApplyType": "dynamic",
                "DataType": "float",
                "AllowedValues": "0-100",
                "IsModifiable": true
            },
            {
                "ParameterName": "aria_block_size",
                "Description": "Block size to be used for Aria index pages.",
                "Source": "engine-default",
                "ApplyType": "static",
                "DataType": "integer",
                "AllowedValues": "1024-32768",
                "IsModifiable": false
            },
            {
                "ParameterName": "aria_checkpoint_interval",
                "Description": "Interval in seconds between automatic checkpoints.",
                "Source": "engine-default",
                "ApplyType": "dynamic",
                "DataType": "integer",
                "AllowedValues": "0-4294967295",
                "IsModifiable": true
            },
        ...
```

Para listar somente os parâmetros modificáveis do RDS for MariaDB versão 10.6, execute o comando a seguir.

Para Linux, macOS ou Unix:

```
aws rds describe-engine-default-parameters --db-parameter-group-family mariadb10.6 \
   --query 'EngineDefaults.Parameters[?IsModifiable==`true`]'
```

Para Windows:

```
aws rds describe-engine-default-parameters --db-parameter-group-family mariadb10.6 ^
   --query "EngineDefaults.Parameters[?IsModifiable==`true`]"
```

Parâmetros do MySQL que não estão disponíveis

Os parâmetros MySQL a seguir não estão disponíveis nos parameter groups de banco de dados específicos do MariaDB:

- bind_address
- binlog_error_action
- binlog_gtid_simple_recovery
- binlog_max_flush_queue_time
- binlog_order_commits
- binlog_row_image
- binlog_rows_query_log_events
- binlogging_impossible_mode
- block_encryption_mode
- core_file
- default_tmp_storage_engine
- div_precision_increment
- end_markers_in_json
- enforce_gtid_consistency
- eq_range_index_dive_limit
- explicit_defaults_for_timestamp
- gtid_executed

- gtid-mode
- gtid_next
- gtid_owned
- gtid_purged
- log_bin_basename
- log_bin_index
- log_bin_use_v1_row_events
- log_slow_admin_statements
- log_slow_slave_statements
- log_throttle_queries_not_using_indexes
- master-info-repository
- optimizer_trace
- optimizer_trace_features
- optimizer_trace_limit
- optimizer_trace_max_mem_size
- optimizer_trace_offset
- relay_log_info_repository
- rpl_stop_slave_timeout
- slave_parallel_workers
- slave_pending_jobs_size_max
- slave_rows_search_algorithms
- storage_engine
- table_open_cache_instances
- timed_mutexes
- transaction_allow_batching
- validate-password
- validate_password_dictionary_file
- validate_password_length
- validate_password_mixed_case_count
- validate_password_number_count

- validate_password_policy
- validate_password_special_char_count

Para obter mais informações sobre os parâmetros do MySQL, consulte a documentação do MySQL.

Migrar dados de um snapshot de banco de dados do MySQL para uma instância de banco de dados do MariaDB

É possível migrar um snapshot de banco de dados do RDS para MySQL para uma nova instância de banco de dados que executa o MariaDB utilizando o AWS Management Console, a AWS CLI ou a API do Amazon RDS. É necessário usar um snapshot de banco de dados criado em uma instância de banco de dados do Amazon RDS executando o MySQL 5.6 ou 5.7. Para saber como criar um snapshot de banco de dados do RDS para MySQL, consulte Criar um snapshot de banco de dados para uma instância de banco de dados single-AZ para o Amazon RDS.

A migração do snapshot não afeta a instância de banco de dados original da qual o snapshot foi tirado. Você pode testar e validar a nova instância de banco de dados antes de desviar o tráfego para ela como um substituto para a instância de banco de dados original.

Após a migração do MySQL para o MariaDB, a instância de banco de dados do MariaDB será associada ao grupo de parâmetros de banco de dados e ao grupo de opções. Depois de restaurar o snapshot de banco de dados, você poderá associar um grupo de parâmetros de banco de dados personalizado à nova instância de banco de dados. No entanto, um grupo de parâmetros do MariaDB apresenta um conjunto diferente de variáveis de sistema configuráveis. Para obter informações sobre as diferenças entre as variáveis de sistema do MySQL e do MariaDB, consulte Diferenças em variáveis de sistema entre o MariaDB e o MySQL. Para saber mais sobre grupos de parâmetros de banco de dados, consulte Grupos de parâmetros para Amazon RDS. Para saber mais sobre option groups, consulte Trabalhar com grupos de opções.

Executar a migração

Você pode migrar um snapshot de banco de dados do RDS para MySQL para uma nova instância de banco de dados MariaDB usando o AWS Management Console, a AWS CLI ou a API do RDS.

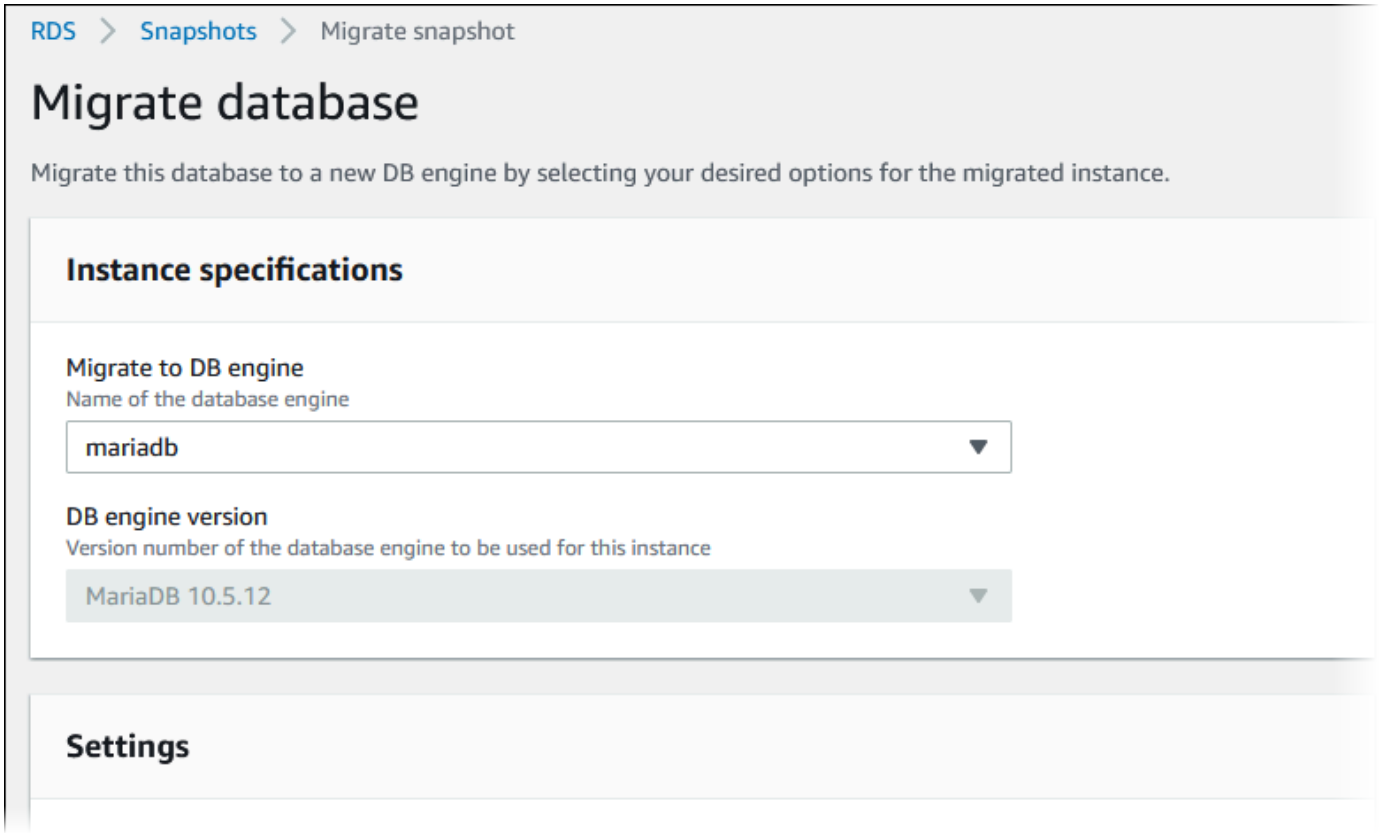
Console

Para migrar um snapshot de banco de dados do MySQL para uma instância de banco de dados do MariaDB

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, selecione Snapshots e, em seguida, selecione o snapshot de banco de dados MySQL que você deseja migrar.

3. Em Actions (Ações), escolha Migrate snapshot (Migrar snapshot). A página Migrate Database (Migrar banco de dados) é exibida.
4. Em Migrate to DB Engine (Migrar para o mecanismo de banco de dados), selecione mariadb.

   O Amazon RDS seleciona uma opção para DB engine version (Versão do mecanismo de banco de dados) automaticamente. Não é possível alterar a versão do mecanismo de banco de dados.



5. Nas seções restantes, especifique suas configurações de instância de banco de dados. Para obter informações sobre cada configuração, consulte Configurações para instâncias de banco de dados.
6. Escolha Migrate (Migrar).

AWS CLI

Para migrar dados de um snapshot de banco de dados do MySQL para uma instância de banco de dados MariaDB, use o comando `restore-db-instance-from-db-snapshot` da AWS CLI com as seguintes opções:

- --db-instance-identifier – Nome da instância de banco de dados para criar a partir do snapshot de banco de dados.

- --db-snapshot-identifier – O identificador para restaurar o snapshot de banco de dados.
- --engine – O mecanismo de banco de dados a utilizar na nova instância.

Example

Para Linux, macOS ou Unix:

```
aws rds restore-db-instance-from-db-snapshot \
    --db-instance-identifier newmariadbinstance \
    --db-snapshot-identifier mysqlsnapshot \
    --engine mariadb
```

Para Windows:

```
aws rds restore-db-instance-from-db-snapshot ^
    --db-instance-identifier newmariadbinstance ^
    --db-snapshot-identifier mysqlsnapshot ^
    --engine mariadb
```

API

Para migrar dados de um snapshot de banco de dados do MySQL para uma instância de banco de dados do MariaDB, chame a operação `RestoreDBInstanceFromDBSnapshot` da API do Amazon RDS.

Incompatibilidades entre o MariaDB e o MySQL

As incompatibilidades entre o MySQL e o MariaDB incluem o seguinte:

- Não é possível migrar um snapshot de banco de dados criado com o MySQL 8.0 para o MariaDB.
- Se o banco de dados MySQL usa um hash de senha SHA256, certifique-se de redefinir senhas de usuário com hash SHA256 antes de se conectar ao banco de dados MariaDB. O código a seguir mostra como redefinir uma senha com hash SHA256.

```
SET old_passwords = 0;
UPDATE mysql.user SET plugin = 'mysql_native_password',
Password = PASSWORD('new_password')
WHERE (User, Host) = ('master_user_name', %);
```

```
FLUSH PRIVILEGES;
```

- Se a sua conta de usuário primária do RDS utilizar o hash de senha SHA-256, redefina a senha utilizando o AWS Management Console, o comando `modify-db-instance` da AWS CLI ou a operação ModifyDBInstance da API do RDS. Para ter mais informações sobre como modificar uma instância de banco de dados , consulte Modificar uma instância de banco de dados do Amazon RDS.
- O MariaDB não é compatível com o plugin Memcached. No entanto, os dados usados pelo plugin Memcached são armazenados como tabelas do InnoDB. Após migrar um snapshot de banco de dados do MySQL, você poderá acessar os dados usados pelo plugin Memcached usando o SQL. Para mais informações sobre o banco de dados innodb_memcache, consulte Elementos internos do plugin memcached do InnoDB.

Referência do MariaDB no SQL do Amazon RDS

A seguir, há descrições de procedimentos armazenados em sistema que estão disponíveis para instâncias do Amazon RDS em execução no mecanismo de banco de dados do MariaDB.

Também é possível usar os procedimentos armazenados em sistema que estão disponíveis para as instâncias de banco de dados MySQL para instâncias de banco de dados MariaDB. Esses procedimentos armazenados estão documentados em Referência de procedimentos armazenados do RDS para MySQL. As instâncias de banco de dados MariaDB são compatíveis com todos os procedimentos armazenados, exceto `mysql.rds_start_replication_until` e `mysql.rds_start_replication_until_gtid`.

Além disso, os procedimentos armazenados em sistema a seguir são compatíveis apenas para instâncias de banco de dados do Amazon RDS que executam o MariaDB:

- mysql.rds_replica_status
- mysql.rds_set_external_master_gtid
- mysql.rds_kill_query_id

mysql.rds_replica_status

Mostra o status de replicação de uma réplica de leitura MariaDB.

Chame este procedimento na réplica de leitura para mostrar informações de status sobre parâmetros essenciais dos threads de réplica.

Sintaxe

```
CALL mysql.rds_replica_status;
```

Observações de uso

Esse procedimento só é compatível com instâncias de banco de dados MariaDB que executam MariaDB versão 10.5 e versões posteriores.

Esse procedimento é o equivalente ao comando `SHOW REPLICA STATUS`. Esse comando não é compatível com instâncias de banco de dados MariaDB versão 10.5 e posterior.

Nas versões anteriores do MariaDB, esse comando `SHOW SLAVE STATUS` exigia o privilégio `REPLICATION SLAVE`. No MariaDB versão 10.5 e posteriores, ele requer o privilégio `REPLICATION REPLICA ADMIN`. Para proteger o gerenciamento do RDS das instâncias de banco de dados MariaDB 10.5, esse novo privilégio não é concedido ao usuário principal do RDS.

Exemplos

O exemplo a seguir mostra o status de uma réplica de leitura MariaDB:

```
call mysql.rds_replica_status;
```

A resposta é semelhante à seguinte:

```
*************************** 1. row ***************************
               Replica_IO_State: Waiting for master to send event
                    Source_Host: XX.XX.XX.XXX
                    Source_User: rdsrepladmin
                    Source_Port: 3306
                  Connect_Retry: 60
                Source_Log_File: mysql-bin-changelog.003988
            Read_Source_Log_Pos: 405
                 Relay_Log_File: relaylog.011024
                  Relay_Log_Pos: 657
          Relay_Source_Log_File: mysql-bin-changelog.003988
             Replica_IO_Running: Yes
            Replica_SQL_Running: Yes
                Replicate_Do_DB:
            Replicate_Ignore_DB:
             Replicate_Do_Table:
         Replicate_Ignore_Table:
mysql.rds_sysinfo,mysql.rds_history,mysql.rds_replication_status
        Replicate_Wild_Do_Table:
    Replicate_Wild_Ignore_Table:
                     Last_Errno: 0
                     Last_Error:
                   Skip_Counter: 0
            Exec_Source_Log_Pos: 405
                Relay_Log_Space: 1016
                Until_Condition: None
                 Until_Log_File:
                  Until_Log_Pos: 0
             Source_SSL_Allowed: No
```

```
              Source_SSL_CA_File:
              Source_SSL_CA_Path:
                 Source_SSL_Cert:
               Source_SSL_Cipher:
                  Source_SSL_Key:
          Seconds_Behind_Master: 0
  Source_SSL_Verify_Server_Cert: No
                  Last_IO_Errno: 0
                  Last_IO_Error:
                 Last_SQL_Errno: 0
                 Last_SQL_Error:
    Replicate_Ignore_Server_Ids:
               Source_Server_Id: 807509301
                 Source_SSL_Crl:
             Source_SSL_Crlpath:
                     Using_Gtid: Slave_Pos
                   Gtid_IO_Pos: 0-807509301-3980
        Replicate_Do_Domain_Ids:
    Replicate_Ignore_Domain_Ids:
                  Parallel_Mode: optimistic
                      SQL_Delay: 0
            SQL_Remaining_Delay: NULL
      Replica_SQL_Running_State: Reading event from the relay log
             Replica_DDL_Groups: 15
Replica_Non_Transactional_Groups: 0
    Replica_Transactional_Groups: 3658
1 row in set (0.000 sec)

Query OK, 0 rows affected (0.000 sec)
```

mysql.rds_set_external_master_gtid

Configura a replicação baseada em GTID de uma instância do MariaDB executada externamente em relação ao Amazon RDS para uma instância de banco de dados MariaDB. Esse procedimento armazenado é compatível apenas com a versão 10.0.24 ou posteriores da instância externa do MariaDB. Ao configurar a replicação na qual uma ou ambas as instâncias não oferecem suporte a identificadores de transações globais (GTIDs) do MariaDB, use mysql.rds_set_external_master (RDS para MariaDB e RDS para MySQL versões principais 8.0 e anteriores).

O uso de GTIDs para replicação oferece recursos de segurança contra falhas não disponibilizados pela replicação de log binário, por isso recomendamos nos casos em que há o suporte pelas instâncias de replicação.

Sintaxe

```
CALL mysql.rds_set_external_master_gtid(
  host_name
  , host_port
  , replication_user_name
  , replication_user_password
  , gtid
  , ssl_encryption
);
```

Parâmetros

host_name

String. O nome de host ou o endereço IP da instância do MariaDB sendo executada externamente ao Amazon RDS que se tornará a instância de origem.

host_port

Inteiro. A porta usada para executar a instância do MariaDB sendo executada externamente ao Amazon RDS a ser configurada como a instância de origem. Se sua configuração de rede inclui replicação de porta SSH que converte o número da porta, especifique o número da porta exposto pelo SSH.

replication_user_name

String. O ID de um usuário com permissões `REPLICATION SLAVE` na instância de banco de dados MariaDB deve ser configurado como a réplica de leitura.

replication_user_password

String. A senha do ID de usuário especificada em `replication_user_name`.

gtid

String. O ID da transação global na instância de origem a partir da qual a replicação deve iniciar.

É possível usar o `@@gtid_current_pos` para obter o GTID atual se a instância de origem tiver sido bloqueada enquanto você estiver configurando a replicação, de modo que o log binário não mude entre os pontos quando você obtiver o GTID e quando a replicação iniciar.

Caso contrário, se você estiver usando o `mysqldump` versão 10.0.13 ou posterior para preencher a instância de réplica antes de iniciar a replicação, será possível obter a posição de GTID na saída usando as opções `--master-data` ou `--dump-slave`. Se você não estiver usando o `mysqldump` versão 10.0.13 ou superior, você poderá executar o `SHOW MASTER STATUS` ou usar as mesmas opções de `mysqldump` para obter o nome e a posição do arquivo de log binário, convertê-los em um GTID executando `BINLOG_GTID_POS` na instância externa do MariaDB:

```
SELECT BINLOG_GTID_POS('<binary log file name>', <binary log file position>);
```

Para obter mais informações sobre a implantação de GTIDs do MariaDB, acesse Global Transaction ID na documentação do MariaDB.

ssl_encryption

Um valor que especifica se a criptografia do Secure Sockets Layer (SSL) será usada na conexão de replicação. 1 especifica para usar criptografia de SSL, 0 especifica para não usar criptografia. O padrão é 0.

> **Note**
>
> A opção `MASTER_SSL_VERIFY_SERVER_CERT` não tem suporte. Essa opção é definida como 0, o que significa que a conexão é criptografada, mas os certificados não são verificados.

Observações de uso

O procedimento `mysql.rds_set_external_master_gtid` deve ser executado pelo usuário mestre. Ele deve ser executado na instância de banco de dados MariaDB que você está configurando como a réplica de uma instância do MariaDB, sendo executada externamente ao Amazon RDS. Antes de executar `mysql.rds_set_external_master_gtid`, é necessário configurar a instância do MariaDB sendo executada externamente ao Amazon RDS como uma instância de origem. Para obter mais informações, consulte Importar dados para uma instância de banco de dados do Amazon RDS para MariaDB.

> **Warning**
>
> Não use o `mysql.rds_set_external_master_gtid` para gerenciar uma replicação entre duas instâncias de banco de dados do Amazon RDS. Use-o apenas ao replicar com

uma instância MariaDB sendo executada externamente ao RDS. Para obter informações sobre como gerenciar a replicação entre instâncias de banco de dados do Amazon RDS, consulte Trabalhar com réplicas de leitura de instância de banco de dados.

Após chamar `mysql.rds_set_external_master_gtid` para configurar uma instância de banco de dados do Amazon RDS como uma réplica de leitura, você poderá chamar mysql.rds_start_replication na réplica para iniciar o processo de replicação. É possível chamar mysql.rds_reset_external_master (RDS para MariaDB e RDS para MySQL versões principais 8.0 e anteriores) para remover a configuração da réplica de leitura.

Quando `mysql.rds_set_external_master_gtid` é chamado, o Amazon RDS grava a hora, o usuário e uma ação do "mestre do conjunto" nas tabelas `mysql.rds_history` e `mysql.rds_replication_status`.

Exemplos

Quando executado em uma instância de banco de dados MariaDB, o exemplo a seguir o configura como a réplica de uma instância do MariaDB sendo executada externamente ao Amazon RDS.

```
call mysql.rds_set_external_master_gtid
 ('Sourcedb.some.com',3306,'ReplicationUser','SomePassW0rd','0-123-456',0);
```

mysql.rds_kill_query_id

Encerra uma consulta em execução no servidor do MariaDB para terminar consultas problemáticas ou de longa duração. É possível identificar o ID da consulta e interromper efetivamente uma consulta específica para resolver problemas de desempenho e manter a operação ideal do banco de dados.

Sintaxe

```
CALL mysql.rds_kill_query_id(queryID);
```

Parâmetros

queryID

Inteiro. A identidade da consulta a ser encerrada.

Observações de uso

Para encerrar uma consulta em execução no servidor do MariaDB, use o procedimento `mysql.rds_kill_query_id` e passe o ID da consulta. Para obter o ID da consulta, consulte a Tabela Information schema PROCESSLIST do MariaDB, como mostrado a seguir:

```
SELECT USER, HOST, COMMAND, TIME, STATE, INFO, QUERY_ID FROM
                INFORMATION_SCHEMA.PROCESSLIST WHERE USER = '<user name>';
```

A conexão ao servidor do MariaDB é retida.

Exemplos

O exemplo a seguir encerra uma consulta com um ID de consulta de 230040:

```
call mysql.rds_kill_query_id(230040);
```

Fuso horário local para as instâncias de banco de dados do MariaDB

Por padrão, o fuso horário de uma instância de banco de dados MariaDB é o Tempo Universal Coordenado (UTC). Você pode definir o fuso horário de sua instância de banco de dados como o fuso horário local de seu aplicativo.

Para definir o fuso horário local de uma instância de banco de dados, defina o parâmetro `time_zone` no grupo de parâmetros para sua instância de banco de dados como um dos valores compatíveis listados posteriormente nesta seção. Quando você configura o parâmetro `time_zone` para um grupo de parâmetros, todas as instâncias de banco de dados e as réplicas de leitura que estiverem usando esse grupo de parâmetros mudarão para usar o novo fuso horário local. Para obter informações sobre como configurar parâmetros em um grupo de parâmetros, consulte Grupos de parâmetros para Amazon RDS.

Após definir o fuso horário local, todas as novas conexões ao banco de dados refletirão a alteração. Se você tiver conexões abertas em seu banco de dados quando mudar o fuso horário local, você não verá a atualização do fuso horário local até que encerre a conexão e abra uma nova conexão.

É possível definir um fuso horário local diferente para uma instância de banco de dados e uma ou mais de suas réplicas de leitura. Para fazer isso, use um grupo de parâmetros diferente para a instância de banco de dados e a(s) replica(s), e defina o parâmetro `time_zone` em cada grupo de parâmetros para um fuso horário local diferente.

Se você estiver replicando entre as Regiões da AWS, a instância de banco de dados de origem e a réplica de leitura usarão grupos de parâmetros diferentes (os grupos de parâmetros são exclusivos de cada Região da AWS). Para usar o mesmo fuso horário local a cada instância, é necessário configurar o parâmetro `time_zone` nos grupos de parâmetros da instância e da réplica de leitura.

Quando você restaura uma instância de banco de dados de um snapshot de banco de dados, o fuso horário local é definido como UTC. Você pode atualizar o fuso horário para o fuso horário local após a conclusão da restauração. Se você restaurar uma instância de banco de dados para um período específico, o fuso horário local para a instância de banco de dados restaurada representará a configuração de fuso horário do grupo de parâmetros da instância de banco de dados restaurada.

A Internet Assigned Numbers Authority (IANA) publica novos fusos horários em https://www.iana.org/time-zones várias vezes por ano. Toda vez que o RDS lança uma nova versão de manutenção secundária do MariaDB, ela vem com os dados de fuso horário mais recentes no momento do

lançamento. Ao usar as versões mais recentes do RDS para MariaDB, você tem dados de fuso horário recentes do RDS. Para garantir que sua instância de banco de dados tenha dados de fuso horário recentes, recomendamos atualizar para uma versão superior do mecanismo de banco de dados. Como alternativa, você pode modificar as tabelas de fuso horário nas instâncias de banco de dados do MariaDB manualmente. Para fazer isso, você pode usar comandos SQL ou executar a ferramenta mysql_tzinfo_to_sql em um cliente SQL. Depois de atualizar os dados do fuso horário manualmente, reinicialize sua instância de banco de dados para que as alterações tenham efeito. O RDS não modifica nem redefine os dados de fuso horário das instâncias de banco de dados em execução. Os novos dados de fuso horário são instalados somente quando você executa uma atualização da versão do mecanismo de banco de dados.

Você pode definir seu fuso horário local para um dos seguintes valores.

Zona	Time zone (Fuso horário)
África	África/Cairo, África/Casablanca, África/Harare, África/Monrovia, África/Nairobi, África/Tripoli, África/Windhoek
América	América/Araguaina, América/Assunção, América/Bogotá, América/Buenos_Aires, América/Caracas, América/Chihuahua, América/Cuiabá, América/Denver, América/Fortaleza, América/Guatemala, América/Halifax, América/Manaus, América/Matamoros, América/Monterrey, América/Montevidéu, América/Phoenix, América/Santiago, América/Tijuana
Ásia	Ásia/Amman, Ásia/Ashgabat, Ásia/Baghdad, Ásia/Baku, Ásia/Bangkok, Ásia/Beirut, Ásia/Calcutta, Ásia/Damascus, Ásia/Dhaka, Ásia/Irkutsk, Ásia/Jerusalem, Ásia/Kabul, Ásia/Karachi, Ásia/Kathmandu, Ásia/Krasnoyarsk, Ásia/Magadan, Ásia/Muscat, Ásia/Novosibirsk, Ásia/Riyadh, Ásia/Seoul, Ásia/Shanghai, Ásia/Singapore, Ásia/Taipei, Ásia/Tehran, Ásia/Tokyo, Ásia/Ulaanbaatar, Ásia/Vladivostok, Ásia/Yakutsk, Ásia/Yerevan
Atlântico	Atlântico/Açores
Austrália	Austrália/Adelaide, Austrália/Brisbane, Austrália/Darwin, Austrália/Hobart, Austrália/Perth, Austrália/Sydney
Brasil	Brasil/DeNoronha, Brasil/Leste
Canadá	Canadá/Newfoundland, Canadá/Saskatchewan, Canadá/Yukon

Zona	Time zone (Fuso horário)
Europa	Europa/Amsterdam, Europa/Athens, Europa/Dublin, Europa/Helsinki, Europa/Istanbul, Europa/Kaliningrad, Europa/Moscow, Europa/Paris, Europa/Prague, Europa/Sarajevo
Pacífico	Pacífico/Auckland, Pacífico/Fiji, Pacífico/Guam, Pacífico/Honolulu, Pacífico/Samoa
EUA	EUA/Alasca, EUA/Central, EUA/East-Indiana, EUA/Eastern, EUA/Pacífico
UTC	UTC

Limitações e problemas conhecidos do RDS para MariaDB

Os itens a seguir referem-se a limitações e problemas conhecidos no uso do RDS para MariaDB.

> **Note**
>
> Essa lista não é exaustiva.

Tópicos

- Limites de tamanho de arquivo do MariaDB no Amazon RDS
- Palavra reservada InnoDB
- Portas personalizadas
- Insights de Performance

Limites de tamanho de arquivo do MariaDB no Amazon RDS

Para instâncias de banco de dados do MariaDB, o tamanho máximo de uma tabela é 16 TB ao usar tablespaces de arquivo por tabela do InnoDB. Esse limite também restringe o espaço de tabela do sistema a um tamanho máximo de 16 TB. Os espaços de tabelas de arquivo por tabela do InnoDB (com cada tabela em seu próprio espaço de tabela) são definidos por padrão para instâncias de bancos de dados MariaDB. Esse limite não está relacionado ao limite máximo de armazenamento para instâncias de banco de dados do MariaDB. Para obter mais informações sobre limites de armazenamento, consulte Armazenamento de instâncias de banco de dados do Amazon RDS.

Existem vantagens e desvantagens na utilização de espaços de tabela de arquivo por tabela do InnoDB, dependendo do seu aplicativo. Para determinar a melhor abordagem para a aplicação, consulte File-per-table tablespaces na documentação do MySQL.

Não recomendamos permitir que as tabelas cresçam até o tamanho máximo do arquivo. Em geral, uma prática recomendada é particionar dados em tabelas menores, o que pode melhorar a performance e os tempos de recuperação.

Uma opção que você pode usar para dividir uma tabela grande em tabelas menores é o particionamento. O particionamento distribui partes da sua tabela grande em arquivos separados com base em regras que você especifica. Por exemplo, se você armazenar transações por data, poderá criar regras de particionamento que distribuem transações antigas em arquivos separados

usando o particionamento. Em seguida, periodicamente, você pode arquivar os dados históricos de transações que não precisam estar prontamente disponíveis para o seu aplicativo. Para ter mais informações, consulte Partitioning na documentação do MySQL.

Como determinar o tamanho de todos os espaços de tabela do InnoDB

- Use o seguinte comando SQL para determinar se algumas das suas tabelas são muito grandes e são candidatas para particionamento.

> **Note**
>
> No MariaDB 10.6 e posterior, essa consulta também retorna o tamanho do espaço de tabela do sistema InnoDB.
> Em versões do MariaDB anteriores à 10.6, não é possível determinar o tamanho do espaço de tabela do sistema InnoDB consultando as tabelas do sistema. Recomendamos fazer upgrade para uma versão posterior.

```
SELECT SPACE,NAME,ROUND((ALLOCATED_SIZE/1024/1024/1024), 2)
as "Tablespace Size (GB)"
FROM information_schema.INNODB_SYS_TABLESPACES ORDER BY 3 DESC;
```

Como determinar o tamanho das tabelas de usuários que não são do InnoDB

- Use o seguinte comando SQL para determinar se algumas das suas tabelas de usuários que não são do InnoDB são muito grandes.

```
SELECT TABLE_SCHEMA, TABLE_NAME, round(((DATA_LENGTH + INDEX_LENGTH+DATA_FREE)
/ 1024 / 1024/ 1024), 2) As "Approximate size (GB)" FROM information_schema.TABLES
WHERE TABLE_SCHEMA NOT IN ('mysql', 'information_schema', 'performance_schema')
and ENGINE<>'InnoDB';
```

Para habilitar espaços de tabela de arquivo por tabela do InnoDB

- Defina o parâmetro `innodb_file_per_table` como 1 no grupo de parâmetros da instância de banco de dados.

Para desabilitar espaços de tabela de arquivo por tabela do InnoDB

- Defina o parâmetro `innodb_file_per_table` como `0` no grupo de parâmetros da instância de banco de dados.

Para obter informações sobre como atualizar um grupo de parâmetros, consulte Grupos de parâmetros para Amazon RDS.

Ao ativar ou desativar os espaços de tabela de arquivo por tabela do InnoDB, você pode emitir um comando `ALTER TABLE`. Você pode usar esse comando para mover uma tabela do espaço de tabela global para seu próprio espaço de tabela. Ou você pode mover uma tabela de seu próprio espaço de tabela para o espaço de tabela global. Veja um exemplo a seguir.

```
ALTER TABLE table_name ENGINE=InnoDB, ALGORITHM=COPY;
```

Palavra reservada InnoDB

`InnoDB` é uma palavra reservada ao RDS para MariaDB. Não é possível usar esse nome para um banco de dados do MariaDB.

Portas personalizadas

O Amazon RDS bloqueia conexões com a porta personalizada 33060 para o mecanismo MariaDB. Escolha uma porta diferente para o seu mecanismo MariaDB.

Insights de Performance

Os contadores do InnoDB não estão visíveis no Insights de Performance para o RDS para MariaDB versão 10.11 porque não há mais suporte para eles na comunidade do MariaDB.

Amazon RDS para Microsoft SQL Server

O Amazon RDS é compatível com as versões e as edições do Microsoft SQL Server. A tabela a seguir mostra a versão secundária compatível mais recente de cada versão principal. Para obter a lista completa das versões, edições e versões de mecanismo de RDS compatíveis, consulte Versões do Microsoft SQL Server no Amazon RDS.

Versão principal	Pacote de serviço/GDR	Atualização cumulativa	Versão secundária	Artigo da base de conhecimento	Data de lançamento
SQL Server 2022	GDR	CU13	16.0.4131.2	KB5040939	12 de agosto de 2024
SQL Server 2019	Não aplicável	CU27	15.0.4382.1	KB5040948	12 de agosto de 2024
SQL Server 2017	GDR	CU31	14.0.3471.2	KB5040940	12 de agosto de 2024
SQL Server 2016	SP3 GDR	Não aplicável	13.0.6441.1	KB5040946	12 de agosto de 2024

Para obter informações sobre licenciamento para o SQL Server, consulte Licenciar o Microsoft SQL Server no Amazon RDS. Consulte informações sobre as compilações do SQL Server neste artigo do suporte da Microsoft sobre Onde encontrar informações sobre os builds de SQL Server mais recentes.

Com o Amazon RDS, você pode criar instâncias de banco de dados e snapshot de banco de dadoss, além de restaurações point-in-time e backups automáticos e manuais. As instâncias de banco de dados que executam o SQL Server podem ser usadas dentro de uma VPC. Também é possível usar o Secure Sockets Layer (SSL) para conectar-se a uma instância de banco de dados executando o SQL Server e usar a criptografia de dados transparente (TDE) para criptografar dados em repouso. O Amazon RDS oferece suporte a implantações multi-AZ para o SQL Server usando o SQL Server

Database Mirroring (DBM) ou grupos de disponibilidade AlwaysOn (AGs) como uma solução de failover de alta disponibilidade.

Para oferecer uma experiência de serviço gerenciado, o Amazon RDS não fornece acesso ao shell para instâncias de bancos de dados e restringe o acesso a determinados procedimentos e tabelas do sistema que exigem privilégios avançados. O Amazon RDS oferece suporte a acesso a bancos de dados em uma instância de banco de dados que usa qualquer aplicação de cliente SQL padrão, como o Microsoft SQL Server Management Studio. O Amazon RDS não permite o acesso direto do host a uma instância de banco de dados pelo Telnet, Secure Shell (SSH) ou Windows Remote Desktop Connection. Quando você cria uma instância de banco de dados, o usuário principal é designado à função db_owner para todos os bancos de dados de usuários nessa instância e possui todas as permissões no nível do banco de dados, exceto as que são usadas para backups. O Amazon RDS gerencia backups para você.

Antes de criar sua primeira instância de banco de dados, você deve concluir as etapas na seção de configuração deste guia. Para obter mais informações, consulte Configuração do ambiente do Amazon RDS.

Tópicos

- Tarefas comuns de gerenciamento para o Microsoft SQL Server no Amazon RDS
- Limitações das instâncias de banco de dados Microsoft SQL Server
- Suporte a classes de instâncias de banco de dados para o Microsoft SQL Server
- Segurança do Microsoft SQL Server
- Suporte ao programa de conformidade de instâncias de banco de dados do Microsoft SQL Server
- Suporte do SSL a instâncias de banco de dados do Microsoft SQL Server
- Versões do Microsoft SQL Server no Amazon RDS
- Gerenciamento de versão no Amazon RDS
- Recursos do Microsoft SQL Server no Amazon RDS
- Compatibilidade entre captura de dados de alteração e instâncias de banco de dados do Microsoft SQL Server
- Recursos não compatíveis e recursos com suporte limitado
- Implantações multi-AZ usando o espelhamento de banco de dados ou os grupos de disponibilidade Always On do Microsoft SQL Server
- Uso da criptografia de dados transparente para criptografar dados em repouso
- Funções e procedimentos armazenados do Amazon RDS para Microsoft SQL Server

- Fuso horário local para instâncias de banco de dados do Microsoft SQL Server
- Licenciar o Microsoft SQL Server no Amazon RDS
- Conectar-se à instância de banco de dados do Microsoft SQL Server
- Trabalhar com o Active Directory com o RDS para SQL Server
- Atualizar aplicações para conexão com instâncias de banco de dados Microsoft SQL Server usando novos certificados SSL/TLS
- Atualizações do mecanismo de banco de dados do Microsoft SQL Server
- Importar e exportar bancos de dados do SQL Server usando backup e restauração nativos
- Trabalhar com réplicas de leitura do Microsoft SQL Server no Amazon RDS
- Implantações multi-AZ para o Amazon RDS for Microsoft SQL Server
- Recursos adicionais do Microsoft SQL Server no Amazon RDS
- Opções para o mecanismo de banco de dados do Microsoft SQL Server
- Tarefas comuns de DBA no Amazon RDS para Microsoft SQL Server

Tarefas comuns de gerenciamento para o Microsoft SQL Server no Amazon RDS

A seguir, você encontra tarefas de gerenciamento comuns que são executadas com uma instância de banco de dados Amazon RDS para SQL Server, com links para documentação relevante de cada tarefa.

Área de tarefa	Documentação relevante
Classes de instância, armazenamento e PIOPS Se você estiver criando uma instância de banco de dados para fins de produção, deve entender como classes de instâncias, tipos de armazenamento e IOPS provisionadas funcionam no Amazon RDS.	Suporte a classes de instâncias de banco de dados para o Microsoft SQL Server Tipos de armazenamento do Amazon RDS
Implantações multi-AZ Uma instância de banco de dados de produção deve usar implantações multi-AZ. As implantações multi-AZ oferecem	Configurar e gerenciar uma implantação multi-AZ para o Amazon RDS

Área de tarefa	Documentação relevante
maior disponibilidade, durabilidade de dados e tolerância a falhas para instâncias de banco de dados. As implantações multi-AZ para o SQL Server são feitas usando a tecnologia nativa DBM ou AGs do SQL Server.	Implantações multi-AZ usando o espelhamento de banco de dados ou os grupos de disponibilidade Always On do Microsoft SQL Server
Amazon Virtual Private Cloud (VPC) Se a sua conta AWS tiver uma VPC padrão, a instância de banco de dados será criada automaticamente dentro da VPC padrão. Se a sua conta não tiver uma VPC padrão e você quiser a instância de banco de dados em uma VPC, deverá criar os grupos de VPC e sub-redes antes de criar a instância de banco de dados.	Trabalhar com uma instância de banco de dados em uma VPC
Grupos de segurança Por padrão, as instâncias de banco de dados são criadas com um firewall que impede o acesso a elas. Você deve criar um grupo de segurança com os endereços IP corretos e uma configuração de rede para acessar a instância de banco de dados.	Controlar acesso com grupos de segurança
Grupos de parâmetros Se a sua instância de banco de dados exigir parâmetros de banco de dados específicos, você poderá criar um grupo de parâmetros antes de criar essa instância de banco de dados.	Grupos de parâmetros para Amazon RDS
Grupos de opções Se a sua instância de banco de dados exigir opções de banco de dados específicas, você deverá criar um grupo de opções antes de criar essa instância de banco de dados.	Opções para o mecanismo de banco de dados do Microsoft SQL Server

Área de tarefa	Documentação relevante
Conexão à instância de banco de dados Após criar um grupo de segurança e associá-lo a uma instância de banco de dados, você poderá se conectar à instância de banco de dados usando qualquer aplicativo cliente SQL padrão, como o Microsoft SQL Server Management Studio.	Conectar-se à instância de banco de dados do Microsoft SQL Server
Backup e restauração Ao criar sua instância de banco de dados, você pode configurá-la para receber backups automáticos. Você também pode fazer backup e restaurar seus bancos de dados manualmente usando arquivos de backup completos (arquivos .bak).	Introdução aos backups Importar e exportar bancos de dados do SQL Server usando backup e restauração nativos
Monitoramento Você pode monitorar sua instância de banco de dados do SQL Server usando as métricas, eventos e o monitoramento aprimorado do CloudWatch Amazon RDS.	Visualizar métricas no console do Amazon RDS Visualizar eventos do Amazon RDS
Arquivos de log Você pode acessar os arquivos de log de sua instância de banco de dados SQL Server.	Monitorar arquivos de log do Amazon RDS Arquivos de log do banco de dados do Amazon RDS para Microsoft SQL Server

Há também tarefas administrativas avançadas para trabalhar com instâncias de banco de dados do SQL Server. Para obter mais informações, consulte a documentação a seguir:

- Tarefas comuns de DBA no Amazon RDS para Microsoft SQL Server.
- Trabalhar com o AWS Managed Active Directory com o RDS para SQL Server
- Acessar o banco de dados tempdb

Limitações das instâncias de banco de dados Microsoft SQL Server

A implantação do Amazon RDS do Microsoft SQL Server em uma instância de banco de dados apresenta algumas limitações das quais você deve estar ciente:

- O número máximo de bancos de dados aceitos em uma instância de banco de dados depende do tipo de classe da instância e do modo de disponibilidade—single-AZ, DBM (Database Mirroring) multi-AZ ou Grupos de disponibilidade (AGs) multi-AZ. Os bancos de dados do sistema Microsoft SQL Server não contam para esse limite.

  A tabela a seguir mostra o número máximo de bancos de dados com suporte para cada tipo de classe de instância e modo de disponibilidade. Use essa tabela para ajudá-lo a decidir se você pode mover de um tipo de classe de instância para outro ou de um modo de disponibilidade para outro. Se sua instância de banco de dados de origem tiver mais bancos de dados do que o tipo de classe de instância de destino ou o modo de disponibilidade puder suportar, a modificação da instância de banco de dados falhará. Você pode ver o status de sua solicitação no painel Events (Eventos).

Tipo de classe de instância	single-AZ	multi-AZ com DBM	multi-AZ com AGs Always On
db.*.micro to db.*.medium	30	N/D	N/D
db.*.large	30	30	30
db.*.xlarge to db.*.16xlarge	100	50	75
db.*.24xlarge	100	50	100

  * Representa os diferentes tipos de classes da instância.

  Por exemplo, digamos que sua instância de banco de dados seja executada em db.*.16xlarge com single-AZ e que possua 76 bancos de dados. Você modifica a instância do banco de dados para atualizar para o uso de AGs Always On multi-AZ. Essa atualização falhará, pois sua instância de banco de dados contém mais bancos de dados do que a configuração de destino pode suportar.

Mas, se você atualizar seu tipo de classe de instância para db.*.24xlarge, a modificação será realizada.

Se a atualização falhar, você verá eventos e mensagens semelhantes aos seguintes:

- Não é possível modificar a classe da instância do banco de dados. A instância possui 76 bancos de dados, mas, após a conversão, ela oferece suporte a apenas 75.
- Não é possível converter a instância de banco de dados para multi-AZ: a instância possui 76 bancos de dados, mas, após a conversão, comportaria apenas 75.

Se a restauração pontual ou o snapshot falhar, você verá eventos e mensagens semelhantes aos seguintes:

- Instância de banco de dados colocada em restauração incompatível. A instância possui 76 bancos de dados, mas, após a conversão, ela oferece suporte a apenas 75.

- As seguintes portas são reservadas para o Amazon RDS, e não é possível usá-las ao criar uma instância de banco de dados: `1234, 1434, 3260, 3343, 3389, 47001,` e `49152-49156`.
- Conexões de cliente de endereços IP dentro do intervalo 169.254.0.0/16 não são permitidas. Esse é o APIPA (Automatic Private IP Addressing Range, Intervalo de endereçamento IP privado automático), usado para o endereçamento de link local.
- O SQL Server Standard Edition usa apenas um subconjunto dos processadores disponíveis, se a instância do banco de dados tiver mais processadores do que os limites de software (24 núcleos, 4 soquetes e 128 GB de RAM). Exemplos disso são as classes de instância db.m5.24xlarge e db.r5.24xlarge.

  Para obter mais informações, consulte a tabela de limites de escala em Edições e recursos compatíveis do SQL Server 2019 (15.x) na documentação da Microsoft.
- O Amazon RDS para SQL Server não oferece suporte para a importação de dados no banco de dados msdb.
- Não renomeie bancos de dados em uma instância de banco de dados em uma implantação multi-AZ do SQL Server.
- Use estas diretrizes ao definir os seguintes parâmetros de banco de dados em RDS for SQL Server:
  - `max server memory (mb)` >= 256 MB
  - `max worker threads` >= (número de CPUs lógicas* 7)

  Para obter mais informações sobre como definir parâmetros de banco de dados, consulte Grupos de parâmetros para Amazon RDS.

- O tamanho de armazenamento máximo para instâncias de banco de dados do SQL Server é o seguinte:
  - Armazenamento de uso geral (SSD) – 16 TiB para todas as edições
  - Armazenamento de IOPS provisionadas – 16 TiB para todas as edições
  - Armazenamento magnético – 1 TiB para todas as edições

  Se for preciso uma maior quantidade de armazenamento, você poderá usar fragmentos em múltiplas instâncias de banco de dados para contornar o limite. Essa abordagem exige uma lógica de roteamento dependente de dados em aplicativos que se conectam ao sistema fragmentado. Você pode usar um framework de fragmentação existente ou pode escrever um código personalizado para permitir a fragmentação. Se você usa um framework existente, o framework não pode instalar nenhum componente no mesmo servidor que a instância de banco de dados.
- O tamanho de armazenamento mínimo para instâncias de banco de dados do SQL Server é o seguinte:
  - Armazenamento de uso geral (SSD) – 20 GiB para edições Enterprise, Standard, Web e Express
  - Armazenamento IOPS provisionado – 20 GiB para edições Enterprise, Standard, Web e Express
  - Armazenamento magnético – 20 GiB para edições Enterprise, Standard, Web e Express
- O Amazon RDS não oferece suporte à execução desses serviços no mesmo servidor que sua instância de banco de dados do RDS:
  - Data Quality Services
  - Master Data Services

  Para usar esses recursos, recomendamos que você instale o SQL Server em uma instância do Amazon EC2 ou use uma instância do SQL Server no local. Nesses casos, a instância do EC2 ou do SQL Server atua como o servidor do Master Data Services para a instância de banco de dados do SQL Server no Amazon RDS. Você pode instalar o SQL Server em uma instância do Amazon EC2 com armazenamento do Amazon EBS, respeitando as políticas de licenciamento da Microsoft.
- Devido às limitações no Microsoft SQL Server, a restauração a determinado momento antes da execução bem-sucedida de `DROP DATABASE` pode não refletir o estado desse banco de dados nesse momento. Por exemplo, o banco de dados descartado normalmente é restaurado ao seu estado até cinco minutos antes do comando `DROP DATABASE` ter sido emitido. Esse tipo de restauração significa que não é possível restaurar as transações feitas durante esses poucos minutos em seu banco de dados descartado. Para contornar essa questão, é possível reeditar o

comando `DROP DATABASE` após a conclusão da operação de restauração. Descartar um banco de dados remove os logs de transações desse banco de dados.

- Para o SQL Server, você cria seus bancos de dados depois de criar sua instância de banco de dados. Os nomes de banco de dados seguem as regras de nomenclatura usuais do SQL Server com as diferenças a seguir.
  - Nomes de banco de dados não podem começar com `rdsadmin`.
  - Não podem começar ou terminar com um espaço ou uma tabulação.
  - Não podem conter nenhum dos caracteres que criam uma nova linha.
  - Não podem conter aspas simples (`'`).
  - Atualmente, o RDS para SQL Server não comporta atualizações automáticas de versões secundárias. Para obter mais informações, consulte Gerenciamento de versão no Amazon RDS.
- O SQL Server Web Edition só permite que você use o modelo Dev/Teste ao criar uma instância de banco de dados do RDS para SQL Server.

Suporte a classes de instâncias de banco de dados para o Microsoft SQL Server

A capacidade de computação e memória de uma instância de banco de dados é determinada de acordo com sua classe de instância de banco de dados. A classe de instância de banco de dados da qual você precisa depende dos requisitos de energia e memória de processamento. Para obter mais informações, consulte Classes de instâncias de banco de dados .

A lista a seguir das classes de instância de banco de dados compatíveis para o Microsoft SQL Server é fornecida aqui para sua comodidade. Para obter a lista mais atual, consulte o console do RDS:https://console.aws.amazon.com/rds/.

Nem todas as classes de instâncias de banco de dados estão disponíveis em todas as versões secundárias do SQL Server compatíveis. Por exemplo, algumas classes de instância de banco de dados mais recentes, como db.r6i, não estão disponíveis em versões secundárias mais antigas. Você pode usar o comando describe-orderable-db-instance-options da AWS CLI para descobrir quais classes de instância de banco de dados estão disponíveis para a sua edição e versão do SQL Server.

Edição do SQL Server	Faixa de suporte para 2022	Faixa de suporte para 2019	Faixa de suporte para 2017 e 2016
Edição Enterprise	db.t3.xlarge –db.t3.2xlarge	db.t3.xlarge –db.t3.2xlarge	db.t3.xlarge –db.t3.2xlarge
	db.r5.large –db.r5.24xlarge	db.r5.xlarge –db.r5.24xlarge	db.r5.xlarge –db.r5.24xlarge
	db.r5b.large –db.r5b.24xlarge	db.r5b.xlarge –db.r5b.24xlarge	db.r5b.xlarge –db.r5b.24xlarge
	db.r5d.large –db.r5d.24xlarge	db.r5d.xlarge –db.r5d.24xlarge	db.r5d.xlarge –db.r5d.24xlarge
	db.r6i.large –db.r6i.32xlarge	db.r6i.xlarge –db.r6i.32xlarge	db.r6i.xlarge –db.r6i.32xlarge
	db.m5.large –db.m5.24xlarge	db.m5.xlarge –db.m5.24xlarge	db.m5.xlarge –db.m5.24xlarge
	db.m5d.large –db.m5d.24xlarge	db.m5d.xlarge –db.m5d.24xlarge	db.m5d.xlarge –db.m5d.24xlarge
	db.m6i.large –db.m6i.32xlarge	db.m6i.xlarge –db.m6i.32xlarge	db.m6i.xlarge –db.m6i.32xlarge
	db.x2iedn.xlarge –db.x2iedn.32xlarge	db.x1.16xlarge –db.x1.32xlarge	db.x1.16xlarge –db.x1.32xlarge

Edição do SQL Server	Faixa de suporte para 2022	Faixa de suporte para 2019	Faixa de suporte para 2017 e 2016
	db.z1d.large –db.z1d.12xlarge	db.x1e.xlarge –db.x1e.32xlarge db.x2iedn.xlarge –db.x2iedn.32xlarge db.z1d.xlarge –db.z1d.12xlarge	db.x1e.xlarge –db.x1e.32xlarge db.x2iedn.xlarge –db.x2iedn.32xlarge db.z1d.xlarge –db.z1d.12xlarge

Edição do SQL Server	Faixa de suporte para 2022	Faixa de suporte para 2019	Faixa de suporte para 2017 e 2016
Edição Standard	db.t3.xlarge –db.t3.2xlarge db.r5.large –db.r5.24xlarge db.r5b.large –db.r5b.8xlarge db.r5d.large –db.r5d.24xlarge db.r6i.large –db.r6i.8xlarge db.m5.large –db.m5.24xlarge db.m5d.large –db.m5d.24xlarge db.m6i.large –db.m6i.8xlarge db.x2iedn.xlarge –db.x2iedn.8xlarge db.z1d.large –db.z1d.12xlarge	db.t3.xlarge –db.t3.2xlarge db.r5.large –db.r5.24xlarge db.r5b.large –db.r5b.24xlarge db.r5d.large –db.r5d.24xlarge db.r6i.large –db.r6i.8xlarge db.m5.large –db.m5.24xlarge db.m5d.large –db.m5d.24xlarge db.m6i.large –db.m6i.8xlarge db.x1.16xlarge –db.x1.32xlarge db.x1e.xlarge –db.x1e.32xlarge	db.t3.xlarge –db.t3.2xlarge db.r5.large –db.r5.24xlarge db.r5b.large –db.r5b.24xlarge db.r5d.large –db.r5d.24xlarge db.r6i.large –db.r6i.8xlarge db.m5.large –db.m5.24xlarge db.m5d.large –db.m5d.24xlarge db.m6i.large –db.m6i.8xlarge db.x1.16xlarge –db.x1.32xlarge db.x1e.xlarge –db.x1e.32xlarge

Edição do SQL Server	Faixa de suporte para 2022	Faixa de suporte para 2019	Faixa de suporte para 2017 e 2016
		db.x2iedn .xlarge –db.x2iedn .32xlarge db.z1d.la rge –db.z1d.12 xlarge	db.x2iedn .xlarge –db.x2iedn .32xlarge db.z1d.la rge –db.z1d.12 xlarge
Edição Web	db.t3.sma ll –db.t3.xlarge db.r5.lar ge –db.r5.4xlarge db.r5b.la rge –db.r5b.4xlarge db.r5d.la rge –db.r5d.4xlarge db.r6i.la rge –db.r6i.4xlarge db.m5.lar ge –db.m5.4xlarge db.m5d.la rge –db.m5d.4xlarge db.m6i.la rge –db.m6i.4xlarge db.z1d.la rge –db.z1d.13 xlarge	db.t3.sma ll –db.t3.2xlarge db.r5.lar ge –db.r5.4xlarge db.r5b.la rge –db.r5b.4xlarge db.r5d.la rge –db.r5d.4xlarge db.r6i.la rge –db.r6i.4xlarge db.m5.lar ge –db.m5.4xlarge db.m5d.la rge –db.m5d.4xlarge db.m6i.la rge –db.m6i.4xlarge db.z1d.la rge –db.z1d.3xlarge	db.t3.sma ll –db.t3.2xlarge db.r5.lar ge –db.r5.4xlarge db.r5b.la rge –db.r5b.4xlarge db.r5d.la rge –db.r5d.4xlarge db.r6i.la rge –db.r6i.4xlarge db.m5.lar ge –db.m5.4xlarge db.m5d.la rge –db.m5d.4xlarge db.m6i.la rge –db.m6i.4xlarge db.z1d.la rge –db.z1d.3xlarge

Edição do SQL Server	Faixa de suporte para 2022	Faixa de suporte para 2019	Faixa de suporte para 2017 e 2016
Edição Express	db.t3.mic ro –db.t3.xlarge	db.t3.mic ro –db.t3.xlarge	db.t3.mic ro –db.t3.xlarge

Segurança do Microsoft SQL Server

O mecanismo de banco de dados do Microsoft SQL Server usa segurança baseada na função. O nome de usuário mestre que você usa ao criar uma instância de banco de dados é um login de autenticação do SQL Server que é um membro das funções fixas de servidor `processadmin`, `public` e `setupadmin`.

Qualquer usuário que crie um banco de dados é atribuído à função db_owner desse banco de dados e tem todas as permissões no nível do banco de dados, exceto para aquelas usadas para backups. O Amazon RDS gerencia backups para você.

As seguintes funções no nível do servidor não estão disponíveis no Amazon RDS para SQL Server:

- bulkadmin
- dbcreator
- diskadmin
- securityadmin
- serveradmin
- sysadmin

As seguintes permissões no nível do servidor não estão disponíveis em instâncias de banco de dados do RDS for SQL Server:

- ALTERAR QUALQUER BANCO DE DADOS
- ALTER ANY EVENT NOTIFICATION
- ALTER RESOURCES

- ALTER SETTINGS (você pode usar as operações de API do grupo de parâmetros de banco de dados para modificar os parâmetro; para obter mais informações, consulte Grupos de parâmetros para Amazon RDS)
- AUTHENTICATE SERVER
- CONTROL_SERVER
- CREATE DDL EVENT NOTIFICATION
- CREATE ENDPOINT
- CREATE SERVER ROLE
- CREATE TRACE EVENT NOTIFICATION
- DESCARTAR QUALQUER BANCO DE DADOS
- EXTERNAL ACCESS ASSEMBLY
- SHUTDOWN (Você pode usar a opção de reinicialização do RDS em vez disso)
- UNSAFE ASSEMBLY
- ALTERAR QUALQUER GRUPO DE DISPONIBILIDADE
- CRIAR QUALQUER GRUPO DE DISPONIBILIDADE

Suporte ao programa de conformidade de instâncias de banco de dados do Microsoft SQL Server

AWSOs serviços da no escopo foram totalmente avaliados por um auditor terceirizado e resultam em uma certificação, declaração de conformidade ou Authority to Operate (ATO, Autoridade para operar). Para obter mais informações, consulte Serviços da AWS no escopo por programa de conformidade.

Suporte HIPAA para instâncias de banco de dados para o Microsoft SQL Server

Você pode usar bancos de dados do Amazon RDS para Microsoft SQL Server para criar aplicativos em conformidade com a norma HIPAA. É possível armazenar informações relacionadas à saúde, inclusive informações de saúde protegidas (PHI), sob um Acordo de associado comercial (BAA) com AWS. Para obter mais informações, consulte Conformidade com a HIPAA.

O Amazon RDS para SQL Server oferece suporte a HIPAA para as seguintes versões e edições:

- SQL Server 2.022 edições Enterprise, Standard e Web
- SQL Server 2019 edições Enterprise, Standard e Web
- SQL Server 2017 edições Enterprise, Standard e Web
- SQL Server 2016 edições Enterprise, Standard e Web

Para permitir suporte a HIPAA na instância de banco de dados, configure os três componentes a seguir.

Componente	Detalhes
Auditoria	Para configurar a auditoria, defina o parâmetro `rds.sqlserver_audit` como o valor `fedramp_hipaa`. Se a instância de banco de dados não estiver usando um grupo de parâmetros de banco de dados personalizado, você deverá criar um grupo de parâmetros personalizado e anexá-lo à instância de banco de dados para modificar o parâmetro `rds.sqlserver_audit`. Para obter mais informações, consulte Grupos de parâmetros para Amazon RDS.
Criptografia de transporte	Para configurar a criptografia de transporte, force todas as conexões para a instância de banco de dados para usar Secure Sockets Layer (SSL). Para obter mais informações, consulte ???.
Criptografia em repouso	Para configurar a criptografia em repouso, você tem duas opções: 1. Se você estiver executando o SQL Server 2016–2022 Enterprise Edition ou 2022 Standard Edition, poderá usar Transparent Data Encryption (TDE) para ter criptografia em repouso. Para obter mais informações, consulte Suporte para criptografia de dados transparente no SQL Server. 2. Você pode configurar a criptografia em repouso usando as chaves de criptografia do AWS Key Management Service (AWS KMS). Para obter mais informações, consulte Criptografar recursos do Amazon RDS.

Suporte do SSL a instâncias de banco de dados do Microsoft SQL Server

Você pode usar SSL para criptografar conexões entre seus aplicativos e suas instâncias de banco de dados do Amazon RDS executando o Microsoft SQL Server. Você também pode forçar todas as conexões à instância de banco de dados a usarem SSL. Se você forçar as conexões a usarem o SSL, isso acontece de forma transparente para o cliente, e o cliente não precisa fazer nada para usar o SSL.

O SSL tem suporte em todas as regiões AWS e em todas as edições compatíveis do SQL Server. Para obter mais informações, consulte ???.

Versões do Microsoft SQL Server no Amazon RDS

Você pode especificar qualquer versão atualmente com suporte do Microsoft SQL Server ao criar uma nova instância de banco de dados. Você pode especificar a versão principal do Microsoft SQL Server (como o Microsoft SQL Server 14.00) e qualquer versão secundária com suporte para a versão principal especificada. Se nenhuma versão for especificada, o Amazon RDS usará uma versão compatível por padrão, geralmente a versão mais recente. Se uma versão principal for especificada, mas uma versão secundária não, o Amazon RDS usará por padrão um lançamento recente da versão principal que você especificou.

A tabela a seguir mostra as versões compatíveis para todas as edições e todas as regiões AWS, exceto onde indicado. Também é possível usar o comando `describe-db-engine-versions` da AWS CLI para ver uma lista de versões compatíveis, bem como os padrões para as instâncias de bancos de dados recém-criadas. A tabela a seguir mostra as versões do SQL Server compatíveis com o RDS:

Versão principal	Versão secundária	API do RDS **`EngineVersion`** e CLI **`engine-version`**
SQL Server 2022	16.00.4131.2 (CU13)	`16.00.4131.2.v1`
	16.00.4125.3 (CU13)	`16.00.4125.3.v1`
	16.00.4120.1 (CU12 GDR)	`16.00.4120.1.v1`
	16.00.4115.5 (CU12)	`16.00.4115.5.v1`

Versão principal	Versão secundária	API do RDS **EngineVersion** e CLI **engine-version**
	16.00.4105.2 (CU11)	16.00.4105.2.v1
	16.00.4095.4 (CU10)	16.00.4095.4.v1
	16.00.4085.2 (CU9)	16.00.4085.2.v1
SQL Server 2019	15.00.4382.1 (CU27)	15.00.4382.1.v1
	15.00.4375.4 (CU27)	15.00.4375.4.v1
	15.00.4365.2 (CU26)	15.00.4365.2.v1
	15.00.4355.3 (CU25)	15.00.4355.3.v1
	15.00.4345.5 (CU24)	15.00.4345.5.v1
	15.00.4335.1 (CU23)	15.00.4335.1.v1
	15.00.4322.2 (CU22)	15.00.4322.2.v1
	15.00.4316.3 (CU21)	15.00.4316.3.v1
	15.00.4312.2 (CU20)	15.00.4312.2.v1
	15.00.4236.7 (CU16)	15.00.4236.7.v1
	15.00.4198.2 (CU15)	15.00.4198.2.v1
	15.00.4153.1 (CU12)	15.00.4153.1.v1
	15.00.4073.23 (CU8)	15.00.4073.23.v1
	15.00.4043.16 (CU5)	15.00.4043.16.v1

Versão principal	Versão secundária	API do RDS **EngineVersion** e CLI **engine-version**
SQL Server 2017	14.00.3471.2 (CU31)	14.00.3471.2.v1
	14.00.3465.1 (CU31)	14.00.3465.1.v1
	14.00.3460.9 (CU31)	14.00.3460.9.v1
	14.00.3451.2 (CU30)	14.00.3451.2.v1
	14.00.3421.10 (CU27)	14.00.3421.10.v1
	14.00.3401.7 (CU25)	14.00.3401.7.v1
	14.00.3381.3 (CU23)	14.00.3381.3.v1
	14.00.3356.20 (22 CU22)	14.00.3356.20.v1
	14.00.3294.2 (CU20)	14.00.3294.2.v1
	14.00.3281.6 (CU19)	14.00.3281.6.v1
SQL Server 2016	13.00.6441.1 (GDR)	13.00.6441.1.v1
	13.00.6435.1 (GDR)	13.00.6435.1.v1
	13.00.6430.49 (GDR)	13.00.6430.49.v1
	13.00.6419.1 (SP3 + Hotfix)	13.00.6419.1.v1
	13.00.6300.2 (SP3)	13.00.6300.2.v1

Gerenciamento de versão no Amazon RDS

O Amazon RDS inclui gerenciamento de versão flexível que permite controlar quando e como sua instância de banco de dados é corrigida ou atualizada. Isso permite que você faça o seguinte para o seu mecanismo de banco de dados:

- Mantenha a compatibilidade com versões de patch do mecanismo de banco de dados

- Teste novas versões de patches para verificar se elas funcionam com o aplicativo antes de implantá-las na produção
- Planeje e realize atualizações de versão para atender aos contratos de nível de serviço e requisitos de tempo

Aplicação de patch de mecanismos do Microsoft SQL Server no Amazon RDS

O Amazon RDS agrega periodicamente patches oficiais do banco de dados do Microsoft SQL Server em uma versão do mecanismo de instância de banco de dados específica para o Amazon RDS. Para obter mais informações sobre os patches do Microsoft SQL Server em cada versão de mecanismo, consulte Suporte a versões e recursos no Amazon RDS.

No momento, você executa manualmente todas as atualizações de mecanismo em sua instância de banco de dados. Para obter mais informações, consulte Atualizações do mecanismo de banco de dados do Microsoft SQL Server.

Programação de descontinuação para versões principais do mecanismo do Microsoft SQL Server no Amazon RDS

A tabela a seguir exibe a programação planejada de reprovações para as principais versões do mecanismo do Microsoft SQL Server.

Data	Informações
9 de julho de 2024	A Microsoft interromperá as atualizações de patch críticas para o SQL Server 2014. I informações, consulte a página Microsoft SQL Server 2014 na documentação da Mic
1.º de junho de 2024	O Amazon RDS planeja encerrar o suporte ao Microsoft SQL Server 2014 no RDS pa ocasião, todas as instâncias restantes serão agendadas para migrar para o SQL Ser secundária mais recente disponível). Para obter mais informações, consulte Anúncio Amazon RDS para SQL Server às versões principais do SQL Server 2014. Para evitar upgrade automático para o Microsoft SQL Server 2014, você poderá faze que for mais conveniente. Para obter mais informações, consulte Atualizar a versão d instância de banco de dados.

Data	Informações
12 de julho de 2022	A Microsoft interromperá as atualizações de patch críticas para o SQL Server 2012. I informações, consulte a página Microsoft SQL Server 2012 na documentação da Mic
1º de junho de 2022	O Amazon RDS planeja encerrar o suporte ao Microsoft SQL Server 2012 no RDS pa ocasião, todas as instâncias restantes serão agendadas para migrar para o SQL Ser secundária mais recente disponível). Para obter mais informações, consulte Anúncio Amazon RDS para SQL Server às versões principais do SQL Server 2012. Para evitar uma atualização automática para o Microsoft SQL Server 2012, você pod momento que for mais conveniente. Para obter mais informações, consulteAtualizar a de uma instância de banco de dados
1º de setembro de 2021	O Amazon RDS está começando a desabilitar a criação de instâncias de banco de da SQL Server usando o Microsoft SQL Server 2012. Para obter mais informações, cons suporte do Amazon RDS para SQL Server às versões principais do SQL Server 2012
12 de julho de 2019	A equipe do Amazon RDS substituiu o suporte do Microsoft SQL Server 2008 R2 em instâncias restantes do Microsoft SQL Server 2008 R2 estão migrando para o SQL S secundária mais recente disponível). Para evitar uma atualização automática para o Microsoft SQL Server 2008 R2, você no momento que for mais conveniente. Para obter mais informações, consulte Atualiz mecanismo de uma instância de banco de dados.
25 de abril de 2019	Antes do final de abril de 2019, não será mais possível criar novas instâncias de ban Amazon RDS para SQL Server usando o Microsoft SQL Server 2008R2.

Recursos do Microsoft SQL Server no Amazon RDS

As versões compatíveis do SQL Server no Amazon RDS incluem os recursos a seguir. Em geral, uma versão também inclui recursos das versões anteriores, a menos que indicado de outra forma na documentação da Microsoft.

Tópicos

- Recursos do Microsoft SQL Server 2022
- Recursos do Microsoft SQL Server 2019

- Recursos do Microsoft SQL Server 2017
- Recursos do Microsoft SQL Server 2016
- Fim do suporte ao Microsoft SQL Server 2014 no Amazon RDS
- Fim do suporte ao Microsoft SQL Server 2012 no Amazon RDS
- Fim do suporte ao Microsoft SQL Server 2008 R2 no Amazon RDS

Recursos do Microsoft SQL Server 2022

O SQL Server 2022 inclui muitos recursos novos, como os seguintes:

- Otimização de planos sensíveis a parâmetros: permite vários planos em cache para uma única declaração parametrizada, reduzindo possivelmente os problemas com a detecção de parâmetros.
- SQL Server Ledger: oferece a capacidade de provar criptograficamente que os dados não foram alterados sem autorização.
- Inicialização instantânea de arquivos para eventos de crescimento de arquivos de log de transações: ocasiona a execução mais rápida de eventos de crescimento de log de até 64 MB, inclusive para bancos de dados com TDE habilitado.
- Aprimoramentos na simultaneidade do bloqueio de página do sistema: reduz a contenção por travamento de página ao mesmo tempo em que aloca e desaloca páginas e extensões de dados, proporcionando melhorias significativas de performance para workloads pesadas `tempdb`.

Para ter a lista completa de recursos do SQL Server 2022, consulte What's new in SQL Server 2022 (16.x) na documentação da Microsoft.

Para ver uma lista de recursos não compatíveis, consulte Recursos não compatíveis e recursos com suporte limitado.

Recursos do Microsoft SQL Server 2019

O SQL Server 2019 inclui muitos recursos novos, como os seguintes:

- Recuperação acelerada de banco de dados (ADR) – reduz o tempo de recuperação de falhas após uma reinicialização ou uma reversão de transação de longa execução.
- Processamento de consultas inteligente (IQP):
  - Feedback de concessão de memória no modo de linha – corrige automaticamente o excesso de concessões que, de outra forma, resultaria em perda de memória e simultaneidade reduzida.

  - Modo de lote no rowstore – permite a execução do modo de lote para workloads analíticas sem a necessidade de índices columnstore.
  - Compilação adiada de variável de tabela – melhora a qualidade do plano e a performance geral para consultas que fazem referência a variáveis de tabela.
- Performance inteligente:
  - `OPTIMIZE_FOR_SEQUENTIAL_KEY`Opção de índice – melhora a taxa de transferência para inserções de alta simultaneidade em índices.
  - Maior escalabilidade de pontos de verificação indiretos – ajuda os bancos de dados com workloads DML pesadas.
  - Atualizações simultâneas de espaço livre de página (PFS) – permite o manuseio como uma trava compartilhada em vez de uma trinca exclusiva.
- Monitorar melhorias:
  - `WAIT_ON_SYNC_STATISTICS_REFRESH`Tipo de espera – mostra o tempo acumulado no nível da instância gasto em operações de atualização de estatísticas síncronas.
  - Configurações com escopo de banco de dados – Incluem `LIGHTWEIGHT_QUERY_PROFILING` e `LAST_QUERY_PLAN_STATS`.
  - Funções de gerenciamento dinâmicas (DMFs) – incluem `sys.dm_exec_query_plan_stats` e `sys.dm_db_page_info`.
- Avisos de truncamento detalhados – a mensagem de erro de truncamento de dados usa como padrão a inclusão de nomes de tabela e de coluna e o valor truncado.
- Criação de índice on-line retomável – no SQL Server 2017, há suporte somente à recriação de índice on-line retomável.

Para obter a lista completa de recursos do SQL Server 2019, consulte Novidades no SQL Server 2019 (15.x) na documentação da Microsoft.

Para ver uma lista de recursos não compatíveis, consulte Recursos não compatíveis e recursos com suporte limitado.

Recursos do Microsoft SQL Server 2017

O SQL Server 2017 inclui muitos recursos novos, como os seguintes:

- Processamento de consulta adaptável
- Correção automática do plano (um recurso de ajuste automático)

- GraphDB
- Recompilações de índice resumível

Para obter a lista completa de recursos do SQL Server 2017, consulte Novidades no SQL Server 2017 na documentação da Microsoft.

Para ver uma lista de recursos não compatíveis, consulte Recursos não compatíveis e recursos com suporte limitado.

Recursos do Microsoft SQL Server 2016

O Amazon RDS é compatível com os seguintes recursos do SQL Server 2016:

- Sempre criptografado
- Suporte para JSON
- Análise operacional
- Armazenamento de consultas
- Tabelas temporais

Para obter a lista completa de recursos do SQL Server 2016, consulte Novidades no SQL Server 2016 na documentação da Microsoft.

Fim do suporte ao Microsoft SQL Server 2014 no Amazon RDS

Não há mais suporte ao SQL Server 2014 no Amazon RDS.

O RDS está atualizando todas as instâncias de banco de dados existentes que ainda estão usando o SQL Server 2014 para a versão secundária mais recente do SQL Server 2016. Para obter mais informações, consulte Gerenciamento de versão no Amazon RDS.

Fim do suporte ao Microsoft SQL Server 2012 no Amazon RDS

Não há mais suporte ao SQL Server 2012 no Amazon RDS.

O RDS está atualizando todas as instâncias de banco de dados existentes que ainda estão usando o SQL Server 2012 para a versão secundária mais recente do SQL Server 2016. Para obter mais informações, consulte Gerenciamento de versão no Amazon RDS.

Fim do suporte ao Microsoft SQL Server 2008 R2 no Amazon RDS

Não há mais suporte ao SQL Server 2008 R2 no Amazon RDS.

O RDS está atualizando todas as instâncias de banco de dados existentes que ainda estão usando o SQL Server 2008 R2 para a versão secundária mais recente do SQL Server 2012. Para obter mais informações, consulte Gerenciamento de versão no Amazon RDS.

Compatibilidade entre captura de dados de alteração e instâncias de banco de dados do Microsoft SQL Server

O Amazon RDS comporta captura de dados de alteração (CDC) em instâncias de banco de dados que estejam executando o Microsoft SQL Server. A CDC captura alterações feitas nos dados de suas tabelas e armazena metadados sobre cada alteração que podem ser acessados posteriormente. Para obter mais informações, consulte Captura de dados de alteração na documentação da Microsoft.

O Amazon RDS comporta captura de dados de alteração (CDC) nas seguintes edições e versões do SQL Server:

- Microsoft SQL Server Enterprise Edition (Todas as versões)
- Microsoft SQL Server Standard Edition:
  - 2022
  - 2019
  - 2017
  - 2016 versão 13.00.4422.0 SP1 CU2 e posterior

Para usar a CDC com instâncias de banco de dados do Amazon RDS, primeiro ative ou desative a CDC no nível de banco de dados usando os procedimentos armazenados fornecidos pelo RDS. Depois disso, qualquer usuário com a função `db_owner` para esse banco de dados pode usar os procedimentos armazenados nativos da Microsoft para controlar a CDC nesse banco de dados. Para obter mais informações, consulte Usar a captura de dados de alteração para o Amazon RDS para SQL Server.

Você pode usar a CDC e o AWS Database Migration Service para ativar a replicação contínua de instâncias de banco de dados do SQL Server.

Recursos não compatíveis e recursos com suporte limitado

Os recursos do Microsoft SQL Server a seguir não são compatíveis com o Amazon RDS:

- Fazer backup no armazenamento de Blobs do Microsoft Azure
- Extensão do grupo de buffer
- Gerenciar políticas de senha
- Data Quality Services
- Envio de logs de banco de dados
- Snapshots de banco de dados (o Amazon RDS aceita somente snapshots de instância de banco de dados)
- Procedimentos armazenados estendidos, incluindo xp_cmdshell
- Suporte a FILESTREAM
- Tabelas de arquivos
- Serviços de Machine Learning e R (requer acesso ao SO para instalar)
- Planos de manutenção
- Coletor de dados de performance
- Gerenciamento baseado em políticas
- PolyBase
- Replicação
- Regulador de recursos
- Triggers no nível do servidor
- Endpoints do Service Broker
- Stretch database
- Propriedade de banco de dados TRUSTWORTHY (requer a função sysadmin)
- Endpoints T-SQL (todas as operações que usam CREATE ENDPOINT não estão disponíveis)
- WCF Data Services

Os seguintes recursos do Microsoft SQL Server têm suporte limitado no Amazon RDS:

- Consultas distribuídas/servidores vinculados. Para obter mais informações, consulte: Implement linked servers with Amazon RDS para Microsoft SQL Server (Implementar servidores vinculados com o Amazon RDS para Microsoft SQL Server).

- Tempo de execução de linguagem comum (CLR). No RDS for SQL Server 2016 e versões inferiores, há suporte para CLR no modo SAFE e usando apenas bits de montagem. Não há suporte para CLR no RDS for SQL Server 2017 e versões posteriores. Para obter mais informações, consulte Integração de tempo de execução de linguagem comum na documentação da Microsoft.
- Servidores vinculados ao Oracle OLEDB no Amazon RDS para SQL Server. Para obter mais informações, consulte Suporte para servidores vinculados ao Oracle OLEDB no Amazon RDS para SQL Server.

Os seguintes recursos não são aceitos no Amazon RDS com o SQL Server 2022:

- Suspender banco de dados para snapshot
- Fonte de dados externa
- Backup e restauração para armazenamento de objetos compatível com o S3
- Integração com armazenamento de objetos
- TLS 1.3 e MS-TDS 8.0
- Descarregamento de compactação de backup com QAT
- SQL Server Analysis Services (SSAS)
- Espelhamento de banco de dados com implantações multi-AZ. O SQL Server Always On é o único método compatível com implantações multi-AZ.

Implantações multi-AZ usando o espelhamento de banco de dados ou os grupos de disponibilidade Always On do Microsoft SQL Server

O Amazon RDS oferece suporte a implantações multi-AZ para instâncias de bancos de dados que executem o Microsoft SQL Server usando o SQL Server Database Mirroring (DBM) ou Grupos de disponibilidade AlwaysOn. As implantações multi-AZ oferecem maior disponibilidade, durabilidade de dados e tolerância a falhas para instâncias de banco de dados. No caso de uma manutenção planejada do banco de dados ou de uma interrupção do serviço não planejada, o Amazon RDS faz failover automático para a réplica secundária atualizada. Dessa maneira, as operações do banco de dados podem ser retomadas rapidamente sem intervenção manual. As instâncias primária e secundária usam o mesmo endpoint, cujo endereço de rede física faz a transição para a réplica

secundária passiva como parte do processo de failover. Não é necessário reconfigurar seu aplicativo quando ocorre um failover.

O Amazon RDS gerencia o failover monitorando ativamente sua implantação multi-AZ e iniciando um failover quando ocorre um problema com o primário. O failover não ocorrerá, a menos que o modo em espera e o primário estejam totalmente sincronizados. O Amazon RDS mantém ativamente a implantação multi-AZ reparando automaticamente instâncias de banco de dados não íntegras e restabelecendo a replicação síncrona. Você não precisa gerenciar tudo. O Amazon RDS processa as instâncias primária, testemunha e em espera para você. Quando você configura o SQL Server multi-AZ, o RDS define instâncias secundárias passivas para todos os bancos de dados na instância.

Para obter mais informações, consulte Implantações multi-AZ para o Amazon RDS for Microsoft SQL Server.

Uso da criptografia de dados transparente para criptografar dados em repouso

O Amazon RDS oferece suporte a criptografia de dados transparente (TDE) do Microsoft SQL Server, que codifica de forma transparente os dados armazenados. O Amazon RDS usa grupos de opções para habilitar e configurar esses recursos. Para mais informações sobre as opções da TDE, consulte Suporte para criptografia de dados transparente no SQL Server.

Funções e procedimentos armazenados do Amazon RDS para Microsoft SQL Server

A seguir, há uma lista das funções e dos procedimentos armazenados do Amazon RDS que ajudam a automatizar tarefas do SQL Server.

Tipo de tarefa	Procedimento ou função	Onde é usado
Tarefas administrativas	`rds_drop_database`	Eliminar um banco de dados em uma instância de banco de dados do Amazon RDS para Microsoft SQL Server

Tipo de tarefa	Procedimento ou função	Onde é usado
	`rds_failover_time`	Determinar o horário do último failover do Amazon RDS para SQL Server
	`rds_modify_db_name`	Renomear um banco de dados do Amazon RDS para Microsoft SQL Server em uma implantação multi-AZ
	`rds_read_error_log`	Visualizar logs do agente ou erros
	`rds_set_configuration`	Essa operação é usada para definir várias configurações de instância de banco de dados: • Change Data Capture para instâncias multi-AZ • Como definir o período de retenção para arquivos de rastreamento e despejo • Compactar arquivos de backup
	`rds_set_database_online`	Transição de um banco de dados do Amazon RDS para SQL Server de OFFLINE para ONLINE

Tipo de tarefa	Procedimento ou função	Onde é usado
	`rds_set_system_database_sync_objects`	Ativar a replicação de trabalhos do SQL Server Agent
	`rds_fn_get_system_database_sync_objects`	
	`rds_fn_server_object_last_sync_time`	
	`rds_show_configuration`	Para ver os valores que são definidos usando `rds_set_configuration` , consulte estes tópicos: • Change Data Capture para instâncias multi-AZ • Como definir o período de retenção para arquivos de rastreamento e despejo
	`rds_shrink_tempdbfile`	Recolher o banco de dados tempdb
Captura de dados de alteração (CDC)	`rds_cdc_disable_db`	Desabilitar CDC
	`rds_cdc_enable_db`	Habilitar o CDC

Tipo de tarefa	Procedimento ou função	Onde é usado
Correspondência de banco de dados	`rds_fn_sysmail_allitems`	Visualizar mensagens, logs e anexos
	`rds_fn_sysmail_event_log`	Visualizar mensagens, logs e anexos
	`rds_fn_sysmail_mailattachments`	Visualizar mensagens, logs e anexos
	`rds_sysmail_control`	Essa operação é usada para iniciar e interromper a fila de email: • Iniciar a fila de e-mails • Parar a fila de e-mails
	`rds_sysmail_delete_mailitems_sp`	Excluir mensagens
Backup e restauração nativos	`rds_backup_database`	Realização de backup de um banco de dados
	`rds_cancel_task`	Cancelar uma tarefa
	`rds_finish_restore`	Concluir uma restauração de banco de dados

Tipo de tarefa	Procedimento ou função	Onde é usado
	`rds_restore_database`	Restaurar um banco de dados
	`rds_restore_log`	Restaurar um log
Transferência de arquivos do Amazon S3	`rds_delete_from_filesystem`	Excluir arquivos na instância de banco de dados do RDS
	`rds_download_from_s3`	Baixar arquivos de um bucket do Amazon S3 em uma instância de banco de dados SQL Server
	`rds_gather_file_details`	Listar arquivos na instância de banco de dados do RDS
	`rds_upload_to_s3`	Fazer upload de arquivos de uma instância de banco de dados SQL Server para um bucket do Amazon S3
Microsoft Distributed Transaction Coordinator (MSDTC)	`rds_msdtc_transaction_tracing`	Usar rastreamento de transações
Auditoria do SQL Server	`rds_fn_get_audit_file`	Visualizar logs de auditoria

Tipo de tarefa	Procedimento ou função	Onde é usado

Transparent Data Encryption	`rds_backup_tde_certificate`
	
	`rds_drop_tde_certificate`
	
	`rds_restore_tde_certificate`
	
	`rds_fn_list_user_tde_certificates`
		Suporte para criptografia de dados transparente no SQL Server

Tipo de tarefa	Procedimento ou função	Onde é usado
Microsoft Business Intelligence (MSBI)	`rds_msbi_task`	Essa operação é usada com o SQL Server Analysis Services (SSAS): • Implantar projetos do SSAS no Amazon RDS • Adicionar um usuário de domínio como administrador de banco de dados • Fazer backup de um banco de dados do SSAS • Restaurar um banco de dados do SSAS Essa operação também é usada com o SQL Server Integration Services (SSIS): • Permissões administrativas no SSISDB • Implantar um projeto do SSIS Essa operação também é usada com o SQL Server Reporting Services (SSRS): • Conceder acesso a usuários de domínio • Revogar permissões no nível do sistema
	`rds_fn_task_status`	Essa operação mostra o status das tarefas do MSBI: • SSAS: Monitorar o status de uma tarefa de implantação • SSIS: Monitorar o status de uma tarefa de implantação • SSRS: Monitorar o status de uma tarefa
SSIS	`rds_drop_ssis_database`	Eliminar o banco de dados SSISDB

Tipo de tarefa	Procedimento ou função	Onde é usado
	`rds_sqlagent_proxy`	Criar um proxy do SSIS
SSRS	`rds_drop_ssrs_databases`	Excluir os bancos de dados do SSRS

Fuso horário local para instâncias de banco de dados do Microsoft SQL Server

O fuso horário de uma instância de banco de dados do Amazon RDS que executa o Microsoft SQL Server é definida por padrão. O padrão atual é Universal Coordinated Time (UTC). Você pode definir o fuso horário de sua instância de banco de dados como um fuso horário local, para combinar o fuso horário de seus aplicativos.

Você define o fuso horário ao criar sua instância de banco de dados. É possível criar a instância de banco de dados usando o AWS Management Console, a ação CreateDBInstance da API do Amazon RDS ou o comando create-db-instance da AWS CLI.

Se a sua instância de banco de dados fizer parte de uma implantação multi-AZ (usando o recurso DBM ou AG do SQL Server), durante o failover, seu fuso horário permanecerá como o fuso horário local que você configurou. Para obter mais informações, consulte Implantações multi-AZ usando o espelhamento de banco de dados ou os grupos de disponibilidade Always On do Microsoft SQL Server.

Ao solicitar uma restauração point-in-time, especifique a hora da restauração. A hora é mostrada no fuso horário local. Para obter mais informações, consulte Restaurar uma instância de banco de dados para um momento especificado no Amazon RDS.

As limitações a seguir se aplicam ao definir o fuso horário local em sua instância de banco de dados:

- Não é possível modificar o fuso horário de uma instância de banco de dados existente do SQL Server.

- Não é possível restaurar um snapshot de uma instância de banco de dados em um fuso horário para uma instância de banco de dados em um fuso horário diferente.
- Recomendamos que não restaure um arquivo de backup de um fuso horário em um fuso horário diferente. Se você restaurar um arquivo de backup a partir de um fuso horário para um fuso horário diferente, é preciso auditar as consultas e aplicativos quanto à alteração de fuso horário. Para obter mais informações, consulte Importar e exportar bancos de dados do SQL Server usando backup e restauração nativos.

Fusos horários compatíveis

Você pode definir seu fuso horário local como um dos valores listados na seguinte tabela.

Fuso horário	Compensação de tempo padrão	Descrição	Observações
Hora Padrão do Afeganistão	(UTC+04:30)	Kabul	Este fuso horário não observa o horário de verão.
Hora Padrão do Alasca	(UTC–09:00)	Alasca	
Hora Padrão Aleutiano	(UTC–10:00)	Ilhas Aleutas	
Hora Padrão de Altai	(UTC+07:00)	Barnaul, Gorno-Altaysk	
Hora Padrão Árabe	(UTC+03:00)	Kuwait, Riade	Este fuso horário não observa o horário de verão.
Hora Padrão da Arábia	(UTC+04:00)	Abu Dhabi, Muscat	
Hora Padrão Árabe	(UTC+03:00)	Bagdá	Este fuso horário não observa o horário de verão.

Fuso horário	Compensação de tempo padrão	Descrição	Observações
Hora Padrão da Argentina	(UTC–03:00)	Cidade de Buenos Aires	Este fuso horário não observa o horário de verão.
Hora Padrão de Astracã	(UTC+04:00)	Astracã, Ulianovsk	
Hora Padrão do Atlântico	(UTC–04:00)	Horário do Atlântico (Canadá)	
Hora Padrão Central da Austrália	(UTC+09:30)	Darwin	Este fuso horário não observa o horário de verão.
Hora Padrão Central da Austrália	(UTC+ 08:45)	Eucla	
Hora Padrão do Leste da Austrália	(UTC+10:00)	Canberra, Melbourne, Sydney	
Hora Padrão do Azerbaijão	(UTC+04:00)	Bacu	
Hora Padrão dos Açores	(UTC–01:00)	Açores	
Hora Padrão da Bahia	(UTC–03:00)	Salvador	
Hora Padrão de Bangladesh	(UTC+06:00)	Daca	Este fuso horário não observa o horário de verão.
Hora Padrão da Bielorrússia	(UTC+03:00)	Minsk	Este fuso horário não observa o horário de verão.
Fuso hora padrão de Bougainville	(UTC+11:00)	Ilha de Bougainville	

Fuso horário	Compensação de tempo padrão	Descrição	Observações
Hora Padrão Central do Canadá	(UTC–06:00)	Saskatchewan	Este fuso horário não observa o horário de verão.
Hora Padrão de Cabo Verde	(UTC–01:00)	Ilhas de Cabo Verde	Este fuso horário não observa o horário de verão.
Hora Padrão do Cáucaso	(UTC+04:00)	Erevan	
Hora Padrão da Austrália Central	(UTC+09:30)	Adelaide	
Hora Padrão da América Central	(UTC–06:00)	América Central	Este fuso horário não observa o horário de verão.
Hora Padrão da Ásia Central	(UTC+06:00)	Astana	Este fuso horário não observa o horário de verão.
Hora Padrão Central do Brasil	(UTC–04:00)	Cuiabá	
Hora Padrão Central da Europa	(UTC+01:00)	Belgrado, Bratislava, Budapeste, Ljubljana, Praga	
Hora Oficial da Europa Central	(UTC+01:00)	Sarajevo, Skopje, Varsóvia, Zagreb	
Hora Padrão Central do Pacífico	(UTC+11:00)	Ilhas Salomão, Nova Caledônia	Este fuso horário não observa o horário de verão.
Hora Padrão Central	(UTC–06:00)	Horário Central (EUA e Canadá)	

Fuso horário	Compensação de tempo padrão	Descrição	Observações
Hora Padrão Central (México)	(UTC–06:00)	Guadalajara, Cidade do México, Monterrey	
Fuso horário padrão das Ilhas Chatham	(UTC+ 12:45)	Ilhas Chatham	
Hora Padrão da China	(UTC+08:00)	Pequim, Chongqing, Hong Kong, Urumqi	Este fuso horário não observa o horário de verão.
Hora Padrão de Cuba	(UTC–05:00)	Havana	
Hora Padrão da Linha de Data	(UTC–12:00)	Linha Internacional de Data Oeste	Este fuso horário não observa o horário de verão.
Horário padrão da África Oriental	(UTC+03:00)	Nairóbi	Este fuso horário não observa o horário de verão.
Horário padrão da Austrália Oriental	(UTC+10:00)	Brisbane	Este fuso horário não observa o horário de verão.
Horário padrão da Europa Oriental	(UTC+02:00)	Chisinau	
Horário padrão da América do Sul Oriental	(UTC–03:00)	Brasília	
Hora Padrão da Ilha de Páscoa	(UTC–06:00)	Ilha de Páscoa	
Hora Oficial do Leste dos EUA	(UTC–05:00)	Hora do Leste (EUA e Canadá)	

Fuso horário	Compensação de tempo padrão	Descrição	Observações
Hora Padrão do Leste (México)	(UTC–05:00)	Chetumal	
Hora Padrão do Egito	(UTC+02:00)	Cairo	
Hora Padrão de Ekaterinburg	(UTC+ 05:00)	Ekaterinburg	
Hora Padrão de Fiji	(UTC+12:00)	Fiji	
Hora Padrão de FLE	(UTC+02:00)	Helsinki, Kiev, Riga, Sofia, Talin, Vilnius	
Hora Oficial da Geórgia	(UTC+04:00)	Tbilisi	Este fuso horário não observa o horário de verão.
Hora Padrão de Greenwich	(UTC)	Dublin, Edimburgo, Lisboa, Londres	Esse fuso horário não é igual ao Horário do Meridiano de Greenwich. Esse fuso horário segue o horário de verão.
Hora Oficial da Groenlândia	(UTC–03:00)	Groenlândia	
Hora Oficial de Greenwich	(UTC)	Monróvia, Reykjavik	Este fuso horário não observa o horário de verão.
Hora Oficial GTB	(UTC+02:00)	Atenas, Bucareste	
Hora Padrão do Haiti	(UTC–05:00)	Haiti	
Hora Oficial do Havaí	(UTC–10:00)	Havaí	

Fuso horário	Compensação de tempo padrão	Descrição	Observações
Hora Oficial da Índia	(UTC+05:30)	Chennai, Kolkata, Mumbai, Nova Deli	Este fuso horário não observa o horário de verão.
Hora Padrão do Irã	(UTC+ 03:30)	Teerão	
Hora Padrão de Israel	(UTC+02:00)	Jerusalém	
Hora Padrão da Jordânia	(UTC+02:00)	Amã	
Hora Padrão de Kaliningrado	(UTC+02:00)	Kaliningrado	
Hora Padrão de Kamchatka	(UTC+12:00)	Petropavlovsk-Kamchatsky – antigo	
Hora Padrão da Coreia	(UTC+09:00)	Seul	Este fuso horário não observa o horário de verão.
Hora Padrão da Líbia	(UTC+02:00)	Trípoli	
Hora Padrão das Ilhas Line	(UTC+ 14:00)	Ilha Kiritimati	
Hora Padrão de Lord Howe	(UTC+ 10:30)	Ilha de Lord Howe	
Hora Padrão de Magadan	(UTC+11:00)	Magadan	Este fuso horário não observa o horário de verão.
Hora Padrão de Magallanes	(UTC–03:00)	Punta Arenas	
Hora Padrão do Marquesas	(UTC–09:30)	Ilhas Marquesas	
Hora Padrão das Ilhas Maurício	(UTC+04:00)	Port Louis	Este fuso horário não observa o horário de verão.

Fuso horário	Compensação de tempo padrão	Descrição	Observações
Hora Padrão do Oriente Médio	(UTC+02:00)	Beirute	
Hora Padrão de Montevidéu	(UTC–03:00)	Montevidéu	
Hora Padrão de Marrocos	(UTC+01:00)	Casablanca	
Hora Oficial das Montanhas	(UTC–07:00)	Horário das Montanhas (EUA e Canadá)	
Hora Oficial das Montanhas (México)	(UTC–07:00)	Chihuahua, La Paz, Mazatlan	
Hora Padrão de Mianmar	(UTC+ 06:30)	Rangum (Rangum)	Este fuso horário não observa o horário de verão.
Horário padrão do Norte da Ásia do Central	(UTC+07:00)	Novosibirsk	
Hora Padrão da Namíbia	(UTC+02:00)	Windhoek	
Hora Padrão do Nepal	(UTC+ 05:45)	Katmandu	Este fuso horário não observa o horário de verão.
Hora Oficial da Nova Zelândia	(UTC+12:00)	Auckland, Wellington	
Hora Oficial de Newfoundland	(UTC–03:30)	Newfoundland	
Hora Padrão de Norfolk	(UTC+11:00)	Ilha Norfolk	
Hora Padrão do Nordeste da Ásia	(UTC+08:00)	Irkutsk	

Fuso horário	Compensação de tempo padrão	Descrição	Observações
Hora Padrão do Norte da Ásia	(UTC+07:00)	Krasnoyarsk	
Hora Padrão da Coreia do Norte	(UTC+09:00)	Pyongyang	
Hora Padrão de Omsk	(UTC+06:00)	Omsk	
Hora Padrão do Pacífico SA	(UTC–03:00)	Santiago	
Hora Oficial do Pacífico	(UTC–08:00)	Horário do Pacífico (EUA e Canadá)	
Hora Padrão Pacífico (México)	(UTC–08:00)	Baixa Califórnia	
Hora Padrão do Paquistão	(UTC+ 05:00)	Islamabad	Este fuso horário não observa o horário de verão.
Hora Padrão do Paraguai	(UTC–04:00)	Assunção	
Hora Padrão de Romance	(UTC+01:00)	Bruxelas, Copenhagen, Madri, Paris	
Fuso Horário da Rússia 10	(UTC+11:00)	Chokurdakh	
Fuso Horário da Rússia 11	(UTC+12:00)	Anadyr, Petropavl ovsk-Kamchatsky	
Fuso Horário da Rússia 3	(UTC+04:00)	Izhevsk, Samara	
Hora Padrão da Rússia	(UTC+03:00)	Moscou, São Petersburgo, Volgograd	Este fuso horário não observa o horário de verão.

Fuso horário	Compensação de tempo padrão	Descrição	Observações
Hora Padrão do Leste de SA	(UTC–03:00)	Cayenne, Fortaleza	Este fuso horário não observa o horário de verão.
Hora Padrão do Pacífico na América do Sul	(UTC–05:00)	Bogotá, Lima, Quito, Rio Branco	Este fuso horário não observa o horário de verão.
Hora Padrão de SA Ocidental	(UTC–04:00)	Georgetown, La Paz, Manaus, San Juan	Este fuso horário não observa o horário de verão.
Hora Padrão de Saint Pierre	(UTC–03:00)	Saint Pierre e Miquelon	
Hora Padrão de Sacalina	(UTC+11:00)	Sacalina	
Hora Padrão de Samoa	(UTC+ 13:00)	Samoa	
Hora Padrão de São Tomé	(UTC+01:00)	São Tomé	
Hora Padrão de Saratov	(UTC+04:00)	Saratov	
Hora Padrão do Sudeste da Ásia	(UTC+07:00)	Bangkok, Hanoi, Jakarta	Este fuso horário não observa o horário de verão.
Hora Padrão de Cingapura	(UTC+08:00)	Kuala Lumpur, Cingapura	Este fuso horário não observa o horário de verão.
Hora Padrão da África do Sul	(UTC+02:00)	Harare, Pretória	Este fuso horário não observa o horário de verão.

Fuso horário	Compensação de tempo padrão	Descrição	Observações
Hora Padrão do Sri Lanka	(UTC+05:30)	Sri Jayawarde nepura	Este fuso horário não observa o horário de verão.
Hora Padrão do Sudão	(UTC+02:00)	Cartum	
Hora Padrão da Síria	(UTC+02:00)	Damasco	
Hora padrão de Taipei	(UTC+08:00)	Taipei	Este fuso horário não observa o horário de verão.
Hora Padrão da Tasmânia	(UTC+10:00)	Hobart	
Hora Padrão do Tocantins	(UTC–03:00)	Araguaina	
Hora Padrão de Tóquio	(UTC+09:00)	Osaka, Sapporo, Tóquio	Este fuso horário não observa o horário de verão.
Hora Padrão de Tomsk	(UTC+07:00)	Tomsk	
Hora Padrão de Tonga	(UTC+ 13:00)	Nuku'alofa	Este fuso horário não observa o horário de verão.
Hora Padrão do Transbaikal	(UTC+09:00)	Chita	
Hora Padrão da Turquia	(UTC+03:00)	Istambul	
Hora Padrão de Turks e Caicos	(UTC–05:00)	Turks e Caicos	
Hora Padrão de Ulaanbaatar	(UTC+08:00)	Ulaanbaatar	Este fuso horário não observa o horário de verão.

Fuso horário	Compensação de tempo padrão	Descrição	Observações
Hora Padrão do Leste dos EUA	(UTC–05:00)	Indiana (Leste)	
Hora Oficial das Montanhas (EUA)	(UTC–07:00)	Arizona	Este fuso horário não observa o horário de verão.
UTC	UTC	Tempo Universal Coordenado	Este fuso horário não observa o horário de verão.
UTC–02	(UTC–02:00)	Tempo Universal Coordenado–02	Este fuso horário não observa o horário de verão.
UTC–08	(UTC–08:00)	Tempo Universal Coordenado–08	
UTC–09	(UTC–09:00)	Tempo Universal Coordenado–09	
UTC–11	(UTC–11:00)	Tempo Universal Coordenado–11	Este fuso horário não observa o horário de verão.
UTC+12	(UTC+12:00)	Tempo Universal Coordenado+12	Este fuso horário não observa o horário de verão.
UTC+13	(UTC+ 13:00)	Tempo Universal Coordenado+13	
Hora Padrão da Venezuela	(UTC–04:00)	Caracas	Este fuso horário não observa o horário de verão.

Fuso horário	Compensação de tempo padrão	Descrição	Observações
Hora Padrão de Vladivostok	(UTC+10:00)	Vladivostok	
Hora Padrão de Volgograd	(UTC+04:00)	Volgograd	
Horário padrão da Austrália Ocidental Ocidental	(UTC+08:00)	Perth	Este fuso horário não observa o horário de verão.
Horário padrão da África Central Ocidental	(UTC+01:00)	Centro-oeste da África	Este fuso horário não observa o horário de verão.
Horário padrão da Europa Ocidental	(UTC+01:00)	Amsterdã, Berlim, Berna, Roma, Estocolmo, Viena	
Horário Padrão da Mongólia Ocidental	(UTC+07:00)	Hovd	
Hora Padrão do Oeste da Ásia	(UTC+ 05:00)	Ashgabat, Tashkent	Este fuso horário não observa o horário de verão.
Hora Padrão da Cisjordânia	(UTC+02:00)	Gaza, Hebron	
Hora Padrão do Oeste do Pacífico	(UTC+10:00)	Guam, Porto Moresby	Este fuso horário não observa o horário de verão.
Hora Padrão de Yakutsk	(UTC+09:00)	Yakutsk	

Licenciar o Microsoft SQL Server no Amazon RDS

Quando você configura uma instância de banco de dados do Amazon RDS para o Microsoft SQL Server, a licença do software é incluída.

Isso significa que você não precisará adquirir licenças do SQL Server separadamente. A AWS detém a licença para o software de banco de dados SQL Server. O preço do Amazon RDS inclui a licença do software, os recursos de hardware subjacentes e os recursos de gerenciamento do Amazon RDS.

Amazon RDS oferece suporte às seguintes edições do Microsoft SQL Server:

- Enterprise
- Padrão
- Web
- Express

> **Note**
>
> O licenciamento para o SQL Server Web Edition é compatível apenas com páginas da web, sites, aplicativos web e serviços da web, acessíveis ao público e pela Internet. Esse nível de suporte é necessário para garantir a conformidade com os direitos de uso da Microsoft. Para obter mais informações, consulte os Termos de serviço da AWS.

O Amazon RDS oferece suporte a implantações Multi-AZ para instâncias de bancos de dados que executem o Microsoft SQL Server usando o SQL Server Database Mirroring (DBM) ou Grupos de disponibilidade AlwaysOn. Não há requisitos adicionais de licenciamento para implantações Multi-AZ. Para obter mais informações, consulte Implantações multi-AZ para o Amazon RDS for Microsoft SQL Server.

Restaurar instâncias de banco de dados com licenças encerradas

O Amazon RDS faz snapshots de instâncias de banco de dados com licenças encerradas. Se sua instância for encerrada devido a problemas de licenciamento, você poderá restaurá-la a partir do snapshot para uma nova instância de base de dados. As novas instâncias de banco já têm licença inclusa.

Para obter mais informações, consulte Restaurar instâncias de banco de dados com licenças encerradas no Amazon RDS para SQL Server.

Desenvolvimento e testes

Devido a requisitos de licenciamento, não é possível utilizar a edição Developer do SQL Server no Amazon RDS. Você pode usar a edição Express para desenvolvimento, teste e outras necessidades não relacionadas a produção. No entanto, se você precisar dos recursos completos de uma instalação de nível empresarial do SQL Server para desenvolvimento, poderá baixar e instalar o SQL Server Developer Edition no RDS Custom para SQL Server usando uma CEV com BYOM. Para obter mais informações, consulte Preparar uma CEV usando o modelo BYOM (traga sua própria mídia). A infraestrutura dedicada não é necessária na edição Developer. Ao usar seu próprio host, você também acessa outros recursos de programabilidade que não podem ser acessados no Amazon RDS. Para ter mais informações sobre a diferença entre as edições do SQL Server, consulte Editions and supported features of SQL Server 2019 na documentação da Microsoft.

Conectar-se à instância de banco de dados do Microsoft SQL Server

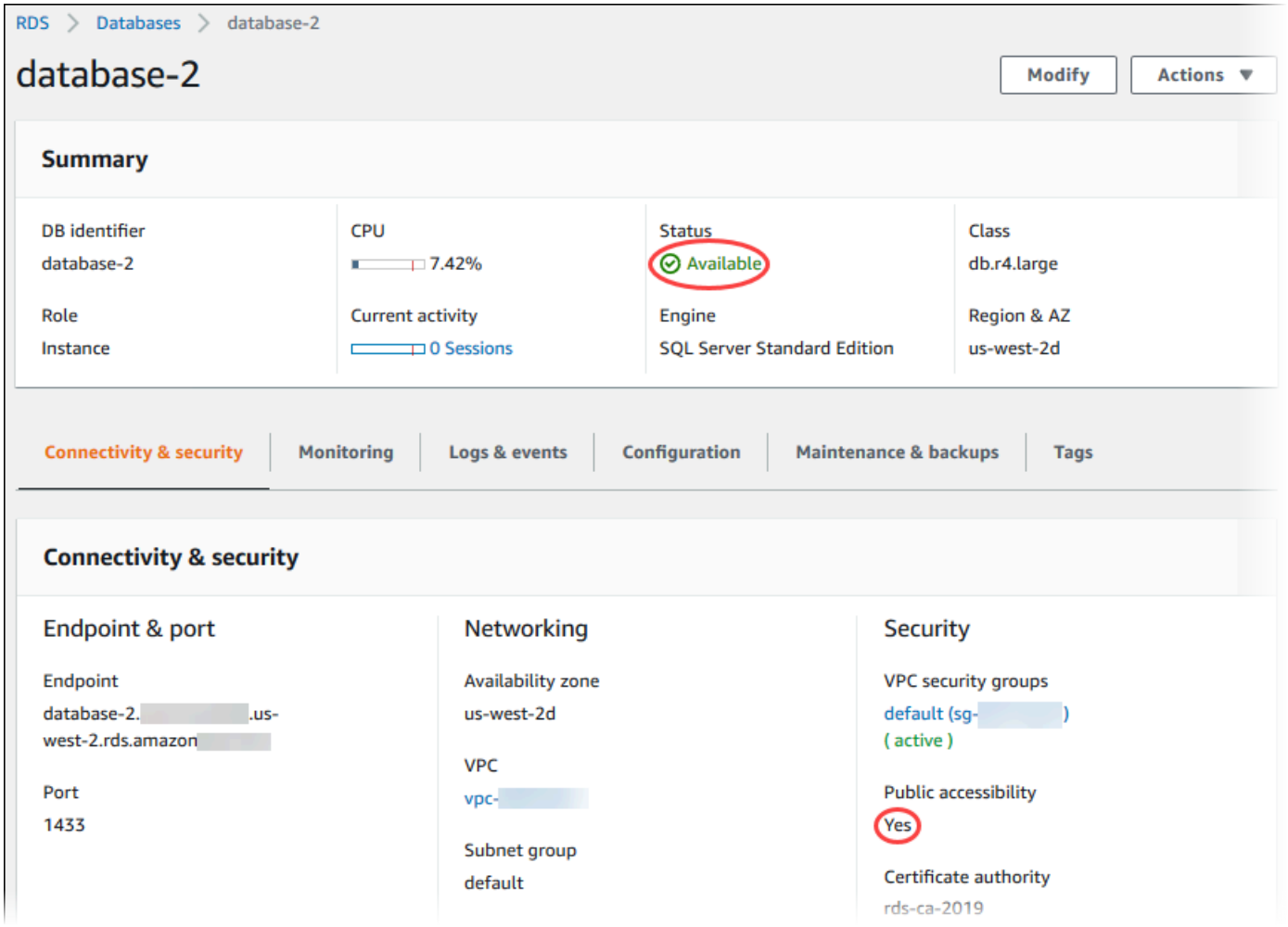
Depois que o Amazon RDS provisiona a sua instância de banco de dados, você pode usar qualquer aplicativo cliente padrão SQL para se conectar à instância de banco de dados. Neste tópico, você se conecta à sua instância de banco de dados usando o Microsoft SQL Server Management Studio (SSMS) ou o SQL Workbench/J.

Consulte e veja um exemplo do processo de criação e conexão com uma instância de banco de dados de amostr Criar uma instância de banco de dados do Microsoft SQL Server e conectar-se a ela.

Antes de se conectar

Antes que você possa se conectar à sua instância de banco de dados, ela precisa estar disponível e acessível.

1. Certifique-se de que seu status seja `available`. Você pode verificar isso na página de detalhes da instância no AWS Management Console ou usando o comando da AWS CLI describe-db-instances.



2. Certifique-se de que esteja acessível à sua origem. Dependendo do seu cenário, talvez não precise ser acessível ao público. Para ter mais informações, consulte Amazon VPC e Amazon RDS.
3. Certifique-se de que as regras de entrada do grupo de segurança da VPC permitam acesso à sua instância de banco de dados. Para obter mais informações, consulte Não é possível conectar-se à instância de banco de dados do Amazon RDS.

Localizando o endpoint e o número da porta da instância de banco de dados

Você precisará do endpoint e do número da porta para conectar-se à instância de banco de dados.

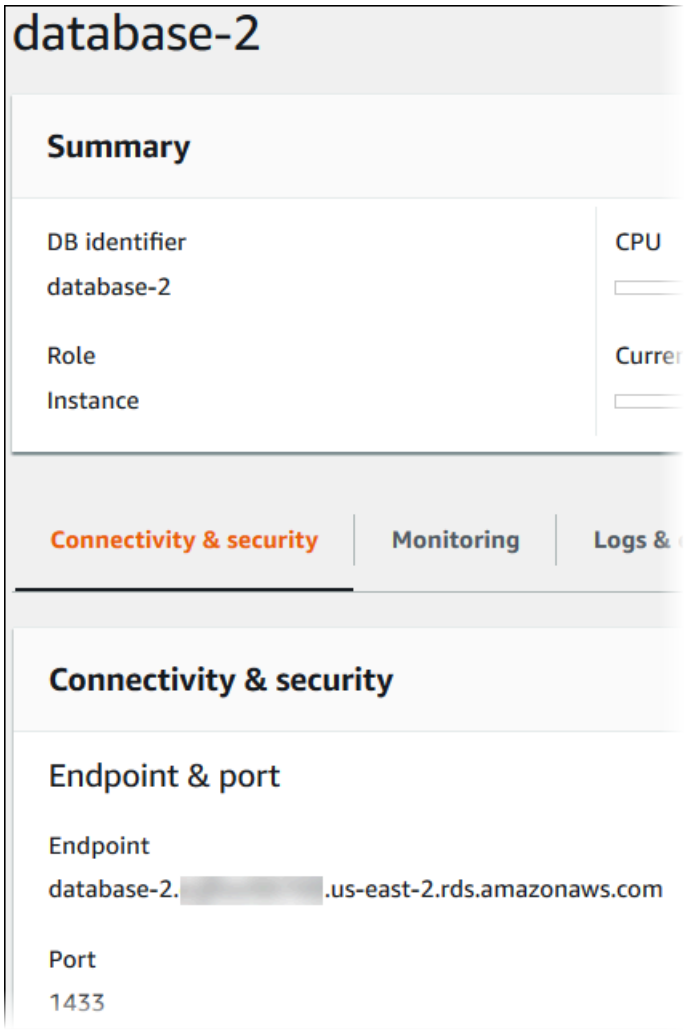
Para encontrar o endpoint e a porta

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No canto superior direito do console do Amazon RDS, escolha a região da AWS da instância de banco de dados.
3. Localize o nome (endpoint) do Sistema de Nomes de Domínio (DNS) e o número da porta da instância de banco de dados:

   a. Abra o console do RDS e escolha Databases (Bancos de dados) para exibir uma lista das instâncias de banco de dados.

   b. Escolha o nome da instância de banco de dados do SQL Server para exibir os detalhes.

   c. Na guia Connectivity & security (Conectividade e segurança), copie o endpoint.



   d. Anote o número da porta.

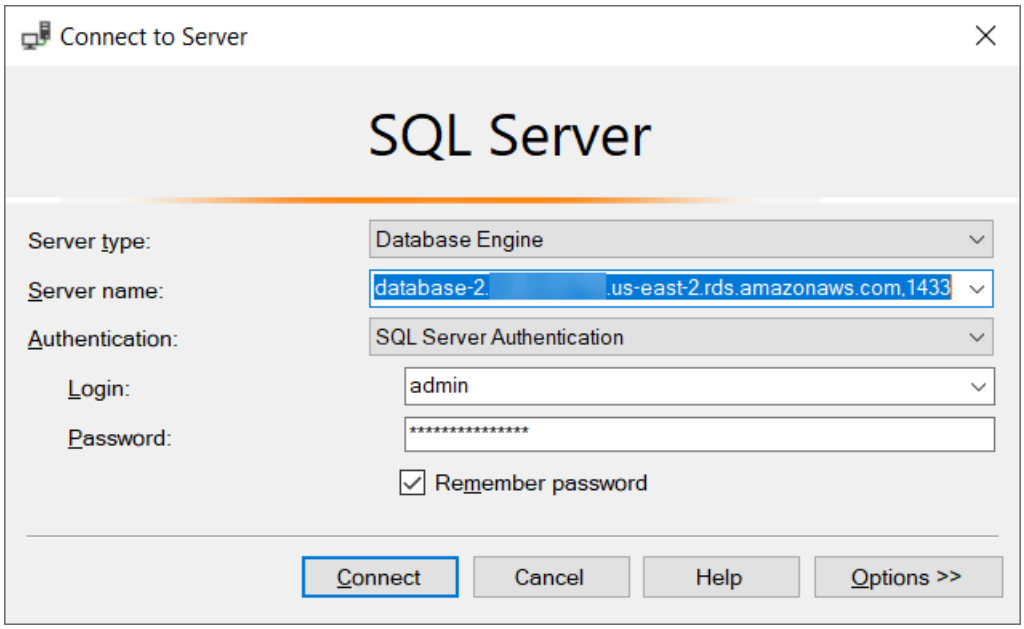
Conectar-se à sua instância de banco de dados com o Microsoft SQL Server Management Studio

Neste procedimento, você se conecta à instância de banco de dados de exemplo usando o Microsoft SQL Server Management Studio (SSMS). Para fazer download de uma versão autônoma desse utilitário, consulte Download do SQL Server Management Studio (SSMS) na documentação da Microsoft.

Como se conectar a uma instância de banco de dados usando o SSMS

1. Inicie o SQL Server Management Studio.

   A caixa de diálogo Connect to Server (Conectar-se ao servidor) é exibida.



2. Forneça as informações da instância de banco de dados:

   a. Para Server type (Tipo de servidor), escolha Database Engine (Mecanismo do banco de dados).

   b. Em Server name (Nome do servidor), digite o nome (endpoint) DNS e o número da porta da instância de banco de dados separados por vírgula.

> **Important**
>
> Substitua o caractere de dois-pontos entre o endpoint e o número da porta por uma vírgula.

O nome do servidor deve ser semelhante ao seguinte exemplo:

```
database-2.cg034itsfake.us-east-1.rds.amazonaws.com,1433
```

c. Para Authentication (Autenticação), escolha SQL Server Authentication (Autenticação do SQL Server).

d. Em Login, digite o nome do usuário mestre da instância de banco de dados.

e. Em Password (Senha), digite a senha da instância de banco de dados.

3. Selecione Conectar.

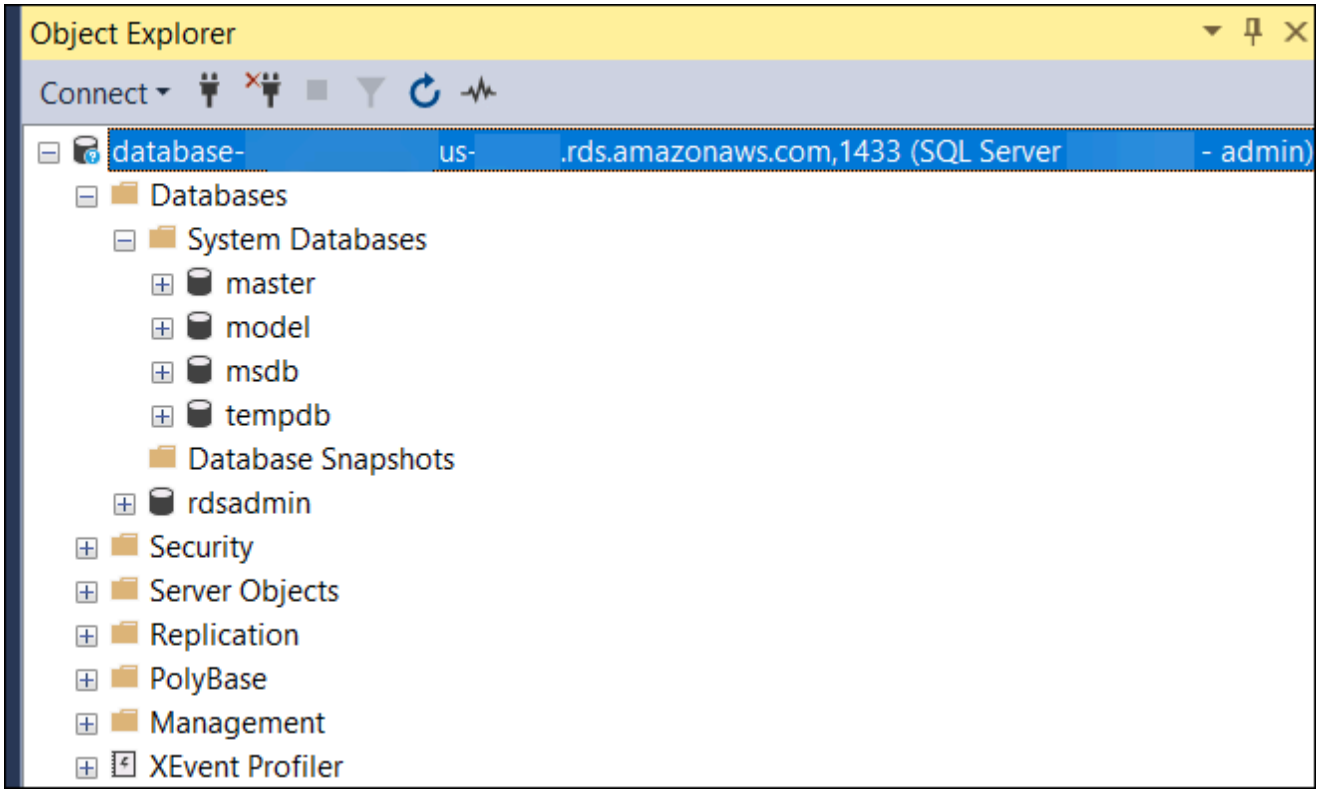
   Depois de alguns momentos, o SSMS conecta-se à sua instância de banco de dados.

   Se não conseguir se conectar à sua instância de banco de dados, consulte Considerações sobre os grupos de segurança e Solução de problemas de conexões com a instância de banco de dados do SQL Server.

4. A instância de banco de dados SQL Server vem com bancos de dados de sistema integrados padrão do SQL Server (`master`, `model`, `msdb` e `tempdb`). Para explorar os bancos de dados do sistema, faça o seguinte:

   a. Em SSMS, no menu View (Exibir), escolha Object Explorer (Navegador de objetos).

   b. Expanda a instância de banco de dados, expanda Bancos de dados e Bancos de dados do sistema.



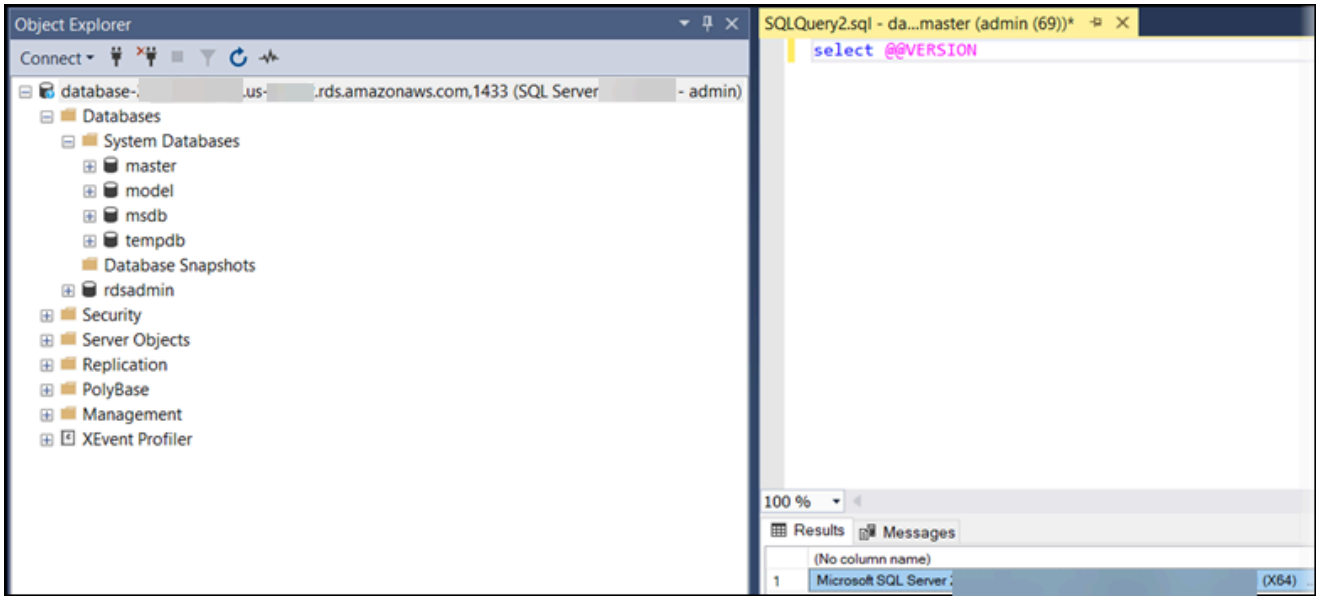
5. Sua instância de banco de dados do SQL Server também conta com um banco de dados chamado `rdsadmin`. O Amazon RDS usa esse banco de dados para armazenar os objetos usados para gerenciar o banco de dados. O banco de dados `rdsadmin` também inclui procedimentos armazenados que você pode executar para realizar tarefas avançadas. Para obter mais informações, consulte Tarefas comuns de DBA no Amazon RDS para Microsoft SQL Server.
6. Agora, você pode começar a criar seus próprios bancos de dados e executar consultas em instâncias de banco de dados e bancos de dados como de costume. Para executar uma consulta de teste na sua instância de banco de dados, faça o seguinte:

   a. No SSMS, no menu File (Arquivo), aponte para New (Novo) e escolha Query with Current Connection (Consultar com a conexão atual).

   b. Digite a consulta SQL a seguir.

   ```
   select @@VERSION
   ```

   c. Execute a consulta. O SSMS retorna a versão do SQL Server da sua instância de banco de dados do Amazon RDS.



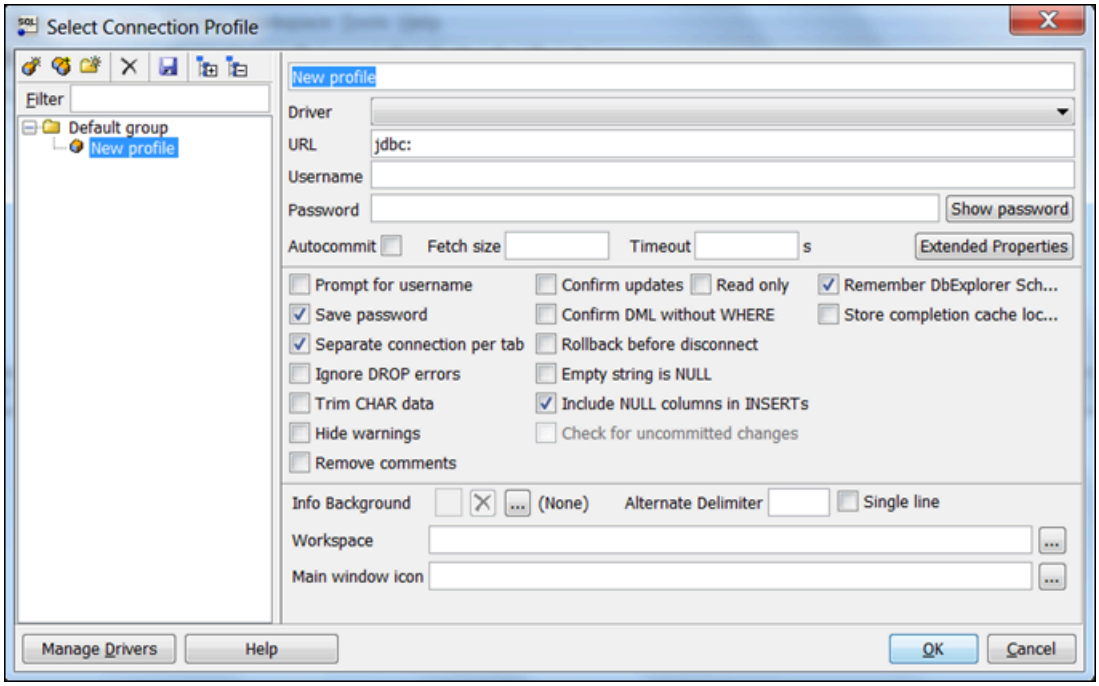
Conectar-se à sua instância de banco de dados com o SQL Workbench/J

Este exemplo mostra como se conectar a uma instância de banco de dados executando o mecanismo de banco de dados do Microsoft SQL Server usando a ferramenta de banco de dados SQL Workbench/J. Para fazer download do SQL Workbench/J, consulte SQL Workbench/J.

O SQL Workbench/J usa o JDBC para se conectar à sua instância de banco de dados. Você também precisa do driver JDBC para o SQL Server. Para baixar esse driver, acesse Baixe o Microsoft JDBC Driver para SQL Server.

Como conectar-se a uma instância de banco de dados usando o SQL Workbench

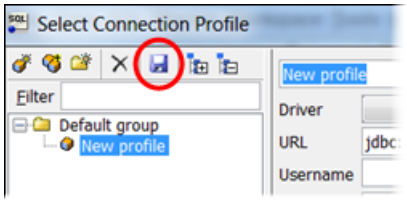
1. Open SQL Workbench/J. A caixa de diálogo Select Connection Profile (Selecionar perfil de conexão) é exibida, conforme mostrado a seguir.



2. Na primeira caixa na parte superior da caixa de diálogo, insira um nome para o perfil.
3. Em Driver, escolha **`SQL JDBC 4.0`**.
4. Em URL, digite **`jdbc:sqlserver://`** e o endpoint da instância de banco de dados. Por exemplo, o valor do URL pode ser este.

```
jdbc:sqlserver://sqlsvr-pdz.abcd12340.us-west-2.rds.amazonaws.com:1433
```

5. Em Username (Nome de usuário), digite o nome do usuário mestre da instância de banco de dados.
6. Em Password (Senha), digite a senha do usuário mestre.
7. Escolha o ícone de gravação na barra de ferramentas da caixa de diálogo, conforme mostrado a seguir.

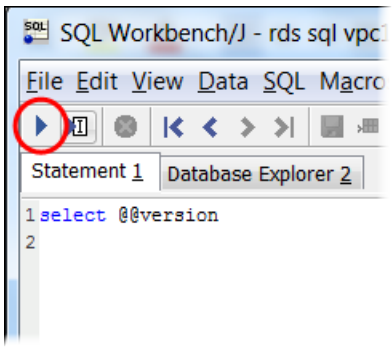


8. Escolha OK. Depois de alguns momentos, o SQL Workbench/J conecta-se à sua instância de banco de dados. Se não conseguir se conectar à sua instância de banco de dados, consulte Considerações sobre os grupos de segurança e Solução de problemas de conexões com a instância de banco de dados do SQL Server.

9. No painel de consultas, digite a consulta SQL a seguir.

```
select @@VERSION
```

10. Escolha o ícone `Execute` na barra de ferramentas, conforme mostrado a seguir.



A consulta retorna as informações de versão da instância de banco de dados, semelhante às informações a seguir.

```
Microsoft SQL Server 2017 (RTM-CU22) (KB4577467) - 14.0.3356.20 (X64)
```

Considerações sobre os grupos de segurança

Para se conectar à sua instância de banco de dados, ela deve estar associada a um grupo de segurança. Esse grupo de segurança contém os endereços IP e a configuração de rede usada para acessar a instância de banco de dados. Talvez você tenha associado sua instância de banco de dados a um grupo de segurança apropriado quando a criou. Se você atribuiu um grupo de segurança padrão não configurado quando criou sua instância de banco de dados, o firewall dessa instância impedirá conexões.

Em alguns casos, talvez seja necessário criar um grupo de segurança para possibilitar o acesso. Para obter instruções sobre como criar um novo security group, consulte Controlar acesso com grupos de segurança. Para obter um tópico que fornece orientação pelo processo de configuração de regras para seu grupo de segurança da VPC, consulte Tutorial: Criar uma VPC para usar com uma instância de banco de dados (somente IPv4).

Depois de criar o novo grupo de segurança, modifique sua instância de banco de dados para associá-la a esse grupo de segurança. Para obter mais informações, consulte Modificar uma instância de banco de dados do Amazon RDS.

Você pode aprimorar a segurança usando o SSL para criptografar conexões com sua a instância de banco de dados. Para ter mais informações, consulte ???.

Solução de problemas de conexões com a instância de banco de dados do SQL Server

A tabela a seguir mostra mensagens de erro que você pode encontrar quando tenta se conectar à sua instância de banco de dados do SQL Server.

Problema	Sugestões de solução de problemas
Não foi possível abrir uma conexão com o SQL Server – Microsoft SQL Server, Erro: 53	Certifique-se de que especificou o nome do servidor corretamente. Em Server name (Nome do servidor), digite o nome DNS e o número da porta da instância de banco de dados de exemplo, separados por vírgula. **Important** Se tiver dois-pontos entre o nome DNS e o número da porta, troque os dois-pontos por uma vírgula. O nome do servidor deve ser semelhante ao seguinte exemplo: `sample-instance.cg034itsfake.us-east-1.rds.amazonaws.com,1433`
Não foi possível estabelecer uma conexão porque a máquina de destino a recusou ativamente – Microsoft SQL Server, erro: 10061	Você conseguiu acessar a instância de banco de dados, mas a conexão foi recusada. Geralmente, esse problema é causado pela especificação do nome de usuário ou da senha incorretamente. Verifique o nome do usuário e a senha e tente novamente.
Ocorreu um erro específico da instância ou relacionado à rede ao estabelecer uma conexão com o SQL	As regras de acesso impostas pelo firewall local e os endereços IP autorizados a acessar a instância de banco de dados podem não corresponder. O problema está provavelmente nas regras de

Problema	Sugestões de solução de problemas
Server. O servidor não foi encontrado ou não estava acessível... A operação de espera expirou – Microsoft SQL Server, Erro: 258	entrada do seu grupo de segurança. Para ter mais informações, consulte Segurança no Amazon RDS.
	A instância de banco de dados deve ser acessível ao público. Para conectar-se a ela de fora da VPC, a instância deve ter um endereço IP público atribuído.

> **Note**
>
> Para ter mais informações sobre problemas de conexão, consulte Não é possível conectar-se à instância de banco de dados do Amazon RDS.

Trabalhar com o Active Directory com o RDS para SQL Server

Você pode unir uma instância de banco de dados do RDS para SQL Server a um domínio do Microsoft Active Directory (AD). Seu domínio do AD pode ser hospedado no AWS Managed AD dentro da AWS ou em um AD autogerenciado em um local de sua escolha, incluindo em seus datacenters corporativos, no AWS EC2 ou em outros provedores de nuvem.

Você pode autenticar usuários do domínio usando a autenticação NTLM com um Active Directory autogerenciado. Você pode usar a autenticação Kerberos e NTLM com o AWS Managed Active Directory.

Nas seções a seguir, você pode encontrar informações sobre como trabalhar com um Active Directory autogerenciado e com o AWS Managed Active Directory para Microsoft SQL Server no Amazon RDS.

Tópicos

- Trabalhar com um Active Directory autogerenciado com uma instância de banco de dados do Amazon RDS para SQL Server
- Trabalhar com o AWS Managed Active Directory com o RDS para SQL Server

Trabalhar com um Active Directory autogerenciado com uma instância de banco de dados do Amazon RDS para SQL Server

Você pode unir suas instâncias de banco de dados do RDS para SQL Server diretamente ao seu domínio autogerenciado do Active Directory (AD), independentemente de onde seu AD esteja hospedado: em datacenters corporativos, no AWS EC2 ou em outros provedores de nuvem. Com um AD autogerenciado, você usa a autenticação NTLM para controlar diretamente a autenticação de usuários e serviços em suas instâncias de banco de dados do RDS para SQL Server sem usar domínios intermediários e relações de confiança de floresta. Quando os usuários se autenticam em uma instância de banco de dados do RDS para SQL Server associada ao seu domínio de AD autogerenciado, as solicitações de autenticação são encaminhadas para um domínio de AD autogerenciado especificado por você.

Tópicos

- Disponibilidade de região e versão
- Requisitos
- Limitações
- Visão geral da configuração de um Active Directory autogerenciado
- Configurar um Active Directory autogerenciado
- Gerenciar uma instância de banco de dados em um domínio de Active Directory autogerenciado
- Entender a associação a um domínio de Active Directory autogerenciado
- Solução de problemas de Active Directory autogerenciado
- Restaurar uma instância de banco de dados do SQL Server e adicioná-la a um domínio de Active Directory autogerenciado

Disponibilidade de região e versão

O Amazon RDS oferece suporte a AD autogerenciado para SQL Server usando NTLM em todas as Regiões da AWS.

Requisitos

Verifique se você atendeu aos seguintes requisitos antes de unir uma instância de banco de dados do RDS para SQL Server ao seu domínio de AD autogerenciado.

Tópicos

- Configurar um AD on-premises
- Configurar sua conectividade de rede
- Configurar uma conta de serviço do domínio de AD

Configurar um AD on-premises

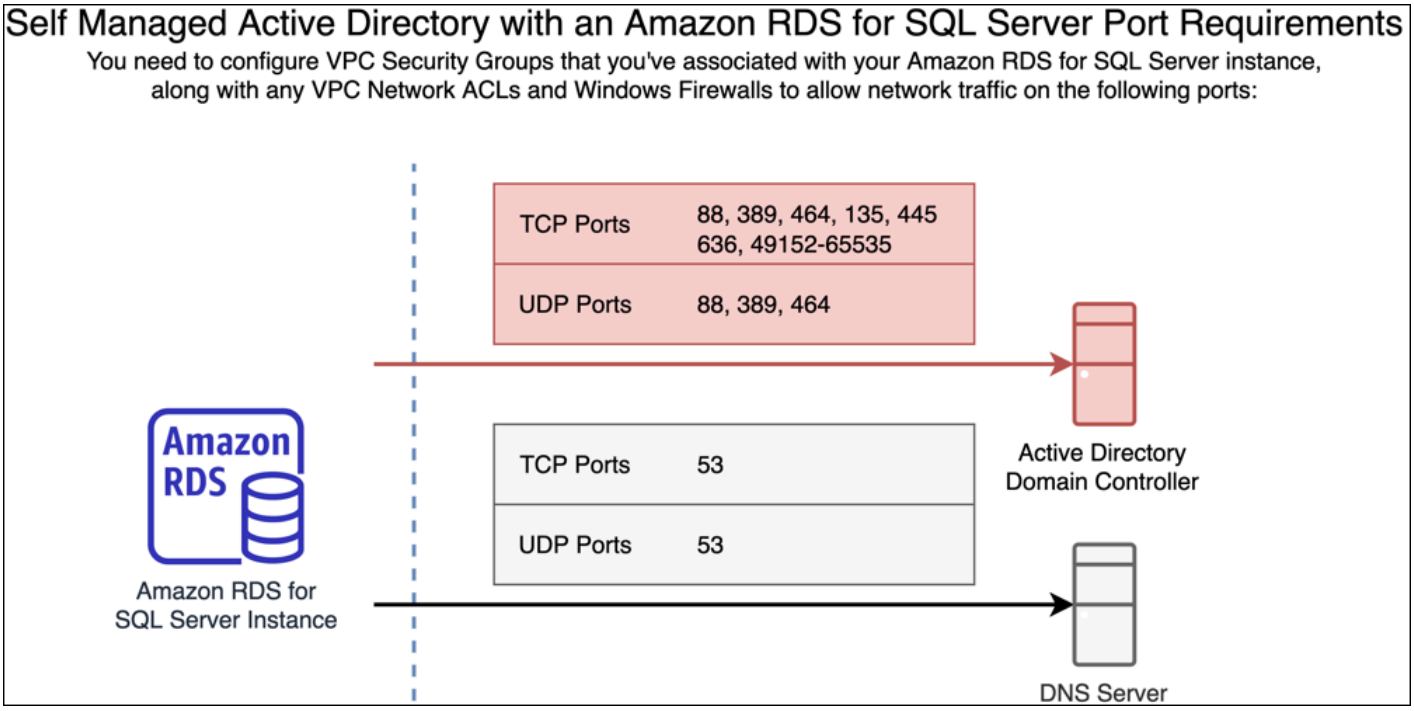
Você precisa ter um Microsoft AD on-premises ou autogerenciado ao qual possa associar a instância do Amazon RDS para SQL Server. O AD on-premises deve ter a seguinte configuração:

- Se você tiver sites do Active Directory definidos, verifique se as sub-redes na VPC associada à sua instância de banco de dados do RDS para SQL Server estão definidas em seu site do Active Directory. Confirme se não há nenhum conflito entre as sub-redes em sua VPC e as sub-redes em seus outros sites do AD.
- Seu controlador de domínios de AD tem um nível funcional de domínio do Windows Server 2008 R2 ou posterior.
- Seu nome de domínio de AD não pode estar no formato de domínio de rótulo único (SLD). O RDS para SQL Server não oferece suporte a domínios SLD.
- O nome de domínio totalmente qualificado (FQDN) do AD não pode exceder 47 caracteres.

Configurar sua conectividade de rede

Você precisa atender às seguintes configurações de rede:

- Conectividade configurada entre a Amazon VPC na qual deseja criar a instância de banco de dados do RDS para SQL Server e o Active Directory autogerenciado. Você pode configurar a conectividade usando o AWS Direct Connect, o AWS VPN, o emparelhamento de VPC ou o AWS Transit Gateway.
- Para grupos de segurança de VPC, o grupo de segurança padrão para sua Amazon VPC padrão já está adicionado à sua instância de banco de dados do RDS para SQL Server no console. Verifique se o grupo de segurança e as ACLs de rede da VPC para as sub-redes em que você vai criar a instância de banco de dados do RDS para SQL Server permitem tráfego nas portas e nas direções mostradas no diagrama a seguir.



A tabela a seguir identifica o perfil de cada porta.

Protocolo	Portas	Função
TCP/UDP	53	Domain Name System (DNS)
TCP/UDP	88	Autenticação de Kerberos
TCP/UDP	464	Alterar/definir senha
TCP/UDP	389	Lightweight Directory Access Protocol (LDAP)
TCP	135	Distributed Computing Environment/End Point Mapper (DCE/EPMAP)
TCP	445	Compartilhamento de arquivos de SMB para serviços de diretório

Protocolo	Portas	Função
TCP	636	Lightweight Directory Access Protocol over TLS/SSL (LDAPS)
TCP	49152 – 65535	Portas efêmeras para RPC

- Em geral, os servidores DNS do domínio estão localizados nos controladores do domínio de AD. Você não precisa configurar o conjunto de opções DHCP da VCP para usar esse atributo. Para obter mais informações, consulte Conjuntos de opções DHCP no Guia do usuário do Amazon VPC.

> **Important**
>
> Se você estiver usando ACLs de rede de VPC, também deverá permitir tráfego de saída em portas dinâmicas (49152-65535) da sua instância de banco de dados do RDS para SQL Server. Confira se essas regras de tráfego também são refletidas nos firewalls que se aplicam a cada um dos controladores do domínio de AD, servidores DNS e instâncias de banco de dados do RDS para SQL Server.
> Embora os grupos de segurança de VPC exijam que as portas sejam abertas somente na direção em que o tráfego de rede é iniciado, a maioria dos firewalls do Windows e das ACLs da rede de VPC exigem que as portas sejam abertas nas duas direções.

Configurar uma conta de serviço do domínio de AD

Verifique se você atendeu aos seguintes requisitos para uma conta de serviço do domínio de AD:

- Confira se você tem uma conta de serviço em seu domínio de AD autogerenciado com permissões delegadas para associar computadores ao domínio. Uma conta de serviço de domínio é uma conta de usuário em seu AD autogerenciado à qual foi delegada permissão para realizar determinadas tarefas.
- É necessário delegar à conta de serviço de domínio as seguintes permissões na unidade organizacional (OU) à qual você está associando a instância de banco de dados do RDS para SQL Server:
  - Capacidade validada para gravar no nome do host DNS

- Capacidade validada para gravar no nome da entidade principal de serviço
- Criar e excluir objetos de computador

Essas permissões representam o conjunto mínimo de permissões necessárias para associar objetos de computador a um Active Directory autogerenciado. Para obter mais informações, consulte Erros ao tentar associar computadores a um domínio na documentação do Microsoft Windows Server.

> **Important**
>
> Não mova objetos de computador criados pelo RDS para SQL Server na unidade organizacional depois da criação da instância de banco de dados. Mover os objetos associados fará com que sua instância de banco de dados do RDS para SQL Server fique malconfigurada. Se você precisar mover os objetos de computador criados pelo Amazon RDS, use a operação de API ModifyDBInstance do RDS para modificar os parâmetros do domínio com a localização desejada dos objetos de computador.

Limitações

As seguintes limitações se aplicam a um AD autogerenciado para SQL Server.

- NTLM é o único tipo de autenticação compatível. A autenticação Kerberos não é compatível. Se você precisar usar a autenticação Kerberos, poderá usar o AWS Managed AD em vez de um AD autogerenciado.
- O serviço Coordenador de Transações Distribuídas da Microsoft (MSDTC) não é compatível, pois requer autenticação Kerberos.
- Suas instâncias de banco de dados do RDS para SQL Server não usam o servidor de protocolo NTP (Network Time Protocol) do seu domínio de AD autogerenciado. Em vez disso, usam um serviço de NTP da AWS.
- Os servidores vinculados ao SQL Server devem usar a autenticação SQL para se conectar a outras instâncias de banco de dados do RDS para SQL Server associadas ao seu domínio de AD autogerenciado.
- As configurações de Objeto de Política de Grupo (GPO) da Microsoft do seu domínio de AD autogerenciado não são aplicadas a instâncias de banco de dados do RDS para SQL Server.

Visão geral da configuração de um Active Directory autogerenciado

Para configurar um AD autogerenciado para uma instância de banco de dados RDS para SQL Server, siga as etapas a seguir, explicadas com mais detalhes em Configurar um Active Directory autogerenciado:

Em seu domínio de AD:

- Crie uma unidade organizacional (OU).
- Crie um usuário do domínio de AD.
- Delegue controle ao usuário do domínio de AD.

Usando o AWS Management Console ou a API:

- Crie uma chave do AWS KMS.
- Crie um segredo usando o AWS Secrets Manager.
- Crie ou modifique uma instância de banco de dados do RDS para SQL Server e associe-a ao domínio de AD autogerenciado.

Configurar um Active Directory autogerenciado

Para configurar um AD autogerenciado, siga as etapas a seguir.

Tópicos

- Etapa 1: Criar uma unidade organizacional no AD
- Etapa 2: Criar um usuário de domínio em seu AD
- Etapa 3: Delegar controle ao usuário de AD
- Etapa 4: Criar uma chave do AWS KMS
- Etapa 5: Criar um segredo da AWS
- Etapa 6: Criar ou modificar a instância de banco de dados do SQL Server
- Etapa 7: Criar logins do SQL Server de Autenticação do Windows

Etapa 1: Criar uma unidade organizacional no AD

> **Important**
>
> Recomendamos criar uma OU e uma credencial de serviço dedicadas com escopo para essa OU para todas as contas da AWS que tenham uma instância de banco de dados do RDS para SQL Server associada ao seu domínio de AD autogerenciado. Ao dedicar uma OU e uma credencial de serviço, você pode evitar permissões conflitantes e seguir o princípio de privilégio mínimo.

Como criar uma OU no AD

1. Conecte-se ao seu domínio de AD como administrador do domínio.
2. Abra Usuários e computadores do Active Directory e selecione o domínio em que deseja criar a OU.
3. Clique com o botão direito do mouse no domínio, escolha Novo e selecione Unidade organizacional.
4. Insira um nome para a OU.
5. Mantenha a caixa Proteger o contêiner contra exclusão acidental selecionada.
6. Clique em OK. A nova OU será exibida em seu domínio.

Etapa 2: Criar um usuário de domínio em seu AD

As credenciais do usuário de domínio serão usadas para o segredo no AWS Secrets Manager.

Como criar um usuário de domínio em seu AD

1. Abra Usuários e computadores do Active Directory e selecione o domínio e a OU em que deseja criar o usuário.
2. Clique com o botão direito do mouse no objeto Usuários, escolha Novo e selecione Usuário.
3. Insira um nome, sobrenome e nome de login para o usuário. Clique em Next.
4. Insira uma senha para o usuário. Não selecione a opção “O usuário deve alterar a senha no próximo login”. Não selecione a opção “A conta está desabilitada”. Clique em Next.
5. Clique em OK. O novo usuário será exibido em seu domínio.

Etapa 3: Delegar controle ao usuário de AD

Como delegar controle ao usuário do domínio de AD

1. Abra o snap-in do MMC Usuários e computadores do Active Directory e selecione o domínio em que deseja criar o usuário.
2. Clique com o botão direito do mouse na OU criada anteriormente e escolha Delegar controle.
3. No Assistente de delegação de controle, clique em Próximo.
4. Na seção Usuários ou grupos, clique em Adicionar.
5. Na seção Selecionar usuários, computadores ou grupos, insira o usuário do AD que você criou e clique em Verificar nomes. Se a verificação de usuário do AD for bem-sucedida, clique em OK.
6. Na seção Usuários ou grupos, confirme se seu usuário do AD foi adicionado e clique em Próximo.
7. Na página Tarefas para delegar, selecione Criar uma tarefa personalizada para delegar e escolha Próximo.
8. Na seção Tipo de objeto do Active Directory:
    a. Selecione Somente os objetos a seguir na pasta.
    b. Selecione Objetos do computador.
    c. Selecione Criar objetos selecionados nesta pasta.
    d. Selecione Excluir objetos selecionados nesta pasta e clique em Próximo.
9. Na seção Permissões:
    a. Mantenha a opção Geral selecionada.
    b. Selecione Gravação validada no nome do host DNS.
    c. Selecione Gravação validada no nome da entidade principal de serviço e clique em Próximo.
10. Em Concluir o assistente de delegação de controle, revise e confirme as configurações e clique em Concluir.

Etapa 4: Criar uma chave do AWS KMS

A chave do KMS é usada para criptografar o segredo da AWS.

Para criar uma chave do AWS KMS

> **Note**
>
> Em Chave de criptografia, não use a chave do KMS padrão da AWS. Crie a chave do AWS KMS na mesma conta da AWS que contém a instância de banco de dados do RDS para SQL Server que você deseja associar ao AD autogerenciado.

1. No console do AWS KMS, escolha Criar chave.
2. Em Tipo de chave, escolha Simétrica.
3. Em Uso da chave, escolha Criptografar e descriptografar.
4. Em Advanced options (Opções avançadas):
   a. Em Origem do material de chaves, escolha Externa.
   b. Em Regionalidade, escolha Chave de região única e clique em Próximo.
5. Em Alias, forneça um nome para a chave do KMS.
6. (Opcional) Em Descrição, forneça uma descrição da chave do KMS.
7. (Opcional) Em Etiquetas, forneça uma etiqueta da chave do KMS e clique em Próximo.
8. Em Administradores de chaves, forneça o nome de um usuário do IAM e selecione-o.
9. Em Exclusão de chaves, mantenha a caixa Permitir que administradores de chaves excluam esta chave selecionada e clique em Próximo.
10. Em Usuários de chaves, informe o mesmo usuário do IAM da etapa anterior e selecione-o. Clique em Next.
11. Revise a configuração.
12. Em Política de chave, inclua o seguinte na Instrução da política:

```
{
    "Sid": "Allow use of the KMS key on behalf of RDS",
    "Effect": "Allow",
    "Principal": {
        "Service": [
            "rds.amazonaws.com"
        ]
    },
```

```
    "Action": "kms:Decrypt",
    "Resource": "*"
}
```

13. Clique em Finish (Concluir).

Etapa 5: Criar um segredo da AWS

Como criar um segredo

> **Note**
>
> Crie o segredo na mesma conta da AWS que contém a instância de banco de dados do RDS para SQL Server que você deseja associar ao AD autogerenciado.

1. No AWS Secrets Manager, escolha Armazenar um novo segredo.
2. Em Tipo de segredo, escolha Outro tipo de segredo.
3. Em Pares de chave/valor, adicione suas duas chaves:
   a. Para a primeira chave, insira `CUSTOMER_MANAGED_ACTIVE_DIRECTORY_USERNAME`.
   b. Para o valor da primeira chave, insira somente o nome de usuário (sem o prefixo do domínio) do usuário do AD. Não inclua o nome do domínio, pois isso faz com que a criação da instância falhe.
   c. Para a segunda chave, insira `CUSTOMER_MANAGED_ACTIVE_DIRECTORY_PASSWORD`.
   d. Para o valor da segunda chave, insira a senha que você criou para o usuário do AD no domínio.
4. Em Chave de criptografia, insira a chave do KMS que você criou em uma etapa anterior e clique em Próximo.
5. Em Nome do secreto, insira um nome descritivo que ajude você a encontrar o segredo posteriormente.
6. (Opcional) Em Descrição, insira uma descrição para o nome do segredo.
7. Em Permissão de recurso, clique em Editar.
8. Adicione a política a seguir à política de permissões:

> **Note**
>
> Recomendamos que você use as condições `aws:sourceAccount` e `aws:sourceArn` na política para evitar o problema de representante confuso. Use sua Conta da AWS em `aws:sourceAccount` e o ARN da instância de banco de dados do RDS para SQL Server em `aws:sourceArn`. Para obter mais informações, consulte Prevenção do problema do substituto confuso entre serviços.

JSON

```
{
    "Version": "2012-10-17",
    "Statement":
    [
        {
            "Effect": "Allow",
            "Principal":
            {
                "Service": "rds.amazonaws.com"
            },
            "Action": "secretsmanager:GetSecretValue",
            "Resource": "*",
            "Condition":
            {
                "StringEquals":
                {
                    "aws:sourceAccount": "123456789012"
                },
                "ArnLike":
                {
                    "aws:sourceArn": "arn:aws:rds:us-
west-2:123456789012:db:*"
                }
            }
        }
    ]
}
```

9. Clique em Salvar, depois em Próximo.

10. Em Definir configurações de rotação, mantenha os valores padrão e escolha Próximo.
11. Revise as configurações do segredo e clique em Armazenar.
12. Escolha o segredo que você criou e copie o valor do ARN do segredo. Isso será usado na próxima etapa para configurar o Active Directory autogerenciado.

Etapa 6: Criar ou modificar a instância de banco de dados do SQL Server

Você pode usar o console, a CLI ou a API do RDS para associar uma instância de banco de dados do RDS para SQL Server a um domínio de AD autogerenciado. Você pode fazer isso por meio de uma das seguintes maneiras:

- Crie uma instância de banco de dados do SQL Server usando o console, o comando create-db-instance da CLI ou a operação da API CreateDBInstance do RDS.

  Para obter instruções, consulte Criar uma instância de banco de dados do Amazon RDS.
- Modifique uma instância de banco de dados existente do SQL Server usando o console, o comando modify-db-instance da CLI ou a operação da API ModifyDBInstance do RDS.

  Para obter instruções, consulte Modificar uma instância de banco de dados do Amazon RDS.
- Restaure uma instância de banco de dados do SQL Server de um snapshot de banco de dados usando o console, o comando restore-db-instance-from-db-snapshot da CLI ou a operação da API RestoreDBInstanceFromDBSnapshot do RDS.

  Para obter instruções, consulte Restaurar uma instância de banco de dados.
- Restaure uma instância de banco de dados SQL Server em um determinado momento usando o console, o comando restore-db-instance-to-point-in-time da CLI ou a operação da API RestoreDBInstanceToPointInTime do RDS.

  Para obter instruções, consulte Restaurar uma instância de banco de dados para um momento especificado no Amazon RDS.

Quando você usa a AWS CLI, são necessários os seguintes parâmetros para que a instância de banco de dados possa usar o domínio de Active Directory autogerenciado que você criou:

- Para o parâmetro `--domain-fqdn`, use o nome de domínio totalmente qualificado (FQDN) do seu Active Directory autogerenciado.
- Para o parâmetro `--domain-ou`, use a OU criada em seu AD autogerenciado.

- Para o parâmetro `--domain-auth-secret-arn`, use o valor do ARN do segredo criado em uma etapa anterior.
- Para o parâmetro `--domain-dns-ips`, use os endereços IPv4 primário e secundário dos servidores DNS para seu AD autogerenciado. Se você não tiver um endereço IP de servidor DNS secundário, insira o endereço IP principal duas vezes.

O exemplo a seguir de comandos da CLI mostra como criar, modificar e remover uma instância de banco de dados do RDS para SQL Server com um domínio de AD autogerenciado.

> **Important**
>
> Se você modificar uma instância de banco de dados para associá-la ou removê-la de um domínio de AD autogerenciado, será necessária uma reinicialização da instância de banco de dados para que a modificação entre em vigor. Você pode optar por aplicar as alterações imediatamente ou esperar até a próxima janela de manutenção. Escolher a opção Aplicar imediatamente causará tempo de inatividade para uma instância de banco de dados single-AZ. Uma instância de banco de dados multi-AZ realizará um failover antes de concluir a reinicialização. Para obter mais informações, consulte Usar a configuração de programação de modificações.

O comando da CLI a seguir cria uma instância de banco de dados do RDS para SQL Server e a associa a um domínio de AD autogerenciado.

Para Linux, macOS ou Unix:

```
aws rds create-db-instance \
    --db-instance-identifier my-DB-instance \
    --db-instance-class db.m5.xlarge \
    --allocated-storage 50 \
    --engine sqlserver-se \
    --engine-version 15.00.4043.16.v1 \
    --license-model license-included \
    --master-username my-master-username \
    --master-user-password my-master-password \
    --domain-fqdn my_AD_domain.my_AD.my_domain \
    --domain-ou OU=my-AD-test-OU,DC=my-AD-test,DC=my-AD,DC=my-domain \
    --domain-auth-secret-arn "arn:aws:secretsmanager:region:account-number:secret:my-
AD-test-secret-123456" \
```

```
    --domain-dns-ips "10.11.12.13" "10.11.12.14"
```

Para Windows:

```
aws rds create-db-instance ^
    --db-instance-identifier my-DB-instance ^
    --db-instance-class db.m5.xlarge ^
    --allocated-storage 50 ^
    --engine sqlserver-se ^
    --engine-version 15.00.4043.16.v1 ^
    --license-model license-included ^
    --master-username my-master-username ^
    --master-user-password my-master-password ^
    --domain-fqdn my-AD-test.my-AD.mydomain ^
    --domain-ou OU=my-AD-test-OU,DC=my-AD-test,DC=my-AD,DC=my-domain ^
    --domain-auth-secret-arn "arn:aws:secretsmanager:region:account-number:secret:my-
AD-test-secret-123456" \ ^
    --domain-dns-ips "10.11.12.13" "10.11.12.14"
```

O comando da CLI a seguir modifica uma instância de banco de dados do RDS para SQL Server existente para usar um domínio de Active Directory autogerenciado.

Para Linux, macOS ou Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier my-DB-instance \
    --domain-fqdn my_AD_domain.my_AD.my_domain \
    --domain-ou OU=my-AD-test-OU,DC=my-AD-test,DC=my-AD,DC=my-domain \
    --domain-auth-secret-arn "arn:aws:secretsmanager:region:account-number:secret:my-
AD-test-secret-123456" \
    --domain-dns-ips "10.11.12.13" "10.11.12.14"
```

Para Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier my-DBinstance ^
    --domain-fqdn my_AD_domain.my_AD.my_domain ^
    --domain-ou OU=my-AD-test-OU,DC=my-AD-test,DC=my-AD,DC=my-domain ^
    --domain-auth-secret-arn "arn:aws:secretsmanager:region:account-number:secret:my-
AD-test-secret-123456" ^
    --domain-dns-ips "10.11.12.13" "10.11.12.14"
```

O comando da CLI a seguir remove uma instância de banco de dados do RDS para SQL Server de um domínio de Active Directory autogerenciado.

Para Linux, macOS ou Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier my-DB-instance \
    --disable-domain
```

Para Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier my-DB-instance ^
    --disable-domain
```

Etapa 7: Criar logins do SQL Server de Autenticação do Windows

Use as credenciais de usuário mestre do Amazon RDS para se conectar à instância de banco de dados do SQL Server como você faria para qualquer outra instância de banco de dados. Como a instância de banco de dados é associada ao domínio de AD autogerenciado, você pode provisionar logins e usuários do SQL Server. Você faz isso usando o utilitário de usuários e grupos de AD em seu domínio de AD autogerenciado. As permissões de banco de dados são gerenciadas por meio de permissões padrão do SQL Server concedidas e revogadas a esses logins do Windows.

Para que um usuário de AD autogerenciado se autentique com o SQL Server, deve existir um login do Windows SQL Server para o usuário de AD autogerenciado ou um grupo de Active Directory autogerenciado do qual o usuário é membro. O controle de acesso refinado é gerenciado por meio da concessão e revogação de permissões nesses logins do SQL Server. Um usuário de AD autogerenciado que não tem um login do SQL Server ou que pertence a um grupo de AD autogerenciado ao qual um login não consegue obter acesso à instância de banco de dados do SQL Server.

A permissão ALTER ANY LOGIN é necessária para criar um login de AD autogerenciado do SQL Server. Se você ainda não criou logins com essa permissão, conecte-se como o usuário mestre da instância de banco de dados usando a autenticação do SQL Server e crie logins de AD autogerenciado do SQL Server no contexto do usuário principal.

Você pode executar um comando de linguagem de definição de dados (DDL) como o seguinte para criar um login do SQL Server para um usuário ou grupo de AD autogerenciado.

> **Note**
>
> Especifique usuários e grupos que usam o nome de login anterior ao Windows 2000 no formato *my_AD_domain**my_AD_domain_user*. Não é possível usar um User Principal Name (UPN – Nome de usuário principal) no formato *my_AD_domain_user*@*my_AD_domain*.

```
USE [master]
GO
CREATE LOGIN [my_AD_domain\my_AD_domain_user] FROM WINDOWS WITH DEFAULT_DATABASE =
 [master], DEFAULT_LANGUAGE = [us_english];
GO
```

Para obter mais informações, consulte CREATE LOGIN (Transact-SQL) na documentação da Microsoft Developer Network.

Os usuários (humanos e aplicações) do seu domínio agora podem se conectar à instância do RDS para SQL Server por meio de uma máquina cliente associada ao domínio de AD autogerenciado usando a autenticação do Windows.

Gerenciar uma instância de banco de dados em um domínio de Active Directory autogerenciado

É possível usar o console, a AWS CLI ou a API do Amazon RDS para gerenciar a instância de banco de dados e a respectiva relação com o domínio de AD autogerenciado. Por exemplo, é possível mover a instância de banco de dados para dentro, para fora, de e entre os domínios.

Por exemplo, usando a API do Amazon RDS, você pode fazer o seguinte:

- Para tentar novamente uma associação a um domínio de AD autogerenciado em caso de falha, use a operação de API ModifyDBInstance e especifique o mesmo conjunto de parâmetros:
  - `--domain-fqdn`
  - `--domain-dns-ips`
  - `--domain-ou`
  - `--domain-auth-secret-arn`
- Para remover uma instância de banco de dados de um domínio autogerenciado, use a operação de API `ModifyDBInstance` e especifique `--disable-domain` como parâmetro do domínio.

- Para mover uma instância de banco de dados de um domínio autogerenciado para outro, use a operação de API `ModifyDBInstance` e especifique os parâmetros do novo domínio:
  - `--domain-fqdn`
  - `--domain-dns-ips`
  - `--domain-ou`
  - `--domain-auth-secret-arn`
- Para listar a associação de cada instância de banco de dados ao domínio de AD autogerenciado, use a operação de API DescribeDBInstances.

Entender a associação a um domínio de Active Directory autogerenciado

Depois de criar ou modificar uma instância de banco de dados, a instância se tornará um membro do domínio de AD autogerenciado. O console da AWS indica o status da associação ao domínio de AD autogerenciado para a instância de banco de dados. O status da instância de banco de dados pode ser um dos seguintes:

- joined: a instância é membro do domínio de AD.
- joining: a instância está em processo de se tornar membro do domínio de AD.
- pending-join – a associação da instância está pendente.
- pending-maintenance-join: a AWS tentará tornar a instância um membro do domínio de AD durante a próxima janela de manutenção agendada.
- pending-removal: a remoção da instância do domínio de AD está pendente.
- pending-maintenance-removal: a AWS tentará remover a instância do domínio de AD durante a próxima janela de manutenção agendada.
- failed: um problema de configuração impediu que a instância se associasse ao domínio de AD. Verifique e corrija sua configuração antes de emitir novamente o comando de modificação da instância.
- removing: a instância está sendo removida do domínio de AD.

Uma solicitação para se tornar um membro de um domínio de AD autogerenciado pode falhar devido a um problema de conectividade de rede. Por exemplo, você pode conseguir criar uma instância de banco de dados ou modificar uma instância existente, mas não conseguir transformar a instância de banco de dados em um membro de um domínio de AD autogerenciado. Nesse caso,

execute novamente o comando para criar ou modificar a instância de banco de dados ou modifique a instância recém-criada para associá-la ao domínio de AD autogerenciado.

Solução de problemas de Active Directory autogerenciado

Veja a seguir alguns problemas que você pode encontrar ao configurar ou modificar um AD autogerenciado.

Código de erro	Descrição	Causas comuns	Sugestões de solução de problemas
Erro 2 / 0x2	O sistema não conseguiu encontrar o arquivo especificado.	O formato ou a localização da unidade organizacional (OU) especificada com o parâmetro –`domain-ou` é inválido. A conta de serviço do domínio especificada por meio do AWS Secrets Manager não tem as permissões necessárias para se associar à OU.	Revise o parâmetro –`domain-ou`. Certifique-se de que a conta de serviço do domínio tenha as permissões corretas para a OU. Para ter mais informações, consulte Configurar uma conta de serviço do domínio de AD.
Erro 5 / 0x5	Acesso negado.	Permissões configuradas incorretamente para a conta de serviço do domínio, ou a conta de computador já existe no domínio.	Revise as permissões da conta de serviço no domínio e verifique se a conta de computador do RDS não está duplicada no domínio. Você pode verificar o nome da conta de computador do RDS executando `SELECT @@SERVERNAME` em sua instância de banco de dados do RDS para SQL Server. Se você estiver usando multi-AZ, tente reinicializar com failover,

Código de erro	Descrição	Causas comuns	Sugestões de solução de problemas
			depois verifique a conta de computador do RDS novamente. Para ter mais informações, consulte Reinicializar uma instância de banco de dados .
Erro 87 / 0x57	O parâmetro está incorreto.	A conta de serviço do domínio especificada por meio do AWS Secrets Manager não tem as permissões corretas. O perfil do usuário também pode estar corrompido.	Revise os requisitos da conta de serviço do domínio. Para ter mais informações, consulte Configurar uma conta de serviço do domínio de AD.
Erro 234 / 0xEA	A unidade organizacional (OU) especificada não existe.	A OU especificada com o parâmetro –`domain-ou` não existe em seu AD autogerenciado.	Revise o parâmetro –`domain-ou` e verifique se a OU especificada existe em seu AD autogerenciado.
Erro 1326 / 0x52E	O nome de usuário ou a senha estão incorretos.	As credenciais da conta de serviço do domínio fornecidas no AWS Secrets Manager contêm um nome de usuário desconhecido ou uma senha incorreta. A conta de domínio também pode estar desabilitada em seu AD autogerenciado.	Verifique se as credenciais fornecidas no AWS Secrets Manager estão corretas e se a conta de domínio está habilitada no Active Directory autogerenciado.

Código de erro	Descrição	Causas comuns	Sugestões de solução de problemas
Erro 1355 / 0x54B	O domínio especificado não existe ou não foi possível estabelecer conexão.	O domínio está inativo, o conjunto especificado de IPs DNS está inacessível ou o FQDN especificado está inacessível.	Revise os parâmetros `–domain-dns-ips` e `–domain-fqdn` para garantir que estejam corretos. Revise a configuração de rede da sua instância de banco de dados do RDS para SQL Server e garanta que seu AD autogerenciado esteja acessível. Para ter mais informações, consulte Configurar sua conectividade de rede.
Erro 1722 / 0x6BA	O servidor RPC não está disponível.	Houve um problema ao acessar o serviço RPC do domínio de AD. Isso pode ser devido a um problema de serviço ou rede.	Valide se o serviço RPC está sendo executado nos controladores de domínio e se as portas TCP `135` e `49152-65535` em seu domínio podem ser acessadas por sua instância de banco de dados do RDS para SQL Server.

Código de erro	Descrição	Causas comuns	Sugestões de solução de problemas
Erro 2224 / 0x8B0	A conta de usuário já existe.	A conta de computador que está tentando ser adicionada ao AD autogerenciado já existe.	Identifique a conta de computador executando o `SELECT @@SERVERNAME` em sua instância de banco de dados do RDS para SQL Server, depois remova-a cuidadosamente do AD autogerenciado.
Erro 2242 / 0x8c2	A senha deste usuário expirou.	A senha da conta de serviço do domínio especificada por meio do AWS Secrets Manager expirou.	Atualize a senha da conta de serviço do domínio usada para associar a instância de banco de dados do RDS para SQL Server ao seu AD autogerenciado.

Restaurar uma instância de banco de dados do SQL Server e adicioná-la a um domínio de Active Directory autogerenciado

Você pode restaurar um snapshot de banco de dados ou fazer uma recuperação para um ponto no tempo (PITR) de uma instância de banco de dados do SQL Server e adicioná-la a um domínio de Active Directory autogerenciado. Depois que a instância de banco de dados tiver sido restaurada, modifique-a usando o processo explicado em Etapa 6: Criar ou modificar a instância de banco de dados do SQL Server para adicioná-la a um domínio de AD autogerenciado.

Trabalhar com o AWS Managed Active Directory com o RDS para SQL Server

Você pode usar AWS Managed Microsoft AD com a autenticação do Windows para autenticar usuários quando se conectam à instância de banco de dados do RDS para SQL Server. A instância de banco de dados funciona com AWS Directory Service for Microsoft Active Directory, também chamado de AWS Managed Microsoft AD, para habilitar a Autenticação do Windows. Quando os usuários se autenticam com uma instância de banco de dados do SQL Server unida ao domínio confiável, as solicitações de autenticação são encaminhadas para o diretório de domínio que você criou com o AWS Directory Service.

Disponibilidade de região e versão

O RDS só aceita o uso de AWS Managed Microsoft AD para autenticação do Windows. O RDS não oferece suporte ao uso de AD Connector. Para obter mais informações, consulte:

- Política de compatibilidade de aplicações do AWS Managed Microsoft AD
- Política de compatibilidade de aplicações do AD Connector

Para obter informações sobre a disponibilidade de versões e regiões, consulte Autenticação de Kerberos com RDS para PostgreSQL.

Visão geral da configuração de autenticação do Windows

O Amazon RDS usa o modo misto para a autenticação do Windows. Essa abordagem significa que o usuário mestre (o nome e a senha usados para criar sua instância de banco de dados do SQL Server) usa a autenticação do SQL. Como a conta de usuário mestre é uma credencial privilegiada, você deve restringir o acesso a essa conta.

Para obter a autenticação do Windows usando um Microsoft Active Directory no local ou auto-hospedado, crie uma confiança de floresta. A confiança pode ser unidirecional ou bidirecional. Para obter mais informações sobre como configurar confianças de floresta usando o AWS Directory Service, consulte Quando criar um relacionamento de confiança no Guia de administração do AWS Directory Service.

Para configurar a autenticação do Windows para uma instância de banco de dados do SQL Server, realize as seguintes etapas, explicadas mais detalhadamente em Configuração da autenticação do Windows para instâncias de banco de dados do SQL Server:

1. Use AWS Managed Microsoft AD, seja no AWS Management Console ou na API do AWS Directory Service, para criar um diretório AWS Managed Microsoft AD.
2. Se você usar a AWS CLI ou a API do Amazon RDS para criar a instância de banco de dados do SQL Server, crie uma função do IAM (AWS Identity and Access Management). Essa função usa a política gerenciada do IAM `AmazonRDSDirectoryServiceAccess` e permite que o Amazon RDS realize chamadas para o diretório. Se você usa o console para criar a instância de banco de dados do SQL Server, a AWS cria a função do IAM para você.

   Para a função permitir o acesso, o endpoint do AWS Security Token Service (AWS STS) deve estar ativado na região da AWS da sua conta da AWS. Os endpoints do AWS STS são ativados por padrão em todas as regiões da AWS e você pode usá-los sem ter que tomar medidas adicionais. Para ter mais informações, consulte Gerenciar o AWS STS em uma Região da AWS no Guia do usuário do IAM.
3. Crie e configure usuários e grupos no diretório do AWS Managed Microsoft AD usando as ferramentas do Microsoft Active Directory. Para obter mais informações sobre como criar usuários e grupos no Active Directory, consulte Gerenciar usuários e grupos no AWS Managed Microsoft AD no Guia de administração do AWS Directory Service.
4. Se você planeja localizar o diretório e a instância de banco de dados em VPCs diferentes, habilite o tráfego entre VPCs.
5. Use o Amazon RDS para criar uma instância de banco de dados do SQL Server a partir do console, da AWS CLI ou da API do Amazon RDS. Na solicitação de criação, forneça o identificador de domínio (identificador "`d-*`") que foi gerado quando você criou seu diretório e o nome da função que você criou. Também é possível modificar uma instância de banco de dados do SQL Server para usar a Autenticação do Windows definindo o domínio e os parâmetros da função do IAM para a instância de banco de dados.
6. Use as credenciais de usuário mestre do Amazon RDS para se conectar à instância de banco de dados do SQL Server como você faria com qualquer outra instância de banco de dados. Como a instância de banco de dados é unida ao domínio do AWS Managed Microsoft AD, você pode provisionar logins e usuários do SQL Server dos usuários e grupos do Active Directory em seu domínio. (Conhecidos como logins “Windows” do SQL Server.) As permissões de banco de dados são gerenciadas por meio de permissões padrão do SQL Server concedidas e revogadas a esses logins do Windows.

Quando você cria uma instância de banco de dados RDS para SQL Server conectada ao domínio usando o console do Amazon RDS, a AWS cria automaticamente o perfil do IAM `rds-`

`directoryservice-access-role`. Esse perfil é essencial para gerenciar instâncias conectadas ao domínio e é necessário para as seguintes operações:

- Fazer alterações de configuração em instâncias do SQL Server conectadas ao domínio
- Gerenciar configurações de integração do Active Directory
- Executar operações de manutenção em instâncias associadas ao domínio

> **Important**
>
> Se você excluir o perfil do IAM `rds-directoryservice-access-role`, não poderá fazer alterações na instância do SQL Server conectada ao domínio por meio do console ou da API do Amazon RDS. A tentativa de modificar a instância resulta em uma mensagem de erro informando: You don't have permission to iam:CreateRole. To request access, copy the following text and send it to your AWS administrator.
> Esse erro ocorre porque o Amazon RDS precisa recriar o perfil para gerenciar a conexão do domínio, mas não tem as permissões necessárias. Além disso, esse erro não está registrado em log no CloudTrail, o que pode dificultar a solução de problemas.

Se você excluir acidentalmente o `rds-directoryservice-access-role`, só poderá fazer quaisquer alterações na instância do SQL Server conectada ao domínio se tiver permissões `iam:CreateRole` para recriá-lo. Para recriar o perfil manualmente, ele deve ter a política gerenciada `AmazonRDSDirectoryServiceAccess` anexada e a relação de confiança apropriada que permita que o serviço RDS assuma o perfil.

Criar o endpoint para autenticação do Kerberos

A autenticação baseada em Kerberos exige que o endpoint seja o nome do host especificado pelo cliente, um ponto e o nome de domínio totalmente qualificado (FQDN). Por exemplo, o exemplo a seguir mostra um endpoint que você pode usar com a autenticação baseada em Kerberos. Neste exemplo, o nome do host da instância de banco de dados do SQL Server é `ad-test` e o nome de domínio é `corp-ad.company.com`:

```
ad-test.corp-ad.company.com
```

Se você deseja verificar se sua conexão está usando o Kerberos, execute a seguinte consulta:

```
SELECT net_transport, auth_scheme
  FROM sys.dm_exec_connections
 WHERE session_id = @@SPID;
```

Configuração da autenticação do Windows para instâncias de banco de dados do SQL Server

Use o AWS Directory Service for Microsoft Active Directory, também chamado de AWS Managed Microsoft AD, para configurar a Autenticação do Windows para uma instância de banco de dados do SQL Server. Para configurar a Autenticação do Windows, execute as seguintes etapas:

Etapa 1: Criar um diretório usando o AWS Directory Service for Microsoft Active Directory

O AWS Directory Service cria um Microsoft Active Directory totalmente gerenciado na Nuvem AWS. Ao criar um diretório do AWS Managed Microsoft AD, o AWS Directory Service cria dois controladores de domínio e servidores do Domain Name Service (DNS) em seu nome. Os servidores do diretório são criados em duas sub-redes em duas zonas de disponibilidade diferentes com uma VPC. Essa redundância ajuda a garantir que o diretório permaneça acessível mesmo se ocorrer uma falha.

Ao criar um diretório do AWS Managed Microsoft AD, o AWS Directory Service executa as seguintes tarefas em seu nome:

- Configura um Microsoft Active Directory dentro da VPC.
- Cria uma conta de administrador do diretório com o nome de usuário Admin e a senha especificada. Use essa conta para gerenciar seu diretório.
- Cria um grupo de segurança para os controladores do diretório.

Quando um AWS Directory Service for Microsoft Active Directory é iniciado, a AWS cria uma Unidade Organizacional (UO) que contém todos os objetos do diretório. Essa OU, que tem o nome de NetBIOS que você digitou quando criou seu diretório, está localizada na raiz do domínio. A raiz do domínio é controlada e de propriedade da AWS.

A conta admin que foi criada com o diretório do AWS Managed Microsoft AD tem permissões para as atividades administrativas mais comuns de sua OU:

- Criar atualização ou excluir usuários, grupos e computadores.

- Adicione recursos ao seu domínio, como servidores de arquivos ou de impressão e atribua permissões para esses recursos a usuários e grupos em sua OU.
- Criar OUs adicionais e contêineres.
- Delegar autoridade.
- Criar e vincular políticas de grupo.
- Restaurar objetos excluídos da Lixeira do Active Directory.
- Execute os módulos AD e DNS do Windows PowerShell no Active Directory Web Service.

A conta admin também possui direitos para executar as seguintes atividades de domínio:

- Gerenciar configurações de DNS (adicionar, remover ou atualizar registros, zonas e encaminhadores).
- Visualizar logs de eventos de DNS.
- Visualizar logs de eventos de segurança.

Como criar um diretório com AWS Managed Microsoft AD

1. No painel de navegação do console do AWS Directory Service, escolha Directories (Diretórios) e selecione Set up directory (Configurar diretório).
2. Selecione AWS Managed Microsoft AD. Essa é a única opção compatível atualmente para uso com o Amazon RDS.
3. Escolha Next (Próximo).
4. Na página Enter directory information (Inserir informações do diretório), forneça as seguintes informações:

   Edição

   Escolha a edição que atenda às suas necessidades.

   Nome do DNS do diretório

   O nome completo do diretório, como `corp.example.com`. Nomes com mais de 47 caracteres não são compatíveis com o SQL Server.

   Nome de NetBIOS do diretório

   O nome curto opcional do diretório, como `CORP`.

Descrição do diretório

Uma descrição opcional do diretório.

Senha do Admin

A senha do administrador do diretório. O processo de criação do diretório cria uma conta de administrador com o nome de usuário Admin e essa senha.

A senha do administrador do diretório não pode incluir a palavra `admin`. A senha diferencia letras maiúsculas de minúsculas e deve ter entre 8 e 64 caracteres. Ela também precisa conter pelo menos um caractere de três das quatro categorias a seguir:

- Letras minúsculas (a-z)
- Letras maiúsculas (A-Z)
- Números (0-9)
- Caracteres não alfanuméricos (~!@#$%^&*_-+=`|\(){}[]:;'"<>,.?/)

Confirmar senha

Digite a senha do administrador novamente.

5. Escolha Next (Próximo).
6. Na página Choose VPC and subnets (Selecionar VPC e sub-redes), forneça as seguintes informações:

VPC

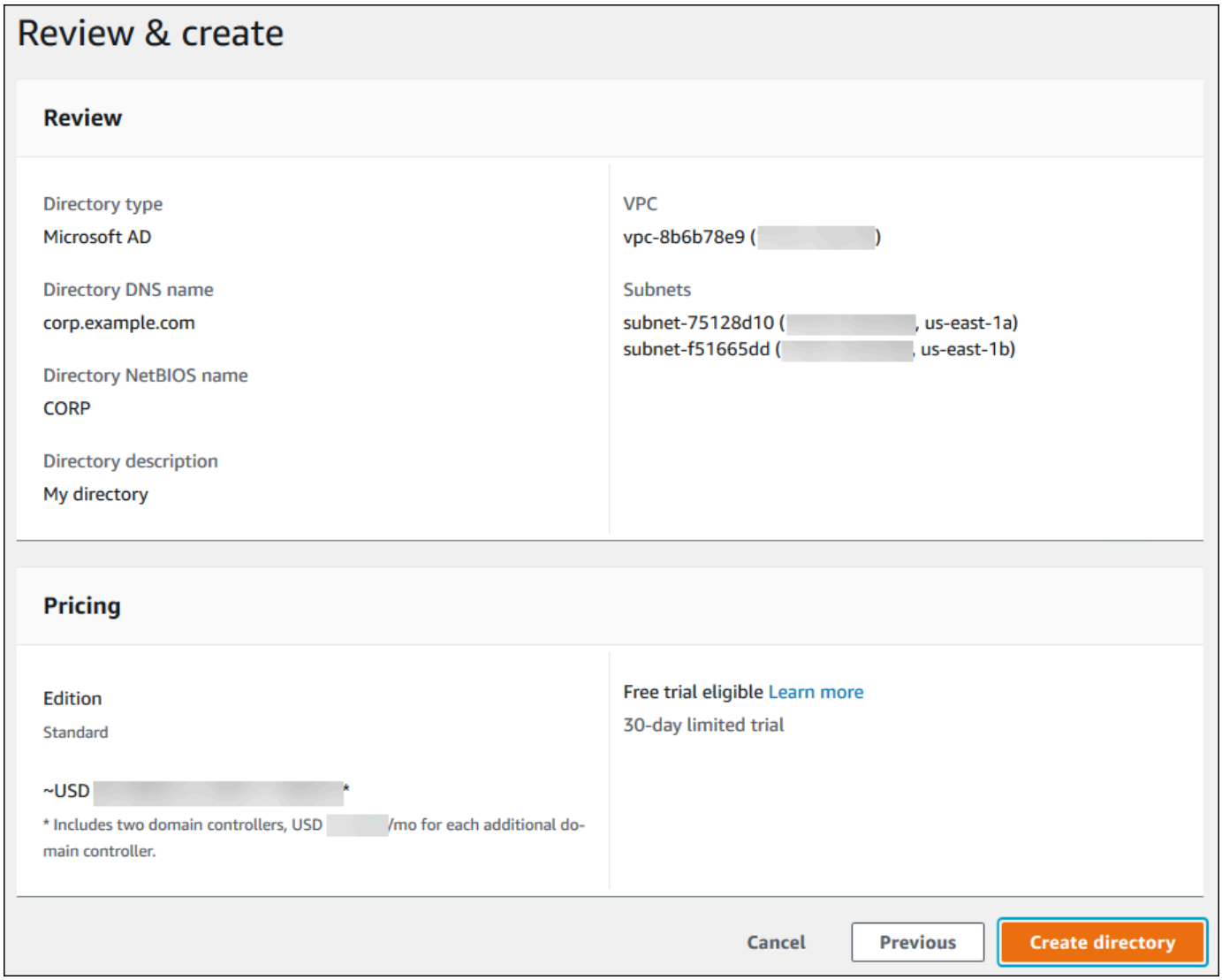
Escolha a VPC do diretório.

> **Note**
>
> É possível localizar o diretório e a instância de banco de dados em VPCs diferentes, mas se o fizer, certifique-se de habilitar o tráfego entre VPCs. Para obter mais informações, consulte Etapa 4: Ativar o tráfego entre VPCs entre o diretório e a instância de banco de dados.

Sub-redes

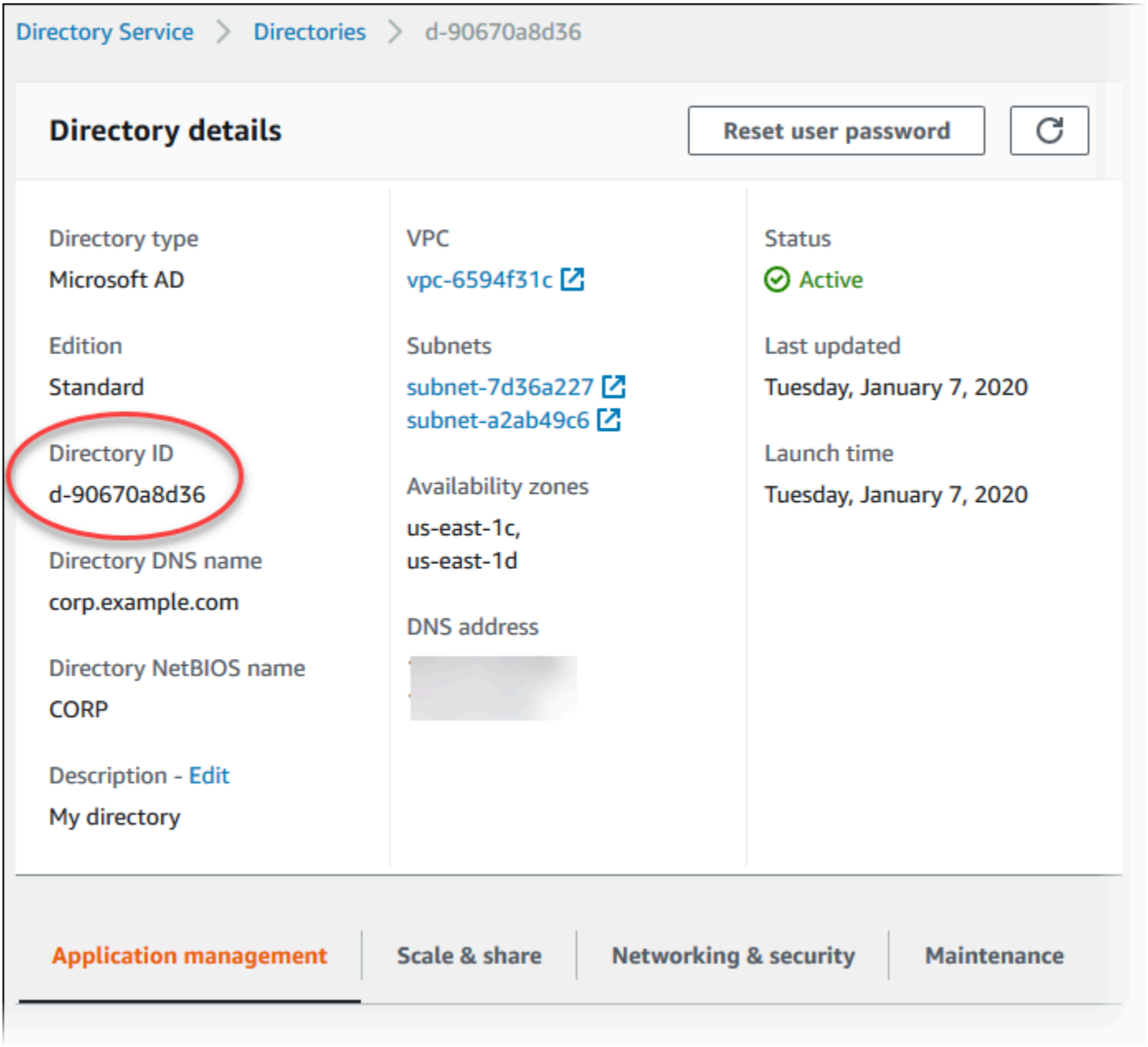
Escolha as sub-redes para os servidores do diretório. As duas sub-redes deve estar em diferentes zonas de disponibilidade.

7. Escolha Next (Próximo).
8. Analise as informações do diretório. Se alterações forem necessárias, escolha Previous (Anterior). Quando as informações estiverem corretas, selecione Create directory (Criar diretório).



A criação do diretório leva vários minutos. Depois que o diretório tiver sido criado com sucesso, o valor de Status muda para Active (Ativo).

Para visualizar informações sobre o diretório, selecione o ID do diretório na listagem de diretórios. Anote o Directory ID (ID do diretório). Esse valor será necessário ao criar ou modificar sua instância de banco de dados do SQL Server.



Etapa 2: Criar a função do IAM para ser usada pelo Amazon RDS

Se usar o console para criar sua instância de banco de dados do SQL Server, você pode pular esta etapa. Se você usar a CLI ou a API do RDS para criar sua instância de banco de dados do SQL Server, você deverá criar uma função do IAM que use a política `AmazonRDSDirectoryServiceAccess` gerenciada do IAM. Esta função permite que o Amazon RDS faça chamadas ao AWS Directory Service para você.

Se você estiver usando uma política personalizada para ingressar em um domínio, em vez de usar a política AWS-managed `AmazonRDSDirectoryServiceAccess` gerenciada, permita a ação `ds:GetAuthorizedApplicationDetails`. Esse requisito entrará em vigor a partir de julho de 2019, devido a uma alteração na API do AWS Directory Service.

A seguinte política `AmazonRDSDirectoryServiceAccess` do IAM, fornece acesso ao AWS Directory Service.

Example Política do IAM para fornecer acesso a AWS Directory Service

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
            "ds:DescribeDirectories",
            "ds:AuthorizeApplication",
            "ds:UnauthorizeApplication",
            "ds:GetAuthorizedApplicationDetails"
        ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

Recomendamos usar as chaves de contexto de condição global `aws:SourceArn` e `aws:SourceAccount` em relações baseadas em recursos para limitar as permissões do serviço a um recurso específico. Essa é a maneira mais eficiente de se proteger contra o problema "confused deputy".

Você pode usar as duas chaves de contexto de condição global e fazer com que o valor `aws:SourceArn` contenha o ID da conta. Nesses casos, verifique se o valor `aws:SourceAccount` e a conta no `aws:SourceArn` usa o mesmo ID de conta quando eles são usados na mesma instrução.

- Use `aws:SourceArn` se quiser acesso entre serviços para um único recurso.

- Use `aws:SourceAccount` se você quiser permitir que qualquer recurso nessa conta seja associado ao uso entre serviços.

Na relação de confiança, certifique-se de usar a chave de contexto de condição global `aws:SourceArn` com o nome do recurso da Amazon (ARN) completo dos recursos que acessam a função. Para Windows Authentication, certifique-se de incluir as instâncias de banco de dados conforme mostrado no exemplo a seguir.

Example relação de confiança com a chave de contexto de condição global para Windows Authentication

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "rds.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceArn": [

 "arn:aws:rds:Region:my_account_ID:db:db_instance_identifier"
                    ]
                }
            }
        }
    ]
}
```

Crie uma função do IAM usando esta política do IAM e relacionamento de confiança. Para obter mais informações sobre como criar funções do IAM, consulte Criação de políticas gerenciadas pelo cliente no Guia do usuário do IAM.

Etapa 3: Criar e configurar usuários e grupos

É possível criar usuários e grupos com a ferramenta Usuários e computadores do Active Directory. Essa ferramenta é uma das ferramentas do Active Directory Domain Services e do Active Directory Lightweight Directory Services. Os usuários representam pessoas ou entidades individuais que têm acesso ao seu diretório. Os grupos são muito úteis para conceder ou negar privilégios a grupos de usuários, em vez de ter que aplicar esses privilégios a cada usuário individual.

Para criar usuários e grupos em um diretório do AWS Directory Service, é necessário estar conectado a uma instância do EC2 do Windows que sejam membro do diretório do AWS Directory Service. Também é necessário estar conectado como um usuário com privilégios para criar usuários e grupos. Para obter mais informações, consulte Adicionar usuários e grupos (Simple AD e AWS Managed Microsoft AD) no Guia de administração do AWS Directory Service.

Etapa 4: Ativar o tráfego entre VPCs entre o diretório e a instância de banco de dados

Se você planeja localizar o diretório e a instância de banco de dados na mesma VPC, ignore esta etapa e prossiga para Etapa 5: Criar ou modificar a instância de banco de dados do SQL Server.

Se você planejar localizar o diretório e a instância de Bancos de Dados em VPCs diferentes, configure o tráfego entre VPCs usando o emparelhamento de VPCs ou o AWS Transit Gateway.

O procedimento a seguir habilita o tráfego entre VPCs usando o emparelhamento de VPCs. Siga as instruções em O que é o emparelhamento de VPCs? no Guia de emparelhamento do Amazon Virtual Private Cloud.

Como habilitar o tráfego entre VPCs usando o emparelhamento de VPCs

1. Configure regras apropriadas de roteamento de VPC para garantir que o tráfego de rede possa fluir em ambos os sentidos.
2. Certifique-se de que o grupo de segurança da instância de banco de dados possa receber o tráfego de entrada do grupo de segurança do diretório.
3. Garanta que não há nenhuma regra na lista de controle de acesso (ACL) de rede para bloquear o tráfego.

Se uma conta diferente da AWS for proprietária do diretório, é necessário compartilhá-lo.

Como compartilhar o diretório entre contas da AWS.

1. Inicie o compartilhamento do diretório com a conta da AWS na qual a instância de banco de dados será criada seguindo as instruções em Tutorial: Compartilhar o diretório AWS Managed Microsoft AD para ingresso perfeito no domínio do EC2 no AWS Directory ServiceGuia de administração.
2. Faça login no console do AWS Directory Service usando a conta para a instância de banco de dados e garanta que o domínio tenha o status `SHARED` antes de prosseguir.
3. Enquanto estiver conectado ao console do AWS Directory Service usando a conta da instância de banco de dados, observe o valor do Directory ID (ID do diretório). Use esse ID do diretório para associar a instância de banco de dados ao domínio.

Etapa 5: Criar ou modificar a instância de banco de dados do SQL Server

Crie ou modifique uma instância de banco de dados do SQL Server para usar com o diretório. É possível usar o console, a CLI ou a API do RDS para associar uma instância de banco de dados a um diretório. Você pode fazer isso por meio de uma das seguintes maneiras:

- Crie uma instância de banco de dados do SQL Server usando o console, o comando create-db-instance da CLI ou a operação da API CreateDBInstance do RDS.

  Para obter instruções, consulte Criar uma instância de banco de dados do Amazon RDS.
- Modifique uma instância de banco de dados existente do SQL Server usando o console, o comando modify-db-instance da CLI ou a operação da API ModifyDBInstance do RDS.

  Para obter instruções, consulte Modificar uma instância de banco de dados do Amazon RDS.
- Restaure uma instância de banco de dados do SQL Server de um snapshot de banco de dados usando o console, o comando restore-db-instance-from-db-snapshot da CLI ou a operação da API RestoreDBInstanceFromDBSnapshot do RDS.

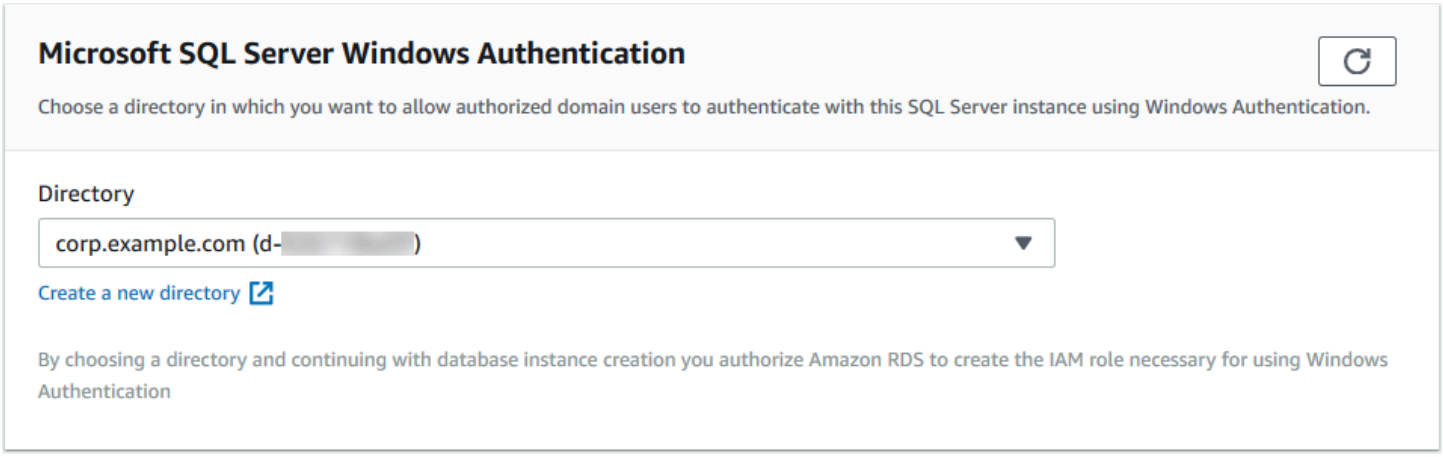
  Para obter instruções, consulte Restaurar uma instância de banco de dados.
- Restaure uma instância de banco de dados SQL Server em um determinado momento usando o console, o comando restore-db-instance-to-point-in-time da CLI ou a operação da API RestoreDBInstanceToPointInTime do RDS.

  Para obter instruções, consulte Restaurar uma instância de banco de dados para um momento especificado no Amazon RDS.

A Autenticação do Windows é compatível somente com instâncias de banco de dados do SQL Server em uma VPC.

Para que a instância de banco de dados possa usar o diretório do domínio criado, o seguinte é necessário:

- Para Directory (Diretório), escolha o identificador de domínio (d-*ID*) gerado ao criar o diretório.
- Verifique se o grupo de segurança da VPC tem uma regra de saída que permite que a instância de banco de dados se comunique com o diretório.



Quando você usa a AWS CLI, são necessários os seguintes parâmetros para que a instância de banco de dados possa usar o diretório criado:

- Para o parâmetro `--domain`, use o identificador de domínio (identificador "d-*ID*") gerado quando o diretório foi criado.
- Para o parâmetro `--domain-iam-role-name`, use a função criada que usa a política gerenciada `AmazonRDSDirectoryServiceAccess` do IAM.

Por exemplo, o comando da CLI a seguir modifica uma instância de banco de dados para usar um diretório.

Para Linux, macOS ou Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --domain d-ID \
    --domain-iam-role-name role-name
```

Para Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --domain d-ID ^
    --domain-iam-role-name role-name
```

> **Important**
>
> Se você modificar uma instância de banco de dados para permitir a autenticação Kerberos, reinicialize a instância de banco de dados após fazer a alteração.

Etapa 6: Criar logins do SQL Server de Autenticação do Windows

Use as credenciais de usuário mestre do Amazon RDS para se conectar à instância de banco de dados do SQL Server como você faria com qualquer outra instância de banco de dados. Como a instância de banco de dados é associada ao domínio AWS Managed Microsoft AD, você pode provisionar logins e usuários do SQL Server. É possível fazer isso a partir de usuários e grupos do Active Directory no domínio. As permissões de banco de dados são gerenciadas por meio de permissões padrão do SQL Server concedidas e revogadas a esses logins do Windows.

Para que um usuário do Active Directory faça a autenticação com o SQL Server, deve existir um login Windows do SQL Server para o usuário ou um grupo do qual o usuário é membro. O controle de acesso refinado é gerenciado por meio da concessão e revogação de permissões nesses logins do SQL Server. Um usuário que não tem um login do SQL Server ou pertence a um grupo ao qual um login não consegue obter acesso à instância de banco de dados do SQL Server.

A permissão ALTER ANY LOGIN é necessária para criar um login do Active Directory SQL Server. Se você ainda não criou logins com essa permissão, conecte-se como o usuário mestre da instância de banco de dados usando a autenticação do SQL Server.

Execute um comando de Data Definition Language (DDL – Linguagem de definição de dados) para criar um login do SQL Server para um usuário ou grupo do Active Directory.

> **Note**
>
> Especifique usuários e grupos que usam o nome de login anterior ao Windows 2000 no formato *domainName**login_name*. Não é possível usar um User Principal Name (UPN – Nome de usuário principal) no formato *login_name*@*DomainName*.
> Você só pode criar um login de autenticação do Windows em uma instância do RDS para SQL Server usando instruções T-SQL. Não é possível usar o SQL Server Management Studio para criar um login de autenticação do Windows.

```
USE [master]
GO
CREATE LOGIN [mydomain\myuser] FROM WINDOWS WITH DEFAULT_DATABASE = [master],
 DEFAULT_LANGUAGE = [us_english];
GO
```

Para obter mais informações, consulte CREATE LOGIN (Transact-SQL) na documentação da Microsoft Developer Network.

Os usuários (humanos e aplicações) do seu domínio agora podem se conectar à instância do RDS for SQL Server a partir de uma máquina cliente conectada ao domínio usando a Autenticação do Windows.

Gerenciamento de uma instância de banco de dados em um domínio

É possível usar o console, a AWS CLI ou a API do Amazon RDS para gerenciar a instância de banco de dados e a respectiva relação no domínio. Por exemplo, é possível mover a instância de banco de dados para dentro, para fora, de e entre os domínios.

Por exemplo, usando a API do Amazon RDS, você pode fazer o seguinte:

- Para tentar uma união de domínio novamente cuja associação falhou, use a operação ModifyDBInstance da API e especifique o ID do diretório da associação atual.
- Para atualizar o nome da função do IAM para a associação, use a operação `ModifyDBInstance` da API e especifique o ID do diretório da associação atual e a nova função do IAM.
- Para remover uma instância de banco de dados de um domínio, use a operação `ModifyDBInstance` da API e especifique `none` como o parâmetro do domínio.

- Para mover uma instância de banco de dados de um domínio para outro, use a operação `ModifyDBInstance` da API e especifique o identificador do novo domínio como o parâmetro do domínio.
- Para listar a associação de cada instância de banco de dados, use a operação DescribeDBInstances da API.

Compreensão da associação de domínio

Após criar ou modificar sua instância de banco de dados, a instância se tornará um membro do domínio. O console da AWS indica o status da associação de domínio para a instância de banco de dados. O status da instância de banco de dados pode ser um dos seguintes:

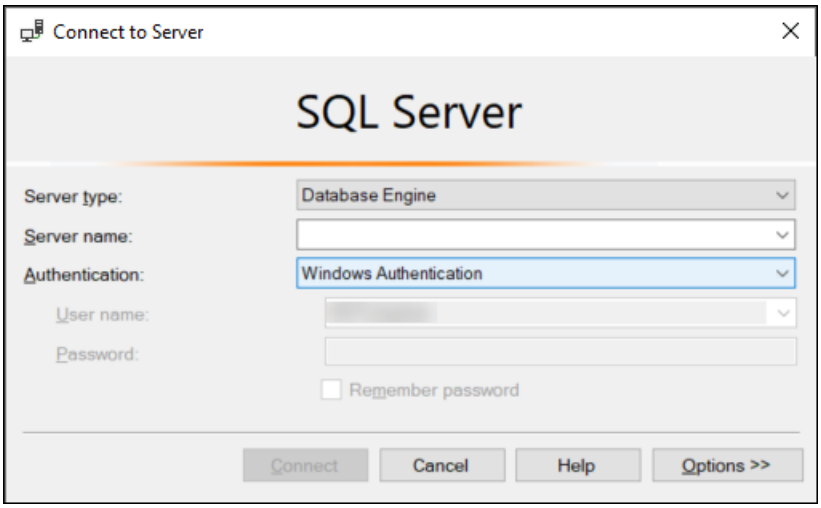
- joined – a instância é membro do domínio.
- joining – a instância está em processo de se tornar membro do domínio.
- pending-join – a associação da instância está pendente.
- pending-maintenance-join: a AWS tentará tornar a instância um membro do domínio durante a próxima janela de manutenção agendada.
- pending-removal – a remoção da instância do domínio está pendente.
- pending-maintenance-removal: a AWS tentará remover a instância do domínio durante a próxima janela de manutenção programada.
- failed – um problema de configuração impediu que a instância se associasse ao domínio. Verifique e corrija sua configuração antes de emitir novamente o comando de modificação da instância.
- removing – a instância está sendo removida do domínio.

Uma solicitação para se tornar um membro de um domínio pode falhar devido a um problema de conectividade de rede ou a uma função do IAM incorreta. Por exemplo, é possível criar uma instância de banco de dados ou modificar uma instância existente e não conseguir transformar a instância de banco de dados em um membro de um domínio. Nesse caso, reexecute o comando para criar ou modificar a instância de banco de dados ou modifique a instância recém-criada para ingressar no domínio.

Conexão ao SQL Server com autenticação do Windows

Para se conectar ao SQL Server com Autenticação do Windows, você deve estar conectado em um computador pertencente ao domínio como um usuário de domínio. Após iniciar o SQL Server

Management Studio, escolha Autenticação do Windows como o tipo de autenticação, conforme mostrado a seguir.



Restauração de uma instância de banco de dados do SQL Server e adição posterior a um domínio

Você pode restaurar um snapshot de banco de dados ou fazer uma Point-In-Time Restore (PITR – Restauração de ponto no tempo) de uma instância de banco de dados do SQL Server e adicioná-la a um domínio. Depois que a instância de banco de dados tiver sido restaurada, modifique a instância usando o processo explicado em Etapa 5: Criar ou modificar a instância de banco de dados do SQL Server para adicionar a instância de banco de dados a um domínio.

Atualizar aplicações para conexão com instâncias de banco de dados Microsoft SQL Server usando novos certificados SSL/TLS

Em 13 de janeiro de 2023, o Amazon RDS publicou novos certificados de autoridade de certificação (CA) para se conectar às suas instâncias de banco de dados do RDS via Secure Socket Layer ou Transport Layer Security (SSL/TLS). A seguir, você pode encontrar informações sobre como atualizar seus aplicativos para usar os novos certificados.

Este tópico pode ajudá-lo a determinar se algum aplicativo cliente usa SSL/TLS para conectar-se às suas instâncias de banco de dados. Em caso positivo, examine mais a fundo se esse aplicativo exige verificação de certificado para se conectar.

> **Note**
>
> Algumas aplicações são configuradas para se conectar a instâncias de banco de dados do SQL Server apenas quando podem verificar com êxito o certificado no servidor.
> Para esses aplicativos, você deve atualizar os repositórios confiáveis de aplicativos cliente para incluir os novos certificados de CA.

Depois de atualizar seus certificados de CA nos armazenamentos confiáveis do aplicativo cliente, você pode fazer o rodízio dos certificados nas suas instâncias de banco de dados. É altamente recomendável testar esses procedimentos em um ambiente de desenvolvimento ou teste antes de implementá-los em seus ambientes de produção.

Para obter mais informações sobre a mudança de certificados, consulte Alternar o certificado SSL/TLS. Para obter mais informações sobre como fazer download de certificados, consulte Usar SSL/TLS para criptografar uma conexão com uma instância ou um cluster de banco de dados. Para obter informações sobre o uso de SSL/TLS com instâncias de bancos de dados Microsoft SQL Server, consulte ???.

Tópicos

- Determinar se alguma aplicação está se conectando à sua instância de banco de dados Microsoft SQL Server usando SSL
- Determinar se um cliente requer verificação de certificado para se conectar
- Atualizar o armazenamento confiável de aplicações

Determinar se alguma aplicação está se conectando à sua instância de banco de dados Microsoft SQL Server usando SSL

Verifique a configuração da instância de banco de dados para obter o valor do parâmetro `rds.force_ssl`. Por padrão, o parâmetro `rds.force_ssl` é definido como 0 (desativado). Se o parâmetro `rds.force_ssl` estiver definido como 1 (ativado), os clientes serão obrigados a usar SSL/TLS para conexões. Para ter mais informações sobre parameter groups, consulte Grupos de parâmetros para Amazon RDS.

Executar a seguinte consulta para obter a opção de criptografia atual para todas as conexões abertas para uma instância de banco de dados. A coluna `ENCRYPT_OPTION` retornará `TRUE` se a conexão estiver criptografada.

```
select SESSION_ID,
    ENCRYPT_OPTION,
    NET_TRANSPORT,
    AUTH_SCHEME
    from SYS.DM_EXEC_CONNECTIONS
```

Essa consulta mostra apenas as conexões atuais. Ela não mostra se os aplicativos que se conectaram e desconectaram no passado usaram SSL.

Determinar se um cliente requer verificação de certificado para se conectar

Você pode verificar se diferentes tipos de clientes exigem verificação do certificado para se conectar.

> **Note**
>
> Se você usar conectores diferentes dos listados, consulte a documentação do conector específico para obter informações sobre como ele impõe conexões criptografadas. Para obter mais informações, consulte Módulos de conexão para bancos de dados Microsoft SQL, na documentação do Microsoft SQL Server.

SQL Server Management Studio

Verifique se a criptografia é aplicada para conexões do SQL Server Management Studio:

1. Inicie o SQL Server Management Studio.
2. Em Connect to server (Conectar-se ao servidor), digite as informações de servidor, o nome de usuário e a senha de login.
3. Escolha Options.
4. Verifique se Encrypt connection (Criptografar conexão) está selecionada na página de conexão.

Para obter mais informações sobre o SQL Server Management Studio, consulte Usar o SQL Server Management Studio.

Sqlcmd

Os exemplos a seguir com o cliente `sqlcmd` mostram como verificar uma conexão SQL Server de um script para determinar se conexões bem-sucedidas requerem um certificado válido. Para obter mais informações, consulte Conectar-se com sqlcmd na documentação do Microsoft SQL Server.

Ao usar `sqlcmd`, uma conexão SSL exigirá verificação com base no certificado do servidor se você usar o argumento de comando `-N` para criptografar conexões, como no exemplo a seguir.

```
$ sqlcmd -N -S dbinstance.rds.amazon.com -d ExampleDB
```

> **Note**
>
> Se `sqlcmd` for chamado com a opção `-C`, ele confiará no certificado do servidor, mesmo que ele não corresponda ao armazenamento confiável no lado do cliente.

ADO.NET

No exemplo a seguir, o aplicativo se conecta usando SSL, e o certificado de servidor deve ser verificado.

```
using SQLC = Microsoft.Data.SqlClient;

...
```

```
    static public void Main()
    {
        using (var connection = new SQLC.SqlConnection(
            "Server=tcp:dbinstance.rds.amazon.com;" +
            "Database=ExampleDB;User ID=LOGIN_NAME;" +
            "Password=YOUR_PASSWORD;" +
            "Encrypt=True;TrustServerCertificate=False;"
            ))
        {
            connection.Open();
            ...
        }
```

Java

No exemplo a seguir, o aplicativo se conecta usando SSL, e o certificado de servidor deve ser verificado.

```
String connectionUrl =
    "jdbc:sqlserver://dbinstance.rds.amazon.com;" +
    "databaseName=ExampleDB;integratedSecurity=true;" +
    "encrypt=true;trustServerCertificate=false";
```

Para habilitar a criptografia SSL para clientes que se conectam usando o JDBC, talvez seja necessário adicionar o certificado do Amazon RDS para a loja de certificados de Java CA. Para obter instruções, consulte o tópico sobre como Configurar o cliente para criptografia, na documentação do Microsoft SQL Server. Você também pode fornecer o nome de arquivo do certificado de CA confiável diretamente, anexando `trustStore=`*`path-to-certificate-trust-store-file`* a string de conexão.

> **Note**
>
> Se você usar `TrustServerCertificate=true` (ou equivalente) na string de conexão, o processo de conexão ignorará a validação da cadeia de confiança. Nesse caso, o aplicativo se conecta mesmo que o certificado não possa ser verificado. O uso de `TrustServerCertificate=false` impõe a validação de certificado e é uma prática recomendada.

Atualizar o armazenamento confiável de aplicações

Você pode atualizar o armazenamento confiável para aplicativos que usam o Microsoft SQL Server. Para obter instruções, consulte ???. Além disso, consulte o tópico sobre como Configurar o cliente para criptografia, na documentação do Microsoft SQL Server.

Se você estiver usando um sistema operacional que não seja o Microsoft Windows, consulte a documentação da distribuição do software para implementação de SSL/TLS para obter informações sobre como adicionar novo certificado de CA raiz. Por exemplo, o OpenSSL e o GnuTLS são opções populares. Use o método de implementação para adicionar confiança ao certificado de CA raiz do RDS. A Microsoft fornece instruções para configurar certificados em alguns sistemas.

Para obter informações sobre como baixar o certificado raiz, consulte Usar SSL/TLS para criptografar uma conexão com uma instância ou um cluster de banco de dados.

Para obter exemplos de scripts que importam certificados, consulte Script de exemplo para importar certificados para o seu armazenamento confiável.

> **Note**
>
> Ao atualizar o armazenamento confiável, é possível reter certificados mais antigos, além de adicionar os novos certificados.

Atualizações do mecanismo de banco de dados do Microsoft SQL Server

Quando o Amazon RDS oferece suporte a uma nova versão de um mecanismo de banco de dados, você pode atualizar suas instâncias de banco de dados para essa nova versão. Há dois tipos de atualizações para instâncias de banco de dados SQL Server: atualizações de versão principal e atualizações de versão secundária.

As atualizações da versão principal podem conter as alterações de banco de dados incompatíveis com os aplicativos existentes. Como resultado, você deve realizar manualmente as atualizações de versões principais das suas instâncias de banco de dados. Você pode iniciar uma atualização de versão principal modificando manualmente sua instância de banco de dados. No entanto, antes de realizar uma atualização de versão principal, recomendamos que você teste a atualização seguindo as etapas descritas em Testar uma atualização do RDS para SQL Server.

As atualizações de versão secundária contêm somente alterações compatíveis com versões anteriores das aplicações existentes. Você pode atualizar a versão secundária da instância de banco de dados de duas maneiras:

- Manualmente: modifique a instância de banco de dados para iniciar a atualização.
- Automaticamente: habilite atualizações automáticas de versão secundária para a instância de banco de dados.

Quando você habilita atualizações automáticas de versões secundárias, o RDS para SQL Server atualiza automaticamente a instância de banco de dados durante janelas de manutenção programada, quando atualizações essenciais de segurança estão disponíveis em uma versão secundária mais recente.

Para versões secundárias do mecanismo após `16.00.4120.1`, `15.00.4365.2`, `14.00.3465.1` e `13.00.6435.1`, os seguintes protocolos de segurança estão desabilitados por padrão:

- `rds.tls10` (protocolo TLS 1.0)
- `rds.tls11` (protocolo TLS 1.1)
- `rds.rc4` (cifra RC4)
- `rds.curve25519` (criptografia Curve25519)
- `rds.3des168` (criptografia Triple DES)

Para versões anteriores do mecanismo, o Amazon RDS habilita esses protocolos de segurança por padrão.

```
...

"ValidUpgradeTarget": [
    {
        "Engine": "sqlserver-se",
        "EngineVersion": "14.00.3281.6.v1",
        "Description": "SQL Server 2017 14.00.3281.6.v1",
        "AutoUpgrade": false,
        "IsMajorVersionUpgrade": false
    }
...
```

Para ter mais informações sobre como executar atualizações, consulte Atualizar de uma instância de banco de dados SQL Server. Para obter informações sobre quais versões do SQL Server estão disponíveis no Amazon RDS, consulte Amazon RDS para Microsoft SQL Server.

Tópicos

- Atualizações da versão principal do RDS para SQL Server
- Considerações sobre atualizações do SQL Server
- Testar uma atualização do RDS para SQL Server
- Atualizar de uma instância de banco de dados SQL Server
- Fazer upgrade de instâncias de banco de dados defasadas antes do fim do suporte

Atualizações da versão principal do RDS para SQL Server

Atualmente, o Amazon RDS oferece suporte para as seguintes atualizações de versão principais para uma instância de banco de dados do Microsoft SQL Server.

Atualize sua instância de banco de dados existente para o SQL Server 2017 ou 2019 a partir de qualquer versão, exceto SQL Server 2008. Para atualizar do SQL Server 2008, primeiro atualize para outra versão.

Versão atual	Versões de upgrade com suporte
SQL Server 2019	SQL Server 2022

Versão atual	Versões de upgrade com suporte
SQL Server 2017	SQL Server 2022 SQL Server 2019
SQL Server 2016	SQL Server 2022 SQL Server 2019 SQL Server 2017

É possível usar uma consulta da AWS CLI, como o exemplo a seguir, para localizar as atualizações disponíveis para uma versão específica do mecanismo de banco de dados.

Example

Para Linux, macOS ou Unix:

```
aws rds describe-db-engine-versions \
    --engine sqlserver-se \
    --engine-version 14.00.3281.6.v1 \
    --query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" \
    --output table
```

Para Windows:

```
aws rds describe-db-engine-versions ^
    --engine sqlserver-se ^
    --engine-version 14.00.3281.6.v1 ^
    --query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" ^
    --output table
```

A saída mostra que é possível atualizar a versão 14.00.3281.6 para as versões mais recentes do SQL Server 2017 ou 2019.

```
--------------------------
|DescribeDBEngineVersions|
+------------------------+
|      EngineVersion     |
```

```
+------------------------+
|  14.00.3294.2.v1       |
|  14.00.3356.20.v1      |
|  14.00.3381.3.v1       |
|  14.00.3401.7.v1       |
|  14.00.3421.10.v1      |
|  14.00.3451.2.v1       |
|  15.00.4043.16.v1      |
|  15.00.4073.23.v1      |
|  15.00.4153.1.v1       |
|  15.00.4198.2.v1       |
|  15.00.4236.7.v1       |
+------------------------+
```

Nível de compatibilidade do banco de dados

Você pode usar os níveis de compatibilidade do banco de dados Microsoft SQL Server para ajustar alguns comportamentos de banco de dados de forma a imitar as versões anteriores do SQL Server. Para ter mais informações, consulte Nível de compatibilidade na documentação da Microsoft. Quando você atualiza sua instância de banco de dados, todos os bancos de dados existentes permanecem em seus níveis de compatibilidade originais.

Você pode alterar o nível de compatibilidade de um banco de dados usando o comando ALTER DATABASE. Por exemplo, para alterar um banco de dados denominado `customeracct` para que ele seja compatível com o SQL Server 2016, emita o seguinte comando:

```
ALTER DATABASE customeracct SET COMPATIBILITY_LEVEL = 130
```

Considerações sobre atualizações do SQL Server

O Amazon RDS usa dois snapshot de banco de dadoss durante o processo de atualização. O primeiro snapshot de banco de dados é da instância de banco de dados antes de qualquer alteração de upgrade ter sido feita. O segundo snapshot de banco de dados é capturado após a conclusão do upgrade.

> **Note**
>
> O Amazon RDS só tirará snapshot de banco de dadoss se você tiver definido o período de retenção de backup para sua instância de banco de dados como um número maior que 0.

Para alterar o período de retenção de backup, consulte Modificar uma instância de banco de dados do Amazon RDS.

Após a conclusão de um upgrade, não será possível reverter para a versão anterior do mecanismo de banco de dados. Se desejar retornar à versão anterior, restaure o snapshot do banco de dados que foi tirado antes do upgrade para criar uma nova instância de banco de dados.

Durante uma atualização de versão secundária ou principal do SQL Server, as métricas Free Storage Space (Espaço de armazenamento gratuito) e Disk Queue Depth (Profundidade da fila do disco) exibirão `-1`. Após a conclusão da atualização, ambas as métricas voltarão ao normal.

Antes de fazer a atualização da instância do SQL Server, analise as informações a seguir.

Tópicos

- Práticas recomendadas antes de iniciar uma atualização
- Considerações Multi-AZ
- Considerações sobre réplicas de leitura
- Considerações de grupos de opções
- Considerações de grupos de parâmetros

Práticas recomendadas antes de iniciar uma atualização

Antes de iniciar o processo de atualização, implemente as seguintes etapas preparatórias para permitir que a atualização ocorra de maneira ideal e para minimizar possíveis problemas:

Gerenciamento do cronograma e de workloads

- Programe as atualizações para períodos de baixo volume de transações.
- Minimize as operações de gravação durante a janela de atualização.

  Isso permite que o Amazon RDS conclua as atualizações mais depressa, reduzindo o número de arquivos de backup do log de transações que o RDS precisa restaurar durante o emparelhamento de secundárias e principais.

Gerenciamento de transações

- Identifique e monitore transações de longa duração.
- Garanta que todas as transações essenciais sejam confirmadas antes de iniciar a atualização.

- Evite transações de longa duração durante a janela de atualização.

Otimização do arquivo de log

Examine e otimize os arquivos de log das transações:

- Reduza os arquivos de log superdimensionados.
- Reduza os padrões de alto consumo de logs.
- Gerencie os arquivos de log virtuais (VLFs).
- Mantenha espaço livre adequado para operações normais.

Considerações Multi-AZ

O Amazon RDS oferece suporte a implantações Multi-AZ para instâncias de bancos de dados que executem o Microsoft SQL Server usando o SQL Server Database Mirroring (DBM) ou Grupos de disponibilidade AlwaysOn. Para obter mais informações, consulte Implantações multi-AZ para o Amazon RDS for Microsoft SQL Server.

Em uma implantação Multi-AZ (espelhamento/AlwaysOn), quando uma atualização é solicitada, o RDS segue uma estratégia de atualização contínua para as instâncias primária e secundária. As atualizações contínuas garantem que pelo menos uma instância esteja disponível para transações enquanto a instância secundária é atualizada. Espera-se que a interrupção dure apenas o tempo de um failover.

Durante a atualização, o RDS remove a instância secundária da configuração Multi-AZ, executa uma atualização da instância secundária e restaura todos os backups de log de transações da primária feitos durante o período em que ela estava desconectada. Depois que todos os backups de log forem restaurados, o RDS une o secundário atualizado ao primário. Quando todos os bancos de dados estiverem em um estado sincronizado, o RDS executa um failover para a instância secundária atualizada. Depois que o failover é concluído, o RDS prossegue com a atualização da antiga instância primária, restaura todos os backups do log de transações e a associa à nova instância primária.

Para minimizar essa duração do failover, recomendamos usar o endpoint do receptor do grupo de disponibilidade do AlwaysOn AG ao usar bibliotecas de cliente que ofereçam suporte à opção de conexão `MultiSubnetFailover` na string de conexão. Ao usar o endpoint do receptor do grupo de disponibilidade, os tempos de failover geralmente são inferiores a 10 segundos, no entanto, essa duração não inclui nenhum tempo adicional de recuperação de falhas.

Considerações sobre réplicas de leitura

Durante uma atualização da versão do banco de dados, o Amazon RDS atualiza todas as réplicas de leitura, bem como a instância de banco de dados primária. O Amazon RDS não é compatível com atualizações de versões de banco de dados nas réplicas de leitura separadamente. Para ter mais informações sobre réplicas de leitura, consulte Trabalhar com réplicas de leitura do Microsoft SQL Server no Amazon RDS.

Ao fazer uma atualização da versão do banco de dados da instância de banco de dados primária, todas as réplicas de leitura também são atualizadas automaticamente. O Amazon RDS vai atualizar todas as réplicas de leitura simultaneamente antes de atualizar a instância de banco de dados primária. As réplicas de leitura podem não estar disponíveis até que a atualização da versão do banco de dados na instância de banco de dados primária seja concluída.

Considerações de grupos de opções

Se a sua instância de banco de dados usar um grupo de opções de banco de dados personalizado, em alguns casos, o Amazon RDS não poderá atribuir automaticamente à instância um novo grupo de opções. Por exemplo, ao atualizar para uma nova versão principal, você deve especificar um novo grupo de opções. Recomendamos que você crie um novo grupo de opções e adicione as mesmas opções a ele que o seu grupo de opções personalizadas existente.

Para ter mais informações, consulte Criar um grupo de opções ou Copiar um grupo de opções.

Considerações de grupos de parâmetros

Se sua instância de banco de dados usa um grupo de parâmetros de banco de dados personalizado:

- O Amazon RDS reinicializa automaticamente a instância de banco de dados após uma atualização.
- Em alguns casos, o RDS não pode atribuir automaticamente um novo grupo de parâmetros à sua instância de banco de dados.

  Por exemplo, ao atualizar para uma nova versão principal, você deve especificar um novo grupo de parâmetros. Recomendamos que você crie um novo parameter group e configure os parâmetros como no seu parameter group personalizado existente.

Para ter mais informações, consulte Criar um grupo de parâmetros de banco de dados no Amazon RDS ou Copiar um grupo de parâmetros de banco de dados no Amazon RDS.

Testar uma atualização do RDS para SQL Server

Antes de executar um upgrade de versão principal na sua instância de banco de dados, você deve testar completamente seu banco de dados e todos os aplicativos que o acessam para verificar a compatibilidade com a nova versão. Recomendamos que você use o procedimento a seguir.

Para testar um upgrade de versão principal

1. Examine a Atualização do SQL Server na documentação da Microsoft para a nova versão do mecanismo de banco de dados para ver se existem problemas de compatibilidade que podem afetar o banco de dados as aplicações.
2. Se a sua instância de banco de dados usar um grupo de opções personalizado, crie um novo grupo de opções compatível com a nova versão para a qual você está atualizando. Para ter mais informações, consulte Considerações de grupos de opções.
3. Se a sua instância de banco de dados usar um parameter group personalizado, crie um novo parameter group compatível com a nova versão para a qual você está atualizando. Para ter mais informações, consulte Considerações de grupos de parâmetros.
4. Crie um snapshot de banco de dados da instância de banco de dados a ser atualizada. Para ter mais informações, consulte Criar um snapshot de banco de dados para uma instância de banco de dados single-AZ para o Amazon RDS.
5. Restaure o snapshot de banco de dados para criar uma nova instância de banco de dados de teste. Para ter mais informações, consulte Restaurar uma instância de banco de dados.
6. Modifique essa nova instância de banco de dados de teste para atualizá-la para a nova versão, usando um dos seguintes métodos:

   - Console
   - AWS CLI
   - API do RDS
7. Avalie o armazenamento usado pela instância atualizada para determinar se o upgrade requer armazenamento adicional.
8. Execute quantos testes de garantia de qualidade na instância de banco de dados atualizada forem necessários para garantir que o seu banco de dados e o seu aplicativo funcionem corretamente com a nova versão. Implemente os novos testes necessários para avaliar o impacto de problemas de compatibilidade que você identificou na etapa 1. Teste todos os procedimentos armazenados e funções. Direcione versões de teste dos seus aplicativos à instância de banco de dados atualizada.

9. Se todos os testes forem aprovados, realize o upgrade na sua instância de banco de dados de produção. Recomendamos que você não permita operações de gravação na instância de banco de dados até confirmar que tudo está funcionando corretamente.

Atualizar de uma instância de banco de dados SQL Server

Para obter informações sobre como atualizar manual ou automaticamente uma instância de banco de dados do SQL Server, consulte o seguinte:

- Atualizar a versão de mecanismo de uma instância de banco de dados
- Práticas recomendadas para atualizar o SQL Server 2008 R2 para o SQL Server 2016 no Amazon RDS for SQL Server

> **Important**
>
> Se você tiver algum snapshot criptografado usando o AWS KMS, recomendamos iniciar um upgrade antes do fim do suporte.

Fazer upgrade de instâncias de banco de dados defasadas antes do fim do suporte

Depois que uma versão principal é defasada, não é possível instalá-la em novas instâncias de banco de dados. O RDS tentará fazer o upgrade automático de todas as instâncias de banco de dados existentes.

Se você precisar restaurar uma instância de banco de dados defasada, poderá fazer uma Point-In-Time Recovery (PITR – Recuperação de ponto no tempo) ou restaurar um snapshot. Isso permite o acesso temporário a uma instância de banco de dados que usa a versão que está sendo preterida. No entanto, depois que uma versão principal for totalmente preterida, essas instâncias de banco de dados também serão automaticamente atualizadas para uma versão com suporte.

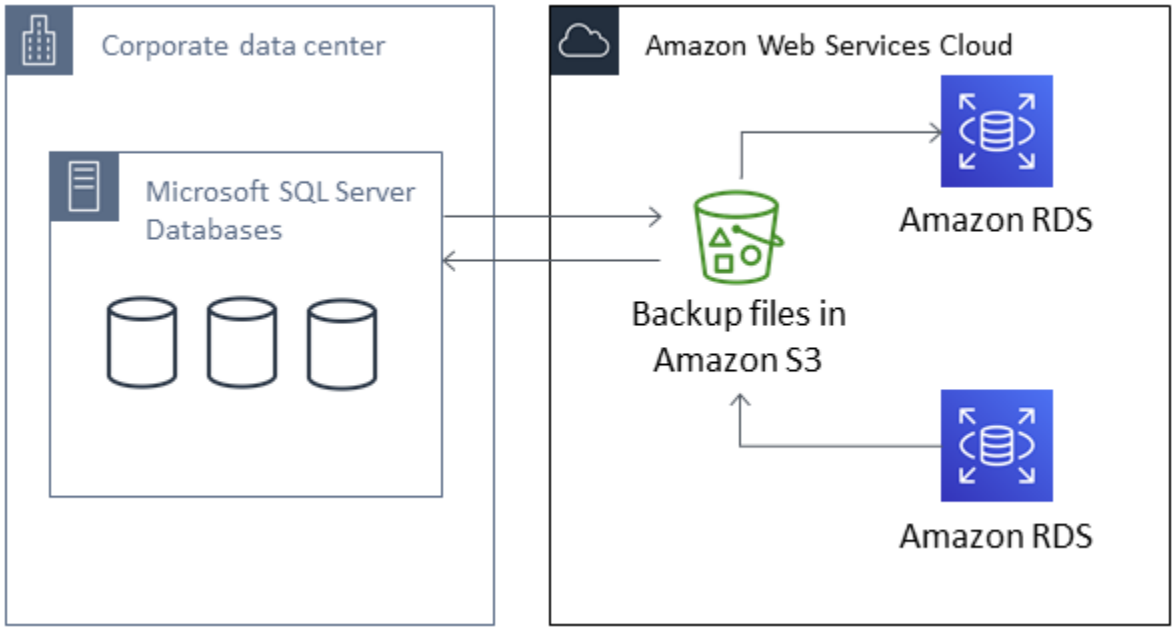
Importar e exportar bancos de dados do SQL Server usando backup e restauração nativos

O Amazon RDS suporta backup e restauração nativos para bancos de dados do Microsoft SQL Server usando arquivos de backup completo (arquivos .bak). Ao usar o RDS, acesse arquivos armazenados no Amazon S3, em vez de usar o sistema de arquivos local no servidor de banco de dados.

Por exemplo, crie um backup completo no servidor local, armazene-o no S3 e o restaure em uma instância de banco de dados do Amazon RDS existente. Também faça backups do RDS, armazene-os no S3 e os restaure sempre que quiser.

O backup e a restauração nativos estão disponíveis em todas as regiões da AWS para as instâncias de banco de dados Mono-AZ e multi-AZ, incluindo instâncias de banco de dados multi-AZ com réplicas de leitura. O backup e a restauração nativos estão disponíveis para todas as edições do Microsoft SQL Server compatíveis com o Amazon RDS.

O diagrama a seguir mostra os cenários compatíveis.



Usar arquivos .bak nativos para fazer backup e restaurar bancos de dados costuma ser a maneira mais rápida de se fazer backup e restauração de bancos de dados. Há muitas vantagens adicionais em se usar o backup e a restauração nativos. Por exemplo, você pode fazer o seguinte:

- Migrar bancos de dados para ou de Amazon RDS.

- Movimentação de bancos de dados entre instâncias de banco de dados do RDS for SQL Server.
- Migrar dados, esquemas, procedimentos armazenados, disparadores e outros códigos de banco de dados dentro de um arquivo .bak.
- Fazer backup e restauração de bancos de dados únicos, em vez de instâncias de banco de dados inteiras.
- Criar cópias de bancos de dados para desenvolvimento, testes, treinamento e demonstrações.
- Armazenar e transferir arquivos de backup com o Amazon S3, para uma camada adicional de proteção para recuperação de desastres.
- Crie backups nativos de bancos de dados com Transparent Data Encryption (TDE) ativada e restaure esses backups em bancos de dados on-premises. Para obter mais informações, consulte Suporte para criptografia de dados transparente no SQL Server.
- Restaure backups nativos de bancos de dados on-premises que têm o TDE ativado para instâncias de banco de dados do RDS para SQL Server. Para obter mais informações, consulte Suporte para criptografia de dados transparente no SQL Server.

Sumário

- Limitações e recomendações
- Configurar o backup e a restauração nativos
  - Criar manualmente uma função do IAM para backup e restauração nativos
- Usar o backup e restauração nativos
  - Realização de backup de um banco de dados
    - Uso
    - Exemplos
  - Restaurar um banco de dados
    - Uso
    - Exemplos
  - Restaurar um log
    - Uso
    - Exemplos
  - Concluir uma restauração de banco de dados
    - Uso
  - Como trabalhar com bancos de dados parcialmente restaurados

        - Descartando um banco de dados parcialmente restaurado
        - Restauração de snapshot e comportamento de recuperação em um ponto anterior no tempo para bancos de dados parcialmente restaurados
    - Cancelar uma tarefa
        - Uso
    - Acompanhar o status de tarefas
        - Uso
        - Exemplos
        - Resposta
- Compactar arquivos de backup
- Solução de problemas
- Importar e exportar dados do SQL Server usando outros métodos
    - Importar dados para o RDS para SQL Server usando um snapshot
        - Importar os dados
            - Assistente para gerar e publicar scripts
            - Assistente de importação e exportação
            - Cópia em massa
    - Exportar dados do RDS para SQL Server
        - Assistente de importação e exportação do SQL Server
        - Utilitário bcp e Assistente para gerar e publicar scripts do SQL Server
- Usar o utilitário BCP do Linux para importar e exportar dados
    - Pré-requisitos
    - Instalar ferramentas de linha de comandos do SQL Server no Linux
    - Exportar dados do RDS para SQL Server
        - Sintaxe básica de exportação
        - Exemplo de exportação
    - Importar dados para o RDS para SQL Server
        - Sintaxe básica de importação
        - Exemplo de importação
    - Opções comuns do BCP

- Práticas recomendadas e considerações
- Solução de problemas comuns do

Limitações e recomendações

A seguir estão algumas limitações para o uso de backup e restauração nativos:

- Você não pode fazer backup ou restaurar de um bucket do Amazon S3 em uma região da AWS diferente de sua instância de banco de dados do Amazon RDS.
- Não é possível restaurar um banco de dados com o mesmo nome de um banco de dados já existente. Os nomes de banco de dados são exclusivos.
- Recomendamos que não restaure backups de um fuso horário em um fuso horário diferente. Se você restaurar backups de um fuso horário em outro fuso horário, deverá auditar as consultas e aplicativos para verificar os efeitos da alteração do fuso horário.
- O Amazon S3 tem um limite de tamanho de 5 TB por arquivo. Para backups nativos de bancos de dados maiores, você pode usar o backup de vários arquivos.
- O tamanho máximo do banco de dados que pode ser feito backup no S3 depende da memória, CPU, E/S e recursos de rede disponíveis na instância de banco de dados. Quanto maior o banco de dados, mais memória o agente de backup consome.
- Não é possível fazer backup ou restaurar mais de 10 arquivos de backup ao mesmo tempo.
- Um backup diferencial baseia-se no último backup completo. Para que os backups diferenciais funcionem, você não pode obter um snapshot entre o último backup completo e o backup diferencial. Se desejar um backup diferencial, mas houver um snapshot manual ou automatizado, faça outro backup completo antes de continuar com o backup diferencial.
- Restaurações diferenciais e de log não são compatíveis com bancos de dados com arquivos que têm seu file_guid (identificador exclusivo) definido como NULL.
- Você pode executar até duas tarefas de backup ou restauração ao mesmo tempo.
- Não é possível executar backups de log nativo do SQL Server no Amazon RDS.
- O RDS oferece suporte a restaurações nativas de bancos de dados de até 64 TiB. Restaurações nativas de bancos de dados no SQL Server Express Edition estão limitadas a 10 GB.
- Não é possível fazer backup de um banco de dados durante a janela de manutenção, nem enquanto o Amazon RDS estiver no processo de tirar um snapshot do banco de dados. Se uma tarefa nativa de backup se sobrepor à janela diária de backup do RDS, a tarefa nativa de backup será cancelada.

- Em instâncias de banco de dados multi-AZ, você só pode restaurar de forma nativa os bancos de dados com backup feito no modelo de recuperação completo.
- Não há suporte para a restauração de backups diferenciais em instâncias multi-AZ.
- Não há suporte para a chamada dos procedimentos do RDS para backup/restauração dentro de uma transação.
- Use uma AWS KMS key de criptografia simétrica para criptografar seus backups. O Amazon RDS não oferece suporte para chaves do KMS assimétricas. Para obter mais informações, consulte Criar chaves do KMS simétricas no Guia do desenvolvedor da AWS Key Management Service.
- Os arquivos de backup nativo são criptografados com a chave do KMS especificada usando o modo de criptografia "Encryption-Only". Ao restaurar arquivos de backup criptografados, saiba que eles foram criptografados com o modo de criptografia "Encryption-Only".
- Não é possível restaurar um banco de dados que contenha um grupo de arquivos FILESTREAM.
- Não é possível usar a criptografia do lado do servidor do Amazon S3 com o AWS KMS (SSE-KMS). Quando você fornece uma chave do KMS para um procedimento armazenado, quaisquer restaurações e backups nativos são criptografados e descriptografados do lado do cliente com a chave do KMS. A AWS armazena os backups no bucket do S3 com SSE-S3.

Se o banco de dados estiver offline enquanto o arquivo de backup for criado, copiado e restaurado, recomendamos que você use o backup e a restauração nativos para migrar seu banco de dados para o RDS. Se o banco de dados on-premises não puder ficar offline, recomendamos que você use o AWS Database Migration Service para migrar seu banco de dados para o Amazon RDS. Para obter mais informações, consulte O que é o AWS Database Migration Service?

O backup e a restauração nativos não se destinam a substituir os recursos de recuperação de dados do recurso de cópia de snapshot entre regiões. Recomendamos que você use uma cópia do snapshot para copiar o snapshot do banco de dados em outra região da AWS para recuperação de desastres entre regiões no Amazon RDS. Para obter mais informações, consulte Copiar um snapshot de banco de dados para o Amazon RDS.

Configurar o backup e a restauração nativos

Para configurar o backup e a restauração nativos, você precisará de três componentes:

1. Um bucket do Amazon S3 para armazenar seus arquivos de backup.

   É necessário ter um bucket do S3 para usar nos arquivos de backup e fazer upload dos backups que deseja migrar para o RDS. Se você já tiver um bucket do Amazon S3, você pode usá-lo. Do

contrário, crie um bucket. Como alternativa, opte por criar um novo bucket ao adicionar a opção SQLSERVER_BACKUP_RESTORE usando o AWS Management Console.

Para obter informações sobre como usar o S3, consulte o Guia do usuário do Amazon Simple Storage Service.

2. Uma função do AWS Identity and Access Management (IAM) para acessar o bucket.

   Se você já tiver uma função do IAM, você pode usá-la. Você pode optar por criar uma nova função do IAM ao adicionar a opção SQLSERVER_BACKUP_RESTORE usando o AWS Management Console. Você também pode criar uma nova manualmente.

   Se você quiser criar uma nova função do IAM manualmente, siga a abordagem discutida na próxima seção. Faça o mesmo se quiser associar relações de confiança e políticas de permissões a uma função existente do IAM.

3. A opção SQLSERVER_BACKUP_RESTORE adicionada a um grupo de opções em sua instância de banco de dados.

   Para habilitar o backup e restauração nativos em sua instância de banco de dados, adicione a opção SQLSERVER_BACKUP_RESTORE a um grupo de opções em sua instância de banco de dados. Para mais informações e instruções, consulte Suporte para backup e restauração nativos no SQL Server.

Criar manualmente uma função do IAM para backup e restauração nativos

Se você quiser criar manualmente uma nova função do IAM para usar com backup e restauração nativos, poderá fazê-lo. Nesse caso, você criará uma função para delegar permissões do serviço do Amazon RDS ao seu bucket do Amazon S3. Ao criar uma função do IAM, associe uma relação de fidedignidade e uma política de permissões. A relação de confiança permite que o RDS assuma essa função. A política de permissões define as ações que essa função pode executar. Para obter mais informações sobre como criar a função, consulte Criar uma função para delegar permissões a um AWSserviço da .

Para o recurso de backup e restauração nativo, use políticas de confiança e permissões semelhantes aos exemplos nesta seção. No exemplo a seguir, usamos o nome principal do serviço `rds.amazonaws.com` como um alias para todas as contas de serviço. Em outros exemplos, especificamos um nome de recurso da Amazon (ARN) para identificar outra conta, usuário ou função para a qual estejamos concedendo acesso na política de confiança.

Recomendamos usar as chaves de contexto de condição global `aws:SourceArn` e `aws:SourceAccount` em relações baseadas em recursos para limitar as permissões do serviço a um recurso específico. Essa é a maneira mais eficiente de se proteger contra o problema "confused deputy".

Você pode usar as duas chaves de contexto de condição global e fazer com que o valor `aws:SourceArn` contenha o ID da conta. Nesses casos, verifique se o valor `aws:SourceAccount` e a conta no `aws:SourceArn` usa o mesmo ID de conta quando eles são usados na mesma instrução.

- Use `aws:SourceArn` se quiser acesso entre serviços para um único recurso.
- Use `aws:SourceAccount` se você quiser permitir que qualquer recurso nessa conta seja associado ao uso entre serviços.

Na relação de confiança, certifique-se de usar a chave de contexto de condição global `aws:SourceArn` com o ARN completo dos recursos que acessam a função. Para backup e restauração nativos, certifique-se de incluir o grupo de opções de banco de dados e as instâncias de banco de dados, conforme mostrado no exemplo a seguir.

Example de relação de confiança com a chave de contexto de condição global para backup e restauração nativos

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "rds.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceArn": [

 "arn:aws:rds:Region:0123456789:db:db_instance_identifier",
                        "arn:aws:rds:Region:0123456789:og:option_group_name",
                  "arn:aws:s3:::amzn-s3-demo-bucket"
```

```
                    ],
                    "aws:SourceAccount": "0123456789"
                }
            }
        }
    ]
}
```

O exemplo a seguir usa um ARN para especificar um recurso. Para obter mais informações sobre como usar ARNs, consulte Nomes de recurso da Amazon (ARNs).

Example de política de permissões para fazer backup e restauração nativos sem suporte de criptografia

JSON

```
{
    "Version": "2012-10-17",
    "Statement":
    [
        {
        "Effect": "Allow",
        "Action":
            [
                "s3:ListBucket",
                "s3:GetBucketLocation"
            ],
        "Resource": "arn:aws:s3:::amzn-s3-demo-bucket"
        },
        {
        "Effect": "Allow",
        "Action":
            [
                "s3:GetObjectAttributes",
                "s3:GetObject",
                "s3:PutObject",
                "s3:ListMultipartUploadParts",
                "s3:AbortMultipartUpload"
            ],
        "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*"
        }
```

```
    ]
}
```

Example política de permissões para fazer backup e restauração nativos com suporte de criptografia

Se quiser criptografar seus arquivos de backup, inclua uma chave de criptografia em sua política de permissões. Para obter mais informações sobre as chaves de criptografia, consulte Getting started (Conceitos básicos) no Guia do desenvolvedor do AWS Key Management Service.

> **Note**
>
> É necessário usar uma chave do KMS de criptografia simétrica para criptografar seus backups. O Amazon RDS não oferece suporte para chaves do KMS assimétricas. Para obter mais informações, consulte Criar chaves do KMS simétricas no Guia do desenvolvedor da AWS Key Management Service.
> A função do IAM também deve ser um usuário de chave e administrador de chave para a chave do KMS, ou seja, ela deve ser especificada na política de chaves. Para obter mais informações, consulte Criar chaves do KMS simétricas no Guia do desenvolvedor da AWS Key Management Service.

JSON

```
{
    "Version": "2012-10-17",
    "Statement":
    [
        {
        "Effect": "Allow",
        "Action":
            [
                "kms:DescribeKey",
                "kms:GenerateDataKey",
                "kms:Encrypt",
                "kms:Decrypt"
            ],
        "Resource": "arn:aws:kms:region:account-id:key/key-id"
        },
        {
```

```
        "Effect": "Allow",
        "Action":
            [
                "s3:ListBucket",
                "s3:GetBucketLocation"
            ],
        "Resource": "arn:aws:s3:::amzn-s3-demo-bucket"
        },
        {
        "Effect": "Allow",
        "Action":
            [
                "s3:GetObjectAttributes",
                "s3:GetObject",
                "s3:PutObject",
                "s3:ListMultipartUploadParts",
                "s3:AbortMultipartUpload"
            ],
        "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*"
        }
    ]
}
```

Usar o backup e restauração nativos

Depois de habilitar e configurar o backup e a restauração nativos, você pode começar a usá-los. Primeiro, conecte-se ao banco de dados do Microsoft SQL Server e chame um procedimento armazenado do Amazon RDS para fazer o trabalho. Para obter instruções sobre como se conectar ao banco de dados, consulte Conectar-se à instância de banco de dados do Microsoft SQL Server.

Alguns dos procedimentos armazenados exigem que você forneça um nome de recurso da Amazon (ARN) ao bucket e arquivo do Amazon S3. O formato do ARN é `arn:aws:s3:::bucket_name/file_name.extension`. O Amazon S3 não exige um número de conta ou região da AWS nos ARNs.

Se você também fornecer uma chave do KMS opcional, o formato do ARN da chave será `arn:aws:kms:region:account-id:key/key-id`. Para obter mais informações, consulte Nomes de recurso da Amazon (ARNs) e namespaces de serviço da AWS. É necessário usar uma chave do KMS de criptografia simétrica para criptografar seus backups. O Amazon RDS não oferece suporte para chaves do KMS assimétricas. Para obter mais informações, consulte Criar chaves do KMS simétricas no Guia do desenvolvedor da AWS Key Management Service.

> **Note**
>
> Independentemente de você usar ou não uma chave do KMS, as tarefas nativas de backup e restauração habilitam a criptografia Advanced Encryption Standard (AES) de 256 bits no lado do servidor por padrão para arquivos carregados no S3.

Para instruções sobre como chamar cada procedimento armazenado, consulte os tópicos a seguir:

- Realização de backup de um banco de dados
- Restaurar um banco de dados
- Restaurar um log
- Concluir uma restauração de banco de dados
- Como trabalhar com bancos de dados parcialmente restaurados
- Cancelar uma tarefa
- Acompanhar o status de tarefas

Realização de backup de um banco de dados

Para fazer backup de seu banco de dados, use o procedimento armazenado `rds_backup_database`.

> **Note**
>
> Você não pode fazer backup de um banco de dados durante a janela de manutenção ou enquanto o Amazon RDS está tirando um snapshot.

Uso

```
exec msdb.dbo.rds_backup_database
 @source_db_name='database_name',
 @s3_arn_to_backup_to='arn:aws:s3:::bucket_name/file_name.extension',
 [@kms_master_key_arn='arn:aws:kms:region:account-id:key/key-id'],
 [@overwrite_s3_backup_file=0|1],
 [@block_size=512|1024|2048|4096|8192|16384|32768|65536],
        [@max_transfer_size=n],
        [@buffer_count=n],
 [@type='DIFFERENTIAL|FULL'],
 [@number_of_files=n];
```

Os seguintes parâmetros são obrigatórios:

- `@source_db_name` – o nome do banco de dados do qual deve ser feito backup
- `@s3_arn_to_backup_to` – o ARN indica o bucket do Amazon S3 a ser usado no backup e o nome do arquivo de backup.

  O arquivo pode ter qualquer extensão, mas a extensão `.bak` costuma ser usada.

Os seguintes parâmetros são opcionais:

- `@kms_master_key_arn`: o ARN da chave do KMS simétrica a ser usada para criptografar o item.
  - Não é possível usar a chave de criptografia padrão. Se você usar a chave padrão, não será feito backup do banco de dados.
  - Se você não especificar um identificador de chave KMS, o arquivo de backup não será criptografado. Para ter mais informações, consulte Criptografar recursos do Amazon RDS.

- Quando você especifica uma chave do KMS, é usada a criptografia no lado do cliente.
- O Amazon RDS não oferece suporte para chaves do KMS assimétricas. Para obter mais informações, consulte Criar chaves do KMS simétricas no Guia do desenvolvedor da AWS Key Management Service.

- `@overwrite_s3_backup_file` – um valor que indica se deseja substituir um arquivo de backup existente.
  - `0` – não substitui um arquivo existente. Esse valor é o padrão.

    Definir `@overwrite_s3_backup_file` como 0 retornará um erro se o arquivo já existir.
  - `1` – substitui um arquivo existente que tenha o nome especificado, ainda que ele não seja um arquivo de backup.
- `@type` – o tipo de backup.
  - `DIFFERENTIAL` – faz backup diferencial.
  - `FULL` – faz backup completo. Esse valor é o padrão.

  Um backup diferencial baseia-se no último backup completo. Para que os backups diferenciais funcionem, você não pode obter um snapshot entre o último backup completo e o backup diferencial. Se desejar um backup diferencial, mas houver um snapshot, faça outro backup completo antes de continuar com o backup diferencial.

  É possível procurar o último backup ou snapshot completo usando o seguinte exemplo de consulta SQL:

```
select top 1
database_name
,  backup_start_date
,  backup_finish_date
from    msdb.dbo.backupset
where   database_name='mydatabase'
and     type = 'D'
order by backup_start_date desc;
```

- `@number_of_files` – O número de arquivos em que o backup será dividido (em blocos). O número máximo é 10.
  - O backup de vários arquivos é compatível com backups completos e diferenciais.
  - Se você inserir um valor de 1 ou omitir o parâmetro, um único arquivo de backup será criado.

Forneça o prefixo que os arquivos têm em comum e depois o sufixo com um asterisco (*). O asterisco pode estar em qualquer lugar na parte *file_name* do ARN do S3. O asterisco é substituído por uma série de strings alfanuméricas nos arquivos gerados, começando com `1-of-`*number_of_files*.

Por exemplo, se os nomes de arquivo no ARN do S3 forem `backup*.bak` e você definir `@number_of_files=4`, os arquivos de backup gerados serão `backup1-of-4.bak`, `backup2-of-4.bak`, `backup3-of-4.bak` e `backup4-of-4.bak`.

- Se algum dos nomes de arquivo já existir e `@overwrite_s3_backup_file` for 0, um erro será retornado.
- Os backups de vários arquivos só podem ter um asterisco na parte *file_name* do ARN do S3.
- Os backups de arquivo único podem ter qualquer número de asteriscos na parte *file_name* do ARN do S3. Os asteriscos não são removidos do nome do arquivo gerado.

- `@block_size`: tamanho do bloco (em bytes) especificando o tamanho do bloco físico para operações de backup. Os valores válidos são 512, 1.024, 2.048, 4.096, 8.192, 16.384, 32.768 e 65.536
- `@max_transfer_size`: o tamanho máximo de transferência indica o limite máximo do volume de dados (em bytes) transmitido por operação de E/S durante o processo de backup. Os valores válidos são múltiplos de 65536 bytes (64 KB) até 4194304 bytes (4 MB).
- `@buffer_count`: número total de buffers de E/S a serem usados no processo de backup.

Exemplos

Example de backup diferencial

```
exec msdb.dbo.rds_backup_database
@source_db_name='mydatabase',
@s3_arn_to_backup_to='arn:aws:s3:::mybucket/backup1.bak',
@overwrite_s3_backup_file=1,
@type='DIFFERENTIAL';
```

Example de backup completo com criptografia

```
exec msdb.dbo.rds_backup_database
@source_db_name='mydatabase',
@s3_arn_to_backup_to='arn:aws:s3:::mybucket/backup1.bak',
@kms_master_key_arn='arn:aws:kms:us-east-1:123456789012:key/AKIAIOSFODNN7EXAMPLE',
```

```
@overwrite_s3_backup_file=1,
@type='FULL';
```

Example de backup de vários arquivos

```
exec msdb.dbo.rds_backup_database
@source_db_name='mydatabase',
@s3_arn_to_backup_to='arn:aws:s3:::mybucket/backup*.bak',
@number_of_files=4;
```

Example de backup diferencial de vários arquivos

```
exec msdb.dbo.rds_backup_database
@source_db_name='mydatabase',
@s3_arn_to_backup_to='arn:aws:s3:::mybucket/backup*.bak',
@type='DIFFERENTIAL',
@number_of_files=4;
```

Example de backup de vários arquivos com criptografia

```
exec msdb.dbo.rds_backup_database
@source_db_name='mydatabase',
@s3_arn_to_backup_to='arn:aws:s3:::mybucket/backup*.bak',
@kms_master_key_arn='arn:aws:kms:us-east-1:123456789012:key/AKIAIOSFODNN7EXAMPLE',
@number_of_files=4;
```

Example de backup de vários arquivos com substituição do S3

```
exec msdb.dbo.rds_backup_database
@source_db_name='mydatabase',
@s3_arn_to_backup_to='arn:aws:s3:::mybucket/backup*.bak',
@overwrite_s3_backup_file=1,
@number_of_files=4;
```

Example de backup com tamanho de bloco

```
exec msdb.dbo.rds_backup_database
@source_db_name='mydatabase',
@s3_arn_to_backup_to='arn:aws:s3:::mybucket/backup*.bak',
@block_size=512;
```

Example de backup de vários arquivos com **`@max_transfer_size`** e **`@buffer_count`**

```
exec msdb.dbo.rds_backup_database
@source_db_name='mydatabase',
@s3_arn_to_backup_to='arn:aws:s3:::mybucket/backup*.bak',
@number_of_files=4,
@max_transfer_size=4194304,
@buffer_count=10;
```

Example de backup de arquivo único com o parâmetro @number_of_files

Este exemplo gera um arquivo de backup chamado `backup*.bak`.

```
exec msdb.dbo.rds_backup_database
@source_db_name='mydatabase',
@s3_arn_to_backup_to='arn:aws:s3:::mybucket/backup*.bak',
@number_of_files=1;
```

Restaurar um banco de dados

Para fazer restaurar seu banco de dados, chame o procedimento armazenado `rds_restore_database`. O Amazon RDS cria um snapshot inicial do banco de dados após a conclusão da tarefa de restauração, e o banco de dados é aberto.

Uso

```
exec msdb.dbo.rds_restore_database
 @restore_db_name='database_name',
 @s3_arn_to_restore_from='arn:aws:s3:::bucket_name/file_name.extension',
 @with_norecovery=0|1,
 [@kms_master_key_arn='arn:aws:kms:region:account-id:key/key-id'],
        [@block_size=512|1024|2048|4096|8192|16384|32768|65536],
        [@max_transfer_size=n],
        [@buffer_count=n],
 [@type='DIFFERENTIAL|FULL'];
```

Os seguintes parâmetros são obrigatórios:

- `@restore_db_name` – o nome do banco de dados a ser restaurado. Os nomes de banco de dados são exclusivos. Não é possível restaurar um banco de dados com o mesmo nome de um banco de dados já existente.

- `@s3_arn_to_restore_from` – o ARN que indica o prefixo do Amazon S3 e os nomes dos arquivos de backup usados para restaurar o banco de dados.
  - Para o backup de um único arquivo, forneça o nome inteiro do arquivo.
  - Para um backup de vários arquivos, forneça o prefixo que os arquivos têm em comum e depois acrescente um sufixo com um asterisco (`*`).
  - Se `@s3_arn_to_restore_from` estiver vazio, a seguinte mensagem de erro será retornada: O prefixo do ARN do S3 não pode estar vazio.

O seguinte parâmetro é necessário para restaurações diferenciais, mas opcional para restaurações completas:

- `@with_norecovery` – a cláusula de recuperação a ser usada para a operação de restauração.
  - Defina como `0` para restauração com RECOVERY. Nesse caso, o banco de dados está on-line após a restauração.
  - Defina como 1 para restauração com NORECOVERY. Nesse caso, o banco de dados permanece no estado RESTORING após a conclusão da tarefa de restauração. Com essa abordagem, você pode fazer restaurações diferenciais posteriores.
  - Para restaurações DIFFERENTIAL, especifique `0` ou 1.
  - Para restaurações `FULL`, esse valor padrão é `0`.

Os seguintes parâmetros são opcionais:

- `@kms_master_key_arn`: se você criptografou o arquivo de backup, será a chave do KMS a ser usada para descriptografar o arquivo.

  Quando você especifica uma chave do KMS, é usada a criptografia no lado do cliente.
- `@type` – o tipo de restauração. Os tipos válidos são `DIFFERENTIAL` e `FULL`. O valor padrão é `FULL`.
- `@block_size`: tamanho do bloco (em bytes) especificando o tamanho do bloco físico para operações de backup. Os valores válidos são 512, 1.024, 2.048, 4.096, 8.192, 16.384, 32.768 e 65.536
- `@max_transfer_size`: o tamanho máximo de transferência indica o limite máximo do volume de dados (em bytes) transmitido por operação de E/S durante o processo de backup. Os valores válidos são múltiplos de 65536 bytes (64 KB) até 4194304 bytes (4 MB).

- `@buffer_count`: número total de buffers de E/S a serem usados no processo de backup.

> **Note**
>
> Para restaurações diferenciais, o banco de dados deve estar no estado RESTORING ou uma tarefa de restauração com NORECOVERY já deve existir.
> Você não pode restaurar backups diferenciais posteriores enquanto o banco de dados estiver on-line.
> Não é possível enviar uma tarefa de restauração para um banco de dados que já tenha uma tarefa de restauração pendente com RECOVERY.
> Não há suporte para restaurações completas com NORECOVERY e restaurações diferenciais em instâncias multi-AZ.
> Restaurar um banco de dados em uma instância multi-AZ com réplicas de leitura é semelhante a restaurar um banco de dados em uma instância multi-AZ. Você não precisa executar nenhuma ação adicional para restaurar um banco de dados em uma réplica.

Exemplos

Example de restauração de único arquivo

```
exec msdb.dbo.rds_restore_database
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/backup1.bak';
```

Example de restauração de vários arquivos

Para evitar erros ao restaurar vários arquivos, certifique-se de que todos os arquivos de backup tenham o mesmo prefixo e que nenhum outro arquivo use esse prefixo.

```
exec msdb.dbo.rds_restore_database
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/backup*';
```

Example de restauração completa de banco de dados com RECOVERY

Os três exemplos a seguir executam a mesma tarefa, restauração completa com RECOVERY.

```
exec msdb.dbo.rds_restore_database
```

```
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/backup1.bak';
```

```
exec msdb.dbo.rds_restore_database
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/backup1.bak',
[@type='DIFFERENTIAL|FULL'];
```

```
exec msdb.dbo.rds_restore_database
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/backup1.bak',
@type='FULL',
@with_norecovery=0;
```

Example de restauração completa de banco de dados com criptografia

```
exec msdb.dbo.rds_restore_database
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/backup1.bak',
@kms_master_key_arn='arn:aws:kms:us-east-1:123456789012:key/AKIAIOSFODNN7EXAMPLE';
```

Example de restauração com tamanho de bloco

```
exec msdb.dbo.rds_restore_database
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/backup1.bak',
@block_size=512;
```

Example da restauração de vários arquivos com @max_transfer_size e @buffer_count

```
exec msdb.dbo.rds_restore_database
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/backup*',
@max_transfer_size=4194304,
@buffer_count=10;
```

Example de restauração completa de banco de dados com NORECOVERY

```
exec msdb.dbo.rds_restore_database
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/backup1.bak',
```

```
@type='FULL',
@with_norecovery=1;
```

Example de restauração diferencial com NORECOVERY

```
exec msdb.dbo.rds_restore_database
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/backup1.bak',
@type='DIFFERENTIAL',
@with_norecovery=1;
```

Example de restauração diferencial com RECOVERY

```
exec msdb.dbo.rds_restore_database
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/backup1.bak',
@type='DIFFERENTIAL',
@with_norecovery=0;
```

Restaurar um log

Para restaurar seu log, chame o procedimento armazenado `rds_restore_log`.

Uso

```
exec msdb.dbo.rds_restore_log
 @restore_db_name='database_name',
 @s3_arn_to_restore_from='arn:aws:s3:::bucket_name/log_file_name.extension',
 [@kms_master_key_arn='arn:aws:kms:region:account-id:key/key-id'],
 [@with_norecovery=0|1],
 [@stopat='datetime'],
 [@block_size=512|1024|2048|4096|8192|16384|32768|65536],
        [@max_transfer_size=n],
        [@buffer_count=n];
```

Os seguintes parâmetros são obrigatórios:

- `@restore_db_name` – o nome do banco de dados cujo log deve ser restaurado.
- `@s3_arn_to_restore_from` – o ARN indicando o prefixo do Amazon S3 e o nome do arquivo de log usado para restaurar o log. O arquivo pode ter qualquer extensão, mas a extensão `.trn` costuma ser usada.

Se `@s3_arn_to_restore_from` estiver vazio, a seguinte mensagem de erro será retornada: O prefixo do ARN do S3 não pode estar vazio.

Os seguintes parâmetros são opcionais:

- `@kms_master_key_arn`: se você criptografou o log, será a chave do KMS a ser usada para descriptografar o log.
- `@with_norecovery` – a cláusula de recuperação a ser usada para a operação de restauração. Esse valor padrão é `1`.
  - Defina como `0` para restauração com RECOVERY. Nesse caso, o banco de dados está on-line após a restauração. Você não pode restaurar backups de log adicionais enquanto o banco de dados estiver on-line.
  - Defina como `1` para restauração com NORECOVERY. Nesse caso, o banco de dados permanece no estado RESTORING após a conclusão da tarefa de restauração. Com essa abordagem, você pode fazer restaurações de log posteriores.
- `@stopat` – um valor que especifica que o banco de dados seja restaurado ao seu estado na data e hora especificadas (no formato de data e hora). Somente registros de log de transações gravados antes da data e hora especificadas são aplicados ao banco de dados.

  Se esse parâmetro não for especificado (é NULL), o log completo será restaurado.
- `@block_size`: tamanho do bloco (em bytes) especificando o tamanho do bloco físico para operações de backup. Os valores válidos são 512, 1.024, 2.048, 4.096, 8.192, 16.384, 32.768 e 65.536
- `@max_transfer_size`: o tamanho máximo de transferência indica o limite máximo do volume de dados (em bytes) transmitido por operação de E/S durante o processo de backup. Os valores válidos são múltiplos de 65536 bytes (64 KB) até 4194304 bytes (4 MB).
- `@buffer_count`: número total de buffers de E/S a serem usados no processo de backup.

> **Note**
>
> Para restaurações de log, o banco de dados deve estar em um estado de restauração ou uma tarefa de restauração com NORECOVERY já deve existir.
> Você não pode restaurar backups de log enquanto o banco de dados estiver on-line.

Não é possível enviar uma tarefa de restauração de log em um banco de dados que já tenha uma tarefa de restauração pendente com RECOVERY.
Não há suporte para restaurações de log em instâncias multi-AZ.

Exemplos

Example de restauração de log

```
exec msdb.dbo.rds_restore_log
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/mylog.trn';
```

Example de restauração de log com criptografia

```
exec msdb.dbo.rds_restore_log
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/mylog.trn',
@kms_master_key_arn='arn:aws:kms:us-east-1:123456789012:key/AKIAIOSFODNN7EXAMPLE';
```

Example de restauração de log com NORECOVERY

Os dois exemplos a seguir executam a mesma tarefa, restauração de log com NORECOVERY.

```
exec msdb.dbo.rds_restore_log
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/mylog.trn',
@with_norecovery=1;
```

```
exec msdb.dbo.rds_restore_log
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/mylog.trn';
```

Example de restauração com tamanho de bloco

```
exec msdb.dbo.rds_restore_log
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/mylog.trn',
@block_size=512;
```

Example de restauração de log com RECOVERY

```
exec msdb.dbo.rds_restore_log
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/mylog.trn',
@with_norecovery=0;
```

Example de restauração de log com cláusula STOPAT

```
exec msdb.dbo.rds_restore_log
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/mylog.trn',
@with_norecovery=0,
@stopat='2019-12-01 03:57:09';
```

Concluir uma restauração de banco de dados

Se a última tarefa de restauração no banco de dados tiver sido executada usando `@with_norecovery=1`, o estado do banco de dados será RESTORING. Abra esse banco de dados para operação normal usando o procedimento armazenado `rds_finish_restore`.

Uso

```
exec msdb.dbo.rds_finish_restore @db_name='database_name';
```

> **Note**
>
> Para usar essa abordagem, o estado do banco de dados deve ser RESTORING sem nenhuma tarefa de restauração pendente.
> Não há suporte para o procedimento `rds_finish_restore` em instâncias multi-AZ.
> Para concluir a restauração do banco de dados, use o login mestre. Ou use o login do usuário que realizou a última restauração do banco de dados ou faça login com NORECOVERY.

Como trabalhar com bancos de dados parcialmente restaurados

Descartando um banco de dados parcialmente restaurado

Para descartar um banco de dados parcialmente restaurado (deixado no estado RESTORING), use o procedimento armazenado `rds_drop_database`.

```
exec msdb.dbo.rds_drop_database @db_name='database_name';
```

> **Note**
>
> Não é possível enviar uma solicitação de banco de dados DROP para um banco de dados que já tenha uma tarefa de restauração pendente, tampouco concluir a tarefa de restauração. Para descartar o banco de dados, use o login mestre: Ou use o login do usuário que realizou a última restauração do banco de dados ou faça login com NORECOVERY.

Restauração de snapshot e comportamento de recuperação em um ponto anterior no tempo para bancos de dados parcialmente restaurados

Os bancos de dados parcialmente restaurados na instância de origem (deixados no estado RESTORING) são descartados da instância de destino durante a restauração do snapshot e a recuperação em um ponto anterior no tempo.

Cancelar uma tarefa

Para cancelar uma tarefa de backup ou restauração, chame o procedimento armazenado `rds_cancel_task`.

> **Note**
>
> Você não pode cancelar uma tarefa FINISH_RESTORE.

Uso

```
exec msdb.dbo.rds_cancel_task @task_id=ID_number;
```

O seguinte parâmetro é obrigatório:

- `@task_id` – o ID da tarefa a ser cancelada. Você pode obter o ID da tarefa chamando `rds_task_status`.

Acompanhar o status de tarefas

Para acompanhar o status de suas tarefas de backup e restauração, chame o procedimento armazenado `rds_task_status`. Se você não fornecer nenhum parâmetro, o procedimento armazenado retornará o status de todas as tarefas. O status das tarefas é atualizado aproximadamente a cada dois minutos. O histórico de tarefas é mantido por 36 dias.

Uso

```
exec msdb.dbo.rds_task_status
 [@db_name='database_name'],
 [@task_id=ID_number];
```

Os seguintes parâmetros são opcionais:

- `@db_name` – o nome do banco de dados do qual o status será exibido.
- `@task_id` – o nome do ID da tarefa do qual o status será exibido.

Exemplos

Example de listagem do status de uma tarefa específica

```
exec msdb.dbo.rds_task_status @task_id=5;
```

Example de listagem do status de uma tarefa e banco de dados específicos

```
exec msdb.dbo.rds_task_status
@db_name='my_database',
@task_id=5;
```

Example de listagem de todas as tarefas e seus status em um banco de dados específico

```
exec msdb.dbo.rds_task_status @db_name='my_database';
```

Example de listagem de todas as tarefas e seus status na instância atual

```
exec msdb.dbo.rds_task_status;
```

Resposta

O procedimento armazenado `rds_task_status` retorna as colunas a seguir.

Coluna	Descrição
`task_id`	O ID da tarefa.

`task_type`	Tipo de tarefa, dependendo dos parâmetros de entrada, conforme segue:
	• Para tarefas de backup:
	  • BACKUP_DB – backup completo do banco de dados
	  • BACKUP_DB_DIFFERENTIAL – backup diferencial de banco de dados
	• Para tarefas de restauração:
	  • RESTORE_DB – restauração completa de banco de dados com RECOVERY
	  • RESTORE_DB_NORECOVERY – restauração completa de banco de dados com NORECOVERY
	  • RESTORE_DB_DIFFERENTIAL – restauração diferencial de banco de dados com RECOVERY
	  • RESTORE_DB_DIFFERENTIAL_NORECOVERY – restauração diferencial de banco de dados com NORECOVERY
	  • RESTORE_DB_LOG – restauração de log com RECOVERY
	  • RESTORE_DB_LOG_NORECOVERY – restauração de log com NORECOVERY

Coluna	Descrição
	• Para tarefas que concluem uma restauração: • FINISH_RESTORE – concluir a restauração e abrir o banco de dados Amazon RDS cria um snapshot inicial do banco de dados após ser aberto mediante a conclusão das seguintes tarefas de restauração: • RESTORE_DB • RESTORE_DB_DIFFERENTIAL • RESTORE_DB_LOG • FINISH_RESTORE
`database_name`	O nome do banco de dados ao qual a tarefa está associada.
`% complete`	O progresso da tarefa em valor de porcentagem.
`duration (mins)`	A quantidade de tempo gasta na tarefa, em minutos.

Coluna	Descrição
`lifecycle`	O status da tarefa. Os possíveis status são os seguintes: • `CREATED` – assim que você chamar `rds_backup_database` ou `rds_restore_database` , uma tarefa é criada e o status é definido para `CREATED`. • `IN_PROGRESS` – após iniciar uma tarefa de backup ou restauração, o status será definido como `IN_PROGRESS` . Pode demorar até 5 minutos para que o status mude de `CREATED` para `IN_PROGRESS` . • `SUCCESS` – após concluir uma tarefa de backup ou restauração, o status será definido como `SUCCESS`. • `ERROR` – se houver falha em uma tarefa de backup ou restauração, o status será definido como `ERROR`. Para obter mais informações sobre o erro, consulte a coluna `task_info` . • `CANCEL_REQUESTED` – assim que você chamar `rds_cancel_task` , o status da tarefa será definido como `CANCEL_REQUESTED` . • `CANCELLED` – após uma tarefa ter sido cancelada com sucesso, o status da tarefa será definido como `CANCELLED` .
`task_info`	Informações adicionais sobre a tarefa. No caso de erro durante o backup ou a restauração de um banco de dados, esta coluna contém as informações sobre o erro. Para obter uma lista de possíveis erros e estratégias de mitigação, consulte Solução de problemas.
`last_updated`	A data e hora em que o status da tarefa foi atualizado pela última vez. O status é atualizado a cada 5% de progresso.
`created_at`	A data e hora em que a tarefa foi criada.
`S3_object_arn`	O ARN indicando o prefixo do Amazon S3 e o nome do arquivo que está sendo copiado ou restaurado.

Coluna	Descrição
`overwrite_s3_backup_file`	O valor do parâmetro `@overwrite_s3_backup_file` especificado ao chamar uma tarefa de backup. Para obter mais informações, consulte Realização de backup de um banco de dados
`KMS_master_key_arn`	O ARN para a chave do KMS usada para criptografia (para backup) e descriptografia (para restauração).
`filepath`	Não aplicável a tarefas de backup e restauração nativos
`overwrite_file`	Não aplicável a tarefas de backup e restauração nativos

Compactar arquivos de backup

Para economizar espaço no bucket do Amazon S3, você pode compactar seus arquivos de backup. Para obter mais informações sobre como compactar arquivos de backup, consulte Compactação de backup na documentação da Microsoft.

As seguintes edições de banco de dados oferecem suporte à compactação de arquivos de backup:

- Microsoft SQL Server Enterprise Edition
- Microsoft SQL Server Standard Edition

Para ativar a compactação dos arquivos de backup, execute o seguinte código:

```
exec rdsadmin.dbo.rds_set_configuration 'S3 backup compression', 'true';
```

Para desativar a compactação dos arquivos de backup, execute o seguinte código:

```
exec rdsadmin.dbo.rds_set_configuration 'S3 backup compression', 'false';
```

Solução de problemas

Os problemas a seguir podem ocorrer ao usar o backup e a restauração nativos.

Problema	Sugestões de solução de problemas
A opção de backup/restauração do banco de dados ainda não está habilitada ou está em processo de habilitação. Tente novamente mais tarde.	Verifique se você adicionou a opção `SQLSERVER_BACKUP_RESTORE` ao grupo de opções de banco de dados associado à sua instância de banco de dados. Para obter mais informações, consulte Adicionar a opção de backup e restauração nativos.
A permissão EXECUTE foi negada no objeto "*`rds_backup_database`*", no banco de dados "msdb" e no esquema "dbo".	Verifique se você está usando o usuário mestre ao executar o procedimento armazenado. Se você encontrar esse erro mesmo depois de fazer login como usuário mestre, pode ser que as permissões do usuário administrador estejam desalinhadas. Para redefinir o usuário mestre, use o AWS Management Console. Consulte Redefinir a associação ao perfil db_owner para o usuário principal no Amazon RDS para SQL Server.
A permissão EXECUTE foi negada no objeto "*`rds_restore_database`*", no banco de dados "msdb" e no esquema "dbo".	Verifique se você está usando o usuário mestre ao executar o procedimento armazenado. Se você encontrar esse erro mesmo depois de fazer login como usuário mestre, pode ser que as permissões do usuário administrador estejam desalinhadas. Para redefinir o usuário mestre, use o AWS Management Console. Consulte Redefinir a associação ao perfil db_owner para o usuário principal no Amazon RDS para SQL Server.
Acesso negado	O processo de backup ou restauração não pode acessar o arquivo de backup. Isso geralmente é causado por problemas como o seguinte: • Fazer referência ao bucket incorreto. Fazer referência ao bucket usando um formato incorreto. Fazer referência a um nome de arquivo sem usar o ARN. • Permissões incorretas no arquivo de bucket. Por exemplo, se ele for criado por uma conta diferente que esteja tentando acessá-lo agora, adicione as permissões corretas.

Problema	Sugestões de solução de problemas
	• Uma política IAM incorreta ou incompleta. Sua função do IAM deve incluir todos os elementos necessários, incluindo, por exemplo, a versão correta. Eles estão destacados em Importar e exportar bancos de dados do SQL Server usando backup e restauração nativos.
O BANCO DE DADOS DE BACKUP COM COMPACTAÇÃO não tem suporte na edição <edition_name>	A compactação de arquivos de backup só tem suporte no Microsoft SQL Server Enterprise Edition e Standard Edition. Para obter mais informações, consulte Compactar arquivos de backup.
A chave <ARN> não existe	Você tentou restaurar um backup criptografado, mas não forneceu uma chave de criptografia válida. Confira sua chave de criptografia e tente novamente. Para obter mais informações, consulte Restaurar um banco de dados.

Problema	Sugestões de solução de problemas
Reemita a tarefa com o tipo correto e sobrescreva a propriedade	Se você tentar fazer backup de seu banco de dados e fornecer o nome de um arquivo que já existe, mas definir a propriedade de substituição como falso, a operação de salvar falhará. Para corrigir esse erro, forneça o nome de um arquivo que ainda não exista ou defina a propriedade de substituição como verdadeira. Para obter mais informações, consulte Realização de backup de um banco de dados. Há casos em que você desejava restaurar seu banco de dados, mas chamou o procedimento armazenado `rds_backup_database` por engano. Nesse caso, chame o procedimento armazenado `rds_restore_database` . Para obter mais informações, consulte Restaurar um banco de dados. Se você pretendia restaurar seu banco de dados e chamou o procedimento armazenado `rds_restore_database` , confira se você forneceu o nome de um arquivo de backup válido. Para obter mais informações, consulte Usar o backup e restauração nativos.
Especifique um bucket que esteja na mesma região que a instância do RDS	Você não pode fazer backup ou restaurar de um bucket do Amazon S3 em uma região da AWS diferente de sua instância de banco de dados do Amazon RDS. Você pode usar a replicação do Amazon S3 para copiar o arquivo de backup na região da AWS correta. Para obter mais informações, consulte Replicação entre regiões na documentação do Amazon S3.
O bucket especificado não existe	Confirme que você forneceu um o ARN correto para o bucket e o arquivo, no formato correto. Para obter mais informações, consulte Usar o backup e restauração nativos.

Problema	Sugestões de solução de problemas
O usuário <ARN> não está autorizado a executar <kms action> no recurso <ARN>	Você solicitou uma operação criptografada, mas não forneceu as permissões de AWS KMS corretas;. Verifique se você tem as permissões corretas, ou adicione-as caso não tenha. Para obter mais informações, consulte Configurar o backup e a restauração nativos.
A tarefa Restore (Restaurar) não restaura mais de 10 arquivos de backup). Reduza o número de arquivos correspondentes e tente novamente.	Reduza o número de arquivos a partir dos quais você está tentando fazer a restauração. Você poderá tornar cada arquivo individual maior, se for necessário.
O banco de dados "*database_name* " já existe. Dois bancos de dados que diferem apenas por caso ou acento não são permitidos. Escolha um nome de banco de dados diferente.	Não é possível restaurar um banco de dados com o mesmo nome de um banco de dados já existente. Os nomes de banco de dados são exclusivos.

Importar e exportar dados do SQL Server usando outros métodos

Em seguida, você pode encontrar informações sobre o uso de snapshots para importar seus dados do Microsoft SQL Server para o Amazon RDS. Você também pode encontrar informações sobre o uso de snapshots para exportar seus dados de uma instância de banco de dados do RDS executando o SQL Server.

Se o seu cenário oferecer suporte, é mais fácil mover dados dentro e fora do Amazon RDS usando a funcionalidade nativa de backup e restauração. Para obter mais informações, consulteImportar e exportar bancos de dados do SQL Server usando backup e restauração nativos

> **Note**
>
> O Amazon RDS para Microsoft SQL Server não é compatível com a importação de dados no banco de dados do `msdb`.

Importar dados para o RDS para SQL Server usando um snapshot

Para importar dados para uma instância de banco de dados SQL Server usando um snapshot

1. Criar uma instância de banco de dados. Para obter mais informações, consulte Criar uma instância de banco de dados do Amazon RDS.
2. Evite que os aplicativos acessem a instância de banco de dados de destino.

   Se você impedir o acesso à sua instância de banco de dados enquanto estiver importando dados, a transferência de dados será mais rápida. Além disso, você não precisa se preocupar com conflitos enquanto os dados estão sendo carregados se outros aplicativos não puderem gravar na instância de banco de dados ao mesmo tempo. Se algo der errado e você tiver que reverter para um snapshot anterior do banco de dados, as únicas alterações que você perderá serão os dados importados. Você poderá importar esses dados novamente depois de resolver o problema.

   Para obter informações sobre como controlar o acesso à sua instância de banco de dados, consulte Controlar acesso com grupos de segurança.
3. Crie um snapshot do banco de dados de destino.

   Se o banco de dados de destino já estiver preenchido com dados, recomendamos que você tire um snapshot dele antes de importar os dados. Se algo der errado com a importação de dados

ou se quiser descartar as alterações, você poderá restaurar o banco de dados para seu estado anterior usando esse snapshot. Para obter informações sobre snapshots de banco de dados, consulte Criar um snapshot de banco de dados para uma instância de banco de dados single-AZ para o Amazon RDS.

> **Note**
>
> Quando você tira um snapshot de banco de dados, as operações de E/S nesse banco de dados são suspensas por um momento (milissegundos) enquanto o backup está em andamento.

4. Desabilite os backups automatizados no banco de dados de destino.

   Desabilitar backups automatizados na instância de banco de dados de destino melhora a performance enquanto você estiver importando seus dados, pois o Amazon RDS não registra transações quando backups automáticos estão desabilitados. No entanto, existem algumas coisas a considerar. Backups automatizados são necessários para executar uma recuperação point-in-time. Portanto, você não poderá restaurar o banco de dados para um ponto específico ao mesmo tempo em que estiver importando dados. Além disso, todos os backups automatizados que foram criados na instância de banco de dados serão apagados, a menos que você opte por mantê-los.

   Escolher manter os backups automatizados pode ajudar a proteger contra a exclusão acidental de dados. O Amazon RDS também salva as propriedades da instância de banco de dados juntamente com cada backup automatizado para facilitar a recuperação. Usar essa opção permite que você restaure uma instância de banco de dados excluída até um ponto específico no período de retenção do backup mesmo depois de excluí-la. Os backups automatizados são excluídos automaticamente ao final da janela de backup especificada, da mesma forma que para uma instância de banco de dados ativa.

   Você ainda poderá usar os snapshots anteriores para recuperar o banco de dados, e todos os snapshots que tiverem sido tirados permanecem disponíveis. Para obter informações sobre backups automatizados, consulte Introdução aos backups.

5. Desabilite restrições de chaves estrangeiras, se aplicável.

   Se você precisar desabilitar restrições de chaves estrangeiras, poderá fazê-lo com o seguinte script.

```
--Disable foreign keys on all tables
    DECLARE @table_name SYSNAME;
    DECLARE @cmd NVARCHAR(MAX);
    DECLARE table_cursor CURSOR FOR SELECT name FROM sys.tables;

    OPEN table_cursor;
    FETCH NEXT FROM table_cursor INTO @table_name;

    WHILE @@FETCH_STATUS = 0 BEGIN
      SELECT @cmd = 'ALTER TABLE '+QUOTENAME(@table_name)+' NOCHECK CONSTRAINT
 ALL';
      EXEC (@cmd);
      FETCH NEXT FROM table_cursor INTO @table_name;
    END

    CLOSE table_cursor;
    DEALLOCATE table_cursor;

    GO
```

6. Descarte índices, se aplicável.
7. Desabilite triggers, se aplicável.

   Se precisar desabilitar triggers, faça isso com o seguinte script.

```
--Disable triggers on all tables
    DECLARE @enable BIT = 0;
    DECLARE @trigger SYSNAME;
    DECLARE @table SYSNAME;
    DECLARE @cmd NVARCHAR(MAX);
    DECLARE trigger_cursor CURSOR FOR SELECT trigger_object.name trigger_name,
     table_object.name table_name
    FROM sysobjects trigger_object
    JOIN sysobjects table_object ON trigger_object.parent_obj = table_object.id
    WHERE trigger_object.type = 'TR';

    OPEN trigger_cursor;
    FETCH NEXT FROM trigger_cursor INTO @trigger, @table;

    WHILE @@FETCH_STATUS = 0 BEGIN
      IF @enable = 1
         SET @cmd = 'ENABLE ';
```

```
      ELSE
         SET @cmd = 'DISABLE ';

      SET @cmd = @cmd + ' TRIGGER dbo.'+QUOTENAME(@trigger)+' ON
dbo.'+QUOTENAME(@table)+' ';
      EXEC (@cmd);
      FETCH NEXT FROM trigger_cursor INTO @trigger, @table;
   END

   CLOSE trigger_cursor;
   DEALLOCATE trigger_cursor;

   GO
```

8. Consulte a instância SQL Server de origem para quaisquer logons que você queira importar para a instância de banco de dados de destino.

   O SQL Server armazena logons e senhas no banco de dados `master`. Como o Amazon RDS não concede acesso ao banco de dados `master`, não é possível importar logins e senhas diretamente para a sua instância de banco de dados de destino. Em vez disso, você deve consultar o banco de dados `master` na instância do SQL Server de origem para gerar um arquivo de linguagem de definição de dados (DDL). Esse arquivo deve incluir todos os logins e senhas que você queira adicionar à instância de banco de dados de destino. Esse arquivo também deve incluir associações de funções e permissões que você deseja transferir.

   Para ter informações sobre como consultar o banco de dados `master`, consulte Transfer logins and passwords between instances of SQL Server, na Base de Conhecimento da Microsoft.

   A saída do script é outro script que você pode executar na instância de banco de dados de destino. O script no artigo da base de dados de conhecimento possui o seguinte código:

   ```
   p.type IN
   ```

   Sem que `p.type` aparecer, use o seguinte código em vez disso:

   ```
   p.type = 'S'
   ```

9. Importe os dados usando o método em Importar os dados.
10. Conceda acesso para aplicativos à instância de banco de dados de destino.

Quando a importação de dados estiver concluída, você poderá conceder acesso para a instância de banco de dados aos aplicativos que você foram bloqueados durante a importação. Para obter informações sobre como controlar o acesso à sua instância de banco de dados, consulte Controlar acesso com grupos de segurança.

11. Habilite backups automatizados na instância de banco de dados de destino.

    Para obter informações sobre backups automatizados, consulte Introdução aos backups.

12. Habilite restrições de chaves estrangeiras.

    Se você tiver desabilitado restrições de chaves estrangeiras anteriormente, agora poderá habilitá-las com o seguinte script.

```
--Enable foreign keys on all tables
    DECLARE @table_name SYSNAME;
    DECLARE @cmd NVARCHAR(MAX);
    DECLARE table_cursor CURSOR FOR SELECT name FROM sys.tables;

    OPEN table_cursor;
    FETCH NEXT FROM table_cursor INTO @table_name;

    WHILE @@FETCH_STATUS = 0 BEGIN
      SELECT @cmd = 'ALTER TABLE '+QUOTENAME(@table_name)+' CHECK CONSTRAINT ALL';
      EXEC (@cmd);
      FETCH NEXT FROM table_cursor INTO @table_name;
    END

    CLOSE table_cursor;
    DEALLOCATE table_cursor;
```

13. Habilite índices, se aplicável.
14. Habilite triggers, se aplicável.

    Se você tiver desabilitado triggers anteriormente, agora poderá habilitá-los com o seguinte script.

```
--Enable triggers on all tables
    DECLARE @enable BIT = 1;
    DECLARE @trigger SYSNAME;
    DECLARE @table SYSNAME;
    DECLARE @cmd NVARCHAR(MAX);
    DECLARE trigger_cursor CURSOR FOR SELECT trigger_object.name trigger_name,
```

```
    table_object.name table_name
  FROM sysobjects trigger_object
  JOIN sysobjects table_object ON trigger_object.parent_obj = table_object.id
  WHERE trigger_object.type = 'TR';

  OPEN trigger_cursor;
  FETCH NEXT FROM trigger_cursor INTO @trigger, @table;

  WHILE @@FETCH_STATUS = 0 BEGIN
    IF @enable = 1
       SET @cmd = 'ENABLE ';
    ELSE
       SET @cmd = 'DISABLE ';

    SET @cmd = @cmd + ' TRIGGER dbo.'+QUOTENAME(@trigger)+' ON
dbo.'+QUOTENAME(@table)+' ';
    EXEC (@cmd);
    FETCH NEXT FROM trigger_cursor INTO @trigger, @table;
  END

  CLOSE trigger_cursor;
  DEALLOCATE trigger_cursor;
```

Importar os dados

O Microsoft SQL Server Management Studio é um cliente SQL Server gráfico que está incluído em todas as edições do Microsoft SQL Server, exceto a Express Edition. O SQL Server Management Studio Express está disponível na Microsoft como download gratuito. Para encontrar esse download, consulte o site da Microsoft.

> **Note**
>
> O SQL Server Management Studio está disponível apenas como um aplicativo baseado no Windows.

O SQL Server Management Studio inclui as seguintes ferramentas, que são úteis na importação de dados para uma instância de banco de dados SQL Server:

- Assistente para gerar e publicar scripts
- Assistente de importação e exportação

- Cópia em massa

Assistente para gerar e publicar scripts

O Assistente para gerar e publicar scripts cria um script que contém o esquema de um banco de dados, os dados propriamente ditos ou ambos. Você pode gerar um script para um banco de dados na sua implantação local do SQL Server. Em seguida, pode executar esse script para transferir as informações que ele contém para uma instância de banco de dados do Amazon RDS.

> **Note**
>
> Para bancos de dados de 1 GiB ou maiores, é mais eficiente definir scripts apenas para o esquema de banco de dados. Depois, use o Assistente de importação e exportação ou o recurso de cópia em massa do SQL Server para transferir os dados.

Para obter informações detalhadas sobre o Assistente para gerar e publicar scripts, consulte a documentação do Microsoft SQL Server.

No assistente, preste especial atenção às opções avançadas na página Definir Opções de Script para garantir que tudo o que você deseja que o script inclua esteja selecionado. Por exemplo, por padrão, triggers de banco de dados não estão incluídos no script.

Quando o script for gerado e salvo, você poderá usar o SQL Server Management Studio para se conectar à sua instância de banco de dados e executar o script.

Assistente de importação e exportação

O Assistente de importação e exportação cria um pacote especial de Serviços de integração, que você pode usar para copiar dados do seu banco de dados SQL Server local para a instância de banco de dados de destino. O assistente pode filtrar quais tabelas e até mesmo quais tuplas em uma tabela são copiadas para a instância de banco de dados de destino.

> **Note**
>
> O Assistente de importação e exportação funciona bem para conjuntos de dados grandes, mas talvez não seja a maneira mais rápida de exportar dados remotamente da sua implantação local. Para uma maneira ainda mais rápida, considere o recurso de cópia em massa do SQL Server.

Para obter informações detalhadas sobre o Assistente de importação e exportação, consulte a documentação do Microsoft SQL Server.

No assistente, na página Choose a Destination (Selecionar um destino), faça o seguinte:

- Em Server Name (Nome do servidor), digite o nome do endpoint da sua instância de banco de dados.
- Para o modo de autenticação do servidor, escolha Use SQL Server Authentication (Usar autenticação do SQL Server).
- Em User name (Nome do usuário) e Password (Senha), digite as credenciais do usuário mestre criado para a instância de banco de dados.

Cópia em massa

O recurso de cópia em massa do SQL Server é um meio eficiente de copiar dados de um banco de dados de origem para a sua instância de banco de dados. A cópia em massa grava os dados que você especifica em um arquivo de dados, como um arquivo ASCII. Você pode então executar cópias em massa novamente para gravar o conteúdo do arquivo na instância de banco de dados de destino.

Esta seção usa o utilitário bcp, que está incluído em todas as edições do SQL Server. Para obter informações detalhadas sobre operações de importação e exportação em massa, consulte a documentação do Microsoft SQL Server.

> **Note**
>
> Antes de usar a cópia em massa, você deve importar seu esquema de banco de dados para a instância de banco de dados de destino. O Assistente para gerar e publicar scripts, descrito anteriormente neste tópico, é uma excelente ferramenta para esse propósito.

O comando a seguir se conecta à instância local do SQL Server. Ele gera um arquivo delimitado por tabulação de uma tabela especificada no diretório raiz C:\ da sua implantação SQL Server existente. A tabela é especificada pelo nome totalmente qualificado e o arquivo de texto tem o mesmo nome que a tabela que está sendo copiada.

```
bcp dbname.schema_name.table_name out C:\table_name.txt -n -S localhost -U username -
P password -b 10000
```

O código anterior inclui as seguintes opções:

- `-n` especifica que a cópia em massa usa os tipos de dados nativos dos dados a serem copiados.
- `-S` especifica a instância SQL Server com a qual o utilitário bcp se conecta.
- `-U` especifica o nome de usuário da conta que faz login na instância SQL Server.
- `-P` especifica a senha para o usuário especificada por `-U`.
- `-b` especifica o número de linhas por lote de dados importados.

> **Note**
>
> Pode haver outros parâmetros importantes para a sua situação de importação. Por exemplo, talvez você precise do parâmetro `-E` que pertence a valores de identidade. Para obter mais informações; consulte a descrição completa da sintaxe da linha de comando para o utilitário bcp na documentação do Microsoft SQL Server.

Por exemplo, suponha que um banco de dados chamado `store`, que usa o esquema padrão, dbo, contém uma tabela chamada `customers`. A conta de usuário `admin`, com a senha `insecure`, copia 10.000 linhas da tabela `customers` para um arquivo chamado `customers.txt`.

```
bcp store.dbo.customers out C:\customers.txt -n -S localhost -U admin -P insecure -b
 10000
```

Depois de gerar o arquivo de dados, você poderá fazer upload dos dados para a sua instância de banco de dados usando um comando semelhante. Antes disso, crie o banco de dados e o esquema na instância de banco de dados de destino. Em seguida, use o argumento `in` para especificar um arquivo de entrada em vez de `out` para especificar um arquivo de saída. Em vez de usar localhost para especificar a instância local SQL Server, especifique o endpoint da sua instância de banco de dados. Se você usar uma porta diferente de 1433, especifique isso também. O nome de usuário e a senha serão aqueles do usuário mestre e a senha da sua instância de banco de dados. A sintaxe é a seguinte.

```
bcp dbname.schema_name.table_name
     in C:\table_name.txt -n -S endpoint,port -U master_user_name -
P master_user_password -b 10000
```

Para continuar com o exemplo anterior, suponha que o nome do usuário mestre seja `admin` e a senha seja `insecure`. O endpoint da instância de banco de dados é `rds.ckz2kqd4qsn1.us-east-1.rds.amazonaws.com`, e você usa a porta 4080. O comando é o seguinte.

```
bcp store.dbo.customers in C:\customers.txt -n -S rds.ckz2kqd4qsn1.us-
east-1.rds.amazonaws.com,4080 -U admin -P insecure -b 10000
```

> **Note**
>
> Especifique uma senha diferente do prompt mostrado aqui como prática recomendada de segurança.

Exportar dados do RDS para SQL Server

Você pode escolher uma das seguintes opções para exportar dados de uma instância de banco de dados RDS for SQL Server:

- Backup de banco de dados nativo usando um arquivo de backup completo (.bak) – o uso de arquivos .bak para bancos de dados de backup é fortemente otimizado e geralmente é a maneira mais rápida de exportar dados. Para obter mais informações, consulte Importar e exportar bancos de dados do SQL Server usando backup e restauração nativos.
- Assistente de importação e exportação do SQL Server – para obter mais informações, consulte Assistente de importação e exportação do SQL Server.
- Utilitário bcp e Assistente para gerar e publicar scripts do SQL Server – para obter mais informações, consulte Utilitário bcp e Assistente para gerar e publicar scripts do SQL Server.

Assistente de importação e exportação do SQL Server

Você pode usar o Assistente de importação e exportação do SQL Server para copiar uma ou mais tabelas, visualizações ou consultas da sua instância de banco de dados do RDS for SQL Server para outro armazenamento de dados. Essa será a melhor escolha se o armazenamento de dados de destino não for o SQL Server. Para obter mais informações, consulte Assistente de importação e exportação do SQL Server na documentação do SQL Server.

O Assistente de importação e exportação do SQL Server está disponível como parte do Microsoft SQL Server Management Studio. Esse cliente SQL Server gráfico está incluído em todas as edições do Microsoft SQL Server, exceto a Express Edition. O SQL Server Management Studio

está disponível apenas como um aplicativo baseado no Windows. O SQL Server Management Studio Express está disponível na Microsoft como download gratuito. Para encontrar esse download, consulte o site da Microsoft.

Para usar o Assistente de importação e exportação do SQL Server para exportar dados

1. No SQL Server Management Studio, conecte-se à sua instância de banco de dados do RDS for SQL Server. Para obter detalhes sobre como fazer isso, consulte Conectar-se à instância de banco de dados do Microsoft SQL Server.
2. No Object Explorer (Navegador de objetos), expanda Databases (Bancos de dados), abra o menu de contexto (botão direito do mouse) para o banco de dados de origem, escolha Tasks (Tarefas) e depois escolha Export Data (Exportar dados). O assistente é exibido.
3. Na página Choose a Data Source (Selecionar uma fonte de dados), faça o seguinte:
   a. Em Data source (Fonte de dados), selecione **`SQL Server Native Client 11.0`**.
   b. Verifique se o campo Server name (Nome do servidor) mostra o endpoint da sua instância de banco de dados RDS for SQL Server.
   c. Selecione Use SQL Server Authentication (Usar autenticação do SQL Server). Em User name (Nome do usuário) e Password (Senha), digite o nome do usuário mestre e a senha da instância do banco de dados.
   d. Verifique se a caixa Database (Banco de dados) mostra o banco de dados a partir do qual você deseja exportar dados.
   e. Escolha Next (Próximo).
4. Na página Choose a Destination (Selecionar um destino), faça o seguinte:
   a. Em Destination (Destino), selecione **`SQL Server Native Client 11.0`**.

> **Note**
>
> Outras fontes de dados de destino estão disponíveis. Elas incluem provedores de dados .NET Framework, provedor de banco de dados OLE, provedores SQL Server Native Client, provedores ADO.NET, Microsoft Office Excel, Microsoft Office Access e a origem de arquivo simples. Se você optar por uma dessas fontes de dados, ignore o restante da etapa 4. Para obter detalhes sobre as informações de conexão a serem fornecidas a seguir, consulte Escolher um destino, na documentação do SQL Server.

b. Para Server name (Nome do servidor), digite o nome do servidor da instância de banco de dados SQL Server de destino.

c. Escolha o tipo de autenticação apropriado. Digite um nome de usuário e uma senha, se necessário.

d. Para Database (Banco de dados), escolha o nome do banco de dados de destino ou escolha New (Novo) para criar um novo banco de dados para conter os dados exportados.

   Se você escolher New (Novo), consulte Criar um banco de dados, na documentação do SQL Server, para obter detalhes sobre as informações de banco de dados que devem ser fornecidas.

e. Escolha Next (Próximo).

5. Na página Table Copy or Query (Cópia ou consulta da tabela), escolha Copy data from one or more tables or views (Copiar dados de uma ou mais tabelas ou visualizações) ou Write a query to specify the data to transfer (Gravar uma consulta para especificar os dados a serem transferidos). Escolha Next (Próximo).
6. Se você escolher Write a query to specify the data to transfer (Gravar uma consulta para especificar os dados a serem transferidos), verá a página Provide a Source Query (Fornecer uma consulta da fonte). Digite ou cole uma consulta SQL e escolha Parse (Analisar) para verificá-la. Quando a consulta for validada, escolha Next (Próximo).
7. Na página Select Source Tables and Views (Selecionar tabelas e visualizações da fonte), faça o seguinte:

   a. Selecione as tabelas e as visualizações que você deseja exportar ou verifique se a consulta fornecida está selecionada.

   b. Escolha Edit Mappings (Editar mapeamentos) e especifique informações de mapeamento de banco de dados e colunas. Para obter mais informações, consulte Mapeamentos de colunas na documentação do SQL Server.

   c. (Opcional) Para ver uma visualização dos dados a serem exportados, selecione a tabela, exibição ou consulta e escolha Preview (Visualizar).

   d. Escolha Next (Próximo).

8. Na página Run Package (Executar pacote), verifique se a opção Run immediately (Executar imediatamente) está selecionada. Escolha Next (Próximo).
9. Na página Complete the Wizard (Concluir o assistente), verifique se os detalhes da exportação de dados são os esperados. Escolha Terminar.

10. Na página The execution was successful (A execução foi bem-sucedida), escolha Close (Fechar).

Utilitário bcp e Assistente para gerar e publicar scripts do SQL Server

Você pode usar o Assistente para gerar e publicar scripts do SQL Server para criar scripts para um banco de dados inteiro ou apenas para objetos selecionados. Você pode executar esses scripts em uma instância de banco de dados SQL Server de destino para recriar os objetos com scripts. Em seguida, você pode usar o utilitário bcp para exportar em massa os dados dos objetos selecionados para a instância do banco de dados de destino. Essa escolha será melhor se você quiser mover um banco de dados inteiro (incluindo objetos que não sejam tabelas) ou grandes quantidades de dados entre duas instâncias de bancos de dados SQL Server. Para obter uma descrição completa da sintaxe da linha de comando do bcp, consulte Utilitário bcp na documentação do Microsoft SQL Server.

O Assistente para gerar e publicar scripts do SQL Server está disponível como parte do Microsoft SQL Server Management Studio. Esse cliente SQL Server gráfico está incluído em todas as edições do Microsoft SQL Server, exceto a Express Edition. O SQL Server Management Studio está disponível apenas como um aplicativo baseado no Windows. O SQL Server Management Studio Express está disponível na Microsoft como download gratuito.

Para usar o Assistente para gerar e publicar scripts e o utilitário bcp do SQL Server para exportar dados

1. No SQL Server Management Studio, conecte-se à sua instância de banco de dados do RDS for SQL Server. Para obter detalhes sobre como fazer isso, consulte Conectar-se à instância de banco de dados do Microsoft SQL Server.
2. No Object Explorer (Navegador de objetos), expanda o nó Databases (Bancos de dados) e selecione o banco de dados para o qual você deseja definir um script.
3. Siga as instruções do Assistente para gerar e publicar scripts na documentação do SQL Server para criar um arquivo de script.
4. No SQL Server Management Studio, conecte-se à sua instância de banco de dados SQL Server de destino.
5. Com a instância de banco de dados SQL Server de destino selecionada no Object Explorer (Navegador de objetos), escolha Open (Abrir) no menu File (Arquivo), escolha File (Arquivo) e abra o arquivo de script.

6. Se você tiver definido um script para o banco de dados inteiro, reveja a instrução CREATE DATABASE no script. Certifique-se de que o banco de dados esteja sendo criado no local e com os parâmetros desejados. Para obter mais informações, consulte CREATE DATABASE na documentação do SQL Server.
7. Se você estiver criando usuários de banco de dados no script, verifique se há logins no servidor na instância de banco de dados de destino para esses usuários. Caso contrário, crie logins para esses usuários. Os comandos com scripts para criar os usuários do banco de dados falharão de outra forma. Para obter mais informações, consulte Criar um login na documentação do SQL Server.
8. Escolha !Execute no menu do SQL Editor para executar o arquivo de script e criar os objetos do banco de dados. Quando o script terminar, verifique se todos os objetos de banco de dados existem conforme o esperado.
9. Use o utilitário bcp para exportar dados da instância de banco de dados do RDS for SQL Server em arquivos. Abra um prompt de comando e digite o seguinte comando.

```
bcp database_name.schema_name.table_name out data_file -n -S aws_rds_sql_endpoint -
U username -P password
```

O código anterior inclui as seguintes opções:

- table_name é o nome de uma das tabelas criadas no banco de dados de destino e agora deseja preencher com dados.
- data_file é o caminho completo e o nome do arquivo de dados a ser criado.
- `-n` especifica que a cópia em massa usa os tipos de dados nativos dos dados a serem copiados.
- `-S` especifica a instância de banco de dados SQL Server da qual exportar.
- `-U` especifica o nome de usuário a ser usado ao se conectar à instância de banco de dados do SQL Server.
- `-P` especifica a senha para o usuário especificada por `-U`.

O seguinte mostra um exemplo de comando .

```
bcp world.dbo.city out C:\Users\JohnDoe\city.dat -n -S sql-jdoe.1234abcd.us-
west-2.rds.amazonaws.com,1433 -U JohnDoe -P ClearTextPassword
```

Repita essa etapa até ter arquivos de dados para todas as tabelas que você deseja exportar.

10. Prepare sua instância de banco de dados de destino para importação em massa de dados, seguindo as instruções em Diretrizes básicas para dados de importação em massa, na documentação do SQL Server.
11. Escolha um método de importação em massa a ser usado depois de considerar a performance e outras preocupações discutidas no tópico Sobre operações de importação e exportação em massa, na documentação do SQL Server.
12. Faça uma importação em massa dos dados dos arquivos de dados criados usando o utilitário bcp. Para fazer isso, siga as instruções em Importar e exportar dados em massa usando o bcp (SQL Server) ou em Usar BULK INSERT ou OPENROWSET(BULK...) para importar dados para o SQL Server, na documentação do SQL Server, dependendo do que você decidiu na etapa 11.

Usar o utilitário BCP do Linux para importar e exportar dados

O utilitário BCP (Bulk Copy Program) oferece uma maneira eficiente de transferir grandes quantidades de dados entre a instância de banco de dados do RDS para SQL Server e os arquivos de dados. Você pode usar o BCP em ambientes Linux para realizar operações de dados em massa, o que o torna útil para migração de dados, processos de ETL e transferências regulares de dados.

O BCP permite a importação de dados de arquivos para tabelas do SQL Server e a exportação de dados de tabelas do SQL Server para arquivos. Isso é particularmente eficaz para transferir dados estruturados em vários formatos, inclusive arquivos de texto delimitados.

Pré-requisitos

Antes de usar o BCP com uma instância de banco de dados do RDS para SQL Server usando o Linux, verifique se você tem o seguinte:

- Um ambiente Linux com conectividade de rede à instância de banco de dados do RDS para SQL Server.
- Ferramentas de linha de comandos do Microsoft SQL Server instaladas no sistema Linux, como:
  - sqlcmd: ferramenta de consulta de linha de comandos do SQL Server
  - bcp: utilitário Bulk Copy Program
- Credenciais válidas para a instância de banco de dados do RDS para SQL Server.
- Acesso à rede configurado por meio de grupos de segurança para permitir conexões na porta do SQL Server (normalmente 1433).
- Permissões de banco de dados apropriadas para as operações que você deseja realizar.

Instalar ferramentas de linha de comandos do SQL Server no Linux

Para usar o BCP no Linux, você precisa instalar as ferramentas de linha de comandos do Microsoft SQL Server. Para obter instruções detalhadas sobre instalação para sua distribuição Linux específica, consulte a seguinte documentação da Microsoft:

- Install sqlcmd and bcp the SQL Server command-line tools on Linux
- bcp utility: referência completa para o utilitário BCP

Após a instalação, confirme se as ferramentas estão disponíveis em seu PATH executando:

```
bcp -v
sqlcmd -?
```

Exportar dados do RDS para SQL Server

Você pode usar o BCP para exportar dados da instância de banco de dados do RDS para SQL Server para arquivos no sistema Linux. Isso é útil para criar backups, analisar dados ou preparar dados para migração.

Sintaxe básica de exportação

A sintaxe básica para exportar dados usando o BCP é:

```
bcp database.schema.table out output_file -S server_name -U username -P password
 [options]
```

Em que:

- `database.schema.table`: o nome totalmente qualificado da tabela.
- `output_file`: o caminho e o nome do arquivo de saída.
- `server_name`: o endpoint do RDS para SQL Server.
- `username`: o nome de usuário do banco de dados.
- `password`: a senha do banco de dados.

Exemplo de exportação

O seguinte exemplo exporta dados de uma tabela denominada `customers` no banco de dados `sales`:

```
bcp sales.dbo.customers out /home/user/customers.txt \
    -S mydb.cluster-abc123.us-east-1.rds.amazonaws.com \
    -U admin \
    -P mypassword \
    -c \
    -t "|" \
    -r "\n"
```

Esse comando:

- Exporta dados da tabela `customers`.
- Salva a saída de `/home/user/customers.txt`.
- Usa o formato de caracteres (`-c`).
- Usa pipe (|) como delimitador de campo (`-t "|"`).
- Usa nova linha como delimitador de linha (`-r "\n"`).

Importar dados para o RDS para SQL Server

Você pode usar o BCP para exportar dados de arquivos em seu sistema Linux para sua instância de banco de dados do RDS para SQL Server. Isso é útil para migração de dados, carregamento de dados de teste ou atualizações regulares de dados.

Sintaxe básica de importação

A sintaxe básica para exportar dados usando o BCP é:

```
bcp database.schema.table in input_file -S server_name -U username -P password
 [options]
```

Em que:

- `database.schema.table`: o nome totalmente qualificado da tabela de destino.
- `input_file`: o caminho e o nome do arquivo de entrada.
- `server_name`: o endpoint do RDS para SQL Server.
- `username`: o nome de usuário do banco de dados.
- `password`: a senha do banco de dados.

Exemplo de importação

O seguinte exemplo importa dados de um arquivo para uma tabela denominada `customers`:

```
bcp sales.dbo.customers in /home/user/customers.txt \
    -S mydb.cluster-abc123.us-east-1.rds.amazonaws.com \
    -U admin \
    -P mypassword \
    -c \
    -t "|" \
```

```
    -r "\n" \
    -b 1000
```

Esse comando:

- Importa dados para a tabela `customers`.
- Lê os dados do `/home/user/customers.txt`
- Usa o formato de caracteres (`-c`).
- Usa pipe (|) como delimitador de campo (`-t "|"`).
- Usa nova linha como delimitador de linha (`-r "\n"`).
- Processa dados em lotes de mil linhas (`-b 1000`).

Opções comuns do BCP

O BCP oferece várias opções para controlar a formatação de dados e o comportamento de transferência. A seguinte tabela descreve as opções comumente usadas:

Opção	Descrição
`-c`	Usa o tipo de dados de caracteres para todas as colunas.
`-n`	Usa tipos de dados de banco de dados nativos.
`-t`	Especifica o delimitador de campo (o padrão é tab).
`-r`	Especifica o delimitador de linha (o padrão é nova linha).
`-b`	Especifica o tamanho do lote para operações em massa.
`-F`	Especifica a primeira linha a ser exportada ou importada.
`-L`	Especifica a última linha a ser exportada ou importada.
`-e`	Especifica um arquivo de erro para capturar linhas rejeitadas.
`-f`	Especifica um arquivo de formato para formatação de dados.
`-q`	Usa identificadores entre aspas para nomes de objeto.

Práticas recomendadas e considerações

Ao usar o BCP com o RDS para SQL Server no Linux, considere as seguintes práticas recomendadas:

- Use o processamento em lote: para grandes conjuntos de dados, use a opção `-b` para processar dados em lote. Isso melhora o desempenho e permite uma recuperação de erros mais apropriada.
- Lide com erros adequadamente: use a opção `-e` para capturar informações de erros e linhas rejeitadas em um arquivo separado para análise.
- Escolha os formatos de dados apropriados: use o formato de caracteres (`-c`) para compatibilidade entre plataformas ou o formato nativo (`-n`) para melhorar desempenho quando a origem e o destino forem o SQL Server.
- Proteja suas credenciais: evite colocar senhas diretamente nas linhas de comandos. Considere a possibilidade de usar variáveis de ambiente ou arquivos de configuração com as permissões apropriadas.
- Teste com conjuntos de dados pequenos: antes de processar grandes quantidades de dados, teste seus comandos do BCP com conjuntos de dados menores para verificar a formatação e a conectividade.
- Monitore a conectividade de rede: garanta conexões de rede estáveis, especialmente para grandes transferências de dados. Considere a possibilidade de usar ferramentas como `screen` ou `tmux` para operações de longa duração.
- Valide a integridade dos dados: após a transferência de dados, verifique as contagens de linhas e os dados de amostra para garantir que a operação foi concluída com êxito.

Solução de problemas comuns do

A seguinte tabela descreve os problemas comuns que você pode encontrar ao usar o BCP no Linux e as respectivas soluções:

Problema	Solução
Tempo limite de conexão ou erros de rede	Verifique seu endpoint do Amazon RDS, as configura ções do grupo de segurança e a conectividade de rede. Confirme se a porta do SQL Server (normalmente 1433) pode ser acessada pelo seu sistema Linux.

Problema	Solução
Falhas de autenticação	Anote seu nome de usuário e senha. Verifique se o usuário do banco de dados tem as permissões apropriad as para as operações que você está executando.
Erros de formato de dados	Verifique seus delimitadores de campo e linha. Confirme se o formato dos dados corresponde ao que o BCP espera. Use arquivos de formato para estruturas de dados complexas.
Erros de permissão negada	Verifique se o usuário do banco de dados tem permissõe s `INSERT` para importações ou permissões `SELECT` para exportações nas tabelas de destino.
Problemas de tratamento de arquivos grandes	Use o processamento em lote com a opção `-b`. Considere a possibilidade de dividir arquivos grandes em partes menores para melhorar o desempenho e a recuperação de erros.
Problemas de codificação de caracteres	Garanta que seus arquivos de dados usem uma codificaç ão de caracteres compatível. Use a opção `-c` para o formato de caracteres ou especifique as páginas de código apropriadas.

Trabalhar com réplicas de leitura do Microsoft SQL Server no Amazon RDS

Normalmente, você usa réplicas de leitura para configurar a replicação entre as instâncias de banco de dados do Amazon RDS. Para obter informações gerais sobre réplicas de leitura, consulte Trabalhar com réplicas de leitura de instância de banco de dados.

Nesta seção, você pode encontrar informações específicas sobre como trabalhar com réplicas de leitura no Amazon RDS for SQL Server.

- Sincronização de usuários e objetos do banco de dados com uma réplica de leitura do SQL Server
- Solução de problemas de uma réplica de leitura do SQL Server

Configurar réplicas de leitura do SQL Server

Antes que uma instância de banco de dados possa servir como uma instância de origem para replicação, é necessário habilitar backups automáticos na instância de banco de dados de origem. Para fazer isso, defina o período de retenção de backup como um valor diferente de 0. Definir esse tipo de implantação também impõe que os backups automáticos estejam habilitados.

Criar uma réplica de leitura do SQL Server não requer a interrupção da instância de banco de dados primária. O Amazon RDS define os parâmetros e as permissões necessários para a instância banco de dados de origem e para a réplica de leitura sem qualquer interrupção do serviço. Um snapshot é feito da instância de banco de dados de origem, e esse snapshot se torna a réplica de leitura. Quando você exclui uma réplica de leitura, não ocorre nenhuma interrupção.

Você pode criar até quinze réplicas de leitura de uma instância de banco de dados de origem. Para que a replicação funcione eficazmente, recomendamos configurar cada réplica de leitura com a mesma quantidade de recursos de computação e de armazenamento que a instância de banco de dados de origem. Se você dimensionar a instância de banco de dados de origem, dimensione as réplicas de leitura também.

A versão do mecanismo de banco de dados SQL Server da instância de banco de dados de origem e todas as réplicas de leitura devem ser iguais. O Amazon RDS atualiza aa primária imediatamente após a atualização das réplicas de leitura, desconsiderando a janela de manutenção de uma réplica. Para ter mais informações sobre como atualizar a versão do mecanismo de banco de dados, consulte Atualizações do mecanismo de banco de dados do Microsoft SQL Server.

Para que uma réplica de leitura receba e aplique alterações da origem, ela deve ter recursos suficientes de computação e armazenamento. Se uma réplica de leitura atingir a capacidade de recursos de computação, rede ou armazenamento, ela deixará de receber ou aplicar alterações de sua origem. É possível modificar os recursos de armazenamento e CPU de uma réplica de leitura independentemente de sua origem e de outras réplicas de leitura.

Para ter mais informações sobre como criar uma réplica de leitura, consulte Como criar uma réplica de leitura.

Limitações de réplica de leitura com o SQL Server

As seguintes limitações se aplicam a réplicas de leitura do SQL Server no Amazon RDS:

- Réplicas de leitura só estão disponíveis no mecanismo do SQL Server Enterprise Edition (EE).
- Réplicas de leitura estão disponíveis para as versões do SQL Server 2016–2022.
- Você pode criar até quinze réplicas de leitura de uma instância de banco de dados de origem. A replicação poderá atrasar quando a instância de banco de dados de origem tiver mais de cinco réplicas de leitura.
- Réplicas de leitura só estão disponíveis para instâncias de banco de dados em execução em classes de instância de banco de dados com quatro ou mais vCPUs.
- Uma réplica de leitura comporta até cem bancos de dados, dependendo do tipo de classe de instância e do modo de disponibilidade. É necessário criar bancos de dados na instância de banco de dados de origem para replicá-las automaticamente nas réplicas de leitura. Não é possível escolher bancos de dados individuais para replicar. Para obter mais informações, consulte Limitações das instâncias de banco de dados Microsoft SQL Server.
- Não é possível descartar um banco de dados de uma réplica de leitura. Para descartar um banco de dados, retire-o da instância de banco de dados de origem com o procedimento armazenado `rds_drop_database`. Para obter mais informações, consulte Eliminar um banco de dados em uma instância de banco de dados do Amazon RDS para Microsoft SQL Server.
- Se a instância de banco de dados de origem usa criptografia de dados transparente (TDE) para criptografar dados, a réplica de leitura também configura a TDE.

  Se a instância de banco de dados de origem usar uma chave do KMS para criptografar dados, réplicas de leitura na mesma região usarão a mesma chave do KMS. Em relação a réplicas de leitura entre regiões, você deve especificar uma chave do KMS da região da réplica de leitura ao criar a réplica de leitura. Não é possível alterar a chave do KMS para uma réplica de leitura.

- As réplicas de leitura têm o mesmo fuso horário e agrupamento da instância de banco de dados de origem, independentemente da zona de disponibilidade em que foram criadas.
- Os seguintes não são compatíveis com o Amazon RDS for SQL Server:
  - Retenção de backup de réplicas de leitura
  - Recuperação point-in-time de réplicas de leitura
  - Snapshots manuais de réplicas de leitura
  - Réplicas de leitura Multi-AZ
  - Criar réplicas de leitura de réplicas de leitura
  - Sincronização de logins de usuários para réplicas de leitura
- O Amazon RDS for SQL Server não intervém para atenuar atrasos altos de replicação entre uma instância de banco de dados de origem e suas réplicas de leitura. Certifique-se de que a instância de banco de dados de origem e suas réplicas de leitura estejam dimensionadas corretamente, em termos de capacidade de computação e armazenamento, para atender a suas cargas operacionais.
- Você pode replicar entre as regiões AWS GovCloud (Oeste dos EUA) e AWS GovCloud (Leste dos EUA), mas não para dentro ou fora da AWS GovCloud (US) Regions.

Considerações opcionais sobre réplicas do RDS para SQL Server

Antes de criar uma réplica do RDS para SQL Server, considere os seguintes requisitos, restrições e recomendações:

- Se a réplica do SQL Server estiver na mesma região da instância de banco de dados de origem, verifique se ela pertence ao mesmo grupo de opções que essa instância. Modificações no grupo de opções de origem ou na associação do grupo de opções de origem são propagadas para as réplicas. Essas alterações serão aplicadas às réplicas logo depois de serem aplicadas à instância de banco de dados de origem, independentemente da janela de manutenção da réplica.

  Para ter mais informações sobre grupos de opções, consulte Trabalhar com grupos de opções.
- Quando você cria uma réplica entre regiões no SQL Server, o Amazon RDS cria um grupo de opções dedicado para ela.

  Não é possível remover uma réplica entre regiões no SQL Server do seu grupo de opções dedicado. Nenhuma outra instância de banco de dados pode usar o grupo de opções dedicado para uma réplica entre regiões no SQL Server.

As opções a seguir são opções replicadas. Para adicionar outras opções a uma réplica de leitura entre regiões do SQL Server, adicione-as ao grupo de opções da instância de banco de dados de origem. A opção também é instalada em todas as réplicas da instância do banco de dados de origem.

- `TDE`

As opções a seguir são opções não replicadas. Você pode adicionar ou remover opções não replicadas de um grupo de opções dedicado.

- `MSDTC`
- `SQLSERVER_AUDIT`
- Para ativar a opção `SQLSERVER_AUDIT` na réplica de leitura entre regiões, adicione a opção `SQLSERVER_AUDIT` no grupo de opções dedicado na réplica de leitura entre regiões e no grupo de opções da instância de origem. Ao adicionar a opção `SQLSERVER_AUDIT` na instância de origem da réplica de leitura entre regiões do SQL Server, você pode criar o Objeto de auditoria em nível de servidor e especificações de auditoria em nível de servidor em cada uma das réplicas de leitura entre regiões da instância de origem. Para permitir que as réplicas de leitura entre regiões tenham acesso para fazer upload dos logs de auditoria concluídos em um bucket do Amazon S3, adicione a opção `SQLSERVER_AUDIT` ao grupo de opções dedicado e defina as configurações da opção. O bucket do Amazon S3 que você usa como destino para arquivos de auditoria deve estar na mesma região que a réplica leitura entre regiões. Você pode modificar a configuração da opção `SQLSERVER_AUDIT` para cada réplica de leitura entre regiões de forma independente para que cada um possa acessar um bucket do Amazon S3 em sua respectiva região.

As opções a seguir não são compatíveis com réplicas de leitura entre regiões.

- `SSRS`
- `SSAS`
- `SSIS`

As opções a seguir são parcialmente compatíveis com réplicas de leitura entre regiões.

- `SQLSERVER_BACKUP_RESTORE`
- A instância de banco de dados de origem de uma réplica entre regiões do SQL Server pode ter a opção `SQLSERVER_BACKUP_RESTORE`, mas você não pode realizar restaurações nativas na instância de banco de dados de origem até excluir todas as réplicas entre regiões. Todas as tarefas de restauração nativa existentes serão canceladas durante a criação de uma réplica

entre regiões. Você não pode adicionar a opção `SQLSERVER_BACKUP_RESTORE` a um grupo de opções dedicado.

Para ter mais informações sobre o backup e a restauração nativos, consulte Importar e exportar bancos de dados do SQL Server usando backup e restauração nativos.

Quando você promove uma réplica de leitura entre regiões no SQL Server, a réplica promovida se comporta da mesma maneira que outras instâncias de banco de dados do SQL Server, inclusive o gerenciamento de suas opções. Para ter mais informações sobre grupos de opções, consulte Trabalhar com grupos de opções.

Sincronização de usuários e objetos do banco de dados com uma réplica de leitura do SQL Server

Espera-se que todos os logins, perfis de servidor personalizados, trabalhos do agente do SQL ou outros objetos no nível do servidor que existam na instância de banco de dados primária no momento da criação de uma réplica de leitura estejam presentes na réplica de leitura recém-criada. No entanto, todos os objetos no nível do servidor criados na instância de banco de dados primária após a criação da réplica de leitura não serão replicados automaticamente, e você deverá criá-los manualmente na réplica de leitura.

Os usuários do banco de dados são replicados automaticamente da instância de banco de dados primária para a réplica de leitura. Como o banco de dados de réplica de leitura está no modo somente leitura, o identificador de segurança (SID) do usuário do banco de dados não pode ser atualizado no banco de dados. Portanto, ao criar logins de SQL na réplica de leitura, é essencial garantir que o SID desse login corresponda ao SID do login SQL correspondente na instância de banco de dados primária. Se você não sincronizar os SIDs dos logins do SQL, eles não poderão acessar o banco de dados na réplica de leitura. Os logins autenticados do Windows Active Directory (AD) não apresentam esse problema porque o SQL Server obtém o SID do Active Directory.

Para sincronizar um login do SQL da instância de banco de dados primária com a réplica de leitura

1. Conecte-se à instância de banco de dados primária.
2. Crie um login do SQL na instância de banco de dados primária.

```
USE [master]
GO
CREATE LOGIN TestLogin1
```

```
WITH PASSWORD = 'REPLACE WITH PASSWORD';
```

> **Note**
>
> Especifique uma senha diferente do prompt mostrado aqui como prática recomendada de segurança.

3. Crie um usuário do banco de dados para o login do SQL no banco de dados.

```
USE [REPLACE WITH YOUR DB NAME]
GO
CREATE USER TestLogin1 FOR LOGIN TestLogin1;
GO
```

4. Verifique o SID do login do SQL recém-criado na instância de banco de dados primária.

```
SELECT name, sid FROM sys.server_principals WHERE name =  TestLogin1;
```

5. Conecte-se à réplica de leitura. Crie o login do SQL.

```
CREATE LOGIN TestLogin1 WITH PASSWORD = 'REPLACE WITH PASSWORD', SID=[REPLACE WITH
 sid FROM STEP #4];
```

Como alternativa, se você tiver acesso ao banco de dados da réplica de leitura, poderá corrigir o usuário órfão da seguinte forma:

1. Conecte-se à réplica de leitura.
2. Identifique os usuários órfãos no banco de dados.

```
USE [REPLACE WITH YOUR DB NAME]
GO
EXEC sp_change_users_login 'Report';
GO
```

3. Crie um login do SQL para o usuário órfão do banco de dados.

```
CREATE LOGIN TestLogin1 WITH PASSWORD = 'REPLACE WITH PASSWORD', SID=[REPLACE WITH
 sid FROM STEP #2];
```

Exemplo:

```
CREATE LOGIN TestLogin1 WITH PASSWORD = 'TestPa$$word#1',
 SID=[0x1A2B3C4D5E6F7G8H9I0J1K2L3M4N5O6P];
```

> **Note**
>
> Especifique uma senha diferente do prompt mostrado aqui como prática recomendada de segurança.

Solução de problemas de uma réplica de leitura do SQL Server

Você pode monitorar o atraso da replicação no Amazon CloudWatch visualizando a métrica `ReplicaLag` do Amazon RDS. Para obter informações sobre o tempo de atraso de replicação, consulte Monitoramento da replicação de leitura.

Se o atraso de replicação for muito longo, use a seguinte consulta para obter informações sobre esse atraso:

```
SELECT AR.replica_server_name
     , DB_NAME (ARS.database_id) 'database_name'
     , AR.availability_mode_desc
     , ARS.synchronization_health_desc
     , ARS.last_hardened_lsn
     , ARS.last_redone_lsn
     , ARS.secondary_lag_seconds
FROM sys.dm_hadr_database_replica_states ARS
INNER JOIN sys.availability_replicas AR ON ARS.replica_id = AR.replica_id
--WHERE DB_NAME(ARS.database_id) = 'database_name'
ORDER BY AR.replica_server_name;
```

Implantações multi-AZ para o Amazon RDS for Microsoft SQL Server

As implantações Multi-AZ oferecem maior disponibilidade, durabilidade de dados e tolerância a falhas para instâncias de banco de dados. No caso de uma manutenção planejada do banco de dados ou de uma interrupção não planejada do serviço, o Amazon RDS faz failover automático para a instância de banco de dados secundário atualizada. Essa funcionalidade permite que as operações do banco de dados sejam retomadas rapidamente sem intervenção manual. As instâncias primária e em espera usam o mesmo endpoint, cujo endereço de rede física faz a transição para a réplica secundária como parte do processo de failover. Não é necessário reconfigurar seu aplicativo quando ocorre um failover.

O Amazon RDS oferece suporte a implantações Multi-AZ para Microsoft SQL Server usando o SQL Server Database Mirroring (DBM) ou grupos de disponibilidade Always On (AGs). O Amazon RDS monitora e mantém a integridade de sua implantação Multi-AZ. Caso ocorram problemas, o RDS repara automaticamente instâncias de banco de dados não íntegras, reestabelece a sincronização e inicia os failovers. O failover só ocorrerá se o modo em espera e o primário estiverem totalmente sincronizados. Você não precisa gerenciar tudo.

Quando você configura o Multi-AZ do SQL Server, o RDS configura automaticamente todos os bancos de dados na instância para usar DBM ou AGs. O Amazon RDS processa as instâncias primária, testemunha e de banco de dados secundária para você. Como a configuração é automática, o RDS selecione DBM ou Always On AGS com base na versão do SQL Server implantada.

O Amazon RDS oferece suporte a Multi-AZ com Always On AGs para as seguintes versões e edições do SQL Server:

- SQL Server 2022:
  - Edição Standard
  - Edição Enterprise
- SQL Server 2019:
  - Standard Edition 15.00.4073.23 e posteriores
  - Edição Enterprise
- SQL Server 2017:
  - Standard Edition 14.00.3401.7 e posteriores

  - Enterprise Edition 14.00.3049.1 e posteriores
- SQL Server 2016: Enterprise Edition 13.00.5216.0 e posterior

O Amazon RDS oferece suporte a Multi-AZ com DBM para as seguintes versões e edições do SQL Server, exceto para as versões indicadas anteriormente:

- SQL Server 2019: Standard Edition 15.00.4043.16
- SQL Server 2017: Standard e Enterprise Editions
- SQL Server 2016: Standard e Enterprise Editions

Você pode usar a seguinte consulta SQL para determinar se sua instância de banco de dados do SQL Server é single-AZ, multi-AZ com DBM ou multi-AZ com AGs Always On:

```
SELECT CASE WHEN dm.mirroring_state_desc IS NOT NULL THEN 'Multi-AZ (Mirroring)'
    WHEN dhdrs.group_database_id IS NOT NULL THEN 'Multi-AZ (AlwaysOn)'
    ELSE 'Single-AZ'
    END 'high_availability'
FROM sys.databases sd
LEFT JOIN sys.database_mirroring dm ON sd.database_id = dm.database_id
LEFT JOIN sys.dm_hadr_database_replica_states dhdrs ON sd.database_id =
 dhdrs.database_id AND dhdrs.is_local = 1
WHERE DB_NAME(sd.database_id) = 'rdsadmin';
```

A saída será semelhante à seguinte.

```
high_availability
Multi-AZ (AlwaysOn)
```

Adicionar Multi-AZ a uma instância de banco de dados do Microsoft SQL Server

Ao criar uma nova instância de banco de dados do SQL Server usando o AWS Management Console, você pode adicionar Multi-AZ com Database Mirroring (DBM) ou AGs Always On. Faça isso selecionando Yes (Mirroring / Always On) (Sim (Mirroring/Always On)) na Multi-AZ deployment (Implantação Multi-AZ). Para ter mais informações, consulte Criar uma instância de banco de dados do Amazon RDS.

Ao modificar uma instância de banco de dados do SQL Server existente usando o console, você pode adicionar multi-AZ com DBM ou AGs escolhendo Yes (Mirroring / Always On) (Sim (Mirroring/ Always On)) na lista Multi-AZ Deployment (Implantação multi-AZ) na página Modify DB Instance (Modificar instância de banco de dados). Para ter mais informações, consulte Modificar uma instância de banco de dados do Amazon RDS.

> **Note**
>
> Se a instância de banco de dados estiver executando o DBM (Database Mirroring) — e não AGs (Grupos de disponibilidade Always On) — talvez seja necessário desabilitar a otimização na memória antes de adicionar Multi-AZ. Desabilite a otimização na memória com DBM antes de adicionar multi-AZ se a instância de banco de dados executar o SQL Server 2016 ou 2017 Enterprise Edition e tiver a otimização na memória habilitada.
> Se a instância de banco de dados estiver executando AGs, não será necessário realizar essa etapa.

Remover multi-AZ de uma instância de banco de dados do Microsoft SQL Server

Ao modificar uma instância de banco de dados do SQL Server existente usando o AWS Management Console, você pode remover multi-AZ com DBM ou AGs. Para fazer isso, escolha No (Mirroring / Always On) (Não (Mirroring/Always On)) em Multi-AZ deployment (Implantação multi-AZ) na página Modify DB instance (Modificar instância de banco de dados). Para ter mais informações, consulte Modificar uma instância de banco de dados do Amazon RDS.

Limitações, observações e recomendações de implantação multi-AZ do Microsoft SQL Server

A seguir você encontrará algumas restrições aplicáveis ao trabalhar com implantações multi-AZ em instâncias de banco de dados do RDS para SQL Server:

- O Multi-AZ entre regiões não é compatível.
- Não há suporte para a interrupção de um para a instância de banco de dados SQL Server em uma implantação Multi-AZ.
- Não é possível configurar a instância de banco de dado secundária para aceitar a atividade de leitura de banco de dados.

- Multi-AZ com grupos de disponibilidade (AGs) Always On oferece suporte à otimização na memória.
- O Multi-AZ com grupos de disponibilidade (AGs) Always On não oferece suporte à autenticação Kerberos para o listener do grupo de disponibilidade. Isso ocorre, pois o listener não tem nome principal do serviço (SPN).
- Não é possível renomear um banco de dados em uma instância de banco de dados do SQL Server que esteja em uma implantação Multi-AZ do SQL Server. Se você precisar renomear um banco de dados em uma instância assim, primeiro desative o Multi-AZ da instância de banco de dados e renomeie o banco de dados. Por fim, reative Multi-AZ para a instância de banco de dados.
- Você só pode restaurar instâncias de banco de dados Multi-AZ com backup feito usando-se o modelo de recuperação completo.
- As implantações multi-AZ têm um limite de cem trabalhos do SQL Server Agent.

  Se um limite mais alto for necessário, solicite um aumento de cota entrando em contato com o Suporte. Abra a página do AWS Support Center, faça login, se necessário, e escolha Create case (Criar caso). Escolha Service limit increase (Aumento de limite do serviço). Preencha e envie o formulário.
- Não é possível ter um banco de dados off-line em uma instância de banco de dados do SQL Server que esteja em uma implantação multi-AZ do SQL Server.
- O RDS para SQL Server não replica as permissões do banco de dados MSDB para a instância secundária. Se você precisar dessas permissões na instância secundária, deverá recriá-las manualmente.

Veja a seguir algumas observações sobre como trabalhar com implantações multi-AZ em instâncias de banco de dados do RDS para SQL Server:

- O Amazon RDS expõe o endpoint de listener do grupo de disponibilidade de Always On AGs. O endpoint está visível no console e é retornado pela operação de API `DescribeDBInstances` como uma entrada no campo de endpoints.
- O Amazon RDS oferece suporte a failovers de sub-rede do grupo de disponibilidade.
- Para usar o multi-AZ do SQL Server com uma instância de banco de dados do SQL Server em uma nuvem privada virtual (VPC), primeiro crie um grupo de sub-rede de banco de dados que tenha sub-redes em pelo menos duas zonas de disponibilidade distintas. Em seguida, atribua o grupo de sub-rede de banco de dados à réplica primária da instância de banco de dados do SQL Server.

- Quando uma instância de banco de dados é modificada para ser uma implantação Multi-AZ, durante a modificação, ela tem o status modifying. O Amazon RDS cria o modo de espera e faz um backup da instância de banco de dados primária. Depois que o processo estiver concluído, o status da instância de banco de dados primária se tornará available (disponível).
- Implantações Multi-AZ mantém todos os bancos de dados no mesmo nó. Se um banco de dados no host primário fizer failover, todos os bancos de dados do SQL Server farão failover como uma unidade atômica para o host em espera. O Amazon RDS provisiona um novo host íntegro e substitui o host não íntegro.
- O Multi-AZ com DBM ou AGs oferece suporte a uma única réplica em espera.
- Os usuários, os logins e as permissões são replicados automaticamente para você na secundária. Não é necessário recriá-los. Os perfis de servidor definidos pelo usuário só são replicados em instâncias de banco de dados que usam AGs Always On para implantações multi-AZ.
- Em implantações multi-AZ, o RDS para SQL Server cria logins do SQL Server para permitir AGs Always On ou o Database Mirroring. O RDS cria logins com o seguinte padrão, `db_<dbiResourceId>_node1_login`, `db_<dbiResourceId>_node2_login` e `db_<dbiResourceId>_witness_login`.
- O RDS para SQL Server cria um login do SQL Server para permitir o acesso às réplicas de leitura. O RDS cria um login com o seguinte padrão: `db_<readreplica_dbiResourceId>_node_login`.
- Em implantações multi-AZ, os trabalhos do SQL Server Agent são replicados do host primário para o host secundário quando o recurso de replicação de trabalhos é ativado. Para obter mais informações, consulte Ativar a replicação de trabalhos do SQL Server Agent.
- Convém observar latências elevadas em comparação com uma implantação de instância de banco de dados padrão (em uma única zona de disponibilidade) por causa da replicação de dados síncrona.
- Os tempos de failover são afetados pelo tempo necessário para completar o processo de recuperação. Transações grandes aumentam o tempo de failover.
- Em implantações Multi-AZ do SQL Server, a reinicialização com failover reinicializa somente a instância de banco de dados principal. Após o failover, a instância de banco de dados primária torna-se a nova instância de banco de dados secundária. Os parâmetros podem não ser atualizados para instâncias Multi-AZ. Para a reinicialização sem failover, as instâncias de banco de dados primárias e secundárias são reinicializadas e os parâmetros são atualizados após a reinicialização. Se a instância de banco de dados não responder, recomendamos reinicializar sem failover.

A seguir você encontrará algumas recomendações para trabalhar com implantações Multi-AZ em instâncias de banco de dados do RDS for Microsoft SQL Server:

- Para bancos de dados usados em produção ou pré-produção, recomendamos as seguintes opções:
  - Implantações Multi-AZ para alta disponibilidade
  - "IOPS provisionadas" para performance rápida e consistente
  - "Memória otimizada" em vez de "Uso geral"
- Não é possível selecionar a zona de disponibilidade (AZ) para a instância secundária. Por isso, quando implantar hosts de aplicativo, leve isso em conta. O banco de dados pode fazer failover para outro AZ, e os hosts de aplicativo podem não estar no mesmo AZ do banco de dados. Por esse motivo, recomendamos equilibrar os hosts de aplicação em todas as AZs na região da AWS indicada.
- Para obter a melhor performance, não habilite o Database Mirroring ou os AGs Always On durante uma operação grande de carregamento de dados. Se quiser que o carregamento de dados seja o mais rápido possível, termine o carregamento de dados antes de converter a instância de banco de dados em uma implantação Multi-AZ.
- Os aplicativos que acessam os bancos de dados do SQL Server devem ter um tratamento de exceção que capte erros de conexão. O exemplo de código a seguir mostra um bloco try/catch que capta um erro de comunicação. Neste exemplo, a `break` instrução sai do `while` loop se a conexão for bem-sucedida, mas tenta novamente até 10 vezes se uma exceção for lançada.

```
int RetryMaxAttempts = 10;
int RetryIntervalPeriodInSeconds = 1;
int iRetryCount = 0;
while (iRetryCount < RetryMaxAttempts)
{
   using (SqlConnection connection = new SqlConnection(DatabaseConnString))
   {
      using (SqlCommand command = connection.CreateCommand())
      {
         command.CommandText = "INSERT INTO SOME_TABLE VALUES ('SomeValue');";
         try
         {
            connection.Open();
            command.ExecuteNonQuery();
            break;
         }
```

```
         catch (Exception ex)
         {
            Logger(ex.Message);
            iRetryCount++;
         }
         finally {
            connection.Close();
         }
      }
   }
   Thread.Sleep(RetryIntervalPeriodInSeconds * 1000);
}
```

- Não use o comando `Set Partner Off` ao trabalhar com instâncias Multi-AZ. Por exemplo, não faça o seguinte.

```
--Don't do this
ALTER DATABASE db1 SET PARTNER off
```

- Não defina o modo de recuperação como `simple`. Por exemplo, não faça o seguinte.

```
--Don't do this
ALTER DATABASE db1 SET RECOVERY simple
```

- Não use o parâmetro `DEFAULT_DATABASE` ao criar novos login em instâncias de banco de dados Multi-AZ, porque essas configurações não podem ser aplicadas ao espelho em espera. Por exemplo, não faça o seguinte.

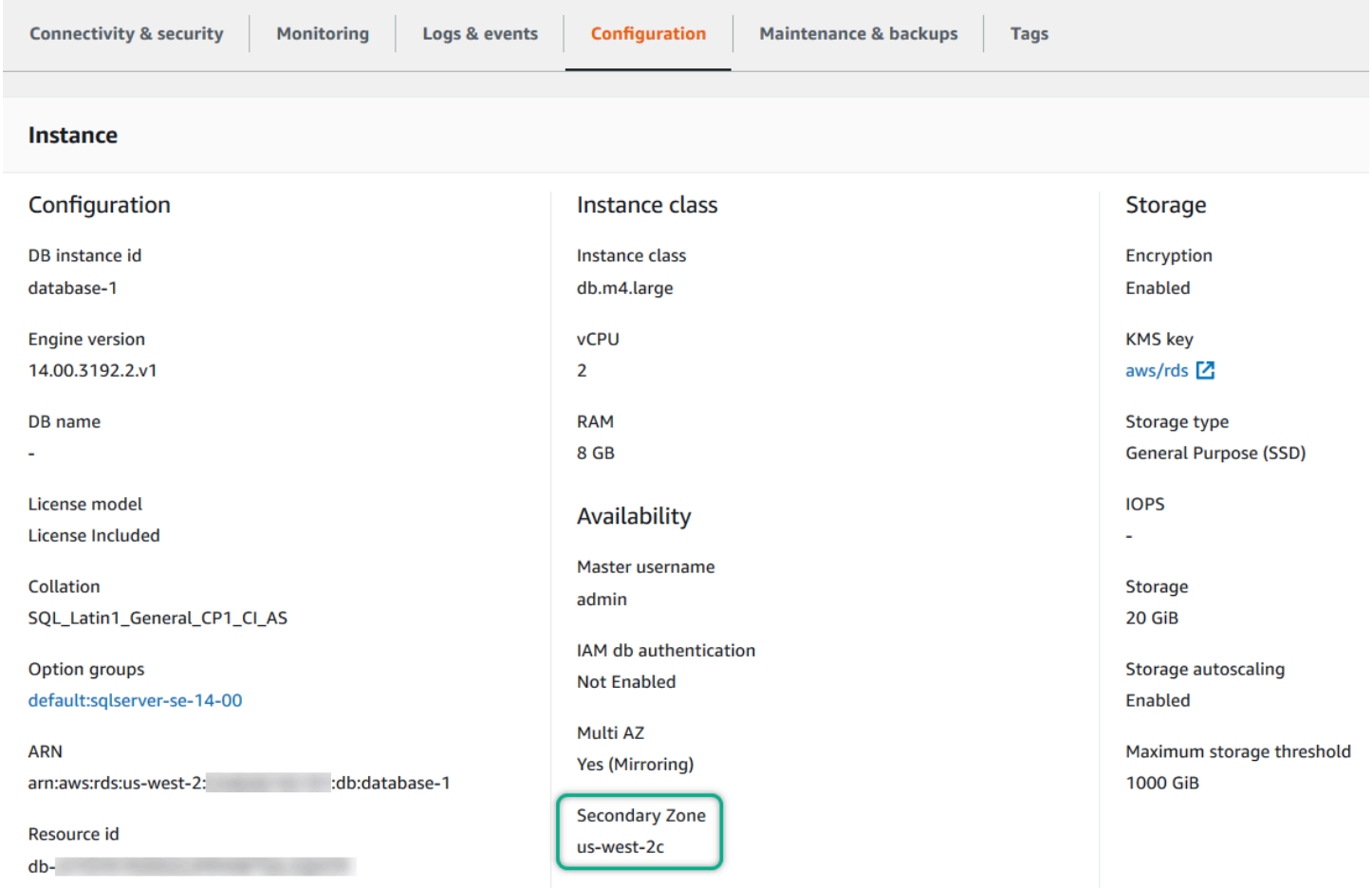
```
--Don't do this
CREATE LOGIN [test_dba] WITH PASSWORD=foo, DEFAULT_DATABASE=[db2]
```

  Além disso, não faça o seguinte.

```
--Don't do this
ALTER LOGIN [test_dba] WITH DEFAULT_DATABASE=[db3]
```

Determinar a localização do secundário

Determine a localização da réplica secundária usando o AWS Management Console. Você precisará saber a localização da secundária se estiver configurando a instância de banco de dados primária em uma VPC.



Também é possível visualizar a zona de disponibilidade da secundária usando o comando AWS CLI da `describe-db-instances` ou a operação da API do RDS `DescribeDBInstances`. O resultado mostra a AZ secundária onde o espelho em espera está localizado.

Migrar do Database Mirroring para Grupos de Disponibilidade Always On

Na versão 14.00.3049.1 do Microsoft SQL Server Enterprise Edition, os Grupos de Disponibilidade (AGs) Always On estão sempre habilitados por padrão.

Para migrar do Database Mirroring (DBM) para AGs, verifique sua versão primeiramente. Se você estiver usando uma instância de banco de dados com uma versão anterior à Enterprise Edition 13.00.5216.0, modifique a instância a fim de atualizá-la para a 13.00.5216.0 ou posterior. Se você

estiver usando uma instância de banco de dados com uma versão anterior à Enterprise Edition 14.00.3049.1, modifique a instância a fim de atualizá-la para a 14.00.3049.1 ou posterior.

Se você deseja atualizar uma instância de banco de dados espelhada para usar AGs, execute a atualização primeiro, modifique a instância para remover o Multi-AZ e depois modifique-a novamente para adicionar o Multi-AZ. Isso converterá a instância para usar AGs Always On.

Recursos adicionais do Microsoft SQL Server no Amazon RDS

Nas seções a seguir, você pode encontrar informações sobre como aumentar as instâncias do Amazon RDS que executam o mecanismo de banco de dados do Microsoft SQL Server.

Tópicos

- Usar política de senha para logins do SQL Server no RDS para SQL Server
- Integrar uma instância de banco de dados do Amazon RDS for SQL Server ao Amazon S3
- Usar o Database Mail no Amazon RDS for SQL Server
- Suporte ao armazenamento de instâncias para o banco de dados tempdb no Amazon RDS for SQL Server
- Usar eventos estendidos com Amazon RDS for Microsoft SQL Server
- Acesso aos backups de logs de transações com o RDS para SQL Server

Usar política de senha para logins do SQL Server no RDS para SQL Server

O Amazon RDS permite que você defina a política de senha para a instância de banco de dados do Amazon RDS que executa o Microsoft SQL Server. Use isso para definir os requisitos de complexidade, duração e bloqueio para logins que usam a autenticação do SQL Server para se autenticar na instância de banco de dados.

Principais termos

Login

No SQL Server, uma entidade principal em nível de servidor que pode se autenticar em uma instância de banco de dados é chamado de login. Outros mecanismos de banco de dados podem se referir a essa entidade principal como usuário. No RDS para SQL Server, um login pode se autenticar usando a Autenticação do SQL Server ou a Autenticação do Windows.

Login do SQL Server

Um login que usa um nome de usuário e uma senha para se autenticar usando a Autenticação do SQL Server é um login do SQL Server. A política de senha que você configura por meio de parâmetros de banco de dados se aplica somente aos logins do SQL Server.

Login do Windows

Um login baseado em uma entidade principal do Windows e se autentica usando a Autenticação do Windows é um login do Windows. É possível configurar a política de senha para os logins do Windows no Active Directory. Para ter mais informações, consulte Trabalhar com o Active Directory com o RDS para SQL Server.

Habilitar e desabilitar a política para cada login

Cada login do SQL Server tem sinalizadores para `CHECK_POLICY` e `CHECK_EXPIRATION`. Por padrão, novos logins são criados com `CHECK_POLICY` definido como `ON` e `CHECK_EXPIRATION` definido como `OFF`.

Se `CHECK_POLICY` estiver habilitado para um login, o RDS para SQL Server valida a senha em relação aos requisitos de complexidade e de tamanho mínimo. As políticas de bloqueio também são aplicáveis. Um exemplo de declaração T-SQL para habilitar `CHECK_POLICY` e `CHECK_EXPIRATION`:

```
ALTER LOGIN [master_user] WITH CHECK_POLICY = ON, CHECK_EXPIRATION = ON;
```

Se CHECK_EXPIRATION estiver habilitado, as senhas estarão sujeitas às políticas de idade da senha. A declaração T-SQL para conferir se CHECK_POLICY e CHECK_EXPIRATION estão definidos:

```
SELECT name, is_policy_checked, is_expiration_checked FROM sys.sql_logins;
```

Parâmetros da política de senha

Todos os parâmetros da política de senha são dinâmicos e não exigem a reinicialização do banco de dados para entrar em vigor. A tabela a seguir lista os parâmetros de banco de dados que você pode definir para modificar a política de senha para logins do SQL Server:

Parâmetro de banco de dados	Descrição	Valores permitidos	Valor padrão	
rds.password_complexity_enabled	Os requisitos de complexidade da senha devem ser atendidos ao criar ou alterar senhas para logins do SQL Server. As seguintes restrições devem ser atendidas: • A senha deve incluir caracteres de três das seguintes categorias: • Letra minúscula do latim (de a a z).	0,1	0	

Parâmetro de banco de dados	Descrição	Valores permitidos	Valor padrão	
	• Letra maiúscula do latim (de A a Z). • Caracteres não alfanuméricos, como: ponto de exclamação (!), cifrão ($), cerquilha (#) ou porcentagem (%). • A senha não contém o nome da conta do usuário.			
rds.password_min_length	O número mínimo de caracteres exigido em uma senha para um login do SQL Server.	0-14	0	

Parâmetro de banco de dados	Descrição	Valores permitidos	Valor padrão	
rds.password_min_age	O número mínimo de dias em que uma senha de login do SQL Server deve ser usada para que o usuário possa alterá-la. As senhas podem ser alteradas imediatamente quando definidas como 0.	0-998	0	
rds.password_max_age	O número máximo de dias em que uma senha de login do SQL Server pode ser usada, após o qual o usuário deve alterá-la. As senhas nunca expiram quando definidas como 0.	0-999	42	

Parâmetro de banco de dados	Descrição	Valores permitidos	Valor padrão	
rds.password_lockout_threshold	O número de tentativas consecutivas de login com falha que fazem com que um login do SQL Server fique bloqueado.	0-999	0	
rds.password_lockout_duration	O número de minutos que um login bloqueado do SQL Server deve esperar antes de ser desbloqueado.	1-60	10	
rds.password_lockout_reset_counter_after	O número de minutos que devem decorrer após uma tentativa de login malsucedida antes que o contador de tentativas de login com falha seja redefinido como 0.	1-60	10	

> **Note**
>
> Para ter mais informações sobre a política de senha do SQL Server, consulte Password Policy.
> As políticas de complexidade e tamanho mínimo da senha também se aplicam aos usuários do banco de dados em bancos de dados contidos. Para ter mais informações, consulte Contained Databases.

As seguintes restrições se aplicam aos parâmetros de política de senha:

- O parâmetro `rds.password_min_age` deve ser menor que `rds.password_max_age parameter`, a menos que `rds.password_max_age` esteja definido como 0.
- O parâmetro `rds.password_lockout_reset_counter_after` deve ser menor que ou igual ao parâmetro `rds.password_lockout_duration`.
- Se `rds.password_lockout_threshold` estiver definido como 0, `rds.password_lockout_duration` e `rds.password_lockout_reset_counter_after` não se aplicarão.

Considerações sobre logins existentes

Depois de modificar a política de senha em uma instância, as senhas existentes para logins não são avaliadas retroativamente em relação aos novos requisitos de complexidade e de tamanho da senha. Somente novas senhas são validadas de acordo com a nova política.

O SQL Server não avalia as senhas existentes de acordo com os requisitos de idade.

É possível que as senhas expirem imediatamente depois que uma política de senha for modificada. Por exemplo, se um login tiver `CHECK_EXPIRATION` habilitado e sua senha foi alterada pela última vez há cem dias e você definiu o parâmetro `rds.password_max_age` como cinco dias, a senha expirará imediatamente e será necessário alterar a senha na próxima tentativa de login.

> **Note**
>
> O RDS para SQL Server não é compatível com políticas de histórico de senha. As políticas de histórico impedem que os logins reutilizem senhas usadas anteriormente.

Considerações para implantações Multi-AZ

O contador de tentativas de login com falha e o estado de bloqueio das instâncias multi-AZ não se replicam entre os nós. No caso de um login ser bloqueado quando uma instância multi-AZ falha, é possível que o login já esteja desbloqueado no novo nó.

Considerações sobre senhas para o login mestre

Ao criar uma instância de banco de dados do RDS para SQL Server, a senha do usuário mestre não é avaliada em relação à política de senha. Uma nova senha mestra também não é avaliada em relação à senha ao realizar operações com o usuário mestre, especificamente ao configurar `MasterUserPassword` no comando `ModifyDBInstance`. Nos dois casos, é possível definir uma senha para o usuário mestre que não atenda à política de senha, e a operação ainda ser bem-sucedida. Se a política não for satisfeita, o RDS tentará gerar um evento do RDS, com a recomendação de definir uma senha forte. Tome cuidado para usar somente senhas fortes para o usuário mestre.

O RDS tenta gerar as seguintes mensagens de evento quando a senha do usuário mestre não atende aos requisitos da política de senha:

- O usuário mestre foi criado, mas a senha não atende ao requisito de tamanho mínimo da política de senha. Pense em usar uma senha mais forte.
- O usuário mestre foi criado, mas a senha não atende ao requisito de complexidade da política de senha. Pense em usar uma senha mais forte.
- A senha do usuário mestre foi redefinida, mas ela não atende ao requisito de tamanho mínimo da política de senha. Pense em usar uma senha mais forte.
- A senha do usuário mestre foi redefinida, mas ela não atende ao requisito de complexidade da política de senha. Pense em usar uma senha mais forte.

Por padrão, o usuário mestre é criado com `CHECK_POLICY` e `CHECK_EXPIRATION` definidos como`OFF`. Para aplicar a política de senha ao usuário mestre, você deve habilitar manualmente esses sinalizadores para o usuário mestre após a criação da instância de banco de dados. Depois de habilitar esses sinalizadores, modifique a senha do usuário mestre diretamente no SQL Server (por exemplo, por meio de declarações T-SQL ou SSMS) para validar a nova senha em relação à política de senha.

> **Note**
>
> Se o usuário mestre for bloqueado, você poderá desbloqueá-lo redefinindo a senha do usuário mestre com o comando `ModifyDBInstance`.

Modificar a senha do usuário mestre

É possível modificar a senha do usuário mestre usando o comando ModifyDBInstance.

> **Note**
>
> Quando você redefine a senha do usuário mestre, o RDS redefine várias permissões para o usuário mestre e este pode perder determinadas permissões. A redefinição da senha do usuário mestre também desbloqueia o usuário mestre, caso ele tenha sido bloqueado.

O RDS valida a nova senha do usuário mestre e tentará emitir um evento do RDS se a senha não atender à política. O RDS define a senha mesmo que ela não atenda à política de senha.

Integrar uma instância de banco de dados do Amazon RDS for SQL Server ao Amazon S3

É possível transferir arquivos entre uma instância de banco de dados do Amazon RDS for SQL Server e um bucket do Amazon S3. Ao fazer isso, é possível usar o Amazon S3 com os recursos do SQL Server, como BULK INSERT. Por exemplo, é possível baixar .csv, .xml, .txt e outros arquivos do Amazon S3 no host da instância de banco de dados de `D:\S3\` para o banco de dados. Todos os arquivos são armazenados em `D:\S3\` na instância de banco de dados.

As limitações a seguir se aplicam a:

> **Note**
>
> O tráfego entre o host do RDS e o S3 é roteado por meio de endpoints da VPC em VPCs internas do RDS para todos os recursos do SQL Server que usam o S3. Esse tráfego não usa a ENI do endpoint da instância do RDS. As políticas de bucket do S3 não podem restringir o tráfego do RDS com base em condições de rede.

- Os arquivos na pasta `D:\S3` são excluídos na réplica em espera após um failover em instâncias Multi-AZ. Para obter mais informações, consulte Limitações Multi-AZ para a integração do S3.
- A instância de banco de dados e o bucket do S3 devem estar na mesma região da AWS.
- Se você executar mais de uma tarefa de integração do S3 por vez, as tarefas serão executadas sequencialmente, não em paralelo.

  > **Note**
  >
  > As tarefas de integração do S3 compartilham a mesma fila que as tarefas nativas de backup e restauração. É possível ter no máximo duas tarefas em andamento a qualquer momento nesta fila. Portanto, duas tarefas nativas de backup e restauração em execução bloquearão todas as tarefas de integração do S3.

- É necessário habilitar novamente o recurso de integração do S3 em instâncias restauradas. A integração do S3 não é propagada da instância de origem para a instância restaurada. Os arquivos em `D:\S3` são excluídos em uma instância restaurada.
- O download na instância de banco de dados é limitado a 100 arquivos. Em outras palavras, não pode haver mais de 100 arquivos em `D:\S3\`.

- Somente arquivos sem extensões de arquivo ou com as seguintes extensões de arquivo são aceitos para download: .abf, .asdatabase, .bcp, .configsettings, .csv, .dat, .deploymentoptions, .deploymenttargets, .fmt e .xmla.
- O bucket do S3 deve ter o mesmo proprietário da função do AWS Identity and Access Management (IAM) relacionada. Portanto, a integração do S3 entre contas não tem suporte.
- O bucket do S3 não pode ser aberto ao público.
- O tamanho do arquivo para uploads do RDS para o S3 é limitado a 50 GB por arquivo.
- O tamanho do arquivo para downloads do S3 para o RDS é limitado ao máximo aceito pelo S3.

Tópicos

- Pré-requisitos para integrar o RDS for SQL Server com o S3
- Habilitação da integração do RDS for SQL Server com o S3
- Transferência de arquivos entre o RDS for SQL Server e o Amazon S3
- Listar arquivos na instância de banco de dados do RDS
- Excluir arquivos na instância de banco de dados do RDS
- Monitorar o status de uma tarefa de transferência de arquivo
- Cancelar uma tarefa
- Limitações Multi-AZ para a integração do S3
- Desabilitação da integração do RDS for SQL Server com o S3

Para obter mais informações sobre como trabalhar com arquivos no Amazon S3, consulte Conceitos básicos do Amazon Simple Storage Service.

Pré-requisitos para integrar o RDS for SQL Server com o S3

Antes de começar, encontre e crie o bucket do S3 que você deseja usar. Além disso, adicione permissões para que a instância de banco de dados do RDS possa acessar o bucket do S3. Para configurar esse acesso, crie uma política do IAM e uma função do IAM.

Console

Como criar uma política do IAM para acesso ao Amazon S3

1. No Console de gerenciamento do IAM, escolha Policies (Políticas) no painel de navegação.

2. Crie uma nova política e use a guia Visual editor (Editor visual) para as etapas a seguir.
3. Para Service (Serviço), insira **S3** e escolha o serviço do S3.
4. Para Actions (Ações), escolha o seguinte para conceder o acesso que a instância de banco de dados exige:

   - `ListAllMyBuckets`: obrigatório.
   - `ListBucket`: obrigatório.
   - `GetBucketAcl`: obrigatório.
   - `GetBucketLocation`: obrigatório.
   - `GetObject`: necessário para baixar arquivos do S3 em `D:\S3\`.
   - `PutObject`: necessário para fazer upload de arquivos de `D:\S3\` no S3.
   - `ListMultipartUploadParts`: necessário para fazer upload de arquivos de `D:\S3\` no S3.
   - `AbortMultipartUpload`: necessário para fazer upload de arquivos de `D:\S3\` no S3.
5. Para Resources (Recursos), as opções exibidas dependem de quais ações são escolhidas na etapa anterior. É possível ver opções para bucket, object (objeto) ou os dois. Para cada um deles, adicione o nome de recurso da Amazon (ARN) adequado.

   Em bucket, adicione o ARN do bucket que você deseja usar. Por exemplo, se o bucket for chamado de *`amzn-s3-demo-bucket`*, defina o ARN como `arn:aws:s3:::`*`amzn-s3-demo-bucket`*.

   Em object (objeto), insira o ARN do bucket e escolha uma das seguintes opções:

   - Para conceder acesso a todos os arquivos no bucket especificado, escolha Any (Qualquer um) para Bucket name (Nome do bucket) e Object name (Nome do objeto).
   - Para conceder acesso a pastas ou arquivos específicos no bucket, forneça os ARNs dos objetos e buckets específicos que você deseja que o SQL Server acesse.
6. Siga as instruções no console até terminar a criação da política.

   O conteúdo anterior é um guia resumido para configurar uma política. Para instruções mais detalhadas sobre como criar políticas do IAM, consulte Criar políticas do IAM no Guia do usuário do IAM.

Como criar uma função do IAM que usa a política do IAM do procedimento anterior

1. No Console de gerenciamento do IAM, escolha Roles (Funções) no painel de navegação.

2. Crie uma nova função do IAM e escolha as seguintes opções à medida que elas forem exibidas no console:

   - Produto da AWS
   - RDS
   - RDS – adicionar função ao banco de dados

   Escolha Next:Permissions (Próximo: permissões) na parte inferior.
3. Para Attach permissions policies (Anexar políticas de permissões), insira o nome da política do IAM criada anteriormente. Depois disso, escolha essa política na lista.
4. Siga as instruções no console até terminar a criação da função.

   O anterior é um guia resumido para configurar uma função. Se quiser informações mais detalhadas sobre como criar funções, consulte Funções do IAM no Guia do usuário do IAM.

AWS CLI

Para conceder acesso do Amazon RDS a um bucket do Simple Storage Service (Simple Storage Service (Amazon S3)), use o seguinte processo:

1. Crie uma política do IAM que conceda ao Amazon RDS acesso a um bucket do S3.
2. Crie uma função do IAM que o Amazon RDS possa assumir em seu nome para acessar seus buckets do S3.

   Para ter mais informações, consulte Criar um perfil para delegar permissões a um usuário do IAM no Guia do usuário do IAM.
3. Anexe a política do IAM que você criou ao perfil do IAM que você criou.

Para criar a política do IAM

Inclua as ações adequadas para conceder o acesso que a instância de banco de dados exige:

- `ListAllMyBuckets`: obrigatório.
- `ListBucket`: obrigatório.
- `GetBucketAcl`: obrigatório.
- `GetBucketLocation`: obrigatório.

- `GetObject` – necessário para baixar arquivos do S3 em `D:\S3\`
- `PutObject` – necessário para fazer upload de arquivos de `D:\S3\` no S3
- `ListMultipartUploadParts` – necessário para fazer upload de arquivos de `D:\S3\` no S3
- `AbortMultipartUpload` – necessário para fazer upload de arquivos de `D:\S3\` no S3

1. O comando da AWS CLI a seguir cria uma política do IAM denominada `rds-s3-integration-policy` com essas opções. Ele concede acesso a um bucket denominado *`amzn-s3-demo-bucket`*.

   Example

   Para Linux, macOS ou Unix:

```
aws iam create-policy \
  --policy-name rds-s3-integration-policy \
  --policy-document '{
        "Version": "2012-10-17",
        "Statement": [
            {
                "Effect": "Allow",
                "Action": "s3:ListAllMyBuckets",
                "Resource": "*"
            },
            {
                "Effect": "Allow",
                "Action": [
                    "s3:ListBucket",
                    "s3:GetBucketAcl",
                    "s3:GetBucketLocation"
                ],
                "Resource": "arn:aws:s3:::amzn-s3-demo-bucket"
            },
            {
                "Effect": "Allow",
                "Action": [
                    "s3:GetObject",
                    "s3:PutObject",
                    "s3:ListMultipartUploadParts",
                    "s3:AbortMultipartUpload"
                ],
                "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/key_prefix/*"
```

```
            }
        ]
    }'
```

Para Windows:

Altere os finais de linha para os que são compatíveis com a interface (^, em vez de \). Além disso, no Windows, é necessário inserir um caractere de escape em todas as aspas duplas com um \. Para evitar a necessidade de inserir um caractere de escape em aspas no JSON, é possível salvá-lo em um arquivo e passá-lo como um parâmetro.

Primeiro, crie o arquivo `policy.json` com a seguinte política de permissão:

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "s3:ListAllMyBuckets",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListBucket",
                "s3:GetBucketACL",
                "s3:GetBucketLocation"
            ],
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:PutObject",
                "s3:ListMultipartUploadParts",
                "s3:AbortMultipartUpload"
            ],
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/key_prefix/*"
        }
```

```
    ]
}
```

Depois disso, use o seguinte comando para criar a política:

```
aws iam create-policy ^
     --policy-name rds-s3-integration-policy ^
     --policy-document file://file_path/assume_role_policy.json
```

2. Após a criação da política, observe o nome de recurso da Amazon (ARN) da política. O ARN é necessário para uma etapa subsequente.

Para criar perfil do IAM

- O comando da AWS CLI a seguir cria a função do IAM `rds-s3-integration-role` para essa finalidade.

  Example

  Para Linux, macOS ou Unix:

```
aws iam create-role \
    --role-name rds-s3-integration-role \
    --assume-role-policy-document '{
      "Version": "2012-10-17",
      "Statement": [
        {
          "Effect": "Allow",
          "Principal": {
             "Service": "rds.amazonaws.com"
           },
          "Action": "sts:AssumeRole"
        }
      ]
    }'
```

  Para Windows:

  Altere os finais de linha para os que são compatíveis com a interface (^, em vez de \). Além disso, no Windows, é necessário inserir um caractere de escape em todas as aspas duplas

com um \. Para evitar a necessidade de inserir um caractere de escape em aspas no JSON, é possível salvá-lo em um arquivo e passá-lo como um parâmetro.

Primeiro, crie o arquivo `assume_role_policy.json` com a seguinte política:

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "rds.amazonaws.com"
                ]
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

Depois disso, use o comando a seguir para criar a função do IAM:

```
aws iam create-role ^
     --role-name rds-s3-integration-role ^
     --assume-role-policy-document file://file_path/assume_role_policy.json
```

Example de usar a chave de contexto de condição global para criar a função do IAM

Recomendamos usar as chaves de contexto de condição global `aws:SourceArn` e `aws:SourceAccount` em políticas baseadas em recursos para limitar as permissões do serviço a um recurso específico. Essa é a maneira mais eficiente de se proteger contra o problema "confused deputy".

Você pode usar as duas chaves de contexto de condição global e fazer com que o valor `aws:SourceArn` contenha o ID da conta. Nesses casos, verifique se o valor `aws:SourceAccount` e a conta no valor `aws:SourceArn` usa o mesmo ID de conta quando eles são usados na mesma instrução de política.

- Use `aws:SourceArn` se quiser acesso entre serviços para um único recurso.
- Use `aws:SourceAccount` se você quiser permitir que qualquer recurso nessa conta seja associado ao uso entre serviços.

Na política, certifique-se de usar a chave de contexto de condição global `aws:SourceArn` com o nome do recurso da Amazon (ARN) completo dos recursos acessando a função. Para integração do S3, certifique-se de incluir os ARNs de instância de banco de dados, conforme mostrado no exemplo a seguir.

Para Linux, macOS ou Unix:

```
aws iam create-role \
    --role-name rds-s3-integration-role \
    --assume-role-policy-document '{
      "Version": "2012-10-17",
      "Statement": [
        {
          "Effect": "Allow",
          "Principal": {
             "Service": "rds.amazonaws.com"
           },
          "Action": "sts:AssumeRole",
                "Condition": {
                    "StringEquals": {

 "aws:SourceArn":"arn:aws:rds:Region:my_account_ID:db:db_instance_identifier"
                    }
                }
        }
      ]
    }'
```

Para Windows:

Adicione as chaves de contexto de condição globais à `assume_role_policy.json`.

JSON

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "rds.amazonaws.com"
                ]
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {

 "aws:SourceArn":"arn:aws:rds:Region:my_account_ID:db:db_instance_identifier"
                }
            }
        }
    ]
}
```

Para anexar a política do IAM à função do IAM

- O seguinte comando da AWS CLI anexa a política à função denominada `rds-s3-integration-role`. Substitua *`your-policy-arn`* pelo ARN da política que você anotou em uma etapa anterior.

  Example

  Para Linux, macOS ou Unix:

  ```
  aws iam attach-role-policy \
      --policy-arn your-policy-arn \
      --role-name rds-s3-integration-role
  ```

  Para Windows:

  ```
  aws iam attach-role-policy ^
      --policy-arn your-policy-arn ^
      --role-name rds-s3-integration-role
  ```

Habilitação da integração do RDS for SQL Server com o S3

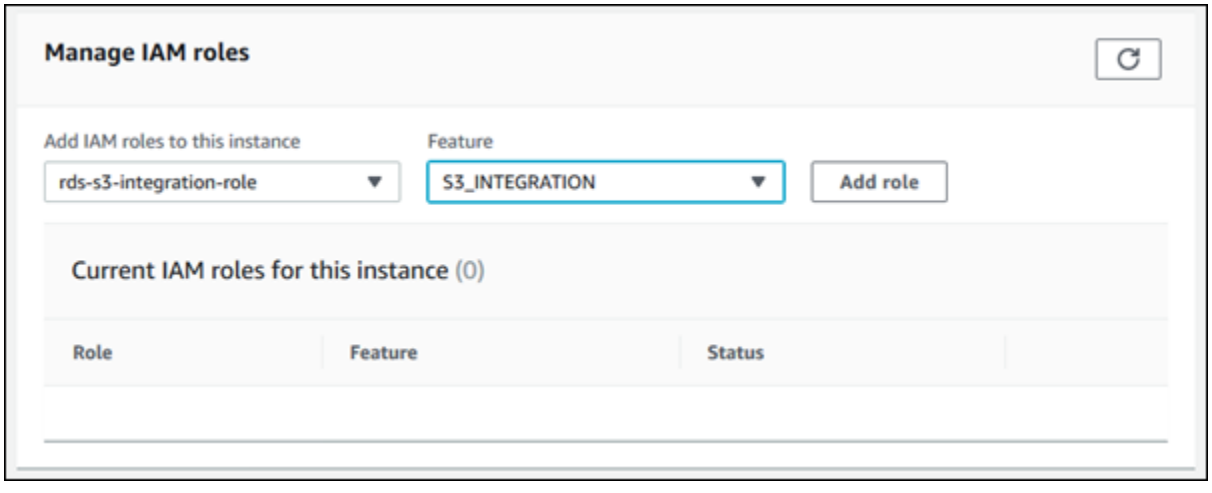
Na seção a seguir, é possível saber como habilitar a integração do Amazon S3 com o Amazon RDS for SQL Server. Para trabalhar com a integração do S3, a instância do banco de dados deve ser associada à função do IAM criada anteriormente para que você possa usar o parâmetro de nome do recurso `S3_INTEGRATION`.

> **Note**
>
> Para adicionar uma função do IAM a uma instância de banco de dados, o status da instância de banco de dados deve ser available (disponível).

Console

Para associar uma função do IAM à sua instância de banco de dados

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. Escolha o nome da instância de banco de dados do RDS for SQL Server para exibir seus detalhes.
3. Na guia Connectivity & security (Conectividade e segurança), na seção Manage IAM roles (Gerenciar funções do IAM), escolha a função do IAM a ser adicionada para Add IAM roles to this instance (Adicionar funções do IAM a essa instância).
4. Em Feature (Recurso), escolha S3_INTEGRATION.



5. Escolha Add role (adicionar função).

AWS CLI

Para adicionar a função do IAM à instância de banco de dados do RDS for SQL Server

- O seguinte comando da AWS CLI adiciona a função do IAM a uma instância de banco de dados do RDS for SQL Server chamada *mydbinstance*.

  Example

  Para Linux, macOS ou Unix:

```
aws rds add-role-to-db-instance \
    --db-instance-identifier mydbinstance \
    --feature-name S3_INTEGRATION \
    --role-arn your-role-arn
```

  Para Windows:

```
aws rds add-role-to-db-instance ^
    --db-instance-identifier mydbinstance ^
    --feature-name S3_INTEGRATION ^
    --role-arn your-role-arn
```

  Substitua *your-role-arn* pelo ARN da função que você anotou em uma etapa anterior. `S3_INTEGRATION` deve ser especificado para a opção `--feature-name`.

Transferência de arquivos entre o RDS for SQL Server e o Amazon S3

É possível usar os procedimentos armazenados do Amazon RDS para baixar e carregar arquivos entre o Amazon S3 e a instância de banco de dados do RDS. Também é possível usar os procedimentos armazenados do Amazon RDS para listar e excluir arquivos na instância do RDS.

Os arquivos que você baixa e carrega no S3 são armazenados na pasta `D:\S3`. Essa é a única pasta que pode ser usada para acessar seus arquivos. É possível organizar seus arquivos em subpastas, que são criadas para você ao incluir a pasta de destino durante o download.

Alguns dos procedimentos armazenados exigem que você forneça um nome de recurso da Amazon (ARN) ao bucket e arquivo do S3. O formato do ARN é `arn:aws:s3:::`*amzn-s3-demo-bucket*`/file_name`. O Amazon S3 não exige um número de conta ou região da AWS nos ARNs.

As tarefas de integração do S3 são executadas sequencialmente e compartilham a mesma fila como tarefas nativas de restauração e backup nativos. É possível ter no máximo duas tarefas em andamento a qualquer momento nesta fila. Pode demorar até cinco minutos para que o processamento da tarefa comece.

Baixar arquivos de um bucket do Amazon S3 em uma instância de banco de dados SQL Server

Para baixar arquivos de um bucket do S3 bucket em uma instância de banco de dados do RDS for SQL Server, use o procedimento armazenado do Amazon RDS `msdb.dbo.rds_download_from_s3` com os parâmetros a seguir.

Nome do parâmetro	Tipo de dados	Padrão	Obrigatório	Descrição
`@s3_arn_of_file`	NVARCHAR	–	Obrigatório	O ARN do S3 do arquivo para download, por exemplo: `arn:aws:s3::: amzn-s3-demo-bucket /mydata.csv`
`@rds_file_path`	NVARCHAR	–	Optional	O caminho do arquivo para a instância do RDS: Se não especificado, o caminho será `D:\S3\<filename in s3>`. O RDS oferece suporte a caminhos absolutos e relativos. Se você deseja criar uma subpasta, inclua-a no caminho do arquivo.
`@overwrite_file`	INT	0	Optional	Substitua o arquivo existente: 0 = não substituir 1 = substituir

É possível baixar arquivos sem uma extensão de arquivo e arquivos com as seguintes extensões: .bcp, .csv, .dat, .fmt, .info, .lst, .tbl, .txt e .xml.

> **Note**
>
> Arquivos com a extensão .ispac são têm suporte para download quando o SQL Server Integration Services está habilitado. Para obter mais informações sobre como habilitar o SSIS, consulte SQL Server Integration Services.
> Arquivos com as seguintes extensões têm suporte para download quando o SQL Server Analysis Services está
> habilitado: .abf, .asdatabase, .configsettings, .deploymentoptions, .deploymenttargets e .xmla. Para obter mais informações sobre como habilitar o SSAS, consulte SQL Server Analysis Services.

O exemplo a seguir mostra o procedimento armazenado para baixar arquivos do S3.

```
exec msdb.dbo.rds_download_from_s3
     @s3_arn_of_file='arn:aws:s3:::amzn-s3-demo-bucket/bulk_data.csv',
     @rds_file_path='D:\S3\seed_data\data.csv',
     @overwrite_file=1;
```

A operação de exemplo `rds_download_from_s3` cria uma pasta chamada `seed_data` em `D:\S3\`, se a pasta não existir ainda. Depois disso, o exemplo baixa o arquivo de origem `bulk_data.csv` do S3 para um novo arquivo chamado `data.csv` na instância de banco de dados. Se o arquivo existia anteriormente, ele será substituído porque o parâmetro `@overwrite_file` está definido como 1.

Fazer upload de arquivos de uma instância de banco de dados SQL Server para um bucket do Amazon S3

Para fazer upload de arquivos de uma instância de banco de dados do RDS for SQL Server para um bucket do S3, use o procedimento armazenado do Amazon RDS `msdb.dbo.rds_upload_to_s3` com os parâmetros a seguir.

Nome do parâmetro	Tipo de dados	Padrão	Obrigatório	Descrição
`@s3_arn_of_file`	NVARCHAR	–	Obrigatório	O ARN do S3 do arquivo a ser criado no S3, por exemplo: `arn:aws:s3::: amzn-s3-demo-bucket /mydata.csv`
`@rds_file_path`	NVARCHAR	–	Obrigatório	O caminho de arquivo do arquivo para fazer upload no S3. Caminhos relativos e absolutos têm suporte.
`@overwrite_file`	INT	–	Optional	Substitua o arquivo existente: 0 = não substituir 1 = substituir

O exemplo a seguir faz upload do arquivo chamado `data.csv` do local especificado em `D:\S3\seed_data\` para um arquivo `new_data.csv` no bucket do S3 especificado pelo ARN.

```
exec msdb.dbo.rds_upload_to_s3
  @rds_file_path='D:\S3\seed_data\data.csv',
  @s3_arn_of_file='arn:aws:s3:::amzn-s3-demo-bucket/new_data.csv',
  @overwrite_file=1;
```

Se o arquivo existia antes no S3, ele será substituído porque o parâmetro @overwrite_file está definido como 1.

Listar arquivos na instância de banco de dados do RDS

Para listar os arquivos disponíveis na instância de banco de dados, use a função e o procedimento armazenado. Primeiro, execute o seguinte procedimento armazenado para coletar detalhes de arquivo dos arquivos em `D:\S3\`.

```
exec msdb.dbo.rds_gather_file_details;
```

O procedimento armazenado retorna o ID da tarefa. Assim como outras tarefas, esse procedimento armazenado é executado de forma assíncrona. Assim que o status da tarefa for SUCCESS, é possível usar o ID da tarefa na função `rds_fn_list_file_details` para listar os diretórios e arquivos existentes em D:\S3\, conforme mostrado a seguir.

```
SELECT * FROM msdb.dbo.rds_fn_list_file_details(TASK_ID);
```

A função `rds_fn_list_file_details` exibe uma tabela com as colunas a seguir.

Parâmetro de saída	Descrição
`filepath`	Caminho absoluto do arquivo (por exemplo, `D:\S3\mydata.csv`)
`size_in_bytes`	Tamanho do arquivo (em bytes)
`last_modified_utc`	A data e a hora da última modificação em formato UTC
`is_directory`	Opção que indica se o item é um diretório (`true`/`false`)

Excluir arquivos na instância de banco de dados do RDS

Para excluir os arquivos disponíveis na instância de banco de dados, use o procedimento armazenado do Amazon RDS `msdb.dbo.rds_delete_from_filesystem` com os seguintes parâmetros.

Nome do parâmetro	Tipo de dados	Padrão	Obrigatório	Descrição
`@rds_file_path`	NVARCHAR	–	Obrigatório	O caminho de arquivo do arquivo a ser excluído. Caminhos relativos e absolutos têm suporte.

Nome do parâmetro	Tipo de dados	Padrão	Obrigatório	Descrição
`@force_delete`	INT	0	Optional	Para excluir um diretório, este sinalizador deve ser incluído e definido como 1. 1 = excluir um diretório Este parâmetro será ignorado se você estiver excluindo um arquivo.

Para excluir um diretório, o `@rds_file_path` deve terminar com uma barra invertida (\), e `@force_delete` deve ser definido como 1.

O exemplo a seguir exclui o arquivo `D:\S3\delete_me.txt`.

```
exec msdb.dbo.rds_delete_from_filesystem
    @rds_file_path='D:\S3\delete_me.txt';
```

O exemplo a seguir exclui o diretório `D:\S3\example_folder\`.

```
exec msdb.dbo.rds_delete_from_filesystem
    @rds_file_path='D:\S3\example_folder\',
    @force_delete=1;
```

Monitorar o status de uma tarefa de transferência de arquivo

Para rastrear o status da tarefa de integração do S3, chame a função `rds_fn_task_status`. Ela leva dois parâmetros. O primeiro parâmetro sempre deve ser `NULL` porque não se aplica à integração do S3. O segundo parâmetro aceita um ID de tarefa.

Para ver uma lista de todas as tarefas, defina o primeiro parâmetro como `NULL` e o segundo parâmetro como `0`, conforme mostrado no seguinte exemplo.

```
SELECT * FROM msdb.dbo.rds_fn_task_status(NULL,0);
```

Para obter uma tarefa específica, defina o primeiro parâmetro como NULL e o segundo parâmetro como o ID da tarefa, conforme mostrado no exemplo a seguir.

```
SELECT * FROM msdb.dbo.rds_fn_task_status(NULL,42);
```

A função rds_fn_task_status retorna as informações a seguir.

Parâmetro de saída	Descrição
task_id	O ID da tarefa.
task_type	Para a integração do S3, as tarefas podem ter os seguintes tipos de tarefa: • DOWNLOAD_FROM_S3 • UPLOAD_TO_S3 • LIST_FILES_ON_DISK • DELETE_FILES_ON_DISK
database_name	Não aplicável a tarefas de integração do S3.
% complete	O progresso da tarefa em porcentagem.
duration(mins)	A quantidade de tempo gasta na tarefa, em minutos.
lifecycle	O status da tarefa. Os possíveis status são os seguintes: • CREATED – Após chamar um dos procedimentos armazenados de integração do S3, uma tarefa será criada, e o status será definido como CREATED. • IN_PROGRESS – Após iniciar uma tarefa, o status será definido como IN_PROGRESS. Pode demorar até 5 minutos para que o status mude de CREATED para IN_PROGRESS.

Parâmetro de saída	Descrição
	• SUCCESS – Após concluir uma tarefa, o status será definido como SUCCESS. • ERROR – Se uma tarefa falhar, o status será definido como ERROR. Para obter mais informações sobre o erro, consulte a coluna task_info . • CANCEL_REQUESTED – Após chamar rds_cancel_task , o status da tarefa será definido como CANCEL_REQUESTED . • CANCELLED – após uma tarefa ter sido cancelada com sucesso, o status da tarefa será definido como CANCELLED .
task_info	Informações adicionais sobre a tarefa. Se um erro ocorrer durante o processamento, essa coluna conterá informações sobre o erro.
last_updated	A data e hora em que o status da tarefa foi atualizado pela última vez.
created_at	A data e hora em que a tarefa foi criada.
S3_object_arn	O ARN do objeto do S3 baixado ou carregado.
overwrite_S3_backup_file	Não aplicável a tarefas de integração do S3.
KMS_master_key_arn	Não aplicável a tarefas de integração do S3.
filepath	O caminho de arquivo na instância de banco de dados do RDS
overwrite_file	Uma opção que indica se um arquivo existente foi substituído.
task_metadata	Não aplicável a tarefas de integração do S3.

Cancelar uma tarefa

Para cancelar tarefas de integração do S3, use o procedimento armazenado `msdb.dbo.rds_cancel_task` com o parâmetro `task_id`. Tarefas de listagem e exclusão que estão em progresso não podem ser canceladas. O exemplo a seguir mostra uma solicitação para cancelar uma tarefa.

```
exec msdb.dbo.rds_cancel_task @task_id = 1234;
```

Para obter uma visão geral de todas as tarefas e seus IDs de tarefa, use a função `rds_fn_task_status` conforme descrito em Monitorar o status de uma tarefa de transferência de arquivo.

Limitações Multi-AZ para a integração do S3

Em instâncias Multi-AZ, os arquivos na pasta `D:\S3` são excluídos na réplica em espera após um failover. Um failover pode ser planejado, por exemplo, durante modificações de instâncias de banco de dados, como alterar a classe de instância ou atualizar a versão do mecanismo. Ou um failover pode ser não planejado, durante uma interrupção do principal.

> **Note**
>
> Não recomendamos usar a pasta `D:\S3` para armazenamento de arquivos. A prática recomendada é carregar arquivos criados no Amazon S3 para torná-los duráveis e baixar arquivos quando precisar importar dados.

Para determinar o último tempo de failover, é possível usar o procedimento `msdb.dbo.rds_failover_time` armazenado. Para obter mais informações, consulte Determinar o horário do último failover do Amazon RDS para SQL Server.

Example de nenhum failover recente

Este exemplo mostra a saída quando não há nenhum failover recente nos logs de erros. Nenhum failover aconteceu desde 29-04-2020 23:59:00.01.

Portanto, todos os arquivos baixados após esse horário que não foram excluídos usando o procedimento `rds_delete_from_filesystem` armazenado ainda estão acessíveis no host atual. Arquivos baixados antes desse horário também podem estar disponíveis.

errorlog_available_from	recent_failover_time
29-04-2020 23:59:00.0100000	nulo

Example de failover recente

Este exemplo mostra a saída quando há um failover nos logs de erros. O failover mais recente foi em 05-05-2020 18:57:51.89.

Todos os arquivos baixados após esse horário que não foram excluídos usando o procedimento `rds_delete_from_filesystem` armazenado ainda estão acessíveis no host atual.

errorlog_available_from	recent_failover_time
29-04-2020 23:59:00.0100000	05-05-2020 18:57:51.8900000

Desabilitação da integração do RDS for SQL Server com o S3

A seguir, é possível saber como desabilitar a integração do Amazon S3 com o Amazon RDS for SQL Server. Os arquivos em `D:\S3\` não são excluídos ao desabilitar a integração do S3.

> **Note**
>
> Para remover uma função do IAM de uma instância de banco de dados, o status da instância de banco de dados deve ser `available`.

Console

Como desassociar uma função do IAM da instância de banco de dados

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. Escolha o nome da instância de banco de dados do RDS for SQL Server para exibir seus detalhes.

3. Na guia Connectivity & security (Conectividade e segurança), na seção Manage IAM roles (Gerenciar funções do IAM), escolha a função do IAM a ser removida.
4. Escolha Excluir.

AWS CLI

Para remover a função do IAM da instância de banco de dados do RDS for SQL Server

- O seguinte comando da AWS CLI remove a função do IAM de uma instância de banco de dados do RDS for SQL Server chamada *mydbinstance*.

  Example

  Para Linux, macOS ou Unix:

```
aws rds remove-role-from-db-instance \
    --db-instance-identifier mydbinstance \
    --feature-name S3_INTEGRATION \
    --role-arn your-role-arn
```

  Para Windows:

```
aws rds remove-role-from-db-instance ^
    --db-instance-identifier mydbinstance ^
    --feature-name S3_INTEGRATION ^
    --role-arn your-role-arn
```

  Substitua *your-role-arn* pelo ARN adequado da função do IAM para a opção `--feature-name`.

Usar o Database Mail no Amazon RDS for SQL Server

Você pode usar o Database Mail para enviar mensagens de e-mail aos usuários a partir da instância de banco de dados do Amazon RDS no SQL Server. As mensagens podem conter arquivos e resultados de consultas. O Database Mail inclui os seguintes componentes:

- Objetos deconfiguração e segurança – esses objetos criam perfis e contas e são armazenados no banco de dados `msdb`.
- Objetos de mensagens – esses objetos incluem o procedimento armazenado sp_send_dbmail usado para enviar mensagens e estruturas de dados que contêm informações sobre mensagens. Eles estão armazenados no banco de dados `msdb`.
- Objetos de registro e auditoria – o Database Mail grava informações de registro no banco de dados `msdb` e no log de eventos de aplicações do Microsoft Windows.
- Executável do Database Mail – `DatabaseMail.exe` lê a partir de uma fila no banco de dados `msdb` e envia mensagens de e-mail.

O RDS oferece suporte ao Database Mail para todas as versões do SQL Server nas edições Web, Standard e Enterprise.

Limitações

As seguintes limitações se aplicam ao uso do Database Mail em sua instância de banco de dados SQL Server:

- O Database Mail não é compatível com o SQL Server Express Edition.
- A modificação dos parâmetros de configuração do Database Mail não é aceita. Para ver os valores predefinidos (padrão), use o procedimento armazenado sysmail_help_configure_sp .
- Os anexos de arquivo não são totalmente compatíveis. Para ter mais informações, consulte Trabalhar com anexos de arquivos.
- O tamanho máximo do anexo do arquivo é de 1 MB.
- O Database Mail requer configuração adicional em instâncias de banco de dados Multi-AZ. Para ter mais informações, consulte Considerações para implantações Multi-AZ.
- Não há suporte para configurar o SQL Server Agent para enviar mensagens de email para operadores predefinidos.

Habilitar Database Mail

Use o seguinte processo para habilitar o Database Mail para a instância de banco de dados:

1. Crie um novo parameter group.
2. Modifique o grupo de parâmetros para definir o parâmetro `database mail xps` como 1.
3. Associe o grupo de parâmetros à instância de banco de dados.

Criar o grupo de parâmetros para o Database Mail

Crie um grupo de parâmetros para o parâmetro `database mail xps` que corresponde à edição e à versão do SQL Server da instância de banco de dados.

> **Note**
>
> Você também pode modificar um grupo de parâmetros existente. Siga o procedimento em Modificar o parâmetro que habilita o Database Mail.

Console

O exemplo a seguir cria um grupo de parâmetros para o SQL Server Standard Edition 2016.

Como criar o grupo de parâmetros

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Parameter groups (Grupos de parâmetros).
3. Escolha Create parameter group (Criar parameter group).
4. No painelCreate parameter group (Criar parameter group), faça o seguinte:
   a. Em Família de grupos de parâmetros, escolha sqlserver-se-13.0.
   b. Em Nome do grupo, insira um identificador para o grupo de parâmetros, como **`dbmail-sqlserver-se-13`**.
   c. Em Descrição, insira **`Database Mail XPs`**.
5. Escolha Criar.

CLI

O exemplo a seguir cria um grupo de parâmetros para o SQL Server Standard Edition 2016.

Como criar o grupo de parâmetros

- Use um dos seguintes comandos.

  Example

  Para Linux, macOS ou Unix:

```
aws rds create-db-parameter-group \
    --db-parameter-group-name dbmail-sqlserver-se-13 \
    --db-parameter-group-family "sqlserver-se-13.0" \
    --description "Database Mail XPs"
```

  Para Windows:

```
aws rds create-db-parameter-group ^
    --db-parameter-group-name dbmail-sqlserver-se-13 ^
    --db-parameter-group-family "sqlserver-se-13.0" ^
    --description "Database Mail XPs"
```

Modificar o parâmetro que habilita o Database Mail

Modifique o parâmetro `database mail xps` no grupo de parâmetros que corresponde à edição e à versão do SQL Server da instância de banco de dados.

Para habilitar o Database Mail, defina o parâmetro `database mail xps` como 1.

Console

O exemplo a seguir modifica o grupo de parâmetros que você criou para o SQL Server Standard Edition 2016.

Como modificar o grupo de parâmetros

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, selecione Parameter groups.

3. Escolha o grupo de parâmetros, como debmail-sqlserver-se-13.
4. Em Parâmetros, filtre a lista de parâmetros para **`mail`**.
5. Escolha xps de email de bancode dados.
6. Escolha Edit parameters.
7. Digite **1**.
8. Escolha Salvar alterações.

CLI

O exemplo a seguir modifica o grupo de parâmetros que você criou para o SQL Server Standard Edition 2016.

Como modificar o grupo de parâmetros

- Use um dos seguintes comandos.

  Example

  Para Linux, macOS ou Unix:

```
aws rds modify-db-parameter-group \
    --db-parameter-group-name dbmail-sqlserver-se-13 \
    --parameters "ParameterName='database mail
 xps',ParameterValue=1,ApplyMethod=immediate"
```

  Para Windows:

```
aws rds modify-db-parameter-group ^
    --db-parameter-group-name dbmail-sqlserver-se-13 ^
    --parameters "ParameterName='database mail
 xps',ParameterValue=1,ApplyMethod=immediate"
```

Associar o grupo de parâmetros à instância de banco de dados

É possível usar o AWS Management Console ou a AWS CLI para associar o grupo de parâmetros do Database Mail à instância de banco de dados.

Console

É possível associar o grupo de parâmetros do Database Mail a uma instância de banco de dados nova ou existente.

- Para uma nova instância de banco de dados, associe-o ao executar a instância. Para ter mais informações, consulte Criar uma instância de banco de dados do Amazon RDS.
- Para uma instância de banco de dados existente, associe-o modificando a instância. Para ter mais informações, consulte Modificar uma instância de banco de dados do Amazon RDS.

CLI

É possível associar o grupo de parâmetros do Database Mail a uma instância de banco de dados nova ou existente.

Para criar uma instância de banco de dados com o grupo de parâmetros do Database Mail

- Especifique o mesmo tipo de mecanismo de banco de dados e a versão principal que você usou ao criar o grupo de parâmetros.

  Example

  Para Linux, macOS ou Unix:

```
aws rds create-db-instance \
    --db-instance-identifier mydbinstance \
    --db-instance-class db.m5.2xlarge \
    --engine sqlserver-se \
    --engine-version 13.00.5426.0.v1 \
    --allocated-storage 100 \
    --manage-master-user-password \
    --master-username admin \
    --storage-type gp2 \
    --license-model li
    --db-parameter-group-name dbmail-sqlserver-se-13
```

  Para Windows:

```
aws rds create-db-instance ^
    --db-instance-identifier mydbinstance ^
    --db-instance-class db.m5.2xlarge ^
```

```
    --engine sqlserver-se ^
    --engine-version 13.00.5426.0.v1 ^
    --allocated-storage 100 ^
    --manage-master-user-password ^
    --master-username admin ^
    --storage-type gp2 ^
    --license-model li ^
    --db-parameter-group-name dbmail-sqlserver-se-13
```

Para modificar uma instância de banco de dados e associar o grupo de parâmetros do Database Mail

- Use um dos seguintes comandos.

  Example

  Para Linux, macOS ou Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --db-parameter-group-name dbmail-sqlserver-se-13 \
    --apply-immediately
```

  Para Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --db-parameter-group-name dbmail-sqlserver-se-13 ^
    --apply-immediately
```

Configurando o Correio do Banco

Execute as seguintes tarefas para configurar o Database Mail:

1. Crie o perfil do Database Mail.
2. Crie a conta Database Mail.
3. Adicione a conta de Correio do Banco de Dados ao perfil de Correio do Banco de Dados.
4. Adicione usuários ao perfil Database Mail.

> **Note**
>
> Para configurar o Database Mail, certifique-se de que você tem `execute` permissão nos procedimentos armazenados no `msdb` banco de dados.

Criando o perfil de Correio de Banco de

Para criar o perfil Database Mail, use o procedimento armazenado sysmail_add_profile_sp . O exemplo a seguir cria um perfil chamado `Notifications`.

Para criar o perfil

- Use a seguinte instrução SQL.

```
USE msdb
GO

EXECUTE msdb.dbo.sysmail_add_profile_sp
    @profile_name         = 'Notifications',
    @description          = 'Profile used for sending outgoing notifications using
 Amazon SES.';
GO
```

Criando a conta Database Mail

Para criar a conta Database Mail, use o procedimento armazenado sysmail_add_account_sp . O exemplo a seguir cria uma conta chamada `SES` em uma instância de banco de dados do RDS para SQL Server em uma VPC privada, usando o Amazon Simple Email Service.

O uso do Amazon SES requer os seguintes parâmetros:

- `@email_address`: uma identidade verificada do Amazon SES. Para ter mais informações, consulte Identidades verificadas no Amazon SES.
- `@mailserver_name`: um endpoint SMTP do Amazon SES. Para ter mais informações, consulte Conexão a um endpoint SMTP do Amazon SES.
- `@username`: um nome de usuário SMTP do Amazon SES. Para ter mais informações, consulte Obtenção de credenciais SMTP do Amazon SES.

Não use um nome de usuário do AWS Identity and Access Management.

- `@password`: uma senha SMTP do Amazon SES. Para ter mais informações, consulte Obtenção de credenciais SMTP do Amazon SES.

Para criar a conta

- Use a seguinte instrução SQL.

```
USE msdb
GO

EXECUTE msdb.dbo.sysmail_add_account_sp
    @account_name        = 'SES',
    @description         = 'Mail account for sending outgoing notifications.',
    @email_address       = 'nobody@example.com',
    @display_name        = 'Automated Mailer',
    @mailserver_name     = 'vpce-0a1b2c3d4e5f-01234567.email-smtp.us-
west-2.vpce.amazonaws.com',
    @port                = 587,
    @enable_ssl          = 1,
    @username            = 'Smtp_Username',
    @password            = 'Smtp_Password';
GO
```

> **Note**
>
> Especifique credenciais diferentes dos prompts mostrados aqui como prática recomendada de segurança.

Adicionar a conta do Database Mail ao perfil do Database Mail

Para adicionar a conta Database Mail ao perfil Database Mail, use o procedimento armazenado sysmail_add_profileaccount_sp. O exemplo a seguir adiciona a conta do SES ao perfil `Notifications`.

Para adicionar a conta ao perfil

- Use a seguinte instrução SQL.

```
USE msdb
GO

EXECUTE msdb.dbo.sysmail_add_profileaccount_sp
    @profile_name        = 'Notifications',
    @account_name        = 'SES',
    @sequence_number     = 1;
GO
```

Adicionar usuários ao perfil Database Mail

Para conceder permissão para que um principal do banco de dados `msdb` use um perfil do Database Mail, use o procedimento armazenado sysmail_add_principalprofile_sp. Um principal é uma entidade que pode solicitar recursos do SQL Server. O principal do banco de dados deve mapear para um usuário de autenticação do SQL Server, um usuário de Autenticação do Windows ou um grupo de autenticação do Windows.

O exemplo a seguir concede acesso público ao perfil `Notifications`.

Para adicionar um usuário ao perfil

- Use a seguinte instrução SQL.

```
USE msdb
GO

EXECUTE msdb.dbo.sysmail_add_principalprofile_sp
    @profile_name       = 'Notifications',
    @principal_name     = 'public',
    @is_default         = 1;
GO
```

Procedimentos armazenados e funções do Amazon RDS para o Database Mail

A Microsoft fornece procedimentos armazenados para usar o Database Mail, como criar, listar, atualizar e excluir contas e perfis. Além disso, o RDS fornece os procedimentos e funções armazenados para o Database Mail mostrados na tabela a seguir.

Procedimento/Função	Descrição
rds_fn_sysmail_allitems	Mostra as mensagens enviadas, incluindo as enviadas por outros usuários.
rds_fn_sysmail_event_log	Mostra eventos, incluindo aqueles para mensagens enviadas por outros usuários.
rds_fn_sysmail_mailattachments	Mostra anexos, incluindo aqueles para mensagens enviadas por outros usuários.
rds_sysmail_control	Inicia e interrompe a fila de correio (processo DatabaseMail.exe).
rds_sysmail_delete_mailitems_sp	Exclui mensagens de e-mail enviadas por todos os usuários das tabelas internas do Database Mail.

Enviando mensagens de e-mail usando o Database

Use o procedimento armazenado sp_send_dbmail para enviar mensagens de email usando o Database Mail.

Uso

```
EXEC msdb.dbo.sp_send_dbmail
@profile_name = 'profile_name',
@recipients = 'recipient1@example.com[; recipient2; ... recipientn]',
@subject = 'subject',
@body = 'message_body',
[@body_format = 'HTML'],
[@file_attachments = 'file_path1; file_path2; ... file_pathn'],
[@query = 'SQL_query'],
[@attach_query_result_as_file = 0|1]';
```

Os seguintes parâmetros são obrigatórios:

- `@profile_name` – o nome do perfil Database Mail a partir do qual enviar a mensagem.
- `@recipients` – a lista delimitada por ponto e vírgula de endereços de e-mail para os quais enviar a mensagem.

- `@subject` – o assunto da mensagem.
- `@body` – o corpo da mensagem. Você também pode usar uma variável declarada como o corpo.

Os seguintes parâmetros são opcionais:

- `@body_format` – esse parâmetro é usado com uma variável declarada para enviar e-mail em formato HTML.
- `@file_attachments` – a lista delimitada por ponto e vírgula de anexos de mensagens. Os caminhos de arquivo devem ser caminhos absolutos.
- `@query` – uma consulta SQL a ser executada. Os resultados da consulta podem ser anexados como um arquivo ou incluídos no corpo da mensagem.
- `@attach_query_result_as_file` – Se deve anexar o resultado da consulta como um arquivo. Defina como 0 para não, 1 para sim. O padrão é 0.

Exemplos

Os exemplos a seguir demonstram como enviar mensagens de e-mail.

Example de enviar uma mensagem para um único destinatário

```
USE msdb
GO

EXEC msdb.dbo.sp_send_dbmail
     @profile_name       = 'Notifications',
     @recipients         = 'nobody@example.com',
     @subject            = 'Automated DBMail message - 1',
     @body               = 'Database Mail configuration was successful.';
GO
```

Example de enviar uma mensagem para vários destinatários

```
USE msdb
GO

EXEC msdb.dbo.sp_send_dbmail
     @profile_name       = 'Notifications',
     @recipients         = 'recipient1@example.com;recipient2@example.com',
     @subject            = 'Automated DBMail message - 2',
```

```
     @body               = 'This is a message.';
GO
```

Example de enviar um resultado de consulta SQL como um anexo de arquivo

```
USE msdb
GO

EXEC msdb.dbo.sp_send_dbmail
     @profile_name       = 'Notifications',
     @recipients         = 'nobody@example.com',
     @subject            = 'Test SQL query',
     @body               = 'This is a SQL query test.',
     @query              = 'SELECT * FROM abc.dbo.test',
     @attach_query_result_as_file = 1;
GO
```

Example de enviar uma mensagem em formato HTML

```
USE msdb
GO

DECLARE @HTML_Body as NVARCHAR(500) = 'Hi, <h4> Heading </h4> </br> See the report. <b>
 Regards </b>';

EXEC msdb.dbo.sp_send_dbmail
     @profile_name       = 'Notifications',
     @recipients         = 'nobody@example.com',
     @subject            = 'Test HTML message',
     @body               = @HTML_Body,
     @body_format        = 'HTML';
GO
```

Example de enviar uma mensagem usando um gatilho quando ocorre um evento específico no banco de dados

```
USE AdventureWorks2017
GO
IF OBJECT_ID ('Production.iProductNotification', 'TR') IS NOT NULL
DROP TRIGGER Purchasing.iProductNotification
GO
```

```
CREATE TRIGGER iProductNotification ON Production.Product
   FOR INSERT
   AS
   DECLARE @ProductInformation nvarchar(255);
   SELECT
   @ProductInformation = 'A new product, ' + Name + ', is now available for $' +
 CAST(StandardCost AS nvarchar(20)) + '!'
   FROM INSERTED i;

EXEC msdb.dbo.sp_send_dbmail
     @profile_name       = 'Notifications',
     @recipients         = 'nobody@example.com',
     @subject            = 'New product information',
     @body               = @ProductInformation;
GO
```

Visualizar mensagens, logs e anexos

Você usa procedimentos armazenados do RDS para visualizar mensagens, logs de eventos e anexos.

Para visualizar todas as mensagens de e-mail

- Use a seguinte consulta SQL:

```
SELECT * FROM msdb.dbo.rds_fn_sysmail_allitems(); --WHERE sent_status='sent' or
 'failed' or 'unsent'
```

Para visualizar todos os logs de eventos de e-mail

- Use a seguinte consulta SQL:

```
SELECT * FROM msdb.dbo.rds_fn_sysmail_event_log();
```

Para visualizar todos os anexos de e-mail

- Use a seguinte consulta SQL:

```
SELECT * FROM msdb.dbo.rds_fn_sysmail_mailattachments();
```

Excluir mensagens

Use o procedimento armazenado `rds_sysmail_delete_mailitems_sp` para excluir mensagens.

> **Note**
>
> O RDS exclui automaticamente os itens da tabela de e-mail quando os dados do histórico do DBMail atingem 1 GB de tamanho, com um período de retenção de pelo menos 24 horas. Se você quiser manter itens de e-mail por um período mais longo, poderá arquivá-los. Para ter mais informações, consulte Create a SQL Server Agent Job to Archive Database Mail Messages and Event Logs na documentação da Microsoft.

Para excluir todas as mensagens de e-mail

- Use a seguinte instrução SQL.

```
DECLARE @GETDATE datetime
SET @GETDATE = GETDATE();
EXECUTE msdb.dbo.rds_sysmail_delete_mailitems_sp @sent_before = @GETDATE;
GO
```

Para excluir todas as mensagens de e-mail com um status específico

- Use a seguinte instrução SQL para excluir todas as mensagens com falha.

```
DECLARE @GETDATE datetime
SET @GETDATE = GETDATE();
EXECUTE msdb.dbo.rds_sysmail_delete_mailitems_sp @sent_status = 'failed';
GO
```

Iniciar e interromper a fila de e-mails

Use as seguintes instruções para iniciar e interromper a fila de e-mails de banco de dados:

Tópicos

- Iniciar a fila de e-mails
- Parar a fila de e-mails

Iniciar a fila de e-mails

Use o procedimento armazenado `rds_sysmail_control` para iniciar o processo Database Mail.

> **Note**
>
> A habilitação do Database Mail inicia automaticamente a fila de e-mails

Para iniciar a fila de correio

- Use a seguinte instrução SQL.

```
EXECUTE msdb.dbo.rds_sysmail_control start;
GO
```

Parar a fila de e-mails

Use o procedimento armazenado `rds_sysmail_control` para interromper o processo Database Mail.

Para interromper a fila de e-mails

- Use a seguinte instrução SQL.

```
EXECUTE msdb.dbo.rds_sysmail_control stop;
GO
```

Trabalhar com anexos de arquivos

As seguintes extensões de anexo de arquivo não são compatíveis com as mensagens do Database Mail do RDS no SQL
Server: .ade, .adp, .apk, .appx, .appxbundle, .bat, .bak, .cab, .chm, .cmd, .com, .cpl, .dll, .dmg, .exe, .hta, .inf
sc, .wsf e .wsh.

O Database Mail usa o contexto de segurança do Microsoft Windows do usuário atual para controlar o acesso aos arquivos. Os usuários que fazem login com o SQL Server Authentication não podem anexar arquivos usando o parâmetro `@file_attachments` com o procedimento

armazenado `sp_send_dbmail`. O Windows não permite que o SQL Server forneça credenciais de um computador remoto para outro computador remoto. Portanto, o Database Mail não pode anexar arquivos de um compartilhamento de rede quando o comando é executado a partir de um computador diferente do computador que esteja executando o SQL Server.

No entanto, você pode usar tarefas do SQL Server Agent para anexar arquivos. Para ter mais informações sobre o SQL Server Agent, consulte Usar o agente do SQL Server para o Amazon RDS e SQL Server Agent na documentação da Microsoft.

Considerações para implantações Multi-AZ

Quando você configura o Database Mail em uma instância de banco de dados Multi-AZ, a configuração não é propagada automaticamente para o secundário. Recomendamos converter a instância Multi-AZ em uma instância Single-AZ, configurar o Database Mail e, em seguida, converter a instância de banco de dados de volta para Multi-AZ. Em seguida, os nós primário e secundário têm a configuração do Database Mail.

Se você criar uma réplica de leitura a partir da instância Multi-AZ que tenha o Database Mail configurado, a réplica herdará a configuração, mas sem a senha para o servidor SMTP. Atualize a conta Database Mail com a senha.

Remover a restrição de SMTP (porta 25)

Por padrão, a AWS bloqueia o tráfego de saída no SMTP (porta 25) para instâncias de banco de dados do RDS para SQL Server. Isso é feito para evitar spam com base nas políticas do proprietário da interface de rede elástica. É possível remover essa restrição, se necessário. Consulte mais informações em Como removo a restrição à porta 25 de uma instância do Amazon EC2 ou de uma função do Lambda?

Suporte ao armazenamento de instâncias para o banco de dados tempdb no Amazon RDS for SQL Server

Um armazenamento de instâncias fornece armazenamento temporário em nível de bloco para a instância de banco de dados. Esse armazenamento está localizado em discos que estão anexados fisicamente ao computador host. Esses discos têm armazenamento de instância de memória expressa não volátil (NVMe) baseado em unidades de estado sólido (SSDs). Esse armazenamento é otimizado para baixa latência, performance de E/S aleatória muito alto e alta taxa de transferência de leitura sequencial.

Ao colocar arquivos de dados `tempdb` e arquivos de log `tempdb` no armazenamento de instâncias, você pode obter latências de leitura e gravação mais baixas em comparação com o armazenamento padrão com base no Amazon EBS.

> **Note**
>
> Os arquivos de banco de dados do SQL Server e de log do banco de dados não são colocados no armazenamento de instâncias.

Habilitar o armazenamento de instâncias

Quando o RDS provisiona instâncias de banco de dados com um dos seguintes tipos de instância, o banco de dados `tempdb` é automaticamente colocado no armazenamento de instâncias:

- db.m5d
- db.r5d
- db.x2iedn

Para habilitar o armazenamento de instâncias, siga um destes procedimentos:

- Crie uma instância de banco de dados do SQL Server usando um desses tipos de instância. Para obter mais informações, consulte Criar uma instância de banco de dados do Amazon RDS.
- Modifique uma instância de banco de dados SQL Server existente para usar uma delas. Para obter mais informações, consulte Modificar uma instância de banco de dados do Amazon RDS.

O armazenamento de instâncias está disponível em todas as regiões da AWS onde um ou mais desses tipos de instância são compatíveis. Para obter mais informações sobre as classes de instância `db.m5d` e `db.r5d`, consulte Classes de instâncias de banco de dados . Para obter mais informações sobre as classes de instância compatíveis pelo Amazon RDS for SQL Server, consulte Suporte a classes de instâncias de banco de dados para o Microsoft SQL Server.

Considerações de localização e tamanho do arquivo

Em instâncias sem armazenamento de instâncias, o RDS armazena os arquivos de dados e de logs `tempdb` no diretório `D:\rdsdbdata\DATA`. Ambos os arquivos começam em 8 MB por padrão.

Em instâncias com um armazenamento de instâncias, o RDS armazena os arquivos de dados e de logs `tempdb` no diretório `T:\rdsdbdata\DATA`.

Quando `tempdb` tem apenas um arquivo de dados (`tempdb.mdf`) e um arquivo de log (`templog.ldf`), `templog.ldf` inicia em 8 MB por padrão e `tempdb.mdf` inicia em 80% ou mais da capacidade de armazenamento da instância. Vinte por cento da capacidade de armazenamento ou 200 GB, o que for menor, é mantido livre para começar. Múltiplos arquivos de dados `tempdb` dividem os 80% do espaço em disco uniformemente, enquanto os arquivos de log sempre têm um tamanho inicial de 8 MB.

Por exemplo, se você modificar sua classe de instância de banco de dados de `db.m5.2xlarge` para `db.m5d.2xlarge`, o tamanho dos arquivos de dados `tempdb` aumenta de 8 MB cada para 234 GB no total.

> **Note**
>
> Além dos arquivos de dados e de log `tempdb` no armazenamento de instâncias (`T:\rdsdbdata\DATA`), você ainda pode criar arquivos de dados e de log `tempdb` extras no volume de dados (`D:\rdsdbdata\DATA`). Esses arquivos sempre têm um tamanho inicial de 8 MB.

Considerações sobre backup

Talvez seja necessário reter backups por longos períodos, incorrendo em custos ao longo do tempo. Os blocos de `tempdb` dados e log podem mudar muitas vezes dependendo da carga de trabalho. Isso pode aumentar consideravelmente o tamanho do snapshot de banco de dados.

Quando `tempdb` está no armazenamento de instâncias, os snapshots não incluem arquivos temporários. Isso significa que os tamanhos de snapshots são menores e consomem menos a alocação de backup gratuita em comparação com o armazenamento de somente EBS.

Erros completos de disco

Se você usar todo o espaço disponível no armazenamento de instâncias, poderá receber erros como os seguintes:

- O log de transações do banco de dados 'tempdb' está cheio devido a 'ACTIVE_TRANSACTION'.
- Não foi possível alocar espaço para o objeto dbo.SORT temporary run storage: 140738941419520' no banco de dados 'tempdb' porque o grupo de arquivos 'PRIMÁRIO' está cheio. Crie espaço em disco excluindo arquivos desnecessários, soltando objetos no grupo de arquivos, adicionando arquivos adicionais ao grupo de arquivos ou definindo o crescimento automático para arquivos existentes no grupo de arquivos.

Você pode executar um ou mais dos procedimentos a seguir quando o armazenamento de instâncias estiver cheio:

- Ajuste sua carga de trabalho ou a maneira como você usa `tempdb`.
- Aumente a escala para usar uma classe de instância de banco de dados com mais armazenamento NVMe.
- Pare de usar o armazenamento de instâncias e use uma classe de instância com apenas armazenamento do EBS.
- Use um modo misto adicionando dados secundários ou arquivos de log para `tempdb` no volume do EBS.

Remover o armazenamento de instâncias

Para remover o armazenamento de instâncias, modifique a instância de banco de dados do SQL Server para usar um tipo de instância que não comporte o armazenamento de instâncias como db.m5, db.r5 ou db.x1e.

> **Note**
>
> Quando você remove o armazenamento de instâncias, os arquivos temporários são movidos para o diretório `D:\rdsdbdata\DATA` e reduzidos em tamanho para 8 MB.

Usar eventos estendidos com Amazon RDS for Microsoft SQL Server

Você pode usar eventos estendidos no Microsoft SQL Server para capturar informações de depuração e solucionar problemas do Amazon RDS for SQL Server. Os eventos estendidos substituem o SQL Trace e o Server Profiler, que foram defasados pela Microsoft. Os eventos estendidos são semelhantes aos traços do Profiler, mas com controle mais granular nos eventos que estão sendo monitorados. Os eventos estendidos são compatíveis com as versões 2016 do SQL Server e posteriores do Amazon RDS. Para obter mais informações, consulte Visão geral de eventos estendidos na documentação da Microsoft.

Os eventos estendidos são ativados automaticamente para usuários com privilégios de usuário primário no Amazon RDS for SQL Server.

Tópicos

- Limitações e recomendações
- Configuração de eventos estendidos no RDS for SQL Server
- Considerações para implantações Multi-AZ
- Consultando arquivos de eventos estendidos

Limitações e recomendações

Ao usar eventos estendidos do RDS for SQL Server, as seguintes limitações se aplicam:

- Os eventos estendidos são compatíveis apenas com as Edições Enterprise e Standard.
- Você não pode alterar as sessões de eventos estendidos padrão.
- Defina o modo de partição de memória de sessão como `NONE`.
- O modo de retenção de eventos de sessão pode ser `ALLOW_SINGLE_EVENT_LOSS` ou `ALLOW_MULTIPLE_EVENT_LOSS`.
- Os destinos de monitoramento de eventos para Windows (ETW) não são compatíveis.
- Verifique se os destinos dos arquivo estão no diretório `D:\rdsdbdata\log`.
- Para alvos de correspondência de pares, defina a propriedade `respond_to_memory_pressure` como `1`.
- A memória de destino do buffer de anel não pode ser maior que 4 MB.
- As seguintes ações não são aceitas:
  - `debug_break`

- `create_dump_all_threads`
- `create_dump_single_threads`

- O evento `rpc_completed` é compatível com as seguintes versões e posteriores: 15.0.4083.2, 14.0.3370.1, 13.0.5865.1, 12.0.6433.1, 11.0.7507.2

Configuração de eventos estendidos no RDS for SQL Server

No RDS for SQL Server, você pode configurar os valores de determinados parâmetros de sessões de eventos estendidos. A tabela a seguir descreve os parâmetros configuráveis.

Nome do parâmetro	Descrição
`xe_session_max_memory`	Especifica a quantidade máxima de memória a ser aloca corresponde à configuração `max_memory` da sessão do
`xe_session_max_event_size`	Especifica o tamanho máximo de memória permitido para configuração `max_event_size` da sessão do evento.
`xe_session_max_dispatch_latency`	Especifica a quantidade de tempo em que os eventos são serem despachados para destinos de sessão de eventos `max_dispatch_latency` da sessão do evento.
`xe_file_target_size`	Especifica o tamanho máximo do destino do arquivo. Ess `size` do destino do arquivo.
`xe_file_retention`	Especifica o tempo de retenção em dias para arquivos ge evento.

> **Note**
>
> Definir `xe_file_retention` como zero remove os arquivos.xel automaticamente depois que o bloqueio nesses arquivos é liberado pelo SQL Server. O bloqueio é liberado sempre que um arquivo.xel atinge o limite de tamanho definido em `xe_file_target_size`.

Você pode usar o procedimento `rdsadmin.dbo.rds_show_configuration` armazenado para mostrar os valores atuais desses parâmetros. Por exemplo, use a instrução SQL a seguir para visualizar a configuração atual do `xe_session_max_memory`.

```
exec rdsadmin.dbo.rds_show_configuration 'xe_session_max_memory'
```

Você pode usar o procedimento `rdsadmin.dbo.rds_set_configuration` armazenado para modificá-los. Por exemplo, use a instrução SQL `xe_session_max_memory` a seguir para definir como 4 MB.

```
exec rdsadmin.dbo.rds_set_configuration 'xe_session_max_memory', 4
```

Considerações para implantações Multi-AZ

Quando você cria uma sessão de evento estendida em uma instância de banco de dados principal, ela não se propaga para a réplica em espera. Você pode fazer failover e criar a sessão de evento estendida na nova instância de banco de dados principal. Ou você pode remover e ler a configuração Multi-AZ para propagar a sessão de evento estendida para a réplica em espera. O RDS interrompe todas as sessões de evento estendidas não padrão na réplica em espera, de modo que essas sessões não consumam recursos no modo de espera. Por isso, depois que uma réplica em espera se tornar a instância de banco de dados principal, inicie manualmente as sessões de eventos estendidas no novo primário.

> **Note**
>
> Essa abordagem se aplica a grupos de disponibilidade sempre ativos e ao espelhamento do banco de dados.

Você também pode usar um trabalho do SQL Server Agent para monitorar a réplica em espera e iniciar as sessões se a espera se tornar a primária. Por exemplo, use a consulta a seguir na etapa de trabalho do SQL Server Agent para reiniciar sessões de evento em uma instância de banco de dados principal.

```
BEGIN
    IF (DATABASEPROPERTYEX('rdsadmin','Updateability')='READ_WRITE'
    AND DATABASEPROPERTYEX('rdsadmin','status')='ONLINE'
    AND (DATABASEPROPERTYEX('rdsadmin','Collation') IS NOT NULL OR
 DATABASEPROPERTYEX('rdsadmin','IsAutoClose')=1)
```

```
    )
    BEGIN
        IF NOT EXISTS (SELECT 1 FROM sys.dm_xe_sessions WHERE name='xe1')
            ALTER EVENT SESSION xe1 ON SERVER STATE=START
        IF NOT EXISTS (SELECT 1 FROM sys.dm_xe_sessions WHERE name='xe2')
            ALTER EVENT SESSION xe2 ON SERVER STATE=START
    END
END
```

Essa consulta reinicia as sessões de evento `xe1` e `xe2` em uma instância de banco de dados principal se essas sessões estiverem em um estado interrompido. Você também pode adicionar uma programação com um intervalo conveniente a essa consulta.

Consultando arquivos de eventos estendidos

Você pode usar o SQL Server Management Studio ou a função `sys.fn_xe_file_target_read_file` para visualizar dados de eventos estendidos que usam destinos de arquivo. Para obter mais informações sobre essa função, consulte sys.fn_xe_file_target_read_file (Transact-SQL) na documentação da Microsoft.

Destinos de arquivos de eventos estendidos só podem gravar arquivos no diretório `D:\rdsdbdata\log` no RDS para SQL Server.

Como exemplo, use a consulta SQL a seguir para listar o conteúdo de todos os arquivos de sessões de eventos estendidos cujos nomes começam com `xe`.

```
SELECT * FROM sys.fn_xe_file_target_read_file('d:\rdsdbdata\log\xe*', null,null,null);
```

Acesso aos backups de logs de transações com o RDS para SQL Server

Com acesso aos backup de logs de transações do RDS para SQL Server, é possível listar os arquivos de log de transações de um banco de dados e copiá-los em um bucket de destino do Amazon S3. Ao copiar backups de logs de transações em um bucket do Amazon S3, você pode usá-los em combinação com backups de banco de dados completos e diferenciais para realizar restaurações do banco de dados a um ponto anterior no tempo. Use procedimentos armazenados no RDS para configurar o acesso aos backups de logs de transações, listar os backups de logs de transações disponíveis e copiá-los em seu bucket do Amazon S3.

O acesso aos backups de logs de transações fornece os seguintes recursos e benefícios:

- Listar e visualizar os metadados dos backups de logs de transações disponíveis de um banco de dados em uma instância de banco de dados do RDS para SQL Server.
- Copiar os backups de logs de transações disponíveis do RDS para SQL Server em um bucket de destino do Amazon S3.
- Executar restaurações a um ponto anterior no tempo de bancos de dados sem a necessidade de restaurar uma instância de banco de dados inteira. Para obter informações sobre como restaurar uma instância de banco de dados, consulte Restaurar uma instância de banco de dados para um momento especificado no Amazon RDS.

Disponibilidade e suporte

O acesso aos backups de logs de transações é compatível em todas as regiões.da AWS. O acesso a backups de logs de transações está disponível para todas as edições e versões do Microsoft SQL Server compatíveis com o Amazon RDS.

Requisitos

Os requisitos a seguir devem ser atendidos antes de ativar o acesso aos backups de logs de transações:

- Os backups automatizados devem ser ativados na instância de banco de dados e a retenção do backup deve ser definida como um valor de um ou mais dias. Para ter mais informações sobre como ativar backups automatizados e configurar uma política de retenção, consulte Ativar backups automáticos.
- Um bucket do Amazon S3 deve existir na mesma conta e região da instância de banco de dados de origem. Antes de ativar o acesso aos backups de logs de transações, selecione um bucket

existente do Amazon S3 ou crie um bucket para usar em seus arquivos de backup de logs de transações.

- Uma política de permissões de bucket do Amazon S3 deve ser configurada da seguinte forma para que o Amazon RDS copie arquivos de log de transações nela:

  1. Defina a propriedade da conta do objeto no bucket como Bucket Owner Preferred (Propriedade do bucket preferencial).
  2. Adicione a política a seguir. Não haverá nenhuma política por padrão, então use as listas de controle de acesso (ACL) do bucket para editar a política do bucket e adicioná-la.

  O exemplo a seguir usa um ARN para especificar um recurso. Recomendamos usar as chaves de contexto de condição global `SourceArn` e `SourceAccount` em relações baseadas em recursos para limitar as permissões do serviço a um recurso específico. Para ter mais informações sobre como trabalhar com ARNs, consulte Amazon resource names (ARNs) (Nomes de recursos da Amazon (ARNs)) e Nomes de recurso da Amazon (ARNs) no Amazon RDS.

  Example de uma política de permissões do Amazon S3 para acesso a backups de logs de transações

  JSON

```
   {
   "Version": "2012-10-17",
   "Statement": [
       {
           "Sid": "Only allow writes to my bucket with bucket owner full
control",
           "Effect": "Allow",
           "Principal": {
               "Service": "backups.rds.amazonaws.com"
           },
           "Action": "s3:PutObject",
           "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/{customer_path}/*",
           "Condition": {
               "StringEquals": {
                   "s3:x-amz-acl": "bucket-owner-full-control",
                   "aws:sourceAccount": "{customer_account}",
                   "aws:sourceArn": "{db_instance_arn}"
               }
```

```
            }
        }
    ]
}
```

- Um AWS Identity and Access Management (perfil do IAM) para acessar o bucket do Amazon S3. Se você já tiver uma função do IAM, você pode usá-la. Você pode optar por criar uma nova função do IAM ao adicionar a opção `SQLSERVER_BACKUP_RESTORE` usando o AWS Management Console. Você também pode criar uma nova manualmente. Para ter mais informações sobre como criar e configurar um perfil do IAM com `SQLSERVER_BACKUP_RESTORE`, consulte Criar manualmente uma função do IAM para backup e restauração nativos.
- A opção `SQLSERVER_BACKUP_RESTORE` deve ser adicionada a um grupo de opções em sua instância de banco de dados. Para ter mais informações sobre a adição da opção `SQLSERVER_BACKUP_RESTORE`, consulte Suporte para backup e restauração nativos no SQL Server.

> **Note**
>
> Se sua instância de banco de dados tiver a criptografia de armazenamento habilitada, será necessário fornecer as ações e a chave do AWS KMS (KMS) no perfil do IAM indicado no grupo nativo de opções de backup e restauração.

Opcionalmente, se você pretende usar o procedimento armazenado `rds_restore_log` para realizar restaurações do banco de dados a um ponto anterior no tempo, recomendamos usar o mesmo caminho do Amazon S3 para o grupo nativo de opções de backup e restauração e acesso aos backups de logs de transações. Esse método garante que, quando o Amazon RDS assume a função do grupo de opções para realizar as funções de log de restauração, ele tenha acesso para recuperar backups de logs de transações do mesmo caminho do Amazon S3.

- Se a instância de banco de dados for criptografada, independentemente do tipo de criptografia (chave gerenciada pela AWS ou chave gerenciada pelo cliente), você deverá fornecer uma chave do KMS gerenciada pelo cliente no perfil do IAM e no procedimento armazenado `rds_tlog_backup_copy_to_S3`.

Limitações e recomendações

O acesso aos backups de logs de transações tem as seguintes limitações e recomendações:

- Você pode listar e copiar até os últimos sete dias de backups de logs de transações para qualquer instância de banco de dados que tenha a retenção de backup configurada entre um a 35 dias.
- Um bucket do Amazon S3 utilizado para acessar backups de logs de transações deve existir na mesma conta e região da instância de banco de dados de origem. Não há suporte à cópia entre contas e regiões.
- Somente um bucket do Amazon S3 pode ser configurado como destino para copiar backups de logs de transações. Você pode selecionar um novo bucket de destino do Amazon S3 com o procedimento armazenado `rds_tlog_copy_setup`. Para ter mais informações sobre como selecionar um novo bucket de destino do Amazon S3, consulte Configurar o acesso aos backups de logs de transações.
- Você não poderá especificar a chave do KMS ao usar o procedimento armazenado `rds_tlog_backup_copy_to_S3` se sua instância do RDS não estiver ativada para criptografia de armazenamento.
- A cópia de várias contas não é compatível. O perfil do IAM utilizado para cópia só permitirá o acesso de gravação aos buckets do Amazon S3 na conta do proprietário da instância de banco de dados.
- Somente duas tarefas simultâneas de qualquer tipo podem ser executadas em uma instância de banco de dados do RDS para SQL Server.
- Somente uma tarefa de cópia pode ser executada em um único banco de dados em um determinado momento. Se você quiser copiar backups de logs de transações para vários bancos de dados na instância de banco de dados, utilize uma tarefa de cópia separada para cada banco de dados.
- Se você copiar um backup de logs de transações que já existe com o mesmo nome no bucket do Amazon S3, o backup de logs de transações existente será substituído.
- Você só pode executar os procedimentos armazenados que são fornecidos com acesso aos backups de logs de transações na instância de banco de dados primária. Não é possível executar esses procedimentos armazenados em uma réplica de leitura do RDS para SQL Server nem em uma instância secundária de um cluster de banco de dados multi-AZ.
- Se a instância de banco de dados do RDS para SQL Server for reinicializada enquanto o procedimento armazenado `rds_tlog_backup_copy_to_S3` estiver em execução, a tarefa será reiniciada automaticamente desde o início, quando a instância de banco de dados estiver on-line novamente. Todos os backups de logs de transações que tenha sido copiado no bucket do Amazon S3 enquanto a tarefa estiver em execução antes da reinicialização serão substituídos.

- Os bancos de dados do sistema Microsoft SQL Server e o banco de dados `RDSAdmin` não podem ser configurados para acesso aos backups de logs de transações.
- A cópia em buckets criptografados pela SSE-KMS não é compatível.

Configurar o acesso aos backups de logs de transações

Para configurar o acesso aos backups de logs de transações, complete a lista de requisitos na seção Requisitos e, depois, execute o procedimento armazenado `rds_tlog_copy_setup`. O procedimento permitirá o acesso ao recurso de backups de logs de transações em nível de instância de banco de dados. Você não precisa executá-lo para cada banco de dados individual na instância de banco de dados.

> **Important**
>
> O usuário do banco de dados deve receber a função `db_owner` no SQL Server em cada banco de dados para configurar e usar o recurso de acesso aos backups de logs de transações.

Example uso:

```
exec msdb.dbo.rds_tlog_copy_setup
@target_s3_arn='arn:aws:s3:::amzn-s3-demo-bucket/myfolder';
```

O seguinte parâmetro é obrigatório:

- `@target_s3_arn`: o ARN do bucket de destino do Amazon S3 no qual copiar arquivos de backup de logs de transações.

Example de definir um bucket de destino do Amazon S3:

```
exec msdb.dbo.rds_tlog_copy_setup @target_s3_arn='arn:aws:s3:::amzn-s3-demo-logging-
bucket/mytestdb1';
```

Para validar a configuração, chame o procedimento armazenado `rds_show_configuration`.

Example de validar a configuração:

```
exec rdsadmin.dbo.rds_show_configuration @name='target_s3_arn_for_tlog_copy';
```

Para modificar o acesso aos backups de logs de transações para apontar para um bucket diferente do Amazon S3, você pode visualizar o valor atual do bucket do Amazon S3 e executar novamente o procedimento armazenado `rds_tlog_copy_setup` utilizando um novo valor para o `@target_s3_arn`.

Example de visualizar o bucket existente do Amazon S3 configurado para acesso aos backups de logs de transações

```
exec rdsadmin.dbo.rds_show_configuration @name='target_s3_arn_for_tlog_copy';
```

Example de atualização para um novo bucket de destino do Amazon S3

```
exec msdb.dbo.rds_tlog_copy_setup @target_s3_arn='arn:aws:s3:::amzn-s3-demo-logging-bucket1/mynewfolder';
```

Listar os backups de logs de transações disponíveis

Com o RDS para SQL Server, os bancos de dados configurados para usar o modelo de recuperação completo e uma retenção de backup de instância de banco de dados definida para um ou mais dias têm os backups de logs de transações ativados automaticamente. Ao permitir o acesso aos backups de logs de transações, eles ficam disponíveis por até sete dias para você copiar em seu bucket do Amazon S3.

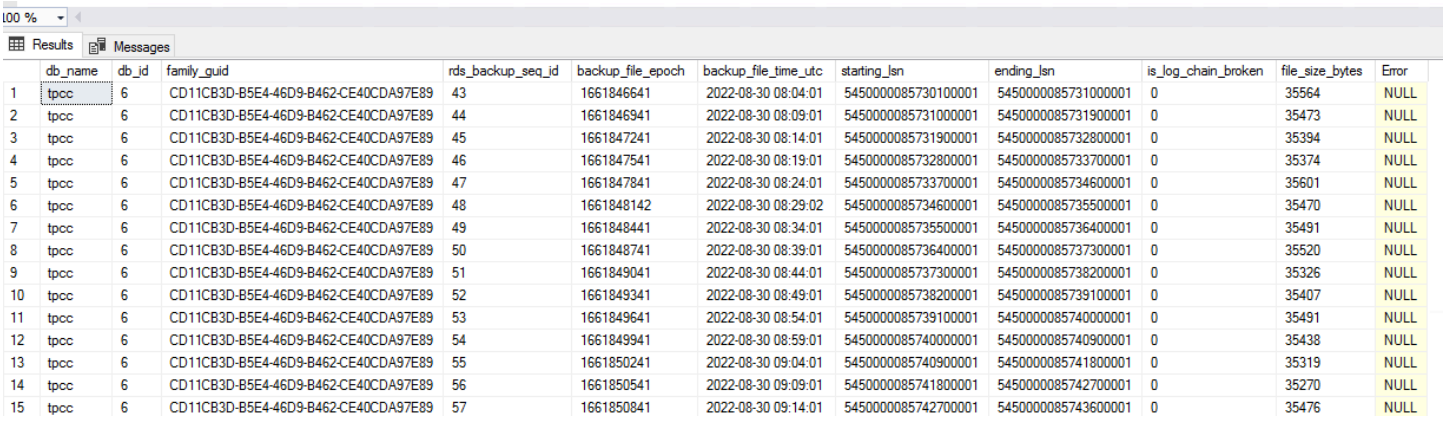
Depois de ativar o acesso aos backups de logs de transações, você pode começar a usá-lo para listar e copiar os arquivos de backup de logs de transações disponíveis.

Listar backups de logs de transações

Para listar todos os backups de logs de transações disponíveis para um banco de dados individual, chame a função `rds_fn_list_tlog_backup_metadata`. Você pode utilizar uma cláusula `ORDER BY` ou `WHERE` ao chamar a função.

Example de listar e filtrar os arquivos de backup de logs de transações disponíveis

```
SELECT * from msdb.dbo.rds_fn_list_tlog_backup_metadata('mydatabasename');
SELECT * from msdb.dbo.rds_fn_list_tlog_backup_metadata('mydatabasename') WHERE
 rds_backup_seq_id = 3507;
SELECT * from msdb.dbo.rds_fn_list_tlog_backup_metadata('mydatabasename') WHERE
 backup_file_time_utc > '2022-09-15 20:44:01' ORDER BY backup_file_time_utc DESC;
```



100 % Results Messages

	db_name	db_id	family_guid	rds_backup_seq_id	backup_file_epoch	backup_file_time_utc	starting_lsn	ending_lsn	is_log_chain_broken	file_size_bytes	Error
1	tpcc	6	CD11CB3D-B5E4-46D9-B462-CE40CDA97E89	43	1661846641	2022-08-30 08:04:01	5450000085730100001	5450000085731000001	0	35564	NULL
2	tpcc	6	CD11CB3D-B5E4-46D9-B462-CE40CDA97E89	44	1661846941	2022-08-30 08:09:01	5450000085731000001	5450000085731900001	0	35473	NULL
3	tpcc	6	CD11CB3D-B5E4-46D9-B462-CE40CDA97E89	45	1661847241	2022-08-30 08:14:01	5450000085731900001	5450000085732800001	0	35394	NULL
4	tpcc	6	CD11CB3D-B5E4-46D9-B462-CE40CDA97E89	46	1661847541	2022-08-30 08:19:01	5450000085732800001	5450000085733700001	0	35374	NULL
5	tpcc	6	CD11CB3D-B5E4-46D9-B462-CE40CDA97E89	47	1661847841	2022-08-30 08:24:01	5450000085733700001	5450000085734600001	0	35601	NULL
6	tpcc	6	CD11CB3D-B5E4-46D9-B462-CE40CDA97E89	48	1661848142	2022-08-30 08:29:02	5450000085734600001	5450000085735500001	0	35470	NULL
7	tpcc	6	CD11CB3D-B5E4-46D9-B462-CE40CDA97E89	49	1661848441	2022-08-30 08:34:01	5450000085735500001	5450000085736400001	0	35491	NULL
8	tpcc	6	CD11CB3D-B5E4-46D9-B462-CE40CDA97E89	50	1661848741	2022-08-30 08:39:01	5450000085736400001	5450000085737300001	0	35520	NULL
9	tpcc	6	CD11CB3D-B5E4-46D9-B462-CE40CDA97E89	51	1661849041	2022-08-30 08:44:01	5450000085737300001	5450000085738200001	0	35326	NULL
10	tpcc	6	CD11CB3D-B5E4-46D9-B462-CE40CDA97E89	52	1661849341	2022-08-30 08:49:01	5450000085738200001	5450000085739100001	0	35407	NULL
11	tpcc	6	CD11CB3D-B5E4-46D9-B462-CE40CDA97E89	53	1661849641	2022-08-30 08:54:01	5450000085739100001	5450000085740000001	0	35491	NULL
12	tpcc	6	CD11CB3D-B5E4-46D9-B462-CE40CDA97E89	54	1661849941	2022-08-30 08:59:01	5450000085740000001	5450000085740900001	0	35438	NULL
13	tpcc	6	CD11CB3D-B5E4-46D9-B462-CE40CDA97E89	55	1661850241	2022-08-30 09:04:01	5450000085740900001	5450000085741800001	0	35319	NULL
14	tpcc	6	CD11CB3D-B5E4-46D9-B462-CE40CDA97E89	56	1661850541	2022-08-30 09:09:01	5450000085741800001	5450000085742700001	0	35270	NULL
15	tpcc	6	CD11CB3D-B5E4-46D9-B462-CE40CDA97E89	57	1661850841	2022-08-30 09:14:01	5450000085742700001	5450000085743600001	0	35476	NULL

A função `rds_fn_list_tlog_backup_metadata` retorna o seguinte resultado:

Nome da coluna	Tipo de dados	Descrição
`db_name`	sysname	O nome do banco de dados fornecido para listar os backups de logs de transações.
`db_id`	int	O identificador interno do banco de dados para o parâmetro de entrada `db_name`.
`family_guid`	uniqueidentifier	O ID exclusivo do banco de dados original na criação. Esse valor permanece o mesmo quando o banco de dados é restaurado, mesmo com um nome de banco de dados diferente.
`rds_backup_seq_id`	int	O ID que o RDS utiliza internamente para manter um número de sequência de cada arquivo de backup de logs de transações.

Nome da coluna	Tipo de dados	Descrição
backup_file_epoch	bigint	A época em que um arquivo de backup de transações foi gerado.
backup_file_time_utc	datetime	O valor UTC convertido em tempo para o valor backup_file_epoch .
starting_lsn	numérico (25,0)	O número de sequência de logs do primeiro registro de log ou do mais antigo de um arquivo de backup de logs de transações.
ending_lsn	numérico (25,0)	O número de sequência de logs do último ou do próximo registro de log de um arquivo de backup de logs de transações.
is_log_chain_broken	bit	Um valor booliano que indica se a cadeia de logs está rompida entre o arquivo de backup de logs de transações atual e o anterior.
file_size_bytes	bigint	O tamanho do conjunto de backup transacional em bytes.
Error	varchar(4000)	Mensagem de erro se a função rds_fn_list_tlog_backup_metadata gerar uma exceção. NULL se não houver exceções.

Copiar backups de logs de transações

Para copiar um conjunto de backups de logs de transações disponíveis para um banco de dados individual em seu bucket do Amazon S3, chame o procedimento armazenado rds_tlog_backup_copy_to_S3. O procedimento armazenado rds_tlog_backup_copy_to_S3 iniciará uma nova tarefa para copiar os backups de logs de transações.

> **Note**
>
> O procedimento armazenado rds_tlog_backup_copy_to_S3 copiará os backups de logs de transações sem validação em relação ao atributo is_log_chain_broken. Por

esse motivo, você deve confirmar manualmente uma cadeia de logs ininterrupta antes de executar o procedimento armazenado `rds_tlog_backup_copy_to_S3`. Para obter mais explicações, consulte Validar a cadeia de logs de backup de logs de transações.

Example uso do procedimento armazenado **`rds_tlog_backup_copy_to_S3`**

```
exec msdb.dbo.rds_tlog_backup_copy_to_S3
 @db_name='mydatabasename',
 [@kms_key_arn='arn:aws:kms:region:account-id:key/key-id'],
 [@backup_file_start_time='2022-09-01 01:00:15'],
 [@backup_file_end_time='2022-09-01 21:30:45'],
 [@starting_lsn=149000000112100001],
 [@ending_lsn=149000000120400001],
 [@rds_backup_starting_seq_id=5],
 [@rds_backup_ending_seq_id=10];
```

Os seguintes parâmetros de entrada estão disponíveis:

Parâmetro	Descrição
`@db_name`	O nome do banco de dados do qual copiar backups de logs de transações
`@kms_key_arn`	Uma chave do KMS gerenciada pelo cliente. Se você criptografar a instância de banco de dados com uma chave do KMS gerenciad a pela AWS, será necessário criar uma chave gerenciada pelo cliente. Se criptografar a instância de banco de dados com uma chave gerenciada pelo cliente, você poderá usar o mesmo ARN da chave do KMS.
`@backup_file_start _time`	O carimbo de data e hora UTC conforme fornecido na coluna `[backup_file_time_utc]` da função `rds_fn_li st_tlog_backup_metadata` .
`@backup_file_end_t ime`	O carimbo de data e hora UTC conforme fornecido na coluna `[backup_file_time_utc]` da função `rds_fn_li st_tlog_backup_metadata` .

Parâmetro	Descrição
`@starting_lsn`	O número de sequência de logs (LSN) conforme fornecido na coluna `[starting_lsn]` da função `rds_fn_list_tlog_backup_metadata`
`@ending_lsn`	O número de sequência de logs (LSN) conforme fornecido na coluna `[ending_lsn]` da função `rds_fn_list_tlog_backup_metadata` .
`@rds_backup_starting_seq_id`	O ID de sequência conforme fornecido na coluna `[rds_backup_seq_id]` da função `rds_fn_list_tlog_backup_metadata` .
`@rds_backup_ending_seq_id`	O ID de sequência conforme fornecido na coluna `[rds_backup_seq_id]` da função `rds_fn_list_tlog_backup_metadata` .

Você pode especificar um conjunto de parâmetros de hora, LSN ou ID de sequência. Somente um conjunto de parâmetros é necessário.

Você também pode especificar apenas um único parâmetro em qualquer um dos conjuntos. Por exemplo, ao fornecer um valor somente para o parâmetro `backup_file_end_time`, todos os arquivos de backup de logs de transações disponíveis antes desse período dentro do limite de sete dias serão copiados em seu bucket do Amazon S3.

Veja a seguir as combinações válidas de parâmetros de entrada para o procedimento armazenado `rds_tlog_backup_copy_to_S3`.

Parâmetros fornecidos	Resultado esperado	
`exec msdb.dbo.rds_tlog_backup_copy_to_S3 @db_name = 'testdb1',`	Copia os backups de logs de transações dos últimos sete dias e existentes entre o	

Parâmetros fornecidos	Resultado esperado	
`@backup_file_start_time='2022-08-23 00:00:00', @backup_file_end_time='2022-08-30 00:00:00';`	intervalo fornecido de `backup_file_start_time` e `backup_file_end_time` . Neste exemplo, o procedimento armazenado copiará os backups de logs de transações que foram gerados entre “2022-08-23 00:00:00” e “2022-08-30 00:00:00”.	

Parâmetros fornecidos	Resultado esperado	
`exec msdb.dbo.rds_tlog_backup_copy_to_S3 @db_name = 'testdb1', @backup_file_start_time='2022-08-23 00:00:00';`	Copia os backups de logs de transações dos últimos sete dias e a partir do `backup_file_start_time` fornecido. Neste exemplo, o procedimento armazenado copiará os backups de logs de transações de “2022-08-23 00:00:00” até o último backup de logs de transações.	

Parâmetros fornecidos	Resultado esperado	
`exec msdb.dbo.rds_tlog_backup_copy_to_S3 @db_name = 'testdb1', @backup_file_end_time='2022-08-30 00:00:00';`	Copia os backups de logs de transações dos últimos sete dias até o `backup_file_end_time` fornecido. Neste exemplo, o procedimento armazenado copiará os backups de logs de transações que foram gerados de “2022-08-23 00:00:00” até “2022-08-30 00:00:00”.	

Parâmetros fornecidos	Resultado esperado	
`exec msdb.dbo.rds_tlog_backup_copy_to_S3 @db_name='testdb1', @starting_lsn =14900000 00040007, @ending_lsn = 149000000 0050009;`	Copia os backups de logs de transações disponíveis dos últimos sete dias e entre o intervalo fornecido do `starting_lsn` e `ending_lsn` . Neste exemplo, o procedimento armazenado copiará backups de logs de transaçõe s dos últimos sete dias com um intervalo de LSN entre 1490000000040007 e 149000000 0050009.	

Parâmetros fornecidos	Resultado esperado	
`exec msdb.dbo.rds_tlog_backup_copy_to_S3 @db_name='testdb1', @starting_lsn =14900000 00040007;`	Copia os backups de logs de transações disponíveis dos últimos sete dias, a partir do `starting_lsn` fornecido. Neste exemplo, o procedimento armazenado copiará os backups de logs de transações do LSN 149000000 0040007 até o último backup de logs de transações.	

Parâmetros fornecidos	Resultado esperado	
`exec msdb.dbo.rds_tlog_backup_copy_to_S3 @db_name='testdb1', @ending_lsn=14900000 00050009;`	Copia os backups de logs de transações disponíveis dos últimos sete dias, até o `ending_lsn` fornecido. Neste exemplo, o procedimento armazenado copiará backups de logs de transações a partir dos últimos sete dias até o lsn 149000000 0050009.	

Parâmetros fornecidos	Resultado esperado	
`exec msdb.dbo.rds_tlog_backup_copy_to_S3 @db_name='testdb1', @rds_backup_starting_seq_id= 2000, @rds_backup_ending_seq_id= 5000;`	Copia os backups de logs de transações disponíveis dos últimos sete dias e existentes entre o intervalo fornecido de `rds_backup_starting_seq_id` e `rds_backup_ending_seq_id` . Neste exemplo, o procedimento armazenado copiará os backups de logs de transações dos últimos sete dias e dentro do intervalo de ID da sequência de backup do rds fornecido, começando de seq_id 2000 até seq_id 5000.	

Parâmetros fornecidos	Resultado esperado	
`exec msdb.dbo.rds_tlog_backup_copy_to_S3 @db_name='testdb1', @rds_backup_starting_seq_id= 2000;`	Copia os backups de logs de transações disponíveis dos últimos sete dias, a partir do `rds_backup_starting_seq_id` fornecido. Neste exemplo, o procedimento armazenado copiará os backups de logs de transações de seq_id 2000 até o último backup de logs de transações.	

Parâmetros fornecidos	Resultado esperado	
`exec msdb.dbo.rds_tlog_backup_copy_to_S3 @db_name='testdb1', @rds_backup_ending_seq_id= 5000;`	Copia os backups de logs de transações disponíveis dos últimos sete dias, até o `rds_backup_ending_seq_id` fornecido. Neste exemplo, o procedimento armazenado copiará backups de logs de transações a partir dos últimos sete dias, até seq_id 5000.	

Parâmetros fornecidos	Resultado esperado	
`exec msdb.dbo.rds_tlog_backup_copy_to_S3 @db_name='testdb1', @rds_backup_starting_seq_id= 2000; @rds_backup_ending_seq_id= 2000;`	Copia um único backup de logs de transações com o `rds_backup_starting_seq_id` fornecido, se disponível nos últimos sete dias. Neste exemplo, o procedimento armazenado copiará um único backup de logs de transações que tenha um seq_id de 2000, se houver nos últimos sete dias.	

Validar a cadeia de logs de backup de logs de transações

Os bancos de dados configurados para acesso aos backups de logs de transações devem ter a retenção automatizada de backup ativada. A retenção automatizada de backup define os bancos de dados na instância de banco de dados de acordo com o modelo de recuperação FULL. Para ser compatível com a restauração a um ponto anterior no tempo de um banco de dados, evite alterar o modelo de recuperação do banco de dados, o que pode ocasionar uma cadeia de registros interrompida. Recomendamos manter o banco de dados definido como o modelo de recuperação FULL.

Para validar manualmente a cadeia de logs antes de copiar os backups de logs de transações, chame a função `rds_fn_list_tlog_backup_metadata` e revise os valores na coluna `is_log_chain_broken`. Um valor de “1" indica que a cadeia de logs foi interrompida entre o backup de logs atual e o backup de logs anterior.

O exemplo a seguir mostra uma cadeia de logs interrompida na saída do procedimento armazenado `rds_fn_list_tlog_backup_metadata`.

rds_sequence_id	first_lsn	last_lsn	is_log_chain_broken
43	90023	90457	0
44	90457	90985	0
45	90987	92034	1

Em uma cadeia de logs normal, o valor do número de sequência de log (LSN) para first_lsn para determinado rds_sequence_id deve corresponder ao valor de last_lsn no rds_sequence_id anterior. Na imagem, o rds_sequence_id de 45 tem um valor first_lsn 90987, que não corresponde ao valor last_lsn de 90985 do rds_sequence_id 44 anterior.

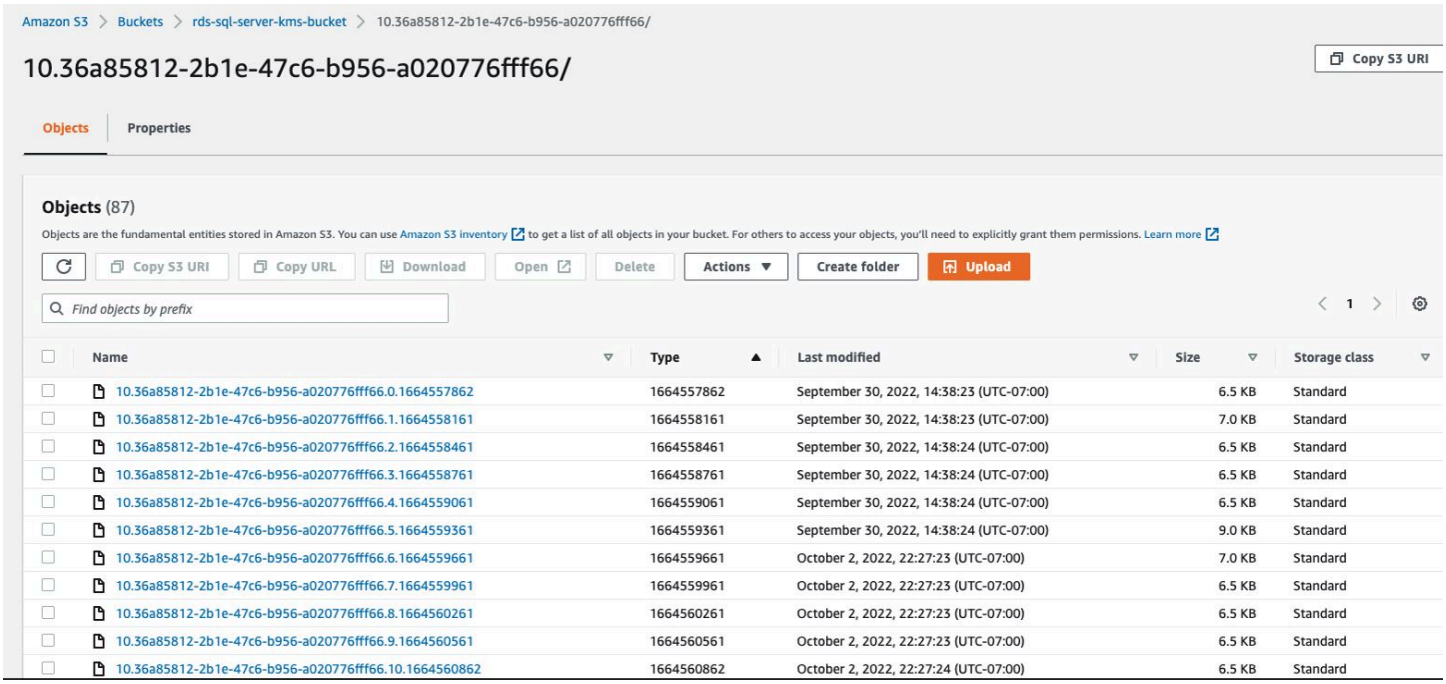
Para ter mais informações sobre a arquitetura de logs de transações e os números de sequência de log do SQL Server, consulte Transaction Log Logical Architecture (Arquitetura lógica do log de transações) na documentação do Microsoft SQL Server.

Estrutura de arquivos e pastas do bucket do Amazon S3

Os backups de logs de transações têm a seguinte estrutura padrão e convenção de nomenclatura em um bucket do Amazon S3:

- Uma nova pasta é criada no caminho `target_s3_arn` de cada banco de dados com a estrutura de nomenclatura `{db_id}.{family_guid}`.
- Dentro da pasta, os backups de logs de transações têm uma estrutura de nome de arquivo `{db_id}.{family_guid}.{rds_backup_seq_id}.{backup_file_epoch}`.
- Você pode ver os detalhes de `family_guid,db_id,rds_backup_seq_id and backup_file_epoch` da função `rds_fn_list_tlog_backup_metadata`.

O exemplo a seguir mostra a estrutura de pastas e arquivos de um conjunto de backups de logs de transações em um bucket do Amazon S3.



Acompanhar o status de tarefas

Para acompanhar o status de suas tarefas de cópia, chame o procedimento armazenado `rds_task_status`. Se você não fornecer nenhum parâmetro, o procedimento armazenado retornará o status de todas as tarefas.

Example uso:

```
exec msdb.dbo.rds_task_status
  @db_name='database_name',
  @task_id=ID_number;
```

Os seguintes parâmetros são opcionais:

- `@db_name` – o nome do banco de dados do qual o status será exibido.
- `@task_id` – o nome do ID da tarefa do qual o status será exibido.

Example de listagem do status de um ID de tarefa específico:

```
exec msdb.dbo.rds_task_status @task_id=5;
```

Example de listagem do status de uma tarefa e banco de dados específicos:

```
exec msdb.dbo.rds_task_status@db_name='my_database',@task_id=5;
```

Example de listagem de todas as tarefas e seus status de um banco de dados específico:

```
exec msdb.dbo.rds_task_status @db_name='my_database';
```

Example de listagem de todas as tarefas e seus status na instância de banco de dados atual

```
exec msdb.dbo.rds_task_status;
```

Cancelar uma tarefa

Para cancelar uma tarefa em execução, chame o procedimento armazenado `rds_cancel_task`.

Example uso:

```
exec msdb.dbo.rds_cancel_task @task_id=ID_number;
```

O seguinte parâmetro é obrigatório:

- `@task_id` – o ID da tarefa a ser cancelada. Você pode visualizar o ID da tarefa chamando o procedimento armazenado `rds_task_status`.

Para ter mais informações sobre como visualizar e cancelar tarefas em execução, consulte Importar e exportar bancos de dados do SQL Server usando backup e restauração nativos.

Solução de problemas de acesso aos backups de logs de transações

Os problemas a seguir podem ocorrer ao usar os procedimentos armazenados para acesso aos backups de logs de transações.

Procedimento armazenado	Mensagem de erro	Problema	Sugestões de solução de problemas
rds_tlog_copy_setup	Os backups estão desativados nessa instância de banco de dados. Ative backups de instâncias de banco de dados com uma retenção de pelo menos “1" e tente novamente.	Backups automatizados não estão ativados para a instância de banco de dados.	A retenção de backup da instância de banco de dados deve ser ativada com uma retenção de pelo menos um dia. Para ter mais informações sobre como ativar backups automatizados e configurar a retenção de backups, consulte Backup retention period (Período de retenção de backup).
rds_tlog_copy_setup	Erro ao executar o procedimento armazenado rds_tlog_copy_setup. Reconecte-se ao endpoint do RDS e tente novamente.	Ocorreu um erro interno.	Reconecte-se ao endpoint do RDS e execute o procedimento armazenado `rds_tlog_copy_setup` novamente.

Procedimento armazenado	Mensagem de erro	Problema	Sugestões de solução de problemas
rds_tlog_copy_setup	A execução do procedimento armazenado rds_tlog_backup_copy_setup em uma transação não é compatível. Verifique se a sessão não tem transações abertas e tente novamente.	Houve uma tentativa de realizar o procedimento armazenado em uma transação com o uso de `BEGIN` e `END`.	Evite utilizar `BEGIN` e `END` ao executar o procedimento armazenado `rds_tlog_copy_setup` .

Procedimento armazenado	Mensagem de erro	Problema	Sugestões de solução de problemas
rds_tlog_copy_setup	O nome do bucket do S3 para o parâmetro de entrada@targ 3_arn deve conter pelo menos um caractere diferente de um espaço.	Um valor incorreto foi fornecido para o parâmetro de entrada `@target_s3_arn` .	O parâmetro de entrada `@target_s3_arn` deve especificar o ARN completo do bucket do Amazon S3.
rds_tlog_copy_setup	A opção `SQLSERVER_BACKUP_RESTORE` não está ativada ou está em processo de ativação. Ative a opção ou tente novamente mais tarde.	A opção `SQLSERVER_BACKUP_RESTORE` não está ativada na instância de banco de dados ou estava apenas ativada e com a ativação interna pendente.	Ative a opção `SQLSERVER_BACKUP_RESTORE` conforme especificado na seção Requirements (Requisitos). Aguarde alguns minutos e execute o procedimento armazenado `rds_tlog_copy_setup` novamente.

Procedimento armazenado	Mensagem de erro	Problema	Sugestões de solução de problemas
rds_tlog_copy_setup	O arn de destino do S3 para o parâmetro de entrada `@target_s3_arn` não pode estar em branco nem ser nulo.	Um valor NULL foi fornecido para o parâmetro de entrada `@target_s3_arn` ou o valor não foi fornecido.	O parâmetro de entrada `@target_s3_arn` deve especificar o ARN completo do bucket do Amazon S3.
rds_tlog_copy_setup	O arn de destino do S3 para o parâmetro de entrada `@target_s3_arn` deve começar com arn:aws.	O parâmetro de entrada `@target_s3_arn` foi fornecido sem `arn:aws` na frente.	O parâmetro de entrada `@target_s3_arn` deve especificar o ARN completo do bucket do Amazon S3.
rds_tlog_copy_setup	O ARN de destino do S3 já está definido com o valor fornecido.	O procedimento armazenado `rds_tlog_copy_setup` foi executado anteriormente e configurado com um ARN do bucket do Amazon S3.	Para modificar o valor do bucket do Amazon S3 para acesso aos backups de logs de transações, forneça um `target S3 ARN` diferente.

Procedimento armazenado	Mensagem de erro	Problema	Sugestões de solução de problemas
rds_tlog_copy_setup	Não é possível gerar credenciais para ativar o acesso aos backups de logs de transações. Confirme o ARN do caminho do S3 fornecido com `rds_tlog_copy_setup` e tente novamente mais tarde.	Houve um erro não especificado ao gerar credenciais para permitir o acesso aos backups de logs de transações.	Revise a configuração e tente novamente.

Procedimento armazenado	Mensagem de erro	Problema	Sugestões de solução de problemas
rds_tlog_copy_setup	Você não pode executar o procedimento armazenado rds_tlog_copy_setup enquanto houver tarefas pendentes. Aguarde até que as tarefas pendentes sejam concluídas e tente novamente.	Somente duas tarefas podem ser executadas por vez. Há tarefas pendentes aguardando conclusão.	Visualize as tarefas pendentes e aguarde a conclusão delas. Para ter mais informações sobre monitoramento do status da tarefa, consulte Acompanhar o status de tarefas.

Procedimento armazenado	Mensagem de erro	Problema	Sugestões de solução de problemas
rds_tlog_backup_copy_to_S3	Uma tarefa de cópia de arquivos de backup T-log já foi emitida para o banco de dados: %s com ID da tarefa: %d, tente novamente mais tarde.	Somente uma tarefa de cópia pode ser executada em um único banco de dados em um determinado momento. Há uma tarefa de cópia pendente aguardando conclusão.	Visualize as tarefas pendentes e aguarde a conclusão delas. Para ter mais informações sobre monitoramento do status da tarefa, consulte Acompanhar o status de tarefas.

Procedimento armazenado	Mensagem de erro	Problema	Sugestões de solução de problemas
rds_tlog_backup_copy_to_S3	Pelo menos um destes três conjuntos de parâmetros deve ser fornecido. SET-1:(@backup_file_start_time, @backup_file_end_time) | SET-2:(@starting_lsn, @ending_lsn) | SET-3:(@rds_backup_starting_seq_id, @rds_backup_ending_seq_id)	Nenhum dos três conjuntos de parâmetros foi fornecido, ou um conjunto de parâmetros fornecido não tem um parâmetro obrigatório.	Você pode especificar parâmetros de hora, lsn ou ID de sequência. É necessário um destes três conjuntos de parâmetros. Para ter mais informações sobre os parâmetros obrigatórios, consulte Copiar backups de logs de transações.

Procedimento armazenado	Mensagem de erro	Problema	Sugestões de solução de problemas
rds_tlog_backup_copy_to_S3	Os backups estão desativados em sua instância. Ative os backups e tente novamente em algum momento.	Backups automatizados não estão ativados para a instância de banco de dados.	Para ter mais informações sobre como ativar backups automatizados e configurar a retenção de backups, consulte Backup retention period (Período de retenção de backup).
rds_tlog_backup_copy_to_S3	Não é possível encontrar o banco de dados %s fornecido.	O valor fornecido para o parâmetro de entrada `@db_name` não corresponde ao nome do banco de dados na instância de banco de dados.	Utilize o nome de banco de dados correto. Para listar todos os bancos de dados por nome, execute `SELECT * from sys.databases`

Procedimento armazenado	Mensagem de erro	Problema	Sugestões de solução de problemas
rds_tlog_backup_copy_to_S3	Não é possível executar o procedimento armazenado rds_tlog_backup_copy_to_S3 para bancos de dados do sistema do SQL Server nem para o banco de dados rdsadmin.	O valor fornecido para o parâmetro de entrada `@db_name` corresponde ao nome do banco de dados do sistema do SQL Server ou ao banco de dados RDSAdmin.	Os seguintes bancos de dados não podem ser usados com acesso aos backups de logs de transações: `master, model, msdb, tempdb, RDSAdmin.`
rds_tlog_backup_copy_to_S3	O nome do banco de dados do parâmetro de entrada @db_name não pode estar em branco nem ser nulo.	Um valor fornecido para o parâmetro de entrada `@db_name` estava em branco ou `NULL`.	Utilize o nome de banco de dados correto. Para listar todos os bancos de dados por nome, execute `SELECT * from sys.databases`

Procedimento armazenado	Mensagem de erro	Problema	Sugestões de solução de problemas
rds_tlog_backup_copy_to_S3	O período de retenção do backup da instância de banco de dados deve ser definido como pelo menos 1 para executar o procedimento armazenado rds_tlog_backup_copy_setup.	Backups automatizados não estão ativados para a instância de banco de dados.	Para ter mais informações sobre como ativar backups automatizados e configurar a retenção de backups, consulte Backup retention period (Período de retenção de backup).

Procedimento armazenado	Mensagem de erro	Problema	Sugestões de solução de problemas
rds_tlog_backup_copy_to_S3	Erro ao executar o procedimento armazenado rds_tlog_backup_copy_to_S3. Reconecte-se ao endpoint do RDS e tente novamente.	Ocorreu um erro interno.	Reconecte-se ao endpoint do RDS e execute o procedimento armazenado `rds_tlog_backup_copy_to_S3` novamente.

Procedimento armazenado	Mensagem de erro	Problema	Sugestões de solução de problemas
rds_tlog_backup_copy_to_S3	Somente um destes três conjuntos de parâmetros deve ser fornecido. SET-1:(@backup_file_start_time, @backup_file_end_time) | SET-2:(@starting_lsn, @ending_lsn) | SET-3:(@rds_backup_starting_seq_id, @rds_backup_ending_seq_id)	Vários conjuntos de parâmetros foram fornecidos.	Você pode especificar parâmetros de hora, lsn ou ID de sequência. É necessário um destes três conjuntos de parâmetros. Para ter mais informações sobre os parâmetros obrigatórios, consulte Copiar backups de logs de transações.

Procedimento armazenado	Mensagem de erro	Problema	Sugestões de solução de problemas
rds_tlog_backup_copy_to_S3	A execução do procedimento armazenado rds_tlog_backup_copy_to_S3 em uma transação não é compatível. Verifique se a sessão não tem transações abertas e tente novamente.	Houve uma tentativa de realizar o procedimento armazenado em uma transação com o uso de `BEGIN` e `END`.	Evite utilizar `BEGIN` e `END` ao executar o procedimento armazenado `rds_tlog_backup_copy_to_S3` .

Procedimento armazenado	Mensagem de erro	Problema	Sugestões de solução de problemas
rds_tlog_backup_copy_to_S3	Os parâmetros fornecidos estão fora do período de retenção de logs de backup de transações. Para listar os arquivos de backup de logs de transações disponíveis, execute a função rds_fn_list_tlog_backup_metadata.	Não há backups de logs transacionais disponíveis para os parâmetros de entrada fornecidos que se encaixem na janela de retenção de cópias.	Tente novamente com um conjunto válido de parâmetros. Para ter mais informações sobre os parâmetros obrigatórios, consulte Copiar backups de logs de transações.

Procedimento armazenado	Mensagem de erro	Problema	Sugestões de solução de problemas
rds_tlog_backup_copy_to_S3	Houve um erro de permissão no processamento da solicitação. O bucket deve estar na mesma conta e região da instância de banco de dados e confirme as permissões da política de bucket do S3 em relação ao modelo na documentação pública.	Foi detectado um problema com o bucket do S3 fornecido ou com suas permissões de política.	Confirme se sua configuração de acesso aos backups de logs de transações está correta. Para ter mais informações sobre os requisitos de configuração para seu bucket do S3, consulte Requisitos.

Procedimento armazenado	Mensagem de erro	Problema	Sugestões de solução de problemas
rds_tlog_backup_copy_to_S3	Não é permitido executar o procedimento armazenado `rds_tlog_backup_copy_to_S3` em uma instância de réplica de leitura do RDS.	Houve uma tentativa de realizar o procedimento armazenado em uma instância de réplica de leitura do RDS.	Conecte-se à instância de banco de dados primária do RDS para executar o procedimento armazenado `rds_tlog_backup_copy_to_S3` .
rds_tlog_backup_copy_to_S3	O LSN para o parâmetro de entrada `@starting_lsn` deve ser menor que `@ending_lsn` .	O valor fornecido para o parâmetro de entrada `@starting_lsn` foi maior do que o valor fornecido para o parâmetro de entrada `@ending_lsn` .	O valor fornecido para o parâmetro de entrada `@starting_lsn` foi menor do que o valor fornecido para o parâmetro de entrada `@ending_lsn` .

Procedimento armazenado	Mensagem de erro	Problema	Sugestões de solução de problemas
rds_tlog_backup_copy_to_S3	O procedimento armazenado `rds_tlog_backup_copy_to_S3` só pode ser executado pelos membros da função `db_owner` no banco de dados de origem.	A função `db_owner` não foi concedida para a conta que está tentando executar o procedimento armazenado `rds_tlog_backup_copy_to_S3` no `db_name` fornecido.	A conta que executa o procedimento armazenado deve ter permissão com a função `db_owner` para o `db_name` fornecido.

Procedimento armazenado	Mensagem de erro	Problema	Sugestões de solução de problemas
rds_tlog_backup_copy_to_S3	O ID de sequência do parâmetro de entrada `@rds_backup_starting_seq_id` deve ser menor ou igual a `@rds_backup_ending_seq_id` .	O valor fornecido para o parâmetro de entrada `@rds_backup_starting_seq_id` foi maior do que o valor fornecido para o parâmetro de entrada `@rds_backup_ending_seq_id` .	O valor fornecido para o parâmetro de entrada `@rds_backup_starting_seq_id` foi menor do que o valor fornecido para o parâmetro de entrada `@rds_backup_ending_seq_id` .
rds_tlog_backup_copy_to_S3	A opção SQLSERVER_BACKUP_RESTORE não está ativada ou está em processo de ativação. Ative a opção ou tente novamente mais tarde.	A opção `SQLSERVER_BACKUP_RESTORE` não está ativada na instância de banco de dados ou estava apenas ativada e com a ativação interna pendente.	Ative a opção `SQLSERVER_BACKUP_RESTORE` conforme especificado na seção Requirements (Requisitos). Aguarde alguns minutos e execute o procedimento armazenado `rds_tlog_backup_copy_to_S3` novamente.

Procedimento armazenado	Mensagem de erro	Problema	Sugestões de solução de problemas
rds_tlog_backup_copy_to_S3	A hora de início do parâmetro de entrada `@backup_file_start_time` deve ser menor do que `@backup_file_end_time` .	O valor fornecido para o parâmetro de entrada `@backup_file_start_time` foi maior do que o valor fornecido para o parâmetro de entrada `@backup_file_end_time` .	O valor fornecido para o parâmetro de entrada `@backup_file_start_time` foi menor do que o valor fornecido para o parâmetro de entrada `@backup_file_end_time` .
rds_tlog_backup_copy_to_S3	Não conseguimos processar a solicitação devido à falta de acesso. Confira sua configuração e permissões para o recurso.	Pode haver um problema com as permissões do bucket do Amazon S3, ou o bucket do Amazon S3 fornecido está em outra conta ou região.	As permissões da política de bucket do Amazon S3 devem ter permissão para permitir o acesso ao RDS. Um bucket do Amazon S3 deve existir na mesma conta e região da instância de banco de dados.

Procedimento armazenado	Mensagem de erro	Problema	Sugestões de solução de problemas
rds_tlog_backup_copy_to_S3	Você não pode fornecer um ARN de chave do KMS como parâmetro de entrada para o procedimento armazenado para instâncias que não são criptografadas para armazenamento.	Quando a criptografia de armazenamento não está ativada na instância de banco de dados, o parâmetro de entrada `@kms_key_arn` não deve ser fornecido.	Não forneça um parâmetro de entrada para `@kms_key_arn` .

Procedimento armazenado	Mensagem de erro	Problema	Sugestões de solução de problemas
rds_tlog_backup_copy_to_S3	Você deve fornecer um ARN de chave do KMS como parâmetro de entrada para o procedimento armazenado para instâncias criptografadas para armazenamento.	Quando a criptografia de armazenamento está ativada na instância de banco de dados, o parâmetro de entrada `@kms_key_arn` deve ser fornecido.	Forneça um parâmetro de entrada para `@kms_key_arn` com um valor que corresponda ao ARN do bucket do Amazon S3 a ser usado para backups de logs de transações.

Procedimento armazenado	Mensagem de erro	Problema	Sugestões de solução de problemas
rds_tlog_backup_copy_to_S3	Você deve executar o procedimento armazenado `rds_tlog_copy_setup` e definir o `@target_s3_arn` antes de executar o procedimento armazenado `rds_tlog_backup_copy_to_S3` .	O procedimento de configuração do acesso aos backups de logs de transações não foi concluído antes da tentativa de executar o procedimento armazenado `rds_tlog_backup_copy_to_S3` .	Execute o procedimento armazenado `rds_tlog_copy_setup` antes de executar o procedimento armazenado `rds_tlog_backup_copy_to_S3` . Para ter mais informações sobre como executar o procedimento de configuração para acesso aos backups de logs de transações, consulte Configurar o acesso aos backups de logs de transações.

Opções para o mecanismo de banco de dados do Microsoft SQL Server

Nesta seção, você pode encontrar descrições de opções que estão disponíveis para instâncias do Amazon RDS que executam o mecanismo do banco de dados do Microsoft SQL Server. Para habilitar essas opções, adicione-as a um grupo de opções e, em seguida, associe o grupo de opções à sua instância de banco de dados. Para obter mais informações, consulte Trabalhar com grupos de opções.

Se você está procurando recursos opcionais que não são adicionados por meio de grupos de opções do RDS (como SSL, Autenticação do Microsoft Windows e integração do Amazon S3), consulte Recursos adicionais do Microsoft SQL Server no Amazon RDS.

O Amazon RDS suporta as seguintes opções de instâncias de banco de dados do Microsoft SQL Server.

Opção	ID da opção	Edições do mecanismo
Servidores vinculados ao Oracle OLEDB	`OLEDB_ORACLE`	SQL Server Enterprise Edition SQL Server Standard Edition
Backup e restauração nativos	`SQLSERVER_BACKUP_RESTORE`	SQL Server Enterprise Edition SQL Server Standard Edition SQL Server Web Edition SQL Server Express Edition

Opção	ID da opção	Edições do mecanismo
Transparent Data Encryption	`TRANSPARENT_DATA_ENCRYPTION` (console do RDS) `TDE` (AWS CLI e API do RDS)	SQL Server 2016–2022 Enterprise Edition SQL Server 2022 Standard Edition

Opção	ID da opção	Edições do mecanismo
Auditoria do SQL Server	SQLSERVER_AUDIT	No RDS, a partir do SQL Server 2016, todas as edições do SQL Server comportam auditorias em nível de servidor, e a edição Enterpris e também aceita auditorias em nível de banco de dados. Começando com o SQL Server SQL Server 2016 (13.x) SP1, todas as edições oferecem suporte a auditorias em nível de servidor e banco de dados. Para obter mais informações, consulte Auditoria do SQL Server (Mecanismo de banco de dados) na documentação do SQL Server.
SQL Server Analysis Services	SSAS	SQL Server Enterpris e Edition SQL Server Standard Edition

Opção	ID da opção	Edições do mecanismo
SQL Server Integration Services	SSIS	SQL Server Enterprise Edition SQL Server Standard Edition
SQL Server Reporting Services	SSRS	SQL Server Enterprise Edition SQL Server Standard Edition
Microsoft Distributed Transaction Coordinator	MSDTC	No RDS, a partir do SQL Server 2016, todas as edições do SQL Server comportam transações distribuídas.

Listas de opções disponíveis para versões e edições do SQL Server

Você pode usar o comando da `describe-option-group-options` AWS CLI para listar as opções disponíveis para versões e edições do SQL Server e as configurações para essas opções.

O exemplo a seguir mostra as opções e as configurações de opção para o SQL Server 2019 Enterprise Edition. A opção `--engine-name` é obrigatória.

```
aws rds describe-option-group-options --engine-name sqlserver-ee --major-engine-version
 15.00
```

A saída será semelhante à seguinte.

```
{
    "OptionGroupOptions": [
        {
            "Name": "MSDTC",
```

```
            "Description": "Microsoft Distributed Transaction Coordinator",
            "EngineName": "sqlserver-ee",
            "MajorEngineVersion": "15.00",
            "MinimumRequiredMinorEngineVersion": "4043.16.v1",
            "PortRequired": true,
            "DefaultPort": 5000,
            "OptionsDependedOn": [],
            "OptionsConflictsWith": [],
            "Persistent": false,
            "Permanent": false,
            "RequiresAutoMinorEngineVersionUpgrade": false,
            "VpcOnly": false,
            "OptionGroupOptionSettings": [
                {
                    "SettingName": "ENABLE_SNA_LU",
                    "SettingDescription": "Enable support for SNA LU protocol",
                    "DefaultValue": "true",
                    "ApplyType": "DYNAMIC",
                    "AllowedValues": "true,false",
                    "IsModifiable": true,
                    "IsRequired": false,
                    "MinimumEngineVersionPerAllowedValue": []
                },
        ...

        {
            "Name": "TDE",
            "Description": "SQL Server - Transparent Data Encryption",
            "EngineName": "sqlserver-ee",
            "MajorEngineVersion": "15.00",
            "MinimumRequiredMinorEngineVersion": "4043.16.v1",
            "PortRequired": false,
            "OptionsDependedOn": [],
            "OptionsConflictsWith": [],
            "Persistent": true,
            "Permanent": false,
            "RequiresAutoMinorEngineVersionUpgrade": false,
            "VpcOnly": false,
            "OptionGroupOptionSettings": []
        }
    ]
}
```

Suporte para servidores vinculados ao Oracle OLEDB no Amazon RDS para SQL Server

Os servidores vinculados ao Oracle Provider para OLEDB no RDS para SQL Server possibilitam que você acesse fontes de dados externas em um banco de dados Oracle. Você pode ler dados de fontes de dados Oracle remotas e executar comandos em servidores de banco de dados Oracle remotos fora de sua instância de banco de dados do RDS para SQL Server. Usando servidores vinculados ao Oracle OLEDB, você pode:

- Acessar diretamente fontes de dados que não sejam o SQL Server
- Consultar diversas fontes de dados do Oracle com a mesma consulta sem mover os dados
- Emitir consultas, atualizações, comandos e transações distribuídos em fontes de dados em um ecossistema corporativo
- Integrar conexões com um banco de dados Oracle a partir do pacote Microsoft Business Intelligence (SSIS, SSRS, SSAS)
- Migrar de um banco de dados Oracle para o RDS para SQL Server

Você pode ativar um ou mais servidores vinculados para Oracle em uma instância de banco de dados existente ou nova do RDS para SQL Server. Depois, é possível integrar fontes de dados Oracle externas à sua instância de banco de dados.

Sumário

- Versões e regiões compatíveis
- Limitações e recomendações
- Ativar servidores vinculados ao Oracle
  - Criar o grupo de opções para OLEDB_ORACLE
  - Adicionar a opção OLEDB_ORACLE ao grupo de opções
  - Modificar a opção de versão do OLEDB_ORACLE para outra versão
  - Associar o grupo de opções à instância de banco de dados
- Modificar propriedades do provedor do OLEDB
- Modificar propriedades do driver OLEDB
- Desativar servidores vinculados ao Oracle

Versões e regiões compatíveis

O RDS para SQL Server é compatível com servidores vinculados ao Oracle OLEDB em todas as regiões para as edições SQL Server Standard e Enterprise Edition nas seguintes versões:

- SQL Server 2022, todas as versões
- SQL Server 2019, todas as versões
- SQL Server 2017, todas as versões

Os servidores vinculados ao Oracle OLEDB são compatíveis com as seguintes versões do banco de dados Oracle:

- Oracle Database 21c, todas as versões
- Oracle Database 19c, todas as versões
- Oracle Database 18c, todas as versões

Os servidores vinculados ao Oracle OLEDB são compatíveis com as seguintes versões do driver OLEDB Oracle:

- 21.7
- 21.16

Limitações e recomendações

Lembre-se das seguintes limitações e recomendações que se aplicam aos servidores vinculados ao Oracle OLEDB:

- Permita o tráfego de rede adicionando a porta TCP aplicável no grupo de segurança para cada instância de banco de dados do RDS para SQL Server. Por exemplo, se você estiver configurando um servidor vinculado entre uma instância de banco de dados Oracle do EC2 e uma instância de banco de dados do RDS para SQL Server, deverá permitir o tráfego do endereço IP da instância de banco de dados Oracle do EC2. Você também deve permitir o tráfego na porta que o SQL Server está usando para escutar a comunicação do banco de dados. Para obter mais informações sobre grupos de segurança, consulte Controlar acesso com grupos de segurança.
- Execute uma reinicialização da instância de banco de dados do RDS para SQL Server depois de ativar, desativar ou modificar a opção `OLEDB_ORACLE` em seu grupo de opções. O status do grupo

de opções é `pending_reboot` para esses eventos e é obrigatório. Para instâncias multi-AZ do RDS para SQL Server com a opção “Sempre ativa” ou “Espelhamento” habilitada, um failover é esperado quando a instância é reiniciada após a criação ou restauração da nova instância.

- Somente a autenticação simples é compatível com nome de usuário e senha para a fonte de dados Oracle.
- Não há suporte aos drivers do Open Database Connectivity (ODBC). Somente as versões do driver OLEDB listadas acima são compatíveis.
- Transações distribuídas (XA) são compatíveis. Para ativar transações distribuídas, ative a opção `MSDTC` no Grupo de Opções para sua instância de banco de dados e garanta que as transações XA estejam ativadas. Para obter mais informações, consulte Suporte para o Microsoft Distributed Transaction Coordinator no RDS for SQL Server.
- A criação de nomes de fonte de dados (DSNs) para usar como atalho para uma cadeia de conexão não é compatível.
- O rastreamento do driver OLEDB não é compatível. Você pode usar o SQL Server Extended Events para rastrear eventos do OLEDB. Para obter mais informações, consulte Set up Extended Events in RDS for SQL Server (Configurar eventos estendidos no RDS para SQL Server).
- O acesso à pasta de catálogos de um servidor vinculado Oracle não é compatível com ouso do SQL Server Management Studio (SSMS).

Ativar servidores vinculados ao Oracle

Ative servidores vinculados ao Oracle adicionando a opção `OLEDB_ORACLE` à sua instância de banco de dados do RDS para SQL Server. Use o seguinte processo:

1. Crie um grupo de opções ou selecione um existente.
2. Adicione a opção `OLEDB_ORACLE` ao grupo de opções.
3. Selecione uma versão do driver OLEDB para usar.
4. Associe o grupo de opções à instância de banco de dados.
5. Reinicialize a instância de banco de dados.

Criar o grupo de opções para OLEDB_ORACLE

Para trabalhar com servidores vinculados ao Oracle, crie um grupo de opções ou modifique um grupo de opções que corresponda à edição do SQL Server e à versão da instância de banco de

dados que você pretende usar. Para concluir esse procedimento, use o AWS Management Console ou a AWS CLI.

Console

O procedimento a seguir cria um grupo de opções para o SQL Server Standard Edition 2019.

Como criar o grupo de opções

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Option groups (Grupos de opções).
3. Escolha Create group (Criar grupo).
4. Na janela Create option group (Criar grupo de opções), faça o seguinte:
    a. Em Name (Nome), insira um nome para o grupo de opções que seja exclusivo na sua conta da AWS, como **`oracle-oledb-se-2019`**. O nome pode conter somente letras, números e hifens.
    b. Em Descrição, insira uma breve descrição do grupo de opções, como **`OLEDB_ORACLE option group for SQL Server SE 2019`**. A descrição é usada para fins de exibição.
    c. Em Engine (Mecanismo), selecione sqlserver-se.
    d. Em Major engine version (Versão do mecanismo principal), selecione 15.00.
5. Escolha Criar.

CLI

O procedimento a seguir cria um grupo de opções para o SQL Server Standard Edition 2019.

Como criar o grupo de opções

- Execute um dos seguintes comandos:

  Example

  Para Linux, macOS ou Unix:

```
aws rds create-option-group \
    --option-group-name oracle-oledb-se-2019 \
    --engine-name sqlserver-se \
```

```
    --major-engine-version 15.00 \
    --option-group-description "OLEDB_ORACLE option group for SQL Server SE 2019"
```

Para Windows:

```
aws rds create-option-group ^
    --option-group-name oracle-oledb-se-2019 ^
    --engine-name sqlserver-se ^
    --major-engine-version 15.00 ^
    --option-group-description "OLEDB_ORACLE option group for SQL Server SE 2019"
```

Adicionar a opção **OLEDB_ORACLE** ao grupo de opções

Depois, use o AWS Management Console ou a AWS CLI para adicionar a opção `OLEDB_ORACLE` ao grupo de opções.

Console

Como adicionar a opção OLEDB_ORACLE

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Option groups (Grupos de opções).
3. Selecione o grupo de opções que você acabou de criar, que é oracle-oledb-se-2019 neste exemplo.
4. Escolha Add option (Adicionar opção).
5. Em Option details (Detalhes da opção), selecione OLEDB_ORACLE em Option name (Nome da opção).
6. Em Versão, escolha a versão do driver OLEDB Oracle que você deseja instalar.
7. Em Scheduling (Programação), escolha se deseja adicionar a opção imediatamente ou na próxima janela de manutenção.
8. Escolha Add option (Adicionar opção).

CLI

Como adicionar a opção OLEDB_ORACLE

- Adicione a opção `OLEDB_ORACLE` ao grupo de opções.

  Example

  Para Linux, macOS ou Unix:

  ```
  aws rds add-option-to-option-group \
      --option-group-name oracle-oledb-se-2019 \
      --options OptionName=OLEDB_ORACLE, OptionVersion=21.16 \
      --apply-immediately
  ```

  Para Windows:

  ```
  aws rds add-option-to-option-group ^
      --option-group-name oracle-oledb-se-2019 ^
      --options OptionName=OLEDB_ORACLE, OptionVersion=21.16 ^
      --apply-immediately
  ```

Modificar a opção de versão do **`OLEDB_ORACLE`** para outra versão

Para modificar a versão da opção `OLEDB_ORACLE` para outra versão, use o AWS Management Console ou a AWS CLI.

Console

Como modificar a opção OLEDB_ORACLE

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Option groups (Grupos de opções).
3. Selecione o grupo de opções com a opção `OLEDB_ORACLE` (oracle-oledb-se-2019 no exemplo anterior).
4. Escolha Modify option (Modificar opção).
5. Em Option details (Detalhes da opção), selecione OLEDB_ORACLE em Option name (Nome da opção).

6. Em Versão, escolha a versão do driver OLEDB Oracle que você deseja usar.
7. Em Programação, escolha se deseja adicionar a opção imediatamente ou na próxima janela de manutenção.
8. Escolha Modify option (Modificar opção).

CLI

Para modificar a versão da opção `OLEDB_ORACLE`, use o comando `rds add-option-to-option-group` da AWS CLI com o grupo de opções e a versão da opção que você deseja usar.

Como modificar a opção OLEDB_ORACLE

- Example

  Para Linux, macOS ou Unix:

```
aws rds add-option-to-option-group \
    --option-group-name oracle-oledb-se-2019 \
    --options OptionName=OLEDB_ORACLE, OptionVersion=21.7 \
    --apply-immediately
```

  Para Windows:

```
aws rds add-option-to-option-group ^
    --option-group-name oracle-oledb-se-2019 ^
    --options OptionName=OLEDB_ORACLE, OptionVersion=21.7 ^
    --apply-immediately
```

Associar o grupo de opções à instância de banco de dados

Para associar o grupo de opções `OLEDB_ORACLE` e o grupo de parâmetros à sua instância de banco de dados, use o AWS Management Console ou a AWS CLI

Console

Para concluir a ativação de servidores vinculados para Oracle, associe o grupo de opções `OLEDB_ORACLE` a uma instância de banco de dados nova ou existente:

- Para uma nova instância de banco de dados, associe-os ao executar a instância. Para obter mais informações, consulte Criar uma instância de banco de dados do Amazon RDS.
- Para uma instância de banco de dados existente, associe-os modificando a instância. Para obter mais informações, consulte Modificar uma instância de banco de dados do Amazon RDS.

CLI

É possível associar o grupo de opções `OLEDB_ORACLE` e o grupo de parâmetros a uma instância de banco de dados nova ou existente.

Como criar uma instância com o grupo de opções **`OLEDB_ORACLE`** e o grupo de parâmetros

- Especifique o mesmo tipo de mecanismo de banco de dados e a versão principal que você usou ao criar o grupo de opções.

  Example

  Para Linux, macOS ou Unix:

```
aws rds create-db-instance \
    --db-instance-identifier mytestsqlserveroracleoledbinstance \
    --db-instance-class db.m5.2xlarge \
    --engine sqlserver-se \
    --engine-version 15.0.4236.7.v1 \
    --allocated-storage 100 \
    --manage-master-user-password \
    --master-username admin \
    --storage-type gp2 \
    --license-model li \
    --domain-iam-role-name my-directory-iam-role \
    --domain my-domain-id \
    --option-group-name oracle-oledb-se-2019 \
    --db-parameter-group-name my-parameter-group-name
```

  Para Windows:

```
aws rds create-db-instance ^
    --db-instance-identifier mytestsqlserveroracleoledbinstance ^
    --db-instance-class db.m5.2xlarge ^
    --engine sqlserver-se ^
    --engine-version 15.0.4236.7.v1 ^
```

```
    --allocated-storage 100 ^
    --manage-master-user-password ^
    --master-username admin ^
    --storage-type gp2 ^
    --license-model li ^
    --domain-iam-role-name my-directory-iam-role ^
    --domain my-domain-id ^
    --option-group-name oracle-oledb-se-2019 ^
    --db-parameter-group-name my-parameter-group-name
```

Como modificar uma instância e associar o grupo de opções **OLEDB_ORACLE**

- Execute um dos seguintes comandos:

  Example

  Para Linux, macOS ou Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier mytestsqlserveroracleoledbinstance \
    --option-group-name oracle-oledb-se-2019 \
    --db-parameter-group-name my-parameter-group-name \
    --apply-immediately
```

  Para Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mytestsqlserveroracleoledbinstance ^
    --option-group-name oracle-oledb-se-2019 ^
    --db-parameter-group-name my-parameter-group-name ^
    --apply-immediately
```

Modificar propriedades do provedor do OLEDB

Você pode visualizar e alterar as propriedades do provedor do OLEDB. Somente o usuário `master` pode realizar essa tarefa. Todos os servidores vinculados para Oracle criados na instância de banco de dados usam as mesmas propriedades desse provedor do OLEDB. Chame o procedimento armazenado `sp_MSset_oledb_prop` para alterar as propriedades do provedor do OLEDB.

Como alterar as propriedades do provedor do OLEDB

```
USE [master]
GO
EXEC sp_MSset_oledb_prop N'OraOLEDB.Oracle', N'AllowInProcess', 1
EXEC sp_MSset_oledb_prop N'OraOLEDB.Oracle', N'DynamicParameters', 0
GO
```

As seguintes propriedades podem ser modificadas:

Nome da propriedade	Valor recomendado (1 = ativado, 0 = desativado)	Descrição
`Dynamic parameter`	1	Permite espaços reservados para SQL (representados por “?”) em consultas parametrizadas.
`Nested queries`	1	Permite declarações `SELECT` aninhadas na cláusula `FROM`, como subconsultas.
`Level zero only`	0	Somente interfaces OLEDB de nível básico são chamadas em relação ao provedor.
`Allow inprocess`	1	Se ativado, o Microsoft SQL Server permite que o provedor seja instanciado como um servidor em processo. Defina essa propriedade como 1 para usar servidores vinculados Oracle.
`Non transacted updates`	0	Se for diferente de zero, o SQL Server permitirá atualizações.
`Index as access path`	Falso	Se for diferente de zero, o SQL Server tentará usar índices do provedor para buscar dados.
`Disallow adhoc access`	Falso	Se definido, o SQL Server não permitirá a execução de consultas de passagem no provedor do OLEDB. Embora essa opção possa ser marcada, às vezes é apropriado executar consultas de passagem.

Nome da propriedade	Valor recomendado (1 = ativado, 0 = desativado)	Descrição
`Supports LIKE operator`	1	Indica que o provedor é compatível com consultas usando a palavra-chave LIKE.

Modificar propriedades do driver OLEDB

Você pode visualizar e alterar as propriedades do driver OLEDB ao criar um servidor vinculado para Oracle. Somente o usuário `master` pode realizar essa tarefa. As propriedades do driver definem como o driver OLEDB manipula os dados ao trabalhar com uma fonte de dados Oracle remota. As propriedades do driver são específicas de cada servidor vinculado Oracle criado na instância de banco de dados. Chame o procedimento armazenado `master.dbo.sp_addlinkedserver` para alterar as propriedades do driver OLEDB.

Exemplo: Para criar um servidor vinculado e alterar a propriedade `FetchSize` do driver OLEDB

```
EXEC master.dbo.sp_addlinkedserver
@server = N'Oracle_link2',
@srvproduct=N'Oracle',
@provider=N'OraOLEDB.Oracle',
@datasrc=N'my-oracle-test.cnetsipka.us-west-2.rds.amazonaws.com:1521/ORCL',
@provstr='FetchSize=200'
GO
```

```
EXEC master.dbo.sp_addlinkedsrvlogin
@rmtsrvname=N'Oracle_link2',
@useself=N'False',
@locallogin=NULL,
@rmtuser=N'master',
@rmtpassword='Test#1234'
GO
```

> **Note**
>
> Especifique uma senha diferente do prompt mostrado aqui como prática recomendada de segurança.

Desativar servidores vinculados ao Oracle

Para desabilitar servidores vinculados ao Oracle, remova a opção `OLEDB_ORACLE` do grupo de opções.

> **Important**
>
> Remover a opção não exclui as configurações de servidor vinculado existentes na instância de banco de dados. Você deve descartá-los manualmente para removê-los da instância de banco de dados.
> Você pode reativar a opção `OLEDB_ORACLE` após a remoção para reutilizar as configurações do servidor vinculado definidas anteriormente na instância de banco de dados.

Console

O procedimento a seguir remove a opção `OLEDB_ORACLE`.

Como remover a opção OLEDB_ORACLE do grupo de opções

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Option groups (Grupos de opções).
3. Selecione o grupo de opções com a opção `OLEDB_ORACLE` (`oracle-oledb-se-2019` nos exemplos anteriores).
4. Selecione Delete option (Excluir opção).
5. Em Deletion options (Opções de exclusão), selecione OLEDB_ORACLE em Options to delete (Opções a serem excluídas).
6. Em Apply immediately (Aplicar imediatamente), selecione Yes (Sim) para excluí-la imediatamente ou No (Não) para excluí-la na próxima janela de manutenção.
7. Escolha Excluir.

CLI

O procedimento a seguir remove a opção `OLEDB_ORACLE`.

Como remover a opção OLEDB_ORACLE do grupo de opções

- Execute um dos seguintes comandos:

  Example

  Para Linux, macOS ou Unix:

```
aws rds remove-option-from-option-group \
    --option-group-name oracle-oledb-se-2019 \
    --options OLEDB_ORACLE \
    --apply-immediately
```

  Para Windows:

```
aws rds remove-option-from-option-group ^
    --option-group-name oracle-oledb-se-2019 ^
    --options OLEDB_ORACLE ^
    --apply-immediately
```

Servidores vinculados ao Teradata ODBC no RDS para SQL Server

O suporte para servidores vinculados ao driver ODBC Teradata no RDS para SQL Server possibilita que você acesse fontes de dados externas em um banco de dados Teradata. É possível ler dados e executar comandos em servidores de banco de dados Teradata remotos fora da instância do RDS para SQL Server. Use servidores vinculados ao ODBC Teradata para habilitar os seguintes recursos:

- Acesse diretamente fontes de dados que não sejam o SQL Server.
- Consulte diversas fontes de dados do Teradata com a mesma consulta sem mover os dados.
- Emita consultas, atualizações, comandos e transações distribuídos em fontes de dados em um ecossistema corporativo.
- Integre conexões a um banco de dados do Teradata por meio do pacote Microsoft Business Intelligence (SSIS, SSRS, SSAS).
- Migre de um banco de dados Teradata para o RDS para SQL Server.

É possível ativar um ou mais servidores vinculados para Teradata em uma instância de banco de dados existente ou nova do RDS para SQL Server. Depois, é possível integrar fontes de dados externas do Teradata à sua instância de banco de dados.

Tópicos

- Versões e regiões compatíveis
- Limitações e recomendações
- Considerações para implantações Multi-AZ
- Ativar servidores vinculados ao Teradata
- Criar servidores vinculados ao Teradata
- Desativar servidores vinculados ao Teradata

Versões e regiões compatíveis

O RDS para SQL Server comporta servidores vinculados ao ODBC Teradata em todas as Regiões da AWS para SQL Server Standard e Enterprise Edition nas seguintes versões:

- SQL Server 2022, todas as versões
- SQL Server 2019, todas as versões
- SQL Server 2017, todas as versões

As versões do banco de dados Teradata a seguir comportam a vinculação com o RDS para SQL Server

- Teradata 17.20, todas as versões

Limitações e recomendações

As seguintes limitações se aplicam aos servidores vinculados ao ODBC Teradata:

- O RDS para SQL Server aceita apenas a autenticação simples com nome de usuário e senha para a origem do Teradata.
- O RDS para SQL Server aceita somente o driver ODBC Teradata versão 17.20.0.33.
- O RDS para SQL Server não comporta a criação de nomes de fonte de dados (DSNs) a serem usados como atalhos para uma string de conexão.
- O RDS para SQL Server não aceita o rastreamento do driver ODBC. Use o SQL Server Extended Events para rastrear eventos do ODBC. Para obter mais informações, consulte Set up Extended Events in RDS for SQL Server (Configurar eventos estendidos no RDS para SQL Server).
- O RDS para SQL Server não aceita o acesso à pasta de catálogos de um servidor vinculado Teradata durante o uso do SQL Server Management Studio (SSMS).

Pense nas seguintes recomendações ao usar servidores vinculados ao ODBC Teradata:

- Permita o tráfego de rede adicionando a porta TCP aplicável no grupo de segurança para cada instância de banco de dados do RDS para SQL Server. Por exemplo, se você estiver configurando um servidor vinculado entre uma instância de banco de dados do EC2 Teradata e uma instância de banco de dados do RDS para SQL Server, deverá permitir o tráfego do endereço IP da instância de banco de dados do EC2 Teradata. Você também deve permitir o tráfego na porta que a instância de banco de dados do RDS para SQL Server está usando para escutar a comunicação do banco de dados. Para obter mais informações sobre grupos de segurança, consulte Controlar acesso com grupos de segurança.
- Transações distribuídas (XA) são compatíveis. Para ativar transações distribuídas, ative a opção `MSDTC` no grupo de opções para sua instância de banco de dados e garanta que as transações XA estejam ativadas. Para ter mais informações, consulte Suporte para o Microsoft Distributed Transaction Coordinator no RDS for SQL Server.
- O ODBC Teradata vinculado aceita SSL/TLS, desde que configurado no Servidor Teradata. Para ter mais informações, consulte Enable TLS Connectivity on Teradata Vantage.

Considerações para implantações Multi-AZ

Atualmente, o RDS para SQL Server não replica servidores vinculados ao servidor de banco de dados espelhado (ou ao servidor secundário do grupo de disponibilidade Always On) em uma implantação Multi-AZ. Se os servidores vinculados forem adicionados antes que a configuração seja alterada para adicionar espelhamento ou Always On, os servidores vinculados serão copiados para os servidores vinculados existentes.

Também é possível criar os servidores vinculados na instância primária, fazer failover para a instância do servidor de alta disponibilidade e, depois, criar os servidores vinculados novamente para que estejam nas duas instâncias do RDS para SQL Server.

Ativar servidores vinculados ao Teradata

Ative servidores vinculados ao Teradata adicionando a opção `ODBC_TERADATA` à sua instância de banco de dados do RDS para SQL Server. Use o seguinte processo:

Tópicos

- Criar o grupo de opções para ODBC_TERADATA
- Adicionar a opção ODBC_TERADATA ao grupo de opções
- Associar a opção ODBC_TERADATA à instância de banco de dados

Criar o grupo de opções para **`ODBC_TERADATA`**

Para trabalhar com servidores vinculados ao Teradata, crie um grupo de opções ou modifique um grupo de opções que corresponda à edição do SQL Server e à versão da instância de banco de dados que você pretende usar. Para concluir esse procedimento, use o AWS Management Console ou a AWS CLI.

Console

Use o procedimento a seguir para criar um grupo de opções para o SQL Server Standard Edition 2019.

Como criar o grupo de opções

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Option groups (Grupos de opções).

3. Escolha Create group (Criar grupo).
4. Na janela Create option group (Criar grupo de opções), faça o seguinte:
    a. Em Name (Nome), insira um nome para o grupo de opções que seja exclusivo na sua Conta da AWS, como `teradata-odbc-se-2019`. O nome pode conter somente letras, números e hifens.
    b. Em Descrição, insira uma breve descrição do grupo de opções.
    c. Em Engine (Mecanismo), selecione sqlserver-se.
    d. Em Major engine version (Versão do mecanismo principal), selecione 15.00.
5. Escolha Criar.

AWS CLI

O procedimento a seguir cria um grupo de opções para o SQL Server Standard Edition 2019.

Example

Para Linux, macOS ou Unix:

```
aws rds create-option-group \
    --option-group-name teradata-odbc-se-2019 \
    --engine-name sqlserver-se \
    --major-engine-version 15.00 \
    --option-group-description "ODBC_TERADATA option group for SQL Server SE 2019"
```

Example

Para Windows:

```
aws rds create-option-group ^
    --option-group-name teradata-odbc-se-2019 ^
    --engine-name sqlserver-se ^
    --major-engine-version 15.00 ^
    --option-group-description "ODBC_TERADATA option group for SQL Server SE 2019"
```

Adicionar a opção **ODBC_TERADATA** ao grupo de opções

Depois, use o AWS Management Console ou a AWS CLI para adicionar a opção `ODBC_Teradata` ao grupo de opções.

Console

Use o procedimento a seguir para criar um grupo de opções para o SQL Server Standard Edition 2019.

Como adicionar a opção **ODBC_TERADATA**

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Option groups (Grupos de opções).
3. Escolha o novo grupo de opções.
4. Escolha Add option (Adicionar opção).
5. Em Detalhes da opção:
   a. Escolha ODBC_TERADATA para Nome da opção.
   b. Em `17.20.33.00` para Versão da opção.
6. Em Programação, escolha se deseja adicionar a opção imediatamente ou na próxima janela de manutenção.
7. Escolha Add option (Adicionar opção).

AWS CLI

O procedimento a seguir adiciona a opção `ODBC_TERADATA` ao grupo de opções.

Example

Para Linux, macOS ou Unix:

```
aws rds add-option-to-option-group \
    --option-group-name teradata-odbc-se-2019 \
    --options "OptionName=ODBC_TERADATA,OptionVersion=17.20.33.00" \
    --apply-immediately
```

Example

Para Windows:

```
aws rds add-option-to-option-group ^
```

```
    --option-group-name teradata-odbc-se-2019 ^
    --options "OptionName=ODBC_TERADATA,OptionVersion=17.20.33.00" ^
    --apply-immediately
```

Associar a opção **ODBC_TERADATA** à instância de banco de dados

Para associar o grupo de opções ODBC_TERADATA à instância de banco de dados, use o AWS Management Console ou a AWS CLI.

Console

Para concluir a ativação de servidores vinculados para Teradata, associe o grupo de opções a uma instância de banco de dados nova ou existente:

- Para uma nova instância de banco de dados, associe-o ao executar a instância. Para ter mais informações, consulte Criar uma instância de banco de dados do Amazon RDS.
- Para uma instância de banco de dados existente, associe-o modificando a instância. Para ter mais informações, consulte Modificar uma instância de banco de dados do Amazon RDS.

AWS CLI

Especifique o mesmo tipo de mecanismo de banco de dados e a versão principal que você usou ao criar o grupo de opções.

Para Linux, macOS ou Unix:

```
aws rds create-db-instance \
    --db-instance-identifier mytestsqlserverteradataodbcinstance \
    --db-instance-class db.m5.2xlarge \
    --engine sqlserver-se \
    --engine-version 15.00 \
    --license-model license-included \
    --allocated-storage 100 \
    --master-username admin \
    --master-user-password password \
    --storage-type gp2 \
    --option-group-name teradata-odbc-se-2019
```

Para Windows:

```
aws rds create-db-instance ^
```

```
    --db-instance-identifier mytestsqlserverteradataodbcinstance ^
    --db-instance-class db.m5.2xlarge ^
    --engine sqlserver-se ^
    --engine-version 15.00 ^
    --license-model license-included ^
    --allocated-storage 100 ^
    --master-username admin ^
    --master-user-password password ^
    --storage-type gp2 ^
    --option-group-name teradata-odbc-se-2019
```

Para modificar uma instância e associar o novo grupo de opções:

Para Linux, macOS ou Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier mytestsqlserverteradataodbcinstance \
    --option-group-name teradata-odbc-se-2019 \
    --apply-immediately
```

Para Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mytestsqlserverteradataodbcinstance ^
    --option-group-name teradata-odbc-se-2019 ^
    --apply-immediately
```

Criar servidores vinculados ao Teradata

Para criar um servidor vinculado ao Teradata, execute os seguintes comandos:

```
EXECUTE master.dbo.sp_addlinkedserver
    @server = N'LinkedServer_NAME',
    @srvproduct=N'',
    @provider=N'MSDASQL',
    @provstr=N'"PROVIDER=MSDASQL;DRIVER={Teradata Database ODBC Driver 17.20};
                DBCName=Server;UID=user_name;PWD=user_password;
                UseDataEncryption=YES/NO;SSLMODE=PREFER/ALLOW/DISABLE>;"',
    @catalog='database'
```

```
EXECUTE master.dbo.sp_addlinkedsrvlogin
```

```
    @rmtsrvname = N'LinkedServer_NAME',
    @locallogin = NULL ,
    @useself = N'False',
    @rmtuser = N'user_name',
    @rmtpassword = N'user_password'
```

Um exemplo dos comandos acima é mostrado aqui:

```
EXECUTE master.dbo.sp_addlinkedserver
    @server = N'LinkedServerToTeradata',
    @srvproduct=N'',
    @provider=N'MSDASQL',
    @provstr=N'"PROVIDER=MSDASQL;DRIVER={Teradata Database ODBC Driver 17.20};
                DBCName=my-teradata-test.cnetsipka.us-west-2.rds.amazonaws.com;
                UID=master;
                PWD=Test#1234;
                UseDataEncryption=YES;
                SSLMODE=PREFER;"',
    @catalog='MyTestTeradataDB'

EXECUTE master.dbo.sp_addlinkedsrvlogin
    @rmtsrvname = N'LinkedServerToTeradata',
    @locallogin = NULL ,
    @useself = N'False',
    @rmtuser = N'master',
    @rmtpassword = N'Test#1234'
```

> **Note**
>
> Especifique uma senha diferente do prompt mostrado aqui como prática recomendada de segurança.

Desativar servidores vinculados ao Teradata

Para desativar servidores vinculados ao Teradata, remova a opção `ODBC_TERADATA` do grupo de opções.

> **Important**
>
> Excluir a opção não exclui as configurações de servidor vinculado existentes na instância de banco de dados. Você deve descartá-los manualmente para removê-los da instância de banco de dados.
>
> Você pode reativar a `ODBC_TERADATA` após a remoção para reutilizar as configurações do servidor vinculado definidas anteriormente na instância de banco de dados.

Console

Como remover a opção `ODBC_TERADATA` do grupo de opções

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Option groups (Grupos de opções).
3. Escolha o grupo de opções com a opção `ODBC_TERADATA`.
4. Escolha Excluir.
5. Em Opções de exclusão, escolha `ODBC_TERADATA` em Opções a serem excluídas.
6. Em Apply immediately (Aplicar imediatamente), selecione Yes (Sim) para excluí-la imediatamente ou No (Não) para excluí-la na próxima janela de manutenção.
7. Escolha Excluir.

AWS CLI

O comando a seguir remove a opção `ODBC_TERADATA`.

Para Linux, macOS ou Unix:

```
aws rds remove-option-from-option-group \
    --option-group-name teradata-odbc-se-2019 \
    --options ODBC_TERADATA \
    --apply-immediately
```

Para Windows:

```
aws rds remove-option-from-option-group ^
```

```
    --option-group-name teradata-odbc-se-2019 ^
    --options ODBC_TERADATA ^
    --apply-immediately
```

Suporte para backup e restauração nativos no SQL Server

Usando backup e restauração nativos para bancos de dados do SQL Server, é possível criar um backup diferencial ou completo do banco de dados no local e armazenar os arquivos de backup no Amazon S3. Em seguida, você pode restaurar para uma instância de banco de dados existente do Amazon RDS que executa o SQL Server. Você também pode fazer backup de um banco de dados do RDS for SQL Server, armazená-lo no Amazon S3 e restaurá-lo em outros locais. Além disso, você pode restaurar o backup em um servidor local ou em uma instância de banco de dados do Amazon RDS diferente que executa o SQL Server. Para obter mais informações, consulte Importar e exportar bancos de dados do SQL Server usando backup e restauração nativos.

O Amazon RDS é compatível com o backup e restauração nativos para bancos de dados do Microsoft SQL Server usando arquivos de backup diferencial e completo (arquivos .bak).

Adicionar a opção de backup e restauração nativos

O processo geral para adicionar a opção de backup e restauração nativos a uma instância de banco de dados é o seguinte:

1. Crie um novo grupo de opções, ou copie ou modifique um existente.
2. Adicione a opção `SQLSERVER_BACKUP_RESTORE` ao grupo de opções.
3. Associe uma função do AWS Identity and Access Management (IAM) à opção. A função do IAM deve ter acesso a um bucket do S3 para armazenar os backups do banco de dados.

   Ou seja, ela deve ter como opção definir um nome de recurso da Amazon (ARN) válido no formato `arn:aws:iam::`*`account-id`*`:role/`*`role-name`*. Para obter mais informações, consulte Nomes de recurso da Amazon (ARNs) na Referência geral da AWS.

   A função do IAM também deve ter um relacionamento de confiança e uma política de permissões anexadas. A relação de confiança permite que o RDS assuma a função e a política de permissões define as ações que a função pode executar. Para obter mais informações, consulte Criar manualmente uma função do IAM para backup e restauração nativos.
4. Associe o grupo de opções à instância de banco de dados.

Após adicionar a opção de backup e restauração nativos, você não precisará reiniciar a instância de banco de dados. Assim que o grupo de opções estiver ativo, você poderá começar a fazer backup e restaurar imediatamente.

Console

Como adicionar a opção de backup e restauração nativos

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Option groups (Grupos de opções).
3. Crie um grupo de opções ou use um existente. Para obter informações sobre como criar um grupo de opções de banco de dados personalizado, consulte Criar um grupo de opções.

   Para usar um grupo de opções existente, vá para a próxima etapa.
4. Adicione a opção SQLSERVER_BACKUP_RESTORE ao grupo de opções. Para mais informações sobre a adição de opções, consulte Adicionar uma opção a um grupo de opções.
5. Execute um destes procedimentos:

   - Para usar uma função do IAM existente e as configurações do Amazon S3, escolha uma função do IAM existente para a IAM Role (Função do IAM). Se você usar uma função do IAM existente, o RDS usará as configurações do Amazon S3 definidas para essa função.
   - Para criar uma nova função e definir novas configurações do Amazon S3, faça o seguinte:
     1. Em Perfil do IAM, selecione Criar uma nova função.
     2. Em S3 bucket name (Nome do bucket do S3), escolha um bucket do S3 na lista.
     3. Em S3 prefix (optional), (Prefixo do S3 - opcional) especifique um prefixo a ser utilizado para os arquivos armazenados no bucket do Amazon S3.

        Esse prefixo pode incluir um caminho de arquivo, mas isso não é necessário. Se você fornecer um prefixo, o RDS anexará esse prefixo a todos os arquivos de backup. O RDS usa o prefixo durante uma restauração para identificar arquivos relacionados e ignorar arquivos irrelevantes. Por exemplo, você pode usar o bucket do S3 para outros fins, além de manter arquivos de backup. Nesse caso, você pode usar o prefixo para que o RDS execute backup nativo e restaure apenas em uma pasta específica e suas subpastas.

        Se você deixar o prefixo em branco, o RDS não usará um prefixo para identificar arquivos de backup ou arquivos a serem restaurados. Como resultado, durante uma restauração com vários arquivos, o RDS tenta restaurar todos os arquivos em todas as pastas do bucket do S3.

4. Marque a caixa de seleção Enable encryption (Habilitar criptografia) para criptografar o arquivo de backup. Deixe a caixa de seleção desmarcada (o padrão) para deixar o arquivo de backup não criptografado.

   Se você escolheu Enable encryption (Habilitar criptografia), escolha uma chave de criptografia para AWS KMS key. Para obter mais informações sobre as chaves de criptografia, consulte Getting started (Conceitos básicos) no Guia do desenvolvedor do AWS Key Management Service.

6. Escolha Add option (Adicionar opção).
7. Aplique o grupo de opções a uma instância de banco de dados nova ou existente:

   - Para uma nova instância de banco de dados, aplique o grupo de opções ao executar a instância. Para obter mais informações, consulte Criar uma instância de banco de dados do Amazon RDS.
   - Para uma instância de banco de dados existente, aplique o grupo de opções modificando a instância e anexando o novo grupo de opções. Para obter mais informações, consulte Modificar uma instância de banco de dados do Amazon RDS.

CLI

Este procedimento faz as seguintes pressuposições:

- Você está adicionando a opção SQLSERVER_BACKUP_RESTORE a um grupo de opções que já existe. Para mais informações sobre a adição de opções, consulte Adicionar uma opção a um grupo de opções.
- Você está associando a opção a uma função do IAM que já existe e tem acesso a um bucket do S3 para armazenar os backups.
- Você está aplicando o grupo de opções a uma instância de banco de dados que já existe. Para obter mais informações, consulte Modificar uma instância de banco de dados do Amazon RDS.

Como adicionar a opção de backup e restauração nativos

1. Adicione a opção `SQLSERVER_BACKUP_RESTORE` ao grupo de opções.

   Example

   Para Linux, macOS ou Unix:

```
aws rds add-option-to-option-group \
 --apply-immediately \
 --option-group-name mybackupgroup \
 --options "OptionName=SQLSERVER_BACKUP_RESTORE, \
   OptionSettings=[{Name=IAM_ROLE_ARN,Value=arn:aws:iam::account-id:role/role-
name}]"
```

Para Windows:

```
aws rds add-option-to-option-group ^
 --option-group-name mybackupgroup ^
 --options "[{\"OptionName\": \"SQLSERVER_BACKUP_RESTORE\", ^
 \"OptionSettings\": [{\"Name\": \"IAM_ROLE_ARN\", ^
 \"Value\": \"arn:aws:iam::account-id:role/role-name"}]}]" ^
 --apply-immediately
```

> **Note**
>
> Ao usar o prompt de comando do Windows, você deve fazer o escape das aspas duplas (") no código JSON, prefixando-as com uma barra invertida (\).

2. Aplique o grupo de opções à instância de banco de dados.

   Example

   Para Linux, macOS ou Unix:

```
aws rds modify-db-instance \
 --db-instance-identifier mydbinstance \
 --option-group-name mybackupgroup \
 --apply-immediately
```

   Para Windows:

```
aws rds modify-db-instance ^
 --db-instance-identifier mydbinstance ^
 --option-group-name mybackupgroup ^
 --apply-immediately
```

Modificar as configurações de opções de backup e restauração nativos

Após ativar a opção de backup e restauração nativos, você poderá modificar as configurações dela. Para mais informações sobre como modificar as configurações da opção, consulte Modificar uma configuração de opção.

Remover a opção de backup e restauração nativos

Você pode desativar o backup e a restauração nativos, removendo a opção da instância de banco de dados. Após remover a opção de backup e restauração nativos, você não precisará reiniciar a instância de banco de dados.

Para remover a opção de backup e restauração nativos de uma instância de banco de dados, siga um destes procedimentos:

- Remova a opção do grupo de opções ao qual ela pertence. Essa alteração afeta todas as instâncias de bancos de dados que usam o grupo de opções. Para obter mais informações, consulte Remover uma opção de um grupo de opções.
- Modifique a instância do banco de dados e especifique um grupo de opções diferente que não inclua a opção de backup e restauração nativos. Essa alteração afeta uma única instância de banco de dados. Você pode especificar um grupo de opções padrão (vazio) ou criar um grupo de opções personalizado diferente. Para ter mais informações, consulte Modificar uma instância de banco de dados do Amazon RDS.

Suporte para criptografia de dados transparente no SQL Server

O Amazon RDS suporta o uso de Criptografia de Dados Transparente (TDE) para criptografar dados armazenados em suas instâncias de banco de dados executando o Microsoft SQL Server. O TDE criptografa os dados automaticamente antes de gravá-los no armazenamento e os descriptografa automaticamente quando os são lidos.

O Amazon RDS oferece suporte a TDE para as seguintes versões e edições do SQL Server:

- SQL Server 2022 Standard e Enterprise Editions
- SQL Server 2019 Standard e Enterprise Edition
- SQL Server 2017 Enterprise Edition
- SQL Server 2016 Enterprise Edition

A criptografia de dados transparente para o SQL Server permite o gerenciamento de chaves de criptografia usando uma arquitetura de chave de duas camadas. Um certificado, que é gerado a partir da chave mestre do banco de dados, é usado para proteger as chaves de criptografia de dados. A chave de criptografia do banco de dados executa a criptografia e a decodificação reais dos dados no banco de dados do usuário. O Amazon RDS faz backup e gerencia a chave primária do banco de dados e o certificado TDE.

A criptografia de dados transparente é usada em cenários em que você precisa criptografar dados confidenciais. Por exemplo, você pode querer fornecer arquivos de dados e backups a terceiros ou abordar problemas de conformidade regulatórios relacionados à segurança. Não é possível criptografar os bancos de dados do sistema para o SQL Server, como os bancos de dados `model` ou `master`.

Uma discussão detalhada sobre a Transparent Data Encryption não está no escopo deste guia, mas você deve entender os pontos fortes e fracos de segurança de cada algoritmo e chave de criptografia. Para obter informações sobre a criptografia de dados transparente para o SQL Server, consulte o tópico sobre a Transparent Data Encryption (TDE) no site da Microsoft.

Tópicos

- Ativar o TDE para o RDS para SQL Server
- Criptografar dados no RDS para SQL Server
- Fazer backup e restaurar certificados TDE no RDS para SQL Server

- Fazer backup e restaurar certificados TDE para bancos de dados on-premises
- Desativar o TDE para o RDS para SQL Server

Ativar o TDE para o RDS para SQL Server

Para ativar a Transparent Data Encryption para uma instância de banco de dados do RDS para SQL Server, especifique a opção TDE em um grupo de opções do RDS associado a essa instância de banco de dados.

1. Determine se a instância de banco de dados já está associada a um grupo de opções que tenha a opção TDE. Para visualizar o grupo de opções ao qual uma instância de banco de dados está associada, use o console do RDS, o comando describe-db-instance da AWS CLI ou a operação DescribeDBInstances da API.
2. Se a instância de banco de dados não estiver associada a um grupo de opções que tenha o TDE ativado, você terá duas opções. Você pode criar um grupo de opções e adicionar a opção TDE ou pode modificar o grupo de opções associado para adicioná-lo.

   > **Note**
   >
   > No console do RDS, a opção é chamada `TRANSPARENT_DATA_ENCRYPTION`. Na AWS CLI e na API do RDS, ele é chamado `TDE`.

   Para obter informações sobre como criar ou modificar um grupo de opções, consulte Trabalhar com grupos de opções. Para obter informações sobre como adicionar uma opção a um grupo de opções, consulte Adicionar uma opção a um grupo de opções.
3. Associe a instância de banco de dados ao grupo de opções com a opção TDE. Para obter informações sobre como associar uma instância de banco de dados a um grupo de opções, consulte Modificar uma instância de banco de dados do Amazon RDS.

Considerações de grupos de opções

A opção TDE é uma opção persistente. Não é possível removê-lo de um grupo de opções, a menos que todas as instâncias e backups de banco de dados não estejam mais associados ao grupo de opções. Quando você adiciona a opção TDE a um grupo de opções, o grupo de opções só pode ser associado a instâncias de banco de dados que usam TDE. Para mais informações sobre opções persistentes em um grupo de opções, consulte Visão geral de grupos de opções.

Como a opção TDE é uma opção persistente, você pode ter um conflito entre o grupo de opções e uma instância de banco de dados associada. Pode haver um conflito nas seguintes situações:

- O grupo de opções atual tem a opção TDE, e você o substitui por um grupo de opções sem a opção TDE.
- Faça uma restauração a partir de um snapshot de banco de dados para uma nova instância de banco de dados sem um grupo de opções que contenha a opção TDE. Para ter mais informações sobre esse cenário, consulte Considerações sobre grupos de opções.

Considerações de performance do SQL Server

O uso da Transparent Data Encryption pode afetar a performance de uma instância de banco de dados do SQL Server.

O performance de bancos de dados descriptografados também pode ser degradado se os bancos de dados estiverem em uma instância de banco de dados que tenha pelo menos um banco de dados criptografado. Como resultado, recomendamos que você mantenha bancos de dados criptografados e descriptografados em instâncias de banco de dados separadas.

Criptografar dados no RDS para SQL Server

Quando a opção TDE é adicionada a um grupo de opções, o Amazon RDS gera um certificado que é usado no processo de criptografia. Você pode então usar o certificado para executar instruções de SQL que criptografam dados em um banco de dados na instância do banco de dados.

O exemplo a seguir usa o certificado criado pelo RDS chamado `RDSTDECertificateName` para criptografar um banco de dados chamado `myDatabase`.

```
---------- Turning on TDE -------------

-- Find an RDS TDE certificate to use
USE [master]
GO
SELECT name FROM sys.certificates WHERE name LIKE 'RDSTDECertificate%'
GO

USE [myDatabase]
GO
-- Create a database encryption key (DEK) using one of the certificates from the
 previous step
```

```
CREATE DATABASE ENCRYPTION KEY WITH ALGORITHM = AES_256
ENCRYPTION BY SERVER CERTIFICATE [RDSTDECertificateName]
GO

-- Turn on encryption for the database
ALTER DATABASE [myDatabase] SET ENCRYPTION ON
GO

-- Verify that the database is encrypted
USE [master]
GO
SELECT name FROM sys.databases WHERE is_encrypted = 1
GO
SELECT db_name(database_id) as DatabaseName, * FROM sys.dm_database_encryption_keys
GO
```

O tempo necessário para criptografar um banco de dados do SQL Server usando o TDE depende de vários fatores. Eles incluem o tamanho da instância de banco de dados, se a instância usa o armazenamento de IOPS provisionadas, o volume de dados e outros fatores.

Fazer backup e restaurar certificados TDE no RDS para SQL Server

O RDS para SQL Server fornece procedimentos armazenados para backup, restauração e eliminação de certificados TDE. O RDS para SQL Server também fornece uma função para visualizar certificados TDE de usuário restaurados.

Os certificados TDE do usuário são usados para restaurar bancos de dados para o RDS para SQL Server que estão on-premises e têm o TDE ativado. Esses certificados têm o prefixo `UserTDECertificate_`. Depois de restaurar bancos de dados e antes de disponibilizá-los para uso, o RDS modifica os bancos de dados que têm o TDE ativado para usar certificados TDE gerados pelo RDS. Esses certificados têm o prefixo `RDSTDECertificate`.

Certificados TDE do usuário permanecem na instância de banco de dados do RDS para SQL Server, a menos que você os descarte usando os procedimentos armazenados `rds_drop_tde_certificate`. Para ter mais informações, consulte Descartar certificados TDE restaurados.

Você pode usar um certificado TDE de usuário para restaurar outros bancos de dados da instância de banco de dados de origem. Os bancos de dados a serem restaurados devem usar o mesmo certificado TDE e ter o TDE ativado. Você não precisa importar (restaurar) o mesmo certificado novamente.

Tópicos

- Pré-requisitos
- Limitações
- Fazer backup de um certificado TDE
- Restaurar um certificado TDE
- Visualizar certificados TDE restaurados
- Descartar certificados TDE restaurados

Pré-requisitos

Para fazer backup ou restaurar certificados TDE no RDS para SQL Server, realize as tarefas a seguir. As três primeiras são descritos em Configurar o backup e a restauração nativos.

1. Crie buckets do Amazon S3 para armazenar arquivos para backup e restauração.

   Recomendamos que você use buckets separados para backups de banco de dados e backups de certificados TDE.
2. Crie um perfil do IAM para fazer backup e restaurar arquivos.

   O perfil do IAM deve ser um usuário e um administrador da AWS KMS key.

   Além das permissões necessárias para backup e restauração nativos do SQL Server, o perfil do IAM também precisa as seguintes permissões:

   - `s3:GetBucketAcl`, `s3:GetBucketLocation` e `s3:ListBucket` no recurso de bucket do S3
   - `s3:ListAllMyBuckets` no recurso `*`
3. Adicionar a opção `SQLSERVER_BACKUP_RESTORE` a um grupo de opções em sua instância de banco de dados.

   Além da opção `TRANSPARENT_DATA_ENCRYPTION` (`TDE`).
4. Verifique se você tem uma chave do KMS de criptografia simétrica. Você tem as seguintes opções:

   - Se você tiver uma chave do KMS em sua conta, poderá usá-la. Nenhuma outra ação é necessária.
   - Se você não tiver uma chave do KMS de criptografia simétrica em sua conta, crie uma seguindo as instruções em Criar chaves, no Guia do desenvolvedor do AWS Key Management Service.

5. Habilite a integração com o Amazon S3 para transferir arquivos entre a instância de banco de dados e o Amazon S3.

   Para ter informações sobre a integração com o Amazon S3, consulte Integrar uma instância de banco de dados do Amazon RDS for SQL Server ao Amazon S3.

Limitações

O uso de procedimentos armazenados para fazer backup e restaurar certificados TDE tem as seguintes limitações:

- As duas opções `SQLSERVER_BACKUP_RESTORE` e `TRANSPARENT_DATA_ENCRYPTION` (`TDE`) devem ser adicionadas ao grupo de opções associado à sua instância de banco de dados.
- O backup e a restauração de certificados TDE não são compatíveis com instâncias de banco de dados multi-AZ.
- O cancelamento de tarefas de backup e restauração de certificados TDE não é compatível.
- Não é possível usar um certificado TDE de usuário para criptografia TDE de nenhum outro banco de dados em sua instância de banco de dados do RDS para SQL Server. Você pode usá-lo para restaurar somente outros bancos de dados da instância de banco de dados de origem que tem o TDE ativado e que usam o mesmo certificado TDE.
- Você pode descartar apenas certificados TDE do usuário.
- O número máximo de certificados TDE do usuário compatível com o RDS é dez. Se o número exceder dez, descarte os certificados TDE não utilizados e tente novamente.
- O nome do certificado não pode estar em branco nem ser nulo.
- Ao restaurar um certificado, o nome dele não pode incluir a palavra-chave `RDSTDECERTIFICATE` e deve iniciar com o prefixo `UserTDECertificate_`.
- O parâmetro `@certificate_name` pode incluir apenas os seguintes caracteres: a-z, 0-9, @, $, # e sublinhado (_).
- A extensão do arquivo `@certificate_file_s3_arn` deve ser .cer (não diferencia maiúsculas de minúsculas).
- A extensão do arquivo para `@private_key_file_s3_arn` deve ser .pvk (não diferencia maiúsculas de minúsculas).
- Os metadados do S3 do arquivo de chave privada devem incluir a tag `x-amz-meta-rds-tde-pwd`. Para ter mais informações, consulte Fazer backup e restaurar certificados TDE para bancos de dados on-premises.

Fazer backup de um certificado TDE

Para fazer backup de certificados TDE, use o procedimento armazenado `rds_backup_tde_certificate`. Ele tem a sintaxe a seguir.

```
EXECUTE msdb.dbo.rds_backup_tde_certificate
    @certificate_name='UserTDECertificate_certificate_name |
 RDSTDECertificatetimestamp',
    @certificate_file_s3_arn='arn:aws:s3:::bucket_name/certificate_file_name.cer',
    @private_key_file_s3_arn='arn:aws:s3:::bucket_name/key_file_name.pvk',
    @kms_password_key_arn='arn:aws:kms:region:account-id:key/key-id',
    [@overwrite_s3_files=0|1];
```

Os seguintes parâmetros são obrigatórios:

- `@certificate_name`: o nome do certificado TDE do qual fazer backup.
- `@certificate_file_s3_arn`: o nome do recurso da Amazon (ARN) de destino do arquivo de backup do certificado no Amazon S3.
- `@private_key_file_s3_arn`: o ARN do S3 de destino do arquivo de chave privada que protege o certificado TDE.
- `@kms_password_key_arn`: o ARN da chave do KMS simétrica usada para criptografar a senha da chave privada.

O parâmetro a seguir é opcional:

- `@overwrite_s3_files`: indica se os arquivos de chave privada e certificados existentes devem ser substituídos no S3:
  - `0`: não substitui os arquivos existentes. Esse valor é o padrão.

    Definir `@overwrite_s3_files` como 0 retornará um erro se já existir um arquivo.
  - `1` – substitui um arquivo existente que tenha o nome especificado, ainda que ele não seja um arquivo de backup.

Example Fazer backup de um certificado TDE

```
EXECUTE msdb.dbo.rds_backup_tde_certificate
    @certificate_name='RDSTDECertificate20211115T185333',
    @certificate_file_s3_arn='arn:aws:s3:::TDE_certs/mycertfile.cer',
```

```
    @private_key_file_s3_arn='arn:aws:s3:::TDE_certs/mykeyfile.pvk',
    @kms_password_key_arn='arn:aws:kms:us-
west-2:123456789012:key/AKIAIOSFODNN7EXAMPLE',
    @overwrite_s3_files=1;
```

Restaurar um certificado TDE

Use o procedimento armazenado `rds_restore_tde_certificate` para restaurar (importar) certificados TDE do usuário. Ele tem a sintaxe a seguir.

```
EXECUTE msdb.dbo.rds_restore_tde_certificate
    @certificate_name='UserTDECertificate_certificate_name',
    @certificate_file_s3_arn='arn:aws:s3:::bucket_name/certificate_file_name.cer',
    @private_key_file_s3_arn='arn:aws:s3:::bucket_name/key_file_name.pvk',
    @kms_password_key_arn='arn:aws:kms:region:account-id:key/key-id';
```

Os seguintes parâmetros são obrigatórios:

- `@certificate_name`: o nome do certificado TDE a ser restaurado. O nome deve iniciar com o prefixo `UserTDECertificate_`.
- `@certificate_file_s3_arn`: o ARN do S3 do arquivo de backup usado para restaurar o certificado TDE.
- `@private_key_file_s3_arn`: o ARN do S3 do arquivo de backup da chave privada do certificado TDE a ser restaurado.
- `@kms_password_key_arn`: o ARN da chave do KMS simétrica usada para criptografar a senha da chave privada.

Example Restaurar um certificado TDE

```
EXECUTE msdb.dbo.rds_restore_tde_certificate
    @certificate_name='UserTDECertificate_myTDEcertificate',
    @certificate_file_s3_arn='arn:aws:s3:::TDE_certs/mycertfile.cer',
    @private_key_file_s3_arn='arn:aws:s3:::TDE_certs/mykeyfile.pvk',
    @kms_password_key_arn='arn:aws:kms:us-
west-2:123456789012:key/AKIAIOSFODNN7EXAMPLE';
```

Visualizar certificados TDE restaurados

Use a função `rds_fn_list_user_tde_certificates` para visualizar certificados TDE do usuário restaurados (importados). Ele tem a sintaxe a seguir.

```
SELECT * FROM msdb.dbo.rds_fn_list_user_tde_certificates();
```

A saída será semelhante à seguinte. Nem todas as colunas são exibidas aqui.

name	certif te_id	princi _id	pvt_ke ncrypt _type_ c	issuer me	cert_s al_num	thumbp t	subjec	start_ e	expiry te	pvt_key_l ast_backu p_date
UserTD rtific _tde_c	343	1	ENCRYP _BY_MA R_KEY	AnyCom y Shippi	79 3e 57 a3 69 fd 1d 9e 47 2c 32 67 1d 9c ca af	0x6BB2 341103 80B FE1BA2 C69509 5B5	AnyCom y Shippi	2022-0 5 19:49: 000000	2023-0 5 19:49: 000000	NULL

Descartar certificados TDE restaurados

Para descartar certificados TDE do usuário restaurados (importados) que você não está usando, utilize o procedimento armazenado `rds_drop_tde_certificate`. Ele tem a sintaxe a seguir.

```
EXECUTE msdb.dbo.rds_drop_tde_certificate
 @certificate_name='UserTDECertificate_certificate_name';
```

O seguinte parâmetro é obrigatório:

- `@certificate_name`: o nome do certificado TDE a ser descartado.

Você só pode descartar certificados TDE restaurados (importados). Você não pode descartar certificados criados pelo RDS.

Example Descartar um certificado TDE

```
EXECUTE msdb.dbo.rds_drop_tde_certificate
 @certificate_name='UserTDECertificate_myTDEcertificate';
```

Fazer backup e restaurar certificados TDE para bancos de dados on-premises

Você pode fazer backup de certificados TDE para bancos de dados on-premises e depois restaurá-los para o RDS para SQL Server. Você também pode restaurar um certificado TDE do RDS para SQL Server para uma instância de banco de dados on-premises.

O procedimento a seguir faz backup de um certificado TDE e uma chave privada. A chave privada é criptografada usando uma chave de dados gerada a partir de sua chave do KMS de criptografia simétrica.

Para fazer backup de um certificado TDE on-premises

1. Gere a chave de dados usando o comando generate-data-key da AWS CLI.

   ```
   aws kms generate-data-key \
       --key-id my_KMS_key_ID \
       --key-spec AES_256
   ```

   A saída será semelhante à seguinte.

   ```
   {
   "CiphertextBlob": "AQIDAHimL2NEoAlOY6Bn7LJfnxi/OZe9kTQo/
   XQXduug1rmerwGiL7g5ux4av9GfZLxYTDATAAAAfjB8BgkqhkiG9w0B
   BwagbzBtAgEAMGgGCSqGSIb3DQEHATAeBglghkgBZQMEAS4wEQQMyCxLMi7GRZgKqD65AgEQgDtjvZLJo2cQ31Vetng
   ```

```
2RezQy3sAS6ZHrCjfnfn0c65bFdhsXxjSMnudIY7AKw==",
"Plaintext": "U/fpGtmzGCYBi8A2+0/9qcRQRK2zmG/aOn939ZnKi/0=",
"KeyId": "arn:aws:kms:us-west-2:123456789012:key/1234abcd-00ee-99ff-88dd-
aa11bb22cc33"
}
```

Use a saída de texto simples na próxima etapa como a senha da chave privada.

2. Faça backup do seu certificado TDE, conforme mostrado no exemplo a seguir.

```
BACKUP CERTIFICATE myOnPremTDEcertificate TO FILE = 'D:\tde-cert-backup.cer'
WITH PRIVATE KEY (
FILE = 'C:\Program Files\Microsoft SQL Server\MSSQL14.MSSQLSERVER\MSSQL\DATA\cert-
backup-key.pvk',
ENCRYPTION BY PASSWORD = 'U/fpGtmzGCYBi8A2+0/9qcRQRK2zmG/aOn939ZnKi/0=');
```

3. Salve o arquivo de backup do certificado no bucket de certificados do Amazon S3.
4. Salve o arquivo de backup de chave privada no bucket de certificados do S3, com a seguinte tag nos metadados do arquivo:

   - Chave: `x-amz-meta-rds-tde-pwd`
   - Valor: o valor `CiphertextBlob` da geração da chave de dados, como no exemplo a seguir.

```
AQIDAHimL2NEoAlOY6Bn7LJfnxi/OZe9kTQo/
XQXduug1rmerwGiL7g5ux4av9GfZLxYTDATAAAAfjB8BgkqhkiG9w0B
BwagbzBtAgEAMGgGCSqGSIb3DQEHATAeBglghkgBZQMEAS4wEQQMyCxLMi7GRZgKqD65AgEQgDtjvZLJo2cQ31Vet
2RezQy3sAS6ZHrCjfnfn0c65bFdhsXxjSMnudIY7AKw==
```

O procedimento a seguir restaura um certificado TDE do RDS para SQL Server para uma instância de banco de dados on-premises. Copie e restaure o certificado TDE na instância de banco de dados de destino usando o backup de certificado, o arquivo de chave privada correspondente e a chave de dados. O certificado restaurado é criptografado pela chave mestra do banco de dados do novo servidor.

Como restaurar um certificado TDE

1. Copie o arquivo de backup do certificado TDE e o arquivo de chave privada do Amazon S3 para a instância de destino. Para ter mais informações sobre como copiar arquivos do Amazon S3, consulte Transferência de arquivos entre o RDS for SQL Server e o Amazon S3.

2. Use sua chave do KMS para descriptografar o texto de criptografia de saída para recuperar o texto simples da chave de dados. O texto de criptografia está localizado nos metadados do S3 do arquivo de backup de chave privada.

```
aws kms decrypt \
    --key-id my_KMS_key_ID \
    --ciphertext-blob fileb://exampleCiphertextFile | base64 -d \
    --output text \
    --query Plaintext
```

   Use a saída de texto simples na próxima etapa como a senha da chave privada.

3. Use o comando SQL a seguir para restaurar seu certificado TDE.

```
CREATE CERTIFICATE myOnPremTDEcertificate FROM FILE='D:\tde-cert-backup.cer'
WITH PRIVATE KEY (FILE = N'D:\tde-cert-key.pvk',
DECRYPTION BY PASSWORD = 'plain_text_output');
```

Para ter mais informações sobre descriptografia do KMS, consulte descriptografar na seção do KMS da Referência de comando AWS CLI.

Depois que o certificado TDE for restaurado na instância de banco de dados de destino, você poderá restaurar bancos de dados criptografados com esse certificado.

> **Note**
>
> Você pode usar o mesmo certificado TDE para criptografar vários bancos de dados do SQL Server na instância de banco de dados de origem. Para migrar vários bancos de dados para uma instância de destino, copie o certificado TDE associado a eles na instância de destino apenas uma vez.

Desativar o TDE para o RDS para SQL Server

Para desativar o TDE em uma instância de banco de dados do RDS para SQL Server, primeiro, verifique se não há objetos criptografados na instância de banco de dados. Para fazer isso, descriptografe os objetos ou descarte-os. Se houver algum objeto criptografado na instância de banco de dados, você não poderá desativar o TDE para a instância de banco de dados. Ao usar o console para remover a opção TDE de um grupo de opções, o console indica que ele está

processando. Além disso, um evento de erro será criado se o grupo de opções estiver associado a uma instância de banco de dados criptografada ou a um DB snapshot.

O exemplo a seguir remove a criptografia de TDE de um banco de dados chamado `customerDatabase`.

```
------------- Removing TDE ----------------

USE [customerDatabase]
GO

-- Turn off encryption of the database
ALTER DATABASE [customerDatabase]
SET ENCRYPTION OFF
GO

-- Wait until the encryption state of the database becomes 1. The state is 5
 (Decryption in progress) for a while
SELECT db_name(database_id) as DatabaseName, * FROM sys.dm_database_encryption_keys
GO

-- Drop the DEK used for encryption
DROP DATABASE ENCRYPTION KEY
GO

-- Alter to SIMPLE Recovery mode so that your encrypted log gets truncated
USE [master]
GO
ALTER DATABASE [customerDatabase] SET RECOVERY SIMPLE
GO
```

Quando todos os objetos forem descriptografados, você terá duas opções:

1. Você poderá modificar a instância de banco de dados para ser associada a um grupo de opções sem a opção TDE.
2. Você pode remover a opção TDE do grupo de opções.

Auditoria do SQL Server

No Amazon RDS, você pode auditar bancos de dados do Microsoft SQL Server usando o mecanismo de auditoria interno do SQL Server. É possível criar auditorias e especificações de auditoria da mesma forma como elas são criadas para servidores de banco de dados no local.

O RDS faz upload dos logs de auditoria concluídos no bucket do S3 usando a função do IAM que você fornece. Se você habilitar a retenção, o RDS manterá seus logs de auditoria na sua instância de banco de dados durante o período de tempo configurado.

Para obter mais informações, consulte Auditoria do SQL Server (Mecanismo de banco de dados) na documentação do Microsoft SQL Server.

Auditoria do SQL Server com o recurso Database Activity Streams

Você pode usar o recurso Database Activity Streams para RDS a fim de integrar eventos de auditoria do SQL Server com ferramentas de monitoramento de atividade de banco de dados da Imperva, McAfee e IBM. Para obter mais informações sobre como realizar auditoria com o recurso Database Activity Streams para o RDS SQL Server, consulte Auditoria no Microsoft SQL Server

Tópicos

- Suporte para auditoria do SQL Server
- Adicionar a auditoria do SQL Server às opções de instância de banco de dados
- Usar a auditoria do SQL Server
- Visualizar logs de auditoria
- Usar a Auditoria do SQL Server com instâncias Multi-AZ
- Configurar um bucket do S3
- Criar manualmente uma função do IAM para a Auditoria do SQL Server

Suporte para auditoria do SQL Server

No Amazon RDS, a partir do SQL Server 2016, todas as edições do SQL Server comportam auditorias em nível de servidor, e a edição Enterprise também aceita auditorias em nível de banco de dados. Começando com o SQL Server 2016 (13.x) SP1, todas as edições oferecem suporte a auditorias em nível de servidor e em banco de dados. Para obter mais informações, consulte Auditoria do SQL Server (Mecanismo de banco de dados) na documentação do SQL Server.

O RDS oferece suporte para a definição das seguintes configurações de opção para o SQL Server Audit.

Configuração da opção	Valores válidos	Descrição
`IAM_ROLE_ARN`	Um nome de recurso da Amazon (ARN) válido no formato `arn:aws:iam::` *`account-id`* `:role/`*`role-name`* .	O ARN da função do IAM que concede acesso ao bucket do S3 no qual você deseja armazenar seus logs de auditoria. Para obter mais informações, consulte Nomes de recurso da Amazon (ARNs) na Referência geral da AWS.
`S3_BUCKET_ARN`	Um ARN válido no formato `arn:aws:s3:::` *`amzn-s3-demo-bucket`* ou `arn:aws:s3:::` *`amzn-s3-demo-bucket`* `/key-prefix`	O ARN do bucket do S3 no qual você deseja armazenar seus logs de auditoria.
`ENABLE_COMPRESSION`	`true` ou `false`	Controla a compactação de logs de auditoria. Por padrão, a compactação está habilitada (definida como `true`).
`RETENTION_TIME`	`0` para `840`	O tempo de retenção (em horas) durante o qual os registros de auditoria do SQL Server são mantidos em sua instância do RDS. Por padrão, a retenção está desabilitada.

Adicionar a auditoria do SQL Server às opções de instância de banco de dados

Habilitar a Auditoria do SQL Server requer duas etapas: habilitar a opção na instância de banco de dados e habilitar o recurso no SQL Server. O processo para adicionar a opção de Auditoria do SQL Server a uma instância de banco de dados é o seguinte:

1. Crie um novo grupo de opções, ou copie ou modifique um existente.
2. Adicione e configure todas as opções necessárias.
3. Associe o grupo de opções à instância de banco de dados.

Depois de adicionar a opção de Auditoria do SQL Server, você não precisará reiniciar sua instância de banco de dados. Assim que o grupo de opções estiver ativo, você poderá criar auditorias e armazenar logs de auditoria no seu bucket do S3.

Para adicionar e configurar a Auditoria do SQL Server em um grupo de opções da instância de banco de dados

1. Escolha uma das seguintes opções:
   - Use um grupo de opções existente.
   - Crie um grupo de opções de banco de dados personalizado e use esse grupo de opções. Para obter mais informações, consulte Criar um grupo de opções.
2. Adicione a opção SQLSERVER_AUDIT ao grupo de opções e defina as configurações da opção. Para mais informações sobre a adição de opções, consulte Adicionar uma opção a um grupo de opções.
   - Em IAM role (Função do IAM), se você já tiver uma função do IAM com as políticas necessárias, poderá escolher essa função. Para criar uma nova função do IAM, escolha Create a New Role (Criar uma nova função). Para obter informações sobre as políticas necessárias, consulte Criar manualmente uma função do IAM para a Auditoria do SQL Server.
   - Em Select S3 destination (Selecionar destino do S3), se você já tiver um bucket do S3 que deseja usar, escolha-o. Para criar um bucket do S3, escolha Create a New S3 Bucket (Criar um bucket do S3).
   - Em Enable Compression (Habilitar compactação), deixe essa opção marcada para compactar arquivos de auditoria. A compactação está habilitada por padrão. Para desabilitar a compactação, desmarque Enable Compression (Habilitar compactação).

- Em Audit log retention (Retenção do log de auditoria), para manter registros de auditoria na instância de banco de dados, escolha essa opção. Especifique um tempo de retenção em horas. O tempo máximo de retenção é de 35 dias.

3. Aplique o grupo de opções a uma instância de banco de dados nova ou existente. Escolha uma das seguintes opções:

   - Se você estiver criando uma nova instância de banco de dados, aplique o grupo de opções ao executar a instância.
   - Em uma instância de banco de dados existente, aplique o grupo de opções modificando a instância e anexando o novo grupo de opções. Para obter mais informações, consulte Modificar uma instância de banco de dados do Amazon RDS.

Modificar a opção de auditoria do SQL Server

Depois de habilitar a opção de Auditoria do SQL Server, você poderá modificar as configurações. Para obter informações sobre como modificar as configurações da opção, consulte Modificar uma configuração de opção.

Remover a Auditoria do SQL Server das opções de instância de banco de dados

Você pode desativar o recurso de Auditoria do SQL Server desabilitando auditorias e, em seguida, excluindo a opção.

Para remover a auditoria

1. Desabilite todas as configurações de auditoria no SQL Server. Para saber onde as auditorias estão sendo executadas, consulte as visualizações do catálogo de segurança do SQL Server. Para obter mais informações, consulte Visualizações do catálogo de segurança na documentação do Microsoft SQL Server.
2. Exclua a opção de Auditoria do SQL Server da instância de banco de dados. Escolha uma das seguintes opções:

   - Exclua a opção de Auditoria do SQL Server do grupo de opções usado pela instância de banco de dados. Essa alteração afeta todas as instâncias de bancos de dados que usam o mesmo grupo de opções. Para obter mais informações, consulte Remover uma opção de um grupo de opções.
   - Modifique a instância de banco de dados e escolha um grupo de opções sem a opção de Auditoria do SQL Server. Essa alteração afeta somente a instância de banco de dados que

você modifica. Você pode especificar um grupo de opções padrão (vazio) ou criar um grupo de opções personalizado diferente. Para obter mais informações, consulte Modificar uma instância de banco de dados do Amazon RDS.

3. Após a exclusão da opção de Auditoria do SQL Server da instância de banco de dados, não é necessário reiniciar a instância. Remova arquivos de auditoria desnecessários do seu bucket do S3.

Usar a auditoria do SQL Server

Você pode controlar auditorias de servidor, especificações de auditorias de servidor e especificações de auditoria de banco de dados da mesma maneira que você as controla para servidores de banco de dados no local.

Criar auditorias

Você cria auditorias de servidor da mesma maneira que as cria para servidores de banco de dados no local. Para obter informações sobre como criar auditorias de servidor, consulte CRIAR AUDITORIA DE SERVIDOR na documentação do Microsoft SQL Server.

Para evitar erros, siga as seguintes limitações:

- Não exceda o número máximo de auditorias de servidor com suporte por instância de 50.
- Instrua o SQL Server a gravar dados em um arquivo binário.
- Não use `RDS_` como um prefixo no nome de auditoria do servidor.
- Para `FILEPATH`, especifique `D:\rdsdbdata\SQLAudit`.
- Para `MAXSIZE`, especifique um tamanho entre 2 MB e 50 MB.
- Não configure `MAX_ROLLOVER_FILES` ou `MAX_FILES`.
- Não configure o SQL Server para encerrar a instância de banco de dados se ele não conseguir gravar o registro de auditoria.

Criar especificações de auditoria

Você cria especificações de auditoria de servidor e especificações de auditoria de banco de dados da mesma maneira que as cria para servidores de banco de dados no local. Para obter informações sobre como criar especificações de auditoria, consulte CRIAR ESPECIFICAÇÃO DE AUDITORIA DE SERVIDOR e CRIAR ESPECIFICAÇÃO DE AUDITORIA DE BANCO DE DADOS na documentação do Microsoft SQL Server.

Para evitar erros, não use `RDS_` como um prefixo no nome da especificação de auditoria de banco de dados ou da especificação de auditoria de servidor.

Visualizar logs de auditoria

Seus logs de auditoria são armazenados em `D:\rdsdbdata\SQLAudit`.

Depois que o SQL Server terminar de gravar em um arquivo de log de auditoria—quando o arquivo atingir seu limite de tamanho—o Amazon RDS o carregará no seu bucket S3. Se a retenção estiver habilitada, o Amazon RDS moverá o arquivo para a pasta de retenção: `D:\rdsdbdata\SQLAudit\transmitted`.

Para obter informações sobre como configurar a retenção, consulte Adicionar a auditoria do SQL Server às opções de instância de banco de dados.

Registros de auditoria são mantidos na instância de banco de dados até que o arquivo de log de auditoria seja carregado. Você pode visualizar os registros de auditoria executando o comando a seguir.

```
SELECT   *
 FROM     msdb.dbo.rds_fn_get_audit_file
             ('D:\rdsdbdata\SQLAudit\*.sqlaudit'
             , default
             , default )
```

É possível usar o mesmo comando para visualizar registros de auditoria na sua pasta de retenção, alterando o filtro para `D:\rdsdbdata\SQLAudit\transmitted\*.sqlaudit`.

```
SELECT   *
 FROM     msdb.dbo.rds_fn_get_audit_file
             ('D:\rdsdbdata\SQLAudit\transmitted\*.sqlaudit'
             , default
             , default )
```

Usar a Auditoria do SQL Server com instâncias Multi-AZ

Para instâncias Multi-AZ, o processo de envio de arquivos de log de auditoria ao Amazon S3 é semelhante ao processo para instâncias Single-AZ. No entanto, existem algumas diferenças importantes:

- Objetos de especificação de auditoria de banco de dados são replicados para todos os nós.
- Auditorias de servidor e especificações de auditoria de servidor não são replicadas para nós secundários. Em vez disso, você precisa criá-las ou modificá-las manualmente.

Para capturar auditorias de servidor ou uma especificação de auditoria de servidor de ambos os nós:

1. Crie uma auditoria de servidor ou uma especificação de auditoria de servidor no nó primário.
2. Faça failover para o nó secundário e crie uma auditoria de servidor ou uma especificação de auditoria de servidor com o mesmo nome e o mesmo GUID no nó secundário. Use o parâmetro `AUDIT_GUID` para especificar o GUID.

Configurar um bucket do S3

Os arquivos de log de auditoria são automaticamente carregados da instância de banco de dados no seu bucket do S3. As seguintes restrições se aplicam ao bucket do S3 que você usa como destino para arquivos de auditoria:

- Ele deve estar na mesma região da AWS que a instância de banco de dados.
- Ele não deve estar aberto ao público.
- O proprietário do bucket também deve ser o proprietário da função do IAM.
- Seu perfil do IAM deve ter permissões para a chave do KMS gerenciada pelo cliente associada à criptografia do lado do servidor do bucket do S3.

A chave de destino usada para armazenar os dados segue este esquema de nomenclatura: *`amzn-s3-demo-bucket`*`/key-prefix/instance-name/audit-name/node_file-name.ext`

> **Note**
>
> Você define o nome do intervalo e os valores do prefixo da chave com a configuração de opção (`S3_BUCKET_ARN`).

O esquema é composto pelos seguintes elementos:

- ***`amzn-s3-demo-bucket`***: o nome do bucket do S3.
- **`key-prefix`** – O prefixo de chave personalizada que você deseja usar para logs de auditoria.

- **`instance-name`** – O nome da sua instância do Amazon RDS.
- **`audit-name`** – O nome da auditoria.
- **`node`** – O identificador do nó que é a origem dos logs de auditoria (`node1` ou `node2`). Há um nó para uma instância Single-AZ e dois nós de replicação para uma instância Multi-AZ. Estes não são nós primários e secundários, porque as funções de primário e secundário mudam ao longo do tempo. Em vez disso, o identificador de nó é um rótulo simples.
  - **`node1`** – O primeiro nó de replicação (Single-AZ tem apenas um nó).
  - **`node2`** – O segundo nó de replicação (Multi-AZ tem dois nós).
- **`file-name`** – O nome do arquivo de destino. O nome do arquivo é obtido como está no SQL Server.
- **`ext`** – A extensão do arquivo (`zip` ou `sqlaudit`):
  - **`zip`** – Se a compactação estiver habilitada (padrão).
  - **`sqlaudit`** – Se a compactação estiver desabilitada.

Criar manualmente uma função do IAM para a Auditoria do SQL Server

Normalmente, quando você cria uma nova opção, o AWS Management Console cria a função do IAM e a política de confiança do IAM para você. No entanto, é possível criar manualmente uma nova função do IAM para uso com Auditorias do SQL Server, para poder personalizá-la com quaisquer requisitos adicionais que você possa ter. Para fazer isso, você cria uma função do IAM e delega permissões para que o serviço do Amazon RDS possa usar seu bucket do Amazon S3. Ao criar essa função do IAM, você anexa as políticas de confiança e permissões. A política de confiança permite que o Amazon RDS assuma essa função. A política de permissão define as ações que essa função pode realizar. Para obter mais informações, consulte Criar uma função para delegar permissões a um produto da AWS no Guia do usuário do AWS Identity and Access Management.

Você pode usar os exemplos nesta seção para criar as relações de confiança e as políticas de permissões necessárias.

O exemplo a seguir mostra uma relação de confiança para o SQL Server Audit. Ela usa o principal de serviço `rds.amazonaws.com` para permitir que o RDS grave no bucket do S3. Um escopo principal do serviço é um identificador que é usado para conceder permissões a um serviço. Sempre que você permitir acesso ao `rds.amazonaws.com` dessa forma, está permitindo que o RDS execute uma ação em seu nome. Para obter mais informações sobre principais de serviço, consulte Elementos da política JSON da AWS: principal.

Example relação de confiança para SQL Server Audit

JSON

```
{
     "Version": "2012-10-17",
     "Statement": [
         {
             "Effect": "Allow",
             "Principal": {
                 "Service": "rds.amazonaws.com"
             },
             "Action": "sts:AssumeRole"
         }
     ]
 }
```

Recomendamos usar as chaves de contexto de condição global `aws:SourceArn` e `aws:SourceAccount` em relações baseadas em recursos para limitar as permissões do serviço a um recurso específico. Essa é a maneira mais eficiente de se proteger contra o problema "confused deputy".

Você pode usar as duas chaves de contexto de condição global e fazer com que o valor `aws:SourceArn` contenha o ID da conta. Nesses casos, verifique se o valor `aws:SourceAccount` e a conta no `aws:SourceArn` usa o mesmo ID de conta quando eles são usados na mesma instrução.

- Use `aws:SourceArn` se quiser acesso entre serviços para um único recurso.
- Use `aws:SourceAccount` se você quiser permitir que qualquer recurso nessa conta seja associado ao uso entre serviços.

Na relação de confiança, certifique-se de usar a chave de contexto de condição global `aws:SourceArn` com o nome do recurso da Amazon (ARN) completo dos recursos que acessam a função. Para o SQL Server Audit, certifique-se de incluir o grupo de opções de banco de dados e as instâncias de banco de dados, conforme mostrado no exemplo a seguir.

Example relação de confiança com a chave de contexto de condição global para SQL Server Audit

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "rds.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceArn": [

 "arn:aws:rds:Region:my_account_ID:db:db_instance_identifier",
                        "arn:aws:rds:Region:my_account_ID:og:option_group_name"
                    ]
                }
            }
        }
    ]
}
```

No exemplo a seguir de uma política de permissões para SQL Server Audit, especificamos um ARN para o bucket do Simple Storage Service (Amazon S3). Você pode usar ARNs para identificar uma conta, usuário ou função específica à qual deseja conceder acesso. Para obter mais informações sobre como usar ARNs, consulte Nomes de recurso da Amazon (ARNs).

Example política de permissões para SQL Server Audit

JSON

```
{
     "Version": "2012-10-17",
     "Statement": [
         {
```

```
            "Effect": "Allow",
            "Action": "s3:ListAllMyBuckets",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListBucket",
                "s3:GetBucketACL",
                "s3:GetBucketLocation"
            ],
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:ListMultipartUploadParts",
                "s3:AbortMultipartUpload"
            ],
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/key_prefix/*"
        }
    ]
}
```

> **Note**
>
> A ação `s3:ListAllMyBuckets` é necessária para verificar se a mesma conta do AWS tem o bucket do S3 e a instância de banco de dados SQL Server. A ação lista os nomes dos buckets na conta.
>
> Os namespaces do bucket do S3 são globais. Se você excluir acidentalmente seu bucket, outro usuário poderá criar um bucket com o mesmo nome em uma conta diferente. Depois disso, os dados de auditoria do SQL Server são gravados no novo bucket.

Suporte para o SQL Server Analysis Services no Amazon RDS for SQL Server

O Microsoft SQL Server Analysis Services (SSAS) faz parte do pacote Microsoft Business Intelligence (MSBI). O SSAS é um processamento analítico online (OLAP) e uma ferramenta de mineração de dados que está instalado no SQL Server. Use o SSAS para analisar dados a fim de ajudar a tomar decisões de negócios. O SSAS difere do banco de dados relacional do SQL Server porque é otimizado para consultas e cálculos comuns em um ambiente de business intelligence.

É possível habilitar o SSAS em instâncias de banco de dados novas ou existentes. Ele é instalado na mesma instância de banco de dados que o mecanismo de banco de dados. Para obter mais informações sobre o SSAS, consulte a Documentação do Analysis Services da Microsoft.

O Amazon RDS oferece suporte às Standard e Enterprise Editions do SSAS for SQL Server nas seguintes versões:

- Modo tabular:
  - Versão 15.00.4043.16.v1 e posterior do SQL Server 2019
  - Versão 14.00.3223.3.v1 e posterior do SQL Server 2017
  - Versão 13.00.5426.0.v1 e posterior do SQL Server 2016
- Modo multidimensional:
  - SQL Server 2019, versão 15.00.4153.1.v1 e posterior
  - SQL Server 2017, versão 14.00.3381.3.v1 e posterior
  - SQL Server 2016, versão 13.00.5882.1.v1 e posterior

Sumário

- Limitações
- Ativar o SSAS
  - Criar um grupo de opções para o SSAS
  - Adicionar a opção SSAS ao grupo de opções
  - Associar o grupo de opções à instância de banco de dados
  - Como conceder acesso de entrada ao grupo de segurança da VPC
  - Habilitar a integração do Simple Storage Service (Simple Storage Service (Amazon S3))
- Implantar projetos do SSAS no Amazon RDS

- Monitorar o status de uma tarefa de implantação
- Como usar o SSAS no Amazon RDS
  - Configurar um usuário autenticado pelo Windows para o SSAS
  - Adicionar um usuário de domínio como administrador de banco de dados
  - Criar um proxy do SSAS
  - Programar o processamento de banco de dados do SSAS usando o SQL Server Agent
  - Revogar o acesso do SSAS do proxy
- Fazer backup de um banco de dados do SSAS
- Restaurar um banco de dados do SSAS
  - Restauração de uma instância de banco de dados para um tempo especificado
- Alterar o modo SSAS
- Desativar o SSAS
- Solução de problemas do SSAS

Limitações

As seguintes limitações se aplicam à utilização do SSAS no RDS for SQL Server:

- O RDS for SQL Server oferece suporte à execução do SSAS no modo Tabular ou Multidimensional. Para obter mais informações, consulte Comparar soluções tabulares e multidimensionais na documentação da Microsoft.
- É possível usar apenas um modo SSAS de cada vez. Antes de alterar os modos, certifique-se de excluir todos os bancos de dados SSAS.

  Para ter mais informações, consulte Alterar o modo SSAS.
- As instâncias Multi-AZ não são compatíveis.
- As instâncias devem usar o Active Directory autogerenciado ou o AWS Directory Service for Microsoft Active Directorypara autenticação SSAS. Para ter mais informações, consulte Trabalhar com o Active Directory com o RDS para SQL Server.
- Os usuários não recebem acesso de administrador do servidor do SSAS, mas podem receber acesso de administrador no nível de banco de dados.
- A única porta compatível para acessar o SSAS é a 2383.

- Não é possível implantar projetos diretamente. Nós fornecemos um procedimento armazenado do RDS para fazer isso. Para obter mais informações, consulte Implantar projetos do SSAS no Amazon RDS.
- O processamento durante a implantação não é compatível.
- O uso de arquivos .xmla para implantação não é compatível.
- Os arquivos de entrada de projeto e os arquivos de saída de backup de banco de dados do SSAS podem estar somente na pasta `D:\S3` na instância de banco de dados.

Ativar o SSAS

Use o seguinte processo para ativar o SSAS para a instância de banco de dados:

1. Crie um grupo de opções ou selecione um existente.
2. Adicione a opção `SSAS` ao grupo de opções.
3. Associe o grupo de opções à instância de banco de dados.
4. Permita acesso de entrada ao grupo de segurança da nuvem privada virtual (VPC) para a porta do ouvinte do SSRS.
5. Ative a integração do Simple Storage Service (Amazon S3).

Criar um grupo de opções para o SSAS

Use o AWS Management Console ou a AWS CLI para criar um grupo de opções que corresponda à versão e ao mecanismo do SQL Server da instância de banco de dados que você planeja usar.

> **Note**
>
> Também é possível usar um grupo de opções existente se ele for para a versão e o mecanismo e corretos do SQL Server.

Console

O procedimento do console a seguir cria um grupo de opções para o SQL Server Standard Edition 2017.

Como criar o grupo de opções

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Option groups (Grupos de opções).
3. Escolha Create group (Criar grupo).
4. No painel Create option group (Criar grupo de opções), faça o seguinte:
   a. Em Name (Nome), insira um nome para o grupo de opções que seja exclusivo na sua conta da AWS, como **`ssas-se-2017`**. O nome pode conter somente letras, números e hifens.
   b. Em Descrição, insira uma breve descrição do grupo de opções, como **`SSAS option group for SQL Server SE 2017`**. A descrição é usada para fins de exibição.
   c. Em Engine (Mecanismo), selecione sqlserver-se.
   d. Em Major engine version (Versão principal do mecanismo), selecione 14.00.
5. Escolha Criar.

CLI

O exemplo de CLI a seguir cria um grupo de opções para o SQL Server Standard Edition 2017.

Como criar o grupo de opções

- Use um dos seguintes comandos.

  Example

  Para Linux, macOS ou Unix:

```
aws rds create-option-group \
    --option-group-name ssas-se-2017 \
    --engine-name sqlserver-se \
    --major-engine-version 14.00 \
    --option-group-description "SSAS option group for SQL Server SE 2017"
```

  Para Windows:

```
aws rds create-option-group ^
    --option-group-name ssas-se-2017 ^
```

```
    --engine-name sqlserver-se ^
    --major-engine-version 14.00 ^
    --option-group-description "SSAS option group for SQL Server SE 2017"
```

Adicionar a opção SSAS ao grupo de opções

Depois, use o AWS Management Console ou a AWS CLI para adicionar a opção SSAS ao grupo de opções.

Console

Como adicionar a opção SSAS

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Option groups (Grupos de opções).
3. Selecione o grupo de opções que você acabou de criar.
4. Escolha Add option (Adicionar opção).
5. Em Option details (Detalhes da opção), selecione SSAS em Option name (Nome da opção).
6. Em Configurações de opção, faça o seguinte:

    a. Para Max memory (Memória máxima), insira um valor de 10 a 80.

    Max memory (Memória máxima) especifica o limite superior acima do qual o SSAS começa a liberar memória de forma mais agressiva para liberar espaço a solicitações que estão em execução e a novas solicitações de alta prioridade. O número é uma porcentagem da memória total da instância de banco de dados. Os valores permitidos são de 10–80 e o padrão é 45.

    b. Para Mode (Modo), escolha o modo de servidor SSAS, Tabular ou Multidimensional.

    Se você não vir a opção de configuração Mode (Modo), significa que o modo multidimensional não é compatível com sua região da AWS. Para ter mais informações, consulte Limitações.

    Tabular é o padrão.

    c. Em Security groups (Grupos de segurança), selecione o grupo de segurança da VPC a ser associado à opção.

> **Note**
>
> A porta para acessar o SSAS, 2383, é preenchida automaticamente.

7. Em Scheduling (Programação), escolha se deseja adicionar a opção imediatamente ou na próxima janela de manutenção.
8. Escolha Add option (Adicionar opção).

CLI

Como adicionar a opção SSAS

1. Crie um arquivo JSON, por exemplo `ssas-option.json`, com os seguintes parâmetros:

   - `OptionGroupName` – O nome do grupo de opções que você criou ou selecionou anteriormente (`ssas-se-2017` no exemplo a seguir).
   - `Port` – A porta que você usa para acessar o SSAS. A única porta compatível é a 2383.
   - `VpcSecurityGroupMemberships` – Associações a grupos de segurança da VPC para a instância de banco de dados do RDS.
   - `MAX_MEMORY` – O limite superior acima do qual o SSAS deve começar a liberar memória de forma mais agressiva para liberar espaço a solicitações que estão em execução e a novas solicitações de alta prioridade. O número é uma porcentagem da memória total da instância de banco de dados. Os valores permitidos são de 10–80 e o padrão é 45.
   - `MODE` – O modo de servidor SSAS, `Tabular` ou `Multidimensional`. `Tabular` é o padrão.

     Se você receber um erro dizendo que a opção de configuração `MODE` não é válida, isso significa que o modo multidimensional não é aceito em sua região da AWS. Para ter mais informações, consulte Limitações.

   Veja a seguir um exemplo de arquivo JSON com configurações de opção SSAS.

```
{
"OptionGroupName": "ssas-se-2017",
"OptionsToInclude": [
 {
 "OptionName": "SSAS",
```

```
 "Port": 2383,
 "VpcSecurityGroupMemberships": ["sg-0abcdef123"],
 "OptionSettings": [{"Name":"MAX_MEMORY","Value":"60"},
{"Name":"MODE","Value":"Multidimensional"}]
 }],
"ApplyImmediately": true
}
```

2. Adicione a opção SSAS ao grupo de opções.

   Example

   Para Linux, macOS ou Unix:

```
aws rds add-option-to-option-group \
    --cli-input-json file://ssas-option.json \
    --apply-immediately
```

   Para Windows:

```
aws rds add-option-to-option-group ^
    --cli-input-json file://ssas-option.json ^
    --apply-immediately
```

Associar o grupo de opções à instância de banco de dados

É possível usar o console ou a CLI para associar o grupo de opções à instância de banco de dados.

Console

Associe o grupo de opções a uma instância de banco de dados nova ou existente:

- Para uma nova instância de banco de dados, associe o grupo de opções à instância de banco de dados ao executar a instância. Para obter mais informações, consulte Criar uma instância de banco de dados do Amazon RDS.
- Para uma instância de banco de dados existente, modifique a instância e associe o novo grupo de opções a ela. Para ter mais informações, consulte Modificar uma instância de banco de dados do Amazon RDS.

> **Note**
>
> Se você usar uma instância existente, ela já deve ter um domínio do Active Directory e uma função do AWS Identity and Access Management (IAM) associados a ela. Se você criar uma nova instância, especifique um domínio do Active Directory e uma função do IAM existentes. Para ter mais informações, consulte Trabalhar com o Active Directory com o RDS para SQL Server.

CLI

É possível associar o grupo de opções a uma instância de banco de dados nova ou existente.

> **Note**
>
> Se você usar uma instância existente, ela já deve ter um domínio do Active Directory e uma função do IAM associados a ela. Se você criar uma nova instância, especifique um domínio do Active Directory e uma função do IAM existentes. Para ter mais informações, consulte Trabalhar com o Active Directory com o RDS para SQL Server.

Como criar uma instância de banco de dados que usa o grupo de opções

- Especifique o mesmo tipo de mecanismo de banco de dados e a versão principal que você usou ao criar o grupo de opções.

  Example

  Para Linux, macOS ou Unix:

```
aws rds create-db-instance \
    --db-instance-identifier myssasinstance \
    --db-instance-class db.m5.2xlarge \
    --engine sqlserver-se \
    --engine-version 14.00.3223.3.v1 \
    --allocated-storage 100 \
    --manage-master-user-password \
    --master-username admin \
    --storage-type gp2 \
    --license-model li \
```

```
    --domain-iam-role-name my-directory-iam-role \
    --domain my-domain-id \
    --option-group-name ssas-se-2017
```

Para Windows:

```
aws rds create-db-instance ^
    --db-instance-identifier myssasinstance ^
    --db-instance-class db.m5.2xlarge ^
    --engine sqlserver-se ^
    --engine-version 14.00.3223.3.v1 ^
    --allocated-storage 100 ^
    --manage-master-user-password ^
    --master-username admin ^
    --storage-type gp2 ^
    --license-model li ^
    --domain-iam-role-name my-directory-iam-role ^
    --domain my-domain-id ^
    --option-group-name ssas-se-2017
```

Como modificar uma instância de banco de dados para associar o grupo de opções

- Use um dos seguintes comandos.

  Example

  Para Linux, macOS ou Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier myssasinstance \
    --option-group-name ssas-se-2017 \
    --apply-immediately
```

  Para Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier myssasinstance ^
    --option-group-name ssas-se-2017 ^
    --apply-immediately
```

Como conceder acesso de entrada ao grupo de segurança da VPC

Crie uma regra de entrada para a porta de listener do SSAS especificada no grupo de segurança da VPC associado à instância de banco de dados. Para obter mais informações sobre como configurar um grupo de segurança, consulte Fornecer acesso à instância de banco de dados na VPC criando um grupo de segurança.

Habilitar a integração do Simple Storage Service (Simple Storage Service (Amazon S3))

Para baixar arquivos de configuração de modelo para seu host para implantação, use a integração do Simple Storage Service (Simple Storage Service (Amazon S3)). Para ter mais informações, consulte Integrar uma instância de banco de dados do Amazon RDS for SQL Server ao Amazon S3.

Implantar projetos do SSAS no Amazon RDS

No RDS, não é possível implantar projetos do SSAS diretamente usando o SQL Server Management Studio (SSMS). Para implantar projetos, use um procedimento armazenado do RDS.

> **Note**
>
> O uso de arquivos .xmla para implantação não é compatível.

Antes de implantar projetos, certifique-se do seguinte:

- A integração do Simple Storage Service (Simple Storage Service (Amazon S3)) está habilitada. Para ter mais informações, consulte Integrar uma instância de banco de dados do Amazon RDS for SQL Server ao Amazon S3.
- A configuração `Processing Option` está definida como `Do Not Process`. Essa configuração significa que nenhum processamento ocorrerá após a implantação.
- Você tem os arquivos *`myssasproject`*`.asdatabase` e *`myssasproject`*`.deploymentoptions`. Eles são gerados automaticamente quando você cria o projeto do SSAS.

Como implantar um projeto do SSAS no RDS

1. Baixe o arquivo `.asdatabase` (modelo do SSAS) no bucket do S3 para a instância de banco de dados, conforme mostrado no exemplo a seguir. Para obter mais informações sobre os

parâmetros de download, consulte Baixar arquivos de um bucket do Amazon S3 em uma instância de banco de dados SQL Server.

```
exec msdb.dbo.rds_download_from_s3
@s3_arn_of_file='arn:aws:s3:::bucket_name/myssasproject.asdatabase',
[@rds_file_path='D:\S3\myssasproject.asdatabase'],
[@overwrite_file=1];
```

2. Baixe o arquivo `.deploymentoptions` no bucket do S3 para a instância de banco de dados.

```
exec msdb.dbo.rds_download_from_s3
@s3_arn_of_file='arn:aws:s3:::bucket_name/myssasproject.deploymentoptions',
[@rds_file_path='D:\S3\myssasproject.deploymentoptions'],
[@overwrite_file=1];
```

3. Implante o projeto.

```
exec msdb.dbo.rds_msbi_task
@task_type='SSAS_DEPLOY_PROJECT',
@file_path='D:\S3\myssasproject.asdatabase';
```

Monitorar o status de uma tarefa de implantação

Para rastrear o status da tarefa de implantação (ou de download), chame a função `rds_fn_task_status`. Ela leva dois parâmetros. O primeiro parâmetro sempre deve ser `NULL` porque ele não se aplica ao SSAS. O segundo parâmetro aceita um ID de tarefa.

Para ver uma lista de todas as tarefas, defina o primeiro parâmetro como `NULL` e o segundo parâmetro como `0`, conforme mostrado no seguinte exemplo.

```
SELECT * FROM msdb.dbo.rds_fn_task_status(NULL,0);
```

Para obter uma tarefa específica, defina o primeiro parâmetro como `NULL` e o segundo parâmetro como o ID da tarefa, conforme mostrado no exemplo a seguir.

```
SELECT * FROM msdb.dbo.rds_fn_task_status(NULL,42);
```

A função `rds_fn_task_status` retorna as informações a seguir.

Parâmetro de saída	Descrição
`task_id`	O ID da tarefa.
`task_type`	No SSAS, as tarefas podem ter os seguintes tipos de tarefa: • `SSAS_DEPLOY_PROJECT` • `SSAS_ADD_DB_ADMIN_MEMBER` • `SSAS_BACKUP_DB` • `SSAS_RESTORE_DB`
`database_name`	Não aplicável a tarefas do SSAS.
`% complete`	O progresso da tarefa em porcentagem.
`duration (mins)`	A quantidade de tempo gasta na tarefa, em minutos.
`lifecycle`	O status da tarefa. Os possíveis status são os seguintes: • `CREATED` – após chamar um dos procedimentos armazenados do SSAS, uma tarefa será criada, e o status será definido como `CREATED`. • `IN_PROGRESS` – Após iniciar uma tarefa, o status será definido como `IN_PROGRESS`. Pode demorar até 5 minutos para que o status mude de `CREATED` para `IN_PROGRESS`. • `SUCCESS` – Após concluir uma tarefa, o status será definido como `SUCCESS`. •

Parâmetro de saída	Descrição
	`ERROR` – Se uma tarefa falhar, o status será definido como `ERROR`. Para obter mais informações sobre o erro, consulte a coluna `task_info` . • `CANCEL_REQUESTED` – Após chamar `rds_cancel_task` , o status da tarefa será definido como `CANCEL_REQUESTED` . • `CANCELLED` – após uma tarefa ter sido cancelada com sucesso, o status da tarefa será definido como `CANCELLED` .
`task_info`	Informações adicionais sobre a tarefa. Se um erro ocorrer durante o processamento, essa coluna conterá informações sobre o erro. Para ter mais informações, consulte Solução de problemas do SSAS.
`last_updated`	A data e hora em que o status da tarefa foi atualizado pela última vez.
`created_at`	A data e hora em que a tarefa foi criada.
`S3_object_arn`	Não aplicável a tarefas do SSAS.
`overwrite_S3_backup_file`	Não aplicável a tarefas do SSAS.
`KMS_master_key_arn`	Não aplicável a tarefas do SSAS.
`filepath`	Não aplicável a tarefas do SSAS.
`overwrite_file`	Não aplicável a tarefas do SSAS.

Parâmetro de saída	Descrição
`task_metadata`	Metadados associados à tarefa do SSAS.

Como usar o SSAS no Amazon RDS

Depois de implantar o projeto do SSAS, será possível processar diretamente o banco de dados OLAP no SSMS.

Como usar o SSAS no RDS

1. No SSMS, conecte-se ao SSAS usando o nome de usuário e a senha do domínio do Active Directory.
2. Expanda Databases (Bancos de dados). Será exibido o banco de dados recém-implantado do SSAS.
3. Localize a string de conexão e atualize o nome de usuário e a senha para dar acesso ao banco de dados da fonte do SQL. Isso é necessário para processar objetos SSAS.
    a. Para o modo Tabular, faça o seguinte:
        1. Escolha a guia Connections (Conexões).
        2. Abra o menu de contexto (clique com o botão direito do mouse) do objeto de conexão e escolha Properties (Propriedades).
        3. Atualize o nome de usuário e a senha na string de conexão.
    b. Para o modo Multidimensional, faça o seguinte:
        1. Expanda a guia Data Sources (Origem dos dados).
        2. Abra o menu de contexto (clique com o botão direito do mouse) para o objeto de origem dos dados e escolha Properties (Propriedades).
        3. Atualize o nome de usuário e a senha na string de conexão.
4. Abra o menu de contexto (clique com o botão direito do mouse) para o banco de dados do SSAS que você criou e selecione Process Database (Processar banco de dados).

    Dependendo do tamanho dos dados de entrada, a operação de processamento pode levar vários minutos para ser concluída.

Tópicos

- Configurar um usuário autenticado pelo Windows para o SSAS
- Adicionar um usuário de domínio como administrador de banco de dados
- Criar um proxy do SSAS
- Programar o processamento de banco de dados do SSAS usando o SQL Server Agent
- Revogar o acesso do SSAS do proxy

Configurar um usuário autenticado pelo Windows para o SSAS

O usuário administrador principal (às vezes chamado de usuário primário) pode usar o exemplo de código a seguir para configurar um logon autenticado pelo Windows e conceder as permissões de procedimento necessárias. Isso concede permissões ao usuário do domínio para executar tarefas do cliente SSAS, usar procedimentos de transferência de arquivos do S3, criar credenciais e trabalhar com o proxy do SQL Server Agent. Para obter mais informações, consulte Credenciais (mecanismo de banco de dados) e Criar um proxy do SQL Server Agent na documentação da Microsoft.

Você pode conceder algumas ou todas as seguintes permissões, conforme necessário, aos usuários autenticados pelo Windows.

Example

```
-- Create a server-level domain user login, if it doesn't already exist
USE [master]
GO
CREATE LOGIN [mydomain\user_name] FROM WINDOWS
GO

-- Create domain user, if it doesn't already exist
USE [msdb]
GO
CREATE USER [mydomain\user_name] FOR LOGIN [mydomain\user_name]
GO

-- Grant necessary privileges to the domain user
USE [master]
GO
GRANT ALTER ANY CREDENTIAL TO [mydomain\user_name]
GO

USE [msdb]
```

```
GO
GRANT EXEC ON msdb.dbo.rds_msbi_task TO [mydomain\user_name] with grant option
GRANT SELECT ON msdb.dbo.rds_fn_task_status TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.rds_task_status TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.rds_cancel_task TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.rds_download_from_s3 TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.rds_upload_to_s3 TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.rds_delete_from_filesystem TO [mydomain\user_name] with grant
 option
GRANT EXEC ON msdb.dbo.rds_gather_file_details TO [mydomain\user_name] with grant
 option
GRANT EXEC ON msdb.dbo.sp_add_proxy TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.sp_update_proxy TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.sp_grant_login_to_proxy TO [mydomain\user_name] with grant
 option
GRANT EXEC ON msdb.dbo.sp_revoke_login_from_proxy TO [mydomain\user_name] with grant
 option
GRANT EXEC ON msdb.dbo.sp_delete_proxy TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.sp_enum_login_for_proxy to [mydomain\user_name] with grant
 option
GRANT EXEC ON msdb.dbo.sp_enum_proxy_for_subsystem TO [mydomain\user_name] with grant
 option
GRANT EXEC ON msdb.dbo.rds_sqlagent_proxy TO [mydomain\user_name] with grant option
ALTER ROLE [SQLAgentUserRole] ADD MEMBER [mydomain\user_name]
GO
```

Adicionar um usuário de domínio como administrador de banco de dados

É possível adicionar um usuário de domínio como administrador de banco de dados do SSAS das seguintes maneiras:

- Um administrador de banco de dados pode usar o SSMS para criar uma função com privilégios `admin` e adicionar usuários a essa função.
- É possível usar o procedimento armazenado a seguir.

```
exec msdb.dbo.rds_msbi_task
@task_type='SSAS_ADD_DB_ADMIN_MEMBER',
@database_name='myssasdb',
@ssas_role_name='exampleRole',
@ssas_role_member='domain_name\domain_user_name';
```

Os seguintes parâmetros são obrigatórios:

- `@task_type` – O tipo da tarefa do MSBI, nesse caso `SSAS_ADD_DB_ADMIN_MEMBER`.
- `@database_name` – O nome do banco de dados do SSAS ao qual você está concedendo privilégios de administrador.
- `@ssas_role_name` – O nome da função de administrador de banco de dados do SSAS. Se a função ainda não existir, ela será criada.
- `@ssas_role_member` – O usuário do banco de dados do SSAS que você está adicionando à função de administrador.

Criar um proxy do SSAS

Para poder programar o processamento do banco de dados do SSAS usando o SQL Server Agent, crie uma credencial do SSAS e um proxy SSAS. Execute esses procedimentos como um usuário autenticado pelo Windows.

Para criar a credencial do SSAS

- Crie a credencial para o proxy. Para fazer isso, você pode usar o SSMS ou a seguinte instrução SQL.

```
USE [master]
GO
CREATE CREDENTIAL [SSAS_Credential] WITH IDENTITY = N'mydomain\user_name', SECRET =
 N'mysecret'
GO
```

> **Note**
>
> `IDENTITY` deve ser um login autenticado por domínio. Substitua *`mysecret`* pela senha do login autenticado por domínio.

Para criar o proxy do SSIS

1. Use a seguinte instrução SQL para criar o proxy.

```
USE [msdb]
GO
```

```
EXEC msdb.dbo.sp_add_proxy
 @proxy_name=N'SSAS_Proxy',@credential_name=N'SSAS_Credential',@description=N''
GO
```

2. Use a seguinte instrução SQL para conceder acesso ao proxy para outros usuários.

```
USE [msdb]
GO
EXEC msdb.dbo.sp_grant_login_to_proxy
 @proxy_name=N'SSAS_Proxy',@login_name=N'mydomain\user_name'
GO
```

3. Use a seguinte instrução SQL para dar ao subsistema do SSAS acesso ao proxy.

```
USE [msdb]
GO
EXEC msdb.dbo.rds_sqlagent_proxy
 @task_type='GRANT_SUBSYSTEM_ACCESS',@proxy_name='SSAS_Proxy',@proxy_subsystem='SSAS'
GO
```

Como visualizar o proxy e as concessões no proxy

1. Use a seguinte instrução SQL para visualizar os beneficiários do proxy.

```
USE [msdb]
GO
EXEC sp_help_proxy
GO
```

2. Use a seguinte instrução SQL para visualizar as concessões do subsistema.

```
USE [msdb]
GO
EXEC msdb.dbo.sp_enum_proxy_for_subsystem
GO
```

Programar o processamento de banco de dados do SSAS usando o SQL Server Agent

Depois de criar a credencial e o proxy e conceder acesso SSAS ao proxy, você pode criar um trabalho do SQL Server Agent para programar o processamento do banco de dados do SSAS.

Para programar processamento de banco de dados do SSAS

- Use o SSMS ou o T-SQL para criar o trabalho do SQL Server Agent. O exemplo a seguir usa T-SQL. Você pode configurar ainda mais seu programa de trabalho por meio do SSMS ou o T-SQL.

  - O parâmetro `@command` descreve o comando XML for Analysis (XMLA) a ser executado pelo trabalho do SQL Server Agent. Este exemplo configura o processamento de banco de dados multidimensional do SSAS.
  - O parâmetro `@server` descreve o nome do servidor do SSAS de destino do trabalho do SQL Server Agent.

    Para chamar o serviço do SSAS na mesma instância de banco de dados do RDS em que o trabalho do SQL Server Agent reside, use `localhost:2383`.

    Para chamar o serviço do SSAS de fora da instância de banco de dados do RDS, use o endpoint do RDS. Você também pode usar o endpoint do Kerberos Active Directory (AD) (*`your-DB-instance-name`*`.`*`your-AD-domain-name`*) se as instâncias de banco de dados do RDS forem unidas pelo mesmo domínio. Para instâncias de banco de dados externas, certifique-se de configurar corretamente o grupo de segurança da VPC associado à instância de banco de dados do RDS para uma conexão segura.

Você pode editar ainda mais a consulta para oferecer suporte a várias operações XMLA. Faça edições modificando diretamente a consulta T-SQL ou usando a interface do usuário do SSMS após a criação do trabalho do SQL Server Agent.

```
USE [msdb]
GO
DECLARE @jobId BINARY(16)
EXEC msdb.dbo.sp_add_job @job_name=N'SSAS_Job',
    @enabled=1,
    @notify_level_eventlog=0,
    @notify_level_email=0,
    @notify_level_netsend=0,
    @notify_level_page=0,
    @delete_level=0,
    @category_name=N'[Uncategorized (Local)]',
    @job_id = @jobId OUTPUT
GO
EXEC msdb.dbo.sp_add_jobserver
```

```
    @job_name=N'SSAS_Job',
    @server_name = N'(local)'
GO
EXEC msdb.dbo.sp_add_jobstep @job_name=N'SSAS_Job',
 @step_name=N'Process_SSAS_Object',
    @step_id=1,
    @cmdexec_success_code=0,
    @on_success_action=1,
    @on_success_step_id=0,
    @on_fail_action=2,
    @on_fail_step_id=0,
    @retry_attempts=0,
    @retry_interval=0,
    @os_run_priority=0, @subsystem=N'ANALYSISCOMMAND',
    @command=N'<Batch xmlns="http://schemas.microsoft.com/analysisservices/2003/
engine">
        <Parallel>
            <Process xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                xmlns:ddl2="http://schemas.microsoft.com/analysisservices/2003/
engine/2" xmlns:ddl2_2="http://schemas.microsoft.com/analysisservices/2003/
engine/2/2"
                xmlns:ddl100_100="http://schemas.microsoft.com/
analysisservices/2008/engine/100/100" xmlns:ddl200="http://schemas.microsoft.com/
analysisservices/2010/engine/200"
                xmlns:ddl200_200="http://schemas.microsoft.com/
analysisservices/2010/engine/200/200" xmlns:ddl300="http://schemas.microsoft.com/
analysisservices/2011/engine/300"
                xmlns:ddl300_300="http://schemas.microsoft.com/
analysisservices/2011/engine/300/300" xmlns:ddl400="http://schemas.microsoft.com/
analysisservices/2012/engine/400"
                xmlns:ddl400_400="http://schemas.microsoft.com/
analysisservices/2012/engine/400/400" xmlns:ddl500="http://schemas.microsoft.com/
analysisservices/2013/engine/500"
                xmlns:ddl500_500="http://schemas.microsoft.com/
analysisservices/2013/engine/500/500">
                <Object>
                    <DatabaseID>Your_SSAS_Database_ID</DatabaseID>
                </Object>
                <Type>ProcessFull</Type>
                <WriteBackTableCreation>UseExisting</WriteBackTableCreation>
            </Process>
        </Parallel>
    </Batch>',
```

```
    @server=N'localhost:2383',
    @database_name=N'master',
    @flags=0,
    @proxy_name=N'SSAS_Proxy'
GO
```

Revogar o acesso do SSAS do proxy

Você pode revogar o acesso ao subsistema SSAS e excluir o proxy SSAS usando os procedimentos armazenados a seguir.

Como revogar o acesso e excluir o proxy

1. Revogue o acesso ao subsistema.

```
USE [msdb]
GO
EXEC msdb.dbo.rds_sqlagent_proxy
 @task_type='REVOKE_SUBSYSTEM_ACCESS',@proxy_name='SSAS_Proxy',@proxy_subsystem='SSAS'
GO
```

2. Revogue as concessões no proxy.

```
USE [msdb]
GO
EXEC msdb.dbo.sp_revoke_login_from_proxy
 @proxy_name=N'SSAS_Proxy',@name=N'mydomain\user_name'
GO
```

3. Exclua o proxy.

```
USE [msdb]
GO
EXEC dbo.sp_delete_proxy @proxy_name = N'SSAS_Proxy'
GO
```

Fazer backup de um banco de dados do SSAS

É possível criar arquivos de backup de banco de dados do SSAS somente na pasta `D:\S3` da instância de banco de dados. Para mover os arquivos de backup para o bucket do S3, use o Amazon S3.

É possível fazer backup de um banco de dados do SSAS da seguinte forma:

- Um usuário de domínio com a função `admin` de um banco de dados específico pode usar o SSMS para fazer backup do banco de dados para a pasta `D:\S3`.

  Para obter mais informações, consulte Adicionar um usuário de domínio como administrador de banco de dados.
- É possível usar o procedimento armazenado a seguir. Este procedimento armazenado não oferece suporte a criptografia.

```
exec msdb.dbo.rds_msbi_task
@task_type='SSAS_BACKUP_DB',
@database_name='myssasdb',
@file_path='D:\S3\ssas_db_backup.abf',
[@ssas_apply_compression=1],
[@ssas_overwrite_file=1];
```

  Os seguintes parâmetros são obrigatórios:

  - `@task_type` – O tipo da tarefa do MSBI, nesse caso `SSAS_BACKUP_DB`.
  - `@database_name` – O nome do banco de dados do SSAS do qual você está fazendo backup.
  - `@file_path` – O caminho para o arquivo de backup do SSAS. É necessária a extensão `.abf`.

  Os seguintes parâmetros são opcionais:

  - `@ssas_apply_compression` – Se deve ser aplicada a compactação do backup do SSAS. Os valores válidos são 1 (Sim) e 0 (Não).
  - `@ssas_overwrite_file` – Se o arquivo de backup do SSAS deve ser substituído. Os valores válidos são 1 (Sim) e 0 (Não).

Restaurar um banco de dados do SSAS

Use o procedimento armazenado a seguir para restaurar um banco de dados do SSAS de um backup.

Não será possível restaurar um banco de dados se houver um banco de dados existente do SSAS com o mesmo nome. O procedimento armazenado para restauração não oferece suporte a arquivos de backup criptografados.

```
exec msdb.dbo.rds_msbi_task
@task_type='SSAS_RESTORE_DB',
@database_name='mynewssasdb',
@file_path='D:\S3\ssas_db_backup.abf';
```

Os seguintes parâmetros são obrigatórios:

- `@task_type` – O tipo da tarefa do MSBI, nesse caso `SSAS_RESTORE_DB`.
- `@database_name` – O nome do novo banco de dados do SSAS para o qual você está restaurando.
- `@file_path` – O caminho para o arquivo de backup do SSAS.

Restauração de uma instância de banco de dados para um tempo especificado

A recuperação em um ponto anterior no tempo (PITR) não se aplica a bancos de dados do SSAS. Se você fizer PITR, somente os dados do SSAS no último snapshot antes do tempo solicitado estarão disponíveis na instância restaurada.

Como ter bancos de dados do SSAS atualizados em uma instância de banco de dados restaurada

1. Faça backup dos bancos de dados do SSAS para a pasta `D:\S3` na instância de origem.
2. Transfira os arquivos de backup para o bucket do S3.
3. Transfira os arquivos de backup do bucket do S3 para a pasta `D:\S3` na instância restaurada.
4. Execute o procedimento armazenado para restaurar os bancos de dados do SSAS na instância restaurada.

   Também é possível reprocessar o projeto do SSAS para restaurar os bancos de dados.

Alterar o modo SSAS

Você pode alterar o modo no qual o SSAS é executado, seja Tabular ou Multidimensional. Para alterar o modo, use o AWS Management Console ou a AWS CLI para modificar as configurações de opções na opção SSAS.

> **Important**
>
> É possível usar apenas um modo SSAS de cada vez. Certifique-se de excluir todos os bancos de dados do SSAS antes de alterar o modo ou você receberá um erro.

Console

O procedimento de console do Amazon RDS a seguir altera o modo SSAS para Tabular e define o parâmetro `MAX_MEMORY` para 70%.

Para modificar a opção SSAS

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Option groups (Grupos de opções).
3. Selecione o grupo de opções com a opção `SSAS` que você quer modificar (`ssas-se-2017` nos exemplos anteriores).
4. Escolha Modify option (Modificar opção).
5. Altere as configurações de opção:
   a. Para Max memory (Memória máxima), insira **`70`**.
   b. Para Mode (Modo), escolha Tabular.
6. Escolha Modify option (Modificar opção).

AWS CLI

O seguinte exemplo da AWS CLI altera o modo SSAS para Tabular e define o parâmetro `MAX_MEMORY` para 70%.

Para que o comando da CLI funcione, certifique-se de incluir todos os parâmetros necessários, mesmo se você não estiver modificando-os.

Para modificar a opção SSAS

- Use um dos seguintes comandos.

Example

Para Linux, macOS ou Unix:

```
aws rds add-option-to-option-group \
    --option-group-name ssas-se-2017 \
    --options
 "OptionName=SSAS,VpcSecurityGroupMemberships=sg-12345e67,OptionSettings=[{Name=MAX_MEMORY,
{Name=MODE,Value=Tabular}]" \
    --apply-immediately
```

Para Windows:

```
aws rds add-option-to-option-group ^
    --option-group-name ssas-se-2017 ^
    --options
 OptionName=SSAS,VpcSecurityGroupMemberships=sg-12345e67,OptionSettings=[{Name=MAX_MEMORY,V
{Name=MODE,Value=Tabular}] ^
    --apply-immediately
```

Desativar o SSAS

Para desativar o SSAS, remova a opção SSAS do grupo de opções.

> **Important**
>
> Antes de remover a opção SSAS, exclua os bancos de dados do SSAS.
> É altamente recomendável que você faça backup dos bancos de dados do SSAS antes de excluí-los e remover a opção SSAS.

Console

Como remover a opção SSAS do grupo de opções

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Option groups (Grupos de opções).

3. Escolha o grupo de opções com a opção SSAS que você quer remover (`ssas-se-2017` nos exemplos anteriores).
4. Selecione Delete option (Excluir opção).
5. Em Deletion options (Opções de exclusão), selecione SSAS em Options to delete (Opções a serem excluídas).
6. Em Apply immediately (Aplicar imediatamente), selecione Yes (Sim) para excluir a opção imediatamente ou No (Não) para excluí-la na próxima janela de manutenção.
7. Escolha Excluir.

AWS CLI

Como remover a opção SSAS do grupo de opções

- Use um dos seguintes comandos.

  Example

  Para Linux, macOS ou Unix:

```
aws rds remove-option-from-option-group \
    --option-group-name ssas-se-2017 \
    --options SSAS \
    --apply-immediately
```

  Para Windows:

```
aws rds remove-option-from-option-group ^
    --option-group-name ssas-se-2017 ^
    --options SSAS ^
    --apply-immediately
```

Solução de problemas do SSAS

Você pode encontrar os problemas a seguir ao usar o SSAS.

Problema	Tipo	Sugestões de solução de problemas
Não foi possível configurar a opção SSAS. O modo SSAS solicitado é *new_mode*, mas a instância de banco de dados atual tem *number* banco de dados *current_mode* . Exclua os bancos de dados existentes antes de mudar para o modo *new_mode*. Para recuperar o acesso ao modo *current_mode* para exclusão do banco de dados, atualize o grupo de opções de banco de dados atual ou anexe um novo grupo de opções com %s como o valor de configuração da opção MODE para a opção SSAS.	Evento RDS	Você não pode alterar o modo SSAS se ainda tiver bancos de dados do SSAS que usam o modo atual. Exclua os bancos de dados do SSAS e tente novamente.
Não é possível remover a opção SSAS porque há *number* banco de dados de *modo* existentes. A opção SSAS não pode ser removida até que todos os bancos de dados SSAS sejam excluídos. Adicione a opção SSAS novamente, exclua todos os bancos de dados SSAS e tente novamente.	Evento RDS	Você não pode desativar o SSAS se ainda tiver bancos de dados do SSAS. Exclua os bancos de dados do SSAS e tente novamente.
A opção SSAS não está habilitada ou está em processo de habilitação. Tente novamente mais tarde.	Procedimentos armazenados do RDS	Você não pode executar procedimentos armazenados do SSAS quando a opção estiver desativada ou quando ela está sendo ativada.
A opção SSAS está configurada incorretamente. Certifique-se de que o status da associação do grupo de opções esteja “em sincronia” e	Procedimentos armazenad	Você não pode executar procedimentos armazenados do SSAS quando sua associação ao grupo de opções não tiver o

Problema	Tipo	Sugestões de solução de problemas
revise os logs de eventos do RDS para mensagens de erro de configuração SSAS relevantes. Após essas investigações, tente novamente. Se os erros continuarem a ocorrer, entre em contato com o AWS Support.	os do RDS	status `in-sync`. Isso coloca a opção SSAS em um estado de configuração incorreto. Se o status de associação do seu grupo de opções mudar para `failed` devido à modificação da opção SSAS, há dois motivos possíveis: 1. A opção SSAS foi removida sem que os bancos de dados do SSAS fossem excluídos. 2. O modo SSAS foi atualizado de Tabular para Multidimensional, ou de Multidimensional para Tabular, sem que os bancos de dados do SSAS existentes fossem excluídos. Reconfigure a opção SSAS, porque o RDS permite apenas um modo SSAS por vez e não suporta a remoção de opções SSAS com bancos de dados do SSAS presentes. Verifique se há erros de configuração nos logs de eventos do RDS para sua instância do SSAS e resolva os problemas de acordo.
Falha na implantação. A alteração só pode ser implantada em um servidor em execução no modo *`deployment_file_mode`* . O modo de servidor atual é *`current_mode`* .	Procedimentos armazenados do RDS	Você não pode implantar um banco de dados Tabular em um servidor Multidimensional ou em um banco de dados Multidimensional em um servidor Tabular. Verifique se você está usando arquivos com o modo correto e verifique se a configuração de opção `MODE` é definida como o valor apropriado.

Problema	Tipo	Sugestões de solução de problemas
A restauração falhou. O arquivo de backup só pode ser restaurado em um servidor em execução no modo *restore_file_mode* . O modo de servidor atual é *current_mode* .	Procedimentos armazenados do RDS	Você não pode restaurar um banco de dados Tabular para um servidor Multidimensional ou um banco de dados Multidimensional para um servidor Tabular. Verifique se você está usando arquivos com o modo correto e verifique se a configuração de opção `MODE` é definida como o valor apropriado.
A restauração falhou. O arquivo de backup e as versões da instância de banco de dados do RDS são incompatíveis.	Procedimentos armazenados do RDS	Você não pode restaurar um banco de dados do SSAS com uma versão incompatível com a versão da instância do SQL Server. Para obter mais informações, consulte Níveis de compatibilidade para modelos tabulares e Nível de compatibilidade de um banco de dados multidimensional na documentação da Microsoft.
A restauração falhou. O arquivo de backup especificado na operação de restauração está danificado ou não é um arquivo de backup do SSAS. Certifique-se de que @rds_file_path esteja formatado corretamente.	Procedimentos armazenados do RDS	Você não pode restaurar um banco de dados do SSAS com um arquivo danificado. Certifique-se de que o arquivo não esteja danificado ou corrompido. Esse erro também pode ser gerado quando `@rds_file_path` não está formatado corretamente (por exemplo, ele tem barras invertidas duplas como em `D:\S3\\incorrect_format.abf`).

Problema	Tipo	Sugestões de solução de problemas
A restauração falhou. O nome do banco de dados restaurado não pode conter palavras reservadas, mais de 100 caracteres ou caracteres inválidos: . , ; ' ` : / \\ * | ? \" & % $! + = ( ) [ ] { } < >	Procedimentos armazenados do RDS	O nome do banco de dados restaurado não pode conter palavras ou caracteres reservados que não sejam válidos ou tenham mais de 100 caracteres. Para convenções de nomenclatura de objetos SSAS, consulte Regras de nomeação de objetos na documentação da Microsoft.
Um nome de função inválido foi fornecido. O nome da função não pode conter strings reservados.	Procedimentos armazenados do RDS	O nome da função não pode conter strings reservados. Para convenções de nomenclatura de objetos SSAS, consulte Regras de nomeação de objetos na documentação da Microsoft.
Um nome de função inválido foi fornecido. O nome da função não pode conter nenhum dos seguintes caracteres reservados: . , ; ' ` : / \\ * | ? \" & % $! + = ( ) [ ] { } < >	Procedimentos armazenados do RDS	O nome da função não pode conter caracteres reservados. Para convenções de nomenclatura de objetos SSAS, consulte Regras de nomeação de objetos na documentação da Microsoft.

Suporte para SQL Server Integration Services no Amazon RDS for SQL Server

O Microsoft SQL Server Integration Services (SSIS) é um componente que você pode usar para executar uma ampla gama de tarefas de migração de dados. O SSIS é uma plataforma para integração de dados e aplicativos de fluxo de trabalho. Ele possui uma ferramenta de data warehousing usada para extração, transformação e carregamento (ETL) de dados. Você também pode usar essa ferramenta para automatizar a manutenção de bancos de dados do SQL Server e atualizações para dados de cubo multidimensional.

Os projetos do SSIS são organizados em pacotes salvos como arquivos .dtsx baseados em XML. Os pacotes podem conter fluxos de controle e fluxos de dados. Você usa fluxos de dados para representar operações ETL. Após a implantação, os pacotes são armazenados no SQL Server no banco de dados SSISDB. SSISDB é um banco de dados de processamento de transações online (OLTP) no modo de recuperação completa.

O Amazon RDS for SQL Server oferece suporte à execução de SSIS diretamente em instâncias de banco de dados do RDS. Você pode habilitar o SSIS em uma instância de banco de dados existente ou nova. O SSIS é instalado na mesma instância de banco de dados que seu mecanismo de banco de dados.

O RDS oferece suporte ao SSIS para as edições Standard e Enterprise do SQL Server nas seguintes versões:

- SQL Server 2022, todas as versões
- Versão 15.00.4043.16.v1 e posterior do SQL Server 2019
- Versão 14.00.3223.3.v1 e posterior do SQL Server 2017
- Versão 13.00.5426.0.v1 e superior do SQL Server 2016

Sumário

- Limitações e recomendações
- Habilitar o SSIS
  - Criar o grupo de opções para o SSIS
  - Adicionar a opção SSIS ao grupo de opções
  - Criar o grupo de parâmetros para o SSIS
  - Modificar o parâmetro para o SSIS

  - Associar o grupo de opções e o grupo de parâmetros à instância de banco de dados
  - Habilitar a integração do S3
- Permissões administrativas no SSISDB
  - Configurar um usuário autenticado pelo Windows para o SSIS
- Implantar um projeto do SSIS
- Monitorar o status de uma tarefa de implantação
- Usar o SSIS
  - Definir gerenciadores de conexão de banco de dados para projetos do SSIS
  - Criar um proxy do SSIS
  - Programar um pacote do SSIS usando o SQL Server Agent
  - Revogar o acesso do SSIS do proxy
- Desabilitar e descartar o banco de dados do SSIS
  - Desabilitar o SSIS
  - Eliminar o banco de dados SSISDB

Limitações e recomendações

As seguintes limitações e recomendações aplicam-se à execução do SSIS no RDS for SQL Server:

- A instância de banco de dados deve ter um grupo de parâmetros associado ao parâmetro `clr enabled` definido como 1. Para obter mais informações, consulte Modificar o parâmetro para o SSIS.

  > **Note**
  >
  > Se você habilitar o parâmetro `clr enabled` no SQL Server 2017 ou 2019, não será possível usar o tempo de execução de linguagem comum (CLR) na sua instância de banco de dados. Para obter mais informações, consulte Recursos não compatíveis e recursos com suporte limitado.

- As seguintes tarefas de fluxo de controle são compatíveis:
  - Tarefa de execução de DDL de serviços de análise
  - Tarefa de processamento de serviços de análise
  - Tarefa de inserção em massa

  - Tarefa de verificação da integridade do banco de dados
  - Tarefa de fluxo de dados
  - Tarefa de consulta de mineração de dados
  - Tarefa de criação de perfil de dados
  - Tarefa de execução do pacote
  - Tarefa de execução de trabalho do SQL Server Agent
  - Tarefa de execução do SQL
  - Tarefa de execução de instrução T-SQL
  - Tarefa de notificação do operador
  - Tarefa de recriação de índice
  - Tarefa de reorganização de índice
  - Tarefa de redução do banco de dados
  - Tarefa de transferência do banco de dados
  - Tarefa de transferência de trabalhos
  - Tarefa de transferência de logins
  - Tarefa de transferência de objetos do SQL Server
  - Tarefa de atualização de estatísticas
- Somente a implantação do projeto é permitida.
- É permitido executar pacotes do SSIS usando o SQL Server Agent.
- Os registros de log do SSIS só podem ser inseridos em bancos de dados criados pelo usuário.
- Use apenas a pasta `D:\S3` para trabalhar com arquivos. Os arquivos colocados em qualquer outro diretório são excluídos. Esteja ciente de alguns outros detalhes de localização do arquivo:
  - Coloque os arquivos de entrada e saída do projeto do SSIS na pasta `D:\S3`.
  - Para a tarefa de fluxo de dados, altere o local para `BLOBTempStoragePath` e `BufferTempStoragePath` para um arquivo dentro da pasta `D:\S3`. O caminho do arquivo deve começar com `D:\S3\`.
  - Certifique-se de que todos os parâmetros, variáveis e expressões usados para conexões de arquivos apontem para a pasta `D:\S3`.
  - Em instâncias Multi-AZ, os arquivos criados pelo SSIS na pasta `D:\S3` são excluídos após um failover. Para obter mais informações, consulte Limitações Multi-AZ para a integração do S3.

  - Carregue os arquivos criados pelo SSIS na pasta `D:\S3` para seu bucket do Amazon S3 para que sejam duráveis.
- As transformações Importar coluna e Exportar coluna e o componente Script na tarefa de fluxo de dados não são compatíveis.
- Não é possível habilitar o despejo na execução do pacote do SSIS e não é possível adicionar toques de dados em pacotes do SSIS.
- Não há suporte para o recurso de expansão do SSIS.
- Não é possível implantar projetos diretamente. Fornecemos procedimentos armazenados do RDS para fazer isso. Para obter mais informações, consulte Implantar um projeto do SSIS.
- Crie arquivos de projeto do SSIS (.ispac) com o modo de proteção `DoNotSavePasswords` para implantação no RDS.
- O SSIS não é compatível com instâncias Always On com réplicas de leitura.
- Não é possível fazer backup do banco de dados SSISDB associado à opção `SSIS`.
- Importar e restaurar o banco de dados SSISDB de outras instâncias do SSIS não é permitido.
- É possível se conectar a outras instâncias de banco de dados do SQL Server ou a uma fonte de dados Oracle. A conexão com outros mecanismos de banco de dados, como o MySQL ou o PostgreSQL, não é compatível com o SSIS no RDS para SQL Server. Para obter mais informações sobre a conexão com uma fonte de dados Oracle, consulte Servidores vinculados ao Oracle OLEDB.
- O SSIS não aceita uma instância associada ao domínio com uma relação de confiança de saída para um domínio on-premises. Ao usar uma relação de confiança de saída, execute o trabalho do SSIS por meio de uma conta no domínio local da AWS.

Habilitar o SSIS

Você habilita o SSIS adicionando a opção SSIS à sua instância de banco de dados. Use o seguinte processo:

1. Crie um grupo de opções ou selecione um existente.
2. Adicione a opção `SSIS` ao grupo de opções.
3. Crie um novo grupo de parâmetros ou escolha um existente.
4. Modifique o grupo de parâmetros para definir o parâmetro `clr enabled` como 1.
5. Associe o grupo de opções e o grupo de parâmetros à instância de banco de dados.

6. Habilite a integração do Amazon S3.

> **Note**
>
> Se um banco de dados com o nome SSISDB ou um login reservado do SSIS já existir na instância de banco de dados, você não poderá habilitar o SSIS na instância.

Criar o grupo de opções para o SSIS

Para trabalhar com o SSIS, crie um grupo de opções ou modifique um grupo de opções que corresponda à edição do SQL Server e à versão da instância de banco de dados que você pretende usar. Para fazer isso, use o AWS Management Console ou a AWS CLI.

Console

O procedimento a seguir cria um grupo de opções para o SQL Server Standard Edition 2016.

Como criar o grupo de opções

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Option groups (Grupos de opções).
3. Escolha Create group (Criar grupo).
4. Na janela Create option group (Criar grupo de opções), faça o seguinte:
   a. Em Name (Nome), insira um nome para o grupo de opções que seja exclusivo na sua conta da AWS, como **`ssis-se-2016`**. O nome pode conter somente letras, números e hifens.
   b. Em Descrição, insira uma breve descrição do grupo de opções, como **`SSIS option group for SQL Server SE 2016`**. A descrição é usada para fins de exibição.
   c. Em Engine (Mecanismo), selecione sqlserver-se.
   d. Em Versão do mecanismo principal, selecione 13.00.
5. Escolha Criar.

CLI

O procedimento a seguir cria um grupo de opções para o SQL Server Standard Edition 2016.

Como criar o grupo de opções

- Execute um dos seguintes comandos:

  Example

  Para Linux, macOS ou Unix:

```
aws rds create-option-group \
    --option-group-name ssis-se-2016 \
    --engine-name sqlserver-se \
    --major-engine-version 13.00 \
    --option-group-description "SSIS option group for SQL Server SE 2016"
```

  Para Windows:

```
aws rds create-option-group ^
    --option-group-name ssis-se-2016 ^
    --engine-name sqlserver-se ^
    --major-engine-version 13.00 ^
    --option-group-description "SSIS option group for SQL Server SE 2016"
```

Adicionar a opção SSIS ao grupo de opções

Depois, use o AWS Management Console ou a AWS CLI para adicionar a opção `SSIS` ao grupo de opções.

Console

Como adicionar a opção SSIS

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Option groups (Grupos de opções).
3. Escolha o grupo de opções que você acabou de criar, ssis-se-2016 neste exemplo.
4. Escolha Add option (Adicionar opção).
5. Em Detalhes da opção, selecione SSIS em Nome da opção.
6. Em Scheduling (Programação), escolha se deseja adicionar a opção imediatamente ou na próxima janela de manutenção.

7. Escolha Add option (Adicionar opção).

CLI

Como adicionar a opção SSIS

- Adicione a opção `SSIS` ao grupo de opções.

  Example

  Para Linux, macOS ou Unix:

  ```
  aws rds add-option-to-option-group \
      --option-group-name ssis-se-2016 \
      --options OptionName=SSIS \
      --apply-immediately
  ```

  Para Windows:

  ```
  aws rds add-option-to-option-group ^
      --option-group-name ssis-se-2016 ^
      --options OptionName=SSIS ^
      --apply-immediately
  ```

Criar o grupo de parâmetros para o SSIS

Crie ou modifique um grupo de parâmetros para o parâmetro `clr enabled` que corresponde à edição e à versão do SQL Server da instância de banco de dados que você pretende usar para o SSIS.

Console

O procedimento a seguir cria um grupo de parâmetros para o SQL Server Standard Edition 2016.

Como criar o grupo de parâmetros

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Parameter groups (Grupos de parâmetros).
3. Escolha Create parameter group (Criar parameter group).

4. No painelCreate parameter group (Criar parameter group), faça o seguinte:

   a. Em Família de grupos de parâmetros, escolha sqlserver-se-13.0.

   b. Em Nome do grupo, insira um identificador para o grupo de parâmetros, como **`ssis-sqlserver-se-13`**.

   c. Em Descrição, insira **`clr enabled parameter group`**.

5. Escolha Criar.

CLI

O procedimento a seguir cria um grupo de parâmetros para o SQL Server Standard Edition 2016.

Como criar o grupo de parâmetros

- Execute um dos seguintes comandos:

  Example

  Para Linux, macOS ou Unix:

```
aws rds create-db-parameter-group \
    --db-parameter-group-name ssis-sqlserver-se-13 \
    --db-parameter-group-family "sqlserver-se-13.0" \
    --description "clr enabled parameter group"
```

  Para Windows:

```
aws rds create-db-parameter-group ^
    --db-parameter-group-name ssis-sqlserver-se-13 ^
    --db-parameter-group-family "sqlserver-se-13.0" ^
    --description "clr enabled parameter group"
```

Modificar o parâmetro para o SSIS

Modifique o parâmetro `clr enabled` no grupo de parâmetros que corresponde à edição e à versão do SQL Server da instância de banco de dados. Para o SSIS, defina o parâmetro `clr enabled` como 1.

Console

O procedimento a seguir modifica o grupo de parâmetros que você criou para o SQL Server Standard Edition 2016.

Como modificar o grupo de parâmetros

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, selecione Parameter groups.
3. Escolha o grupo de parâmetros, como ssis-sqlserver-se-13.
4. Em Parâmetros, filtre a lista de parâmetros para **`clr`**.
5. Escolha clr ativado.
6. Escolha Edit parameters.
7. Em Valores, escolha 1.
8. Escolha Salvar alterações.

CLI

O procedimento a seguir modifica o grupo de parâmetros que você criou para o SQL Server Standard Edition 2016.

Como modificar o grupo de parâmetros

- Execute um dos seguintes comandos:

  Example

  Para Linux, macOS ou Unix:

```
aws rds modify-db-parameter-group \
    --db-parameter-group-name ssis-sqlserver-se-13 \
    --parameters "ParameterName='clr
 enabled',ParameterValue=1,ApplyMethod=immediate"
```

  Para Windows:

```
aws rds modify-db-parameter-group ^
```

```
    --db-parameter-group-name ssis-sqlserver-se-13 ^
    --parameters "ParameterName='clr
enabled',ParameterValue=1,ApplyMethod=immediate"
```

Associar o grupo de opções e o grupo de parâmetros à instância de banco de dados

Para associar o grupo de opções e o grupo de parâmetros do SSIS à sua instância de banco de dados, use o AWS Management Console ou a AWS CLI

> **Note**
>
> Se você usar uma instância existente, ela já deve ter um domínio do Active Directory e uma função do AWS Identity and Access Management (IAM) associados a ela. Se você criar uma nova instância, especifique um domínio do Active Directory e uma função do IAM existentes. Para obter mais informações, consulte Trabalhar com o Active Directory com o RDS para SQL Server.

Console

Para concluir a ativação do SSIS, associe o grupo de opções e o grupo de parâmetros do SSIS a uma instância de banco de dados nova ou existente:

- Para uma nova instância de banco de dados, associe-os ao executar a instância. Para obter mais informações, consulte Criar uma instância de banco de dados do Amazon RDS.
- Para uma instância de banco de dados existente, associe-os modificando a instância. Para obter mais informações, consulte Modificar uma instância de banco de dados do Amazon RDS.

CLI

É possível associar o grupo de opções e o grupo de parâmetros do SSIS a uma instância de banco de dados nova ou existente.

Como criar uma instância com o grupo de opções e o grupo de parâmetros do SSIS

- Especifique o mesmo tipo de mecanismo de banco de dados e a versão principal que você usou ao criar o grupo de opções.

Example

Para Linux, macOS ou Unix:

```
aws rds create-db-instance \
    --db-instance-identifier myssisinstance \
    --db-instance-class db.m5.2xlarge \
    --engine sqlserver-se \
    --engine-version 13.00.5426.0.v1 \
    --allocated-storage 100 \
    --manage-master-user-password \
    --master-username admin \
    --storage-type gp2 \
    --license-model li \
    --domain-iam-role-name my-directory-iam-role \
    --domain my-domain-id \
    --option-group-name ssis-se-2016 \
    --db-parameter-group-name ssis-sqlserver-se-13
```

Para Windows:

```
aws rds create-db-instance ^
    --db-instance-identifier myssisinstance ^
    --db-instance-class db.m5.2xlarge ^
    --engine sqlserver-se ^
    --engine-version 13.00.5426.0.v1 ^
    --allocated-storage 100 ^
    --manage-master-user-password ^
    --master-username admin ^
    --storage-type gp2 ^
    --license-model li ^
    --domain-iam-role-name my-directory-iam-role ^
    --domain my-domain-id ^
    --option-group-name ssis-se-2016 ^
    --db-parameter-group-name ssis-sqlserver-se-13
```

Como modificar uma instância de banco de dados e associar o grupo de opções e o grupo de parâmetros do SSIS

- Execute um dos seguintes comandos:

Example

Para Linux, macOS ou Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier myssisinstance \
    --option-group-name ssis-se-2016 \
    --db-parameter-group-name ssis-sqlserver-se-13 \
    --apply-immediately
```

Para Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier myssisinstance ^
    --option-group-name ssis-se-2016 ^
    --db-parameter-group-name ssis-sqlserver-se-13 ^
    --apply-immediately
```

Habilitar a integração do S3

Para baixar arquivos de projeto (.ispac) do SSIS para o host para implantação, use a integração de arquivos do S3. Para obter mais informações, consulte Integrar uma instância de banco de dados do Amazon RDS for SQL Server ao Amazon S3.

Permissões administrativas no SSISDB

Quando a instância é criada ou modificada com a opção SSIS, o resultado é um banco de dados SSISDB com as funções ssis_admin e ssis_logreader concedidas ao usuário mestre. O usuário mestre tem os seguintes privilégios no SSISDB:

- alterar na função ssis_admin
- alterar na função ssis_logreader
- alterar qualquer usuário

Como o usuário mestre é um usuário autenticado pelo SQL, você não pode usar o usuário mestre para executar pacotes do SSIS. O usuário mestre pode usar esses privilégios para criar usuários do SSISDB e adicioná-los às funções ssis_admin e ssis_logreader. Fazer isso é útil para dar acesso aos usuários do domínio para usar o SSIS.

Configurar um usuário autenticado pelo Windows para o SSIS

O usuário mestre pode usar o exemplo de código a seguir para configurar um logon autenticado pelo Windows no SSISDB e conceder as permissões de procedimento necessárias. Fazer isso concede permissões ao usuário do domínio para implantar e executar pacotes do SSIS, usar procedimentos de transferência de arquivos do S3, criar credenciais e trabalhar com o proxy do SQL Server Agent. Para obter mais informações, consulte Credenciais (mecanismo de banco de dados) e Criar um proxy do SQL Server Agent na documentação da Microsoft.

> **Note**
>
> Você pode conceder algumas ou todas as seguintes permissões, conforme necessário, aos usuários autenticados pelo Windows.

Example

```
-- Create a server-level SQL login for the domain user, if it doesn't already exist
USE [master]
GO
CREATE LOGIN [mydomain\user_name] FROM WINDOWS
GO

-- Create a database-level account for the domain user, if it doesn't already exist

USE [SSISDB]
GO
CREATE USER [mydomain\user_name] FOR LOGIN [mydomain\user_name]

-- Add SSIS role membership to the domain user
ALTER ROLE [ssis_admin] ADD MEMBER [mydomain\user_name]
ALTER ROLE [ssis_logreader] ADD MEMBER [mydomain\user_name]
GO

-- Add MSDB role membership to the domain user
USE [msdb]
GO
CREATE USER [mydomain\user_name] FOR LOGIN [mydomain\user_name]

-- Grant MSDB stored procedure privileges to the domain user
GRANT EXEC ON msdb.dbo.rds_msbi_task TO [mydomain\user_name] with grant option
GRANT SELECT ON msdb.dbo.rds_fn_task_status TO [mydomain\user_name] with grant option
```

```
GRANT EXEC ON msdb.dbo.rds_task_status TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.rds_cancel_task TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.rds_download_from_s3 TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.rds_upload_to_s3 TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.rds_delete_from_filesystem TO [mydomain\user_name] with grant
 option
GRANT EXEC ON msdb.dbo.rds_gather_file_details TO [mydomain\user_name] with grant
 option
GRANT EXEC ON msdb.dbo.sp_add_proxy TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.sp_update_proxy TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.sp_grant_login_to_proxy TO [mydomain\user_name] with grant
 option
GRANT EXEC ON msdb.dbo.sp_revoke_login_from_proxy TO [mydomain\user_name] with grant
 option
GRANT EXEC ON msdb.dbo.sp_delete_proxy TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.sp_enum_login_for_proxy to [mydomain\user_name] with grant
 option
GRANT EXEC ON msdb.dbo.sp_enum_proxy_for_subsystem TO [mydomain\user_name]  with grant
 option
GRANT EXEC ON msdb.dbo.rds_sqlagent_proxy TO [mydomain\user_name] WITH GRANT OPTION


-- Add the SQLAgentUserRole privilege to the domain user
USE [msdb]
GO
ALTER ROLE [SQLAgentUserRole] ADD MEMBER [mydomain\user_name]
GO

-- Grant the ALTER ANY CREDENTIAL privilege to the domain user
USE [master]
GO
GRANT ALTER ANY CREDENTIAL TO [mydomain\user_name]
GO
```

Implantar um projeto do SSIS

No RDS, não é possível implantar projetos do SSIS diretamente usando o SQL Server Management Studio (SSMS) ou procedimentos do SSIS. Para baixar arquivos de projeto do Amazon S3 e implantá-los, use procedimentos armazenados do RDS.

Para executar os procedimentos armazenados, faça login como qualquer usuário ao qual você concedeu permissões para executar os procedimentos armazenados. Para obter mais informações, consulte Configurar um usuário autenticado pelo Windows para o SSIS.

Como implantar o projeto do SSIS

1. Baixe o arquivo do projeto (.ispac).

```
exec msdb.dbo.rds_download_from_s3
@s3_arn_of_file='arn:aws:s3:::bucket_name/ssisproject.ispac',
@rds_file_path='D:\S3\ssisproject.ispac',
@overwrite_file=1;
```

2. Envie a tarefa de implantação, certificando-se do seguinte:

   - A pasta está presente no catálogo do SSIS.
   - O nome do projeto corresponde ao nome do projeto usado durante o desenvolvimento do projeto do SSIS.

```
exec msdb.dbo.rds_msbi_task
@task_type='SSIS_DEPLOY_PROJECT',
@folder_name='DEMO',
@project_name='ssisproject',
@file_path='D:\S3\ssisproject.ispac';
```

Monitorar o status de uma tarefa de implantação

Para rastrear o status da tarefa de implantação, chame a função `rds_fn_task_status`. Ela leva dois parâmetros. O primeiro parâmetro sempre deve ser `NULL` porque ele não se aplica ao SSIS. O segundo parâmetro aceita um ID de tarefa.

Para ver uma lista de todas as tarefas, defina o primeiro parâmetro como `NULL` e o segundo parâmetro como `0`, conforme mostrado no seguinte exemplo.

```
SELECT * FROM msdb.dbo.rds_fn_task_status(NULL,0);
```

Para obter uma tarefa específica, defina o primeiro parâmetro como `NULL` e o segundo parâmetro como o ID da tarefa, conforme mostrado no exemplo a seguir.

```
SELECT * FROM msdb.dbo.rds_fn_task_status(NULL,42);
```

A função `rds_fn_task_status` retorna as informações a seguir.

Parâmetro de saída	Descrição
`task_id`	O ID da tarefa.
`task_type`	`SSIS_DEPLOY_PROJECT`
`database_name`	Não aplicável a tarefas do SSIS.
`% complete`	O progresso da tarefa em porcentagem.
`duration (mins)`	A quantidade de tempo gasta na tarefa, em minutos.
`lifecycle`	O status da tarefa. Os possíveis status são os seguintes: • `CREATED` – depois de chamar o procedimento `msdb.dbo.rds_msbi_task` armazenado, uma tarefa é criada e o status é definido como `CREATED`. • `IN_PROGRESS` – Após iniciar uma tarefa, o status será definido como `IN_PROGRESS`. Pode demorar até 5 minutos para que o status mude de `CREATED` para `IN_PROGRESS`. • `SUCCESS` – Após concluir uma tarefa, o status será definido como `SUCCESS`. • `ERROR` – Se uma tarefa falhar, o status será definido como `ERROR`. Para obter mais informações sobre o erro, consulte a coluna `task_info`. •

Parâmetro de saída	Descrição
	`CANCEL_REQUESTED` – Após chamar `rds_cancel_task` , o status da tarefa será definido como `CANCEL_REQUESTED` . • `CANCELLED` – após uma tarefa ter sido cancelada com sucesso, o status da tarefa será definido como `CANCELLED` .
`task_info`	Informações adicionais sobre a tarefa. Se um erro ocorrer durante o processamento, essa coluna conterá informações sobre o erro.
`last_updated`	A data e hora em que o status da tarefa foi atualizado pela última vez.
`created_at`	A data e hora em que a tarefa foi criada.
`S3_object_arn`	Não aplicável a tarefas do SSIS.
`overwrite_S3_backup_file`	Não aplicável a tarefas do SSIS.
`KMS_master_key_arn`	Não aplicável a tarefas do SSIS.
`filepath`	Não aplicável a tarefas do SSIS.
`overwrite_file`	Não aplicável a tarefas do SSIS.
`task_metadata`	Metadados associados à tarefa do SSIS.

Usar o SSIS

Depois de implantar o projeto do SSIS no catálogo do SSIS, você pode executar pacotes diretamente do SSMS ou programá-los usando o SQL Server Agent. Você deve usar um login autenticado pelo

Windows para executar pacotes do SSIS. Para obter mais informações, consulte Configurar um usuário autenticado pelo Windows para o SSIS.

Tópicos

- Definir gerenciadores de conexão de banco de dados para projetos do SSIS
- Criar um proxy do SSIS
- Programar um pacote do SSIS usando o SQL Server Agent
- Revogar o acesso do SSIS do proxy

Definir gerenciadores de conexão de banco de dados para projetos do SSIS

Ao utilizar um gerenciador de conexões, você pode usar estes tipos de autenticação:

- Para conexões de banco de dados local usando o AWS Managed Active Directory, você pode usar autenticação do SQL ou autenticação do Windows. Para autenticação do Windows, use *DB_instance_name*.*fully_qualified_domain_name* como o nome do servidor da string de conexão.

  Um exemplo é `myssisinstance.corp-ad.example.com`, onde `myssisinstance` é o nome da instância de banco de dados e `corp-ad.example.com` é o nome de domínio totalmente qualificado.
- Para conexões remotas, use sempre a autenticação do SQL.
- Para conexões de banco de dados local usando um Active Directory autogerenciado, você pode usar autenticação do SQL ou autenticação do Windows. Para autenticação do Windows, use `.` ou *`LocalHost`* como o nome do servidor da string de conexão.

Criar um proxy do SSIS

Para poder agendar pacotes do SSIS usando o SQL Server Agent, crie uma credencial do SSIS e um proxy do SSIS. Execute esses procedimentos como um usuário autenticado pelo Windows.

Como criar a credencial do SSIS

- Crie a credencial para o proxy. Para fazer isso, você pode usar o SSMS ou a seguinte instrução SQL.

```
USE [master]
```

```
GO
CREATE CREDENTIAL [SSIS_Credential] WITH IDENTITY = N'mydomain\user_name', SECRET =
 N'mysecret'
GO
```

> **Note**
>
> IDENTITY deve ser um login autenticado por domínio. Substitua *mysecret* pela senha do login autenticado por domínio.
> Sempre que o host primário do SSISDB for alterado, altere as credenciais de proxy do SSIS para permitir que o novo host as acesse.

Como criar o proxy do SSIS

1. Use a seguinte instrução SQL para criar o proxy.

```
USE [msdb]
GO
EXEC msdb.dbo.sp_add_proxy
 @proxy_name=N'SSIS_Proxy',@credential_name=N'SSIS_Credential',@description=N''
GO
```

2. Use a seguinte instrução SQL para conceder acesso ao proxy para outros usuários.

```
USE [msdb]
GO
EXEC msdb.dbo.sp_grant_login_to_proxy
 @proxy_name=N'SSIS_Proxy',@login_name=N'mydomain\user_name'
GO
```

3. Use a seguinte instrução SQL para dar ao subsistema do SSIS acesso ao proxy.

```
USE [msdb]
GO
EXEC msdb.dbo.rds_sqlagent_proxy
 @task_type='GRANT_SUBSYSTEM_ACCESS',@proxy_name='SSIS_Proxy',@proxy_subsystem='SSIS'
GO
```

Como visualizar o proxy e as concessões no proxy

1. Use a seguinte instrução SQL para visualizar os beneficiários do proxy.

```
USE [msdb]
GO
EXEC sp_help_proxy
GO
```

2. Use a seguinte instrução SQL para visualizar as concessões do subsistema.

```
USE [msdb]
GO
EXEC msdb.dbo.sp_enum_proxy_for_subsystem
GO
```

Programar um pacote do SSIS usando o SQL Server Agent

Depois de criar a credencial e proxy e conceder ao SSIS acesso ao proxy, você pode criar um trabalho do SQL Server Agent para programar o pacote do SSIS.

Como programar o pacote do SSIS

- Você pode usar SSMS ou T-SQL para criar o trabalho do SQL Server Agent. O exemplo a seguir usa T-SQL.

```
USE [msdb]
GO
DECLARE @jobId BINARY(16)
EXEC msdb.dbo.sp_add_job @job_name=N'MYSSISJob',
@enabled=1,
@notify_level_eventlog=0,
@notify_level_email=2,
@notify_level_page=2,
@delete_level=0,
@category_name=N'[Uncategorized (Local)]',
@job_id = @jobId OUTPUT
GO
EXEC msdb.dbo.sp_add_jobserver @job_name=N'MYSSISJob',@server_name=N'(local)'
GO
```

```
EXEC msdb.dbo.sp_add_jobstep
 @job_name=N'MYSSISJob',@step_name=N'ExecuteSSISPackage',
@step_id=1,
@cmdexec_success_code=0,
@on_success_action=1,
@on_fail_action=2,
@retry_attempts=0,
@retry_interval=0,
@os_run_priority=0,
@subsystem=N'SSIS',
@command=N'/ISSERVER "\"\SSISDB\MySSISFolder\MySSISProject\MySSISPackage.dtsx\"" /
SERVER "\"my-rds-ssis-instance.corp-ad.company.com/\""
/Par "\"$ServerOption::LOGGING_LEVEL(Int16)\"";1 /Par
 "\"$ServerOption::SYNCHRONIZED(Boolean)\"";True /CALLERINFO SQLAGENT /REPORTING
 E',
@database_name=N'master',
@flags=0,
@proxy_name=N'SSIS_Proxy'
GO
```

Revogar o acesso do SSIS do proxy

Você pode revogar o acesso ao subsistema do SSIS e excluir o proxy do SSIS usando os seguintes procedimentos armazenados.

Como revogar o acesso e excluir o proxy

1. Revogue o acesso ao subsistema.

```
USE [msdb]
GO
EXEC msdb.dbo.rds_sqlagent_proxy
 @task_type='REVOKE_SUBSYSTEM_ACCESS',@proxy_name='SSIS_Proxy',@proxy_subsystem='SSIS'
GO
```

2. Revogue as concessões no proxy.

```
USE [msdb]
GO
EXEC msdb.dbo.sp_revoke_login_from_proxy
 @proxy_name=N'SSIS_Proxy',@name=N'mydomain\user_name'
```

```
GO
```

3. Exclua o proxy.

```
USE [msdb]
GO
EXEC dbo.sp_delete_proxy @proxy_name = N'SSIS_Proxy'
GO
```

Desabilitar e descartar o banco de dados do SSIS

Siga estas etapas para desabilitar ou descartar bancos de dados do SSIS:

Tópicos

- Desabilitar o SSIS
- Eliminar o banco de dados SSISDB

Desabilitar o SSIS

Para desabilitar o SSIS, remova a opção `SSIS` do grupo de opções.

> **Important**
>
> Remover a opção não exclui o banco de dados SSISDB, portanto, você pode removê-la com segurança sem perder os projetos do SSIS.
> Você pode reativar a opção `SSIS` após a remoção para reutilizar os projetos do SSIS que foram implantados anteriormente no catálogo do SSIS.

Console

O procedimento a seguir remove a opção `SSIS`.

Como remover a opção SSIS do grupo de opções

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Option groups (Grupos de opções).
3. Selecione o grupo de opções com a opção `SSIS` (`ssis-se-2016` nos exemplos anteriores).

4. Selecione Delete option (Excluir opção).
5. Em Opções de exclusão, selecione SSIS em Opções a serem excluídas.
6. Em Apply immediately (Aplicar imediatamente), selecione Yes (Sim) para excluir a opção imediatamente ou No (Não) para excluí-la na próxima janela de manutenção.
7. Escolha Excluir.

CLI

O procedimento a seguir remove a opção `SSIS`.

Como remover a opção SSIS do grupo de opções

- Execute um dos seguintes comandos:

  Example

  Para Linux, macOS ou Unix:

```
aws rds remove-option-from-option-group \
    --option-group-name ssis-se-2016 \
    --options SSIS \
    --apply-immediately
```

  Para Windows:

```
aws rds remove-option-from-option-group ^
    --option-group-name ssis-se-2016 ^
    --options SSIS ^
    --apply-immediately
```

Eliminar o banco de dados SSISDB

Depois de remover a opção SSIS, o banco de dados SSISDB não é excluído. Para descartar o banco de dados SSISDB, use o procedimento `rds_drop_ssis_database` armazenado depois de remover a opção SSIS.

Como descartar o banco de dados do SSIS

- Use o procedimento armazenado a seguir.

```
USE [msdb]
GO
EXEC dbo.rds_drop_ssis_database
GO
```

Depois de descartar o banco de dados SSISDB, se você reativar a opção SSIS, obterá um novo catálogo do SSISDB.

Suporte para SQL Server Reporting Services no Amazon RDS for SQL Server

O Microsoft SQL Server Reporting Services (SSRS) é um aplicativo baseado em servidor usado para geração e distribuição de relatórios. Ele faz parte de um conjunto de serviços do SQL Server que também inclui o SQL Server Analysis Services (SSAS) e o SQL Server Integration Services (SSIS). O SSRS é um serviço criado com base no SQL Server. É possível usá-lo para coletar dados de várias fontes de dados e apresentá-los de uma forma facilmente compreensível e pronta para análise.

O Amazon RDS for SQL Server oferece suporte à execução de SSRS diretamente em instâncias de banco de dados do RDS. Você pode usar o SSRS em instâncias de banco de dados novas ou existentes.

O RDS oferece suporte ao SSRS para as edições Standard e Enterprise do SQL Server nas seguintes versões:

- SQL Server 2022, todas as versões
- Versão 15.00.4043.16.v1 e posterior do SQL Server 2019
- Versão 14.00.3223.3.v1 e posterior do SQL Server 2017
- SQL Server 2016, versão 13.00.5820.21.v1 e posteriores

Sumário

- Limitações e recomendações
- Ativar o SSRS
  - Criar um grupo de opções para o SSRS
  - Adicionar a opção SSRS ao grupo de opções
  - Associar o grupo de opções à instância de banco de dados
  - Como conceder acesso de entrada ao grupo de segurança da VPC
- Bancos de dados do servidor de relatórios
- Arquivos de log do SSRS
- Acessar o portal da Web do SSRS
  - Como usar SSL no RDS
  - Conceder acesso a usuários de domínio

    - Acessar o portal da Web
- Implantação de relatórios e configuração de fontes de dados de relatórios
    - Implantar relatórios no SSRS
    - Configurar a fonte de dados de relatórios
- Como usar o SSRS Email para enviar relatórios
- Revogar permissões no nível do sistema
- Monitorar o status de uma tarefa
- Desabilitar e excluir bancos de dados do SSRS
    - Desativar o SSRS
    - Excluir os bancos de dados do SSRS

Limitações e recomendações

As seguintes limitações e recomendações aplicam-se à execução do SSRS no RDS for SQL Server:

- Você não pode usar SSRS em instâncias de banco de dados que tenham réplicas de leitura.
- As instâncias devem usar o Active Directory autogerenciado ou AWS Directory Service for Microsoft Active Directory para o portal da web do SSRS e a autenticação de servidor web. Para ter mais informações, consulte Trabalhar com o Active Directory com o RDS para SQL Server.
- Não é possível fazer backup dos bancos de dados do servidor de relatórios criados com a opção SSRS.
- Não há suporte para importar e restaurar bancos de dados do servidor de relatório de outras instâncias do SSRS. Para ter mais informações, consulte Bancos de dados do servidor de relatórios.
- Não é possível configurar o SSRS para escutar na porta SSL padrão (443). Os valores permitidos são 1150–49511, exceto 1234, 1434, 3260, 3343, 3389 e 47001.
- Não há suporte para assinaturas por compartilhamento de arquivos do Microsoft Windows.
- Não há suporte para o uso do Gerenciador de configuração do Reporting Services.
- Não há suporte para a criação e a modificação de funções.
- Não há suporte para a modificação das propriedades do servidor de relatório.
- Não são concedidas funções de administrador e usuário do sistema.
- Não é possível editar atribuições de função no nível do sistema pelo portal da web.

Ativar o SSRS

Use o seguinte processo para ativar o SSRS para a instância de banco de dados:

1. Crie um grupo de opções ou selecione um existente.
2. Adicione a opção SSRS ao grupo de opções.
3. Associe o grupo de opções à instância de banco de dados.
4. Conceda acesso de entrada ao grupo de segurança da nuvem privada virtual (VPC) para a porta do listener do SSRS.

Criar um grupo de opções para o SSRS

Para trabalhar com o SSRS, crie um grupo de opções que corresponda à versão e ao mecanismo do SQL Server da instância de banco de dados que você planeja usar. Para fazer isso, use o AWS Management Console ou a AWS CLI.

> **Note**
>
> Também é possível usar um grupo de opções existente se ele for para a versão e o mecanismo e corretos do SQL Server.

Console

O procedimento a seguir cria um grupo de opções para o SQL Server Standard Edition 2017.

Como criar o grupo de opções

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Option groups (Grupos de opções).
3. Escolha Create group (Criar grupo).
4. No painel Create option group (Criar grupo de opções), faça o seguinte:
   a. Em Name (Nome), insira um nome para o grupo de opções que seja exclusivo na sua Conta da AWS, como **`ssrs-se-2017`**. O nome pode conter somente letras, números e hifens.
   b. Em Descrição, insira uma breve descrição do grupo de opções, como **`SSRS option group for SQL Server SE 2017`**. A descrição é usada para fins de exibição.

c. Em Engine (Mecanismo), selecione sqlserver-se.

d. Em Major engine version (Versão principal do mecanismo), selecione 14.00.

5. Escolha Criar.

CLI

O procedimento a seguir cria um grupo de opções para o SQL Server Standard Edition 2017.

Como criar o grupo de opções

- Execute um dos seguintes comandos:

Example

Para Linux, macOS ou Unix:

```
aws rds create-option-group \
    --option-group-name ssrs-se-2017 \
    --engine-name sqlserver-se \
    --major-engine-version 14.00 \
    --option-group-description "SSRS option group for SQL Server SE 2017"
```

Para Windows:

```
aws rds create-option-group ^
    --option-group-name ssrs-se-2017 ^
    --engine-name sqlserver-se ^
    --major-engine-version 14.00 ^
    --option-group-description "SSRS option group for SQL Server SE 2017"
```

Adicionar a opção SSRS ao grupo de opções

Depois, use o AWS Management Console ou a AWS CLI para adicionar a opção SSRS ao grupo de opções.

Console

Como adicionar a opção SSRS

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Option groups (Grupos de opções).
3. Selecione o grupo de opções que você acabou de criar, depois escolha Add Option (Adicionar opção).
4. Em Detalhes da opção, selecione SSRS em Nome da opção.
5. Em Configurações de opção, faça o seguinte:

   a. Insira a porta de escuta do serviço SSRS. O padrão é 8443. Para obter uma lista de valores permitidos, consulte Limitações e recomendações.

   b. Insira um valor em Máximo de memória.

      O Máximo de memória especifica o limite superior acima do qual não são concedidas novas solicitações de alocação de memória aos aplicativos do servidor de relatório. O número é uma porcentagem da memória total da instância de banco de dados. Os valores permitidos são de 10 a 80.

   c. Em Security groups (Grupos de segurança), selecione o grupo de segurança da VPC a ser associado à opção. Use o mesmo grupo de segurança associado à instância de banco de dados.

6. Para usar o SSRS Email para enviar relatórios, marque a caixa de seleção Configure email delivery options (Configurar opções de entrega por e-mail) em Email delivery in reporting services (Entrega por e-mail em serviços de relatórios), depois faça o seguinte:

   a. Para Sender email address (Endereço de e-mail do remetente), insira o endereço de e-mail a ser usado no campo From (De) das mensagens enviadas pelo SSRS Email.

      Especifique uma conta de usuário que tenha permissão para enviar e-mails do servidor SMTP.

   b. Para SMTP server (Servidor SMTP), especifique o gateway ou servidor SMTP a ser usado.

      Pode ser um endereço IP, o nome NetBIOS de um computador na intranet corporativa ou um nome de domínio totalmente qualificado.

c. Para SMTP port (Porta SMTP), insira a porta a ser usada para se conectar ao servidor de e-mail. O padrão é 25.

d. Para usar a autenticação:

i. Marque a caixa de seleção Use authentication (Usar autenticação).

ii. Para Secret Amazon Resource Name (ARN) (Nome do recurso da Amazon (ARN) do segredo), insira o ARN do AWS Secrets Manager para as credenciais do usuário.

Use o formato a seguir:

`arn:aws:secretsmanager:`*`Region`*`:`*`AccountId`*`:secret:`*`SecretName`*`-6RandomChara`

Por exemplo:

`arn:aws:secretsmanager:`*`us-west-2`*`:`*`123456789012`*`:secret:`*`MySecret-a1b2c3`

Para obter mais informações sobre como criar o segredo, consulte Como usar o SSRS Email para enviar relatórios.

e. Marque a caixa de seleção Use Secure Sockets Layer (SSL) [Usar Secure Sockets Layer (SSL)] para criptografar as mensagens de e-mail usando SSL.

7. Em Scheduling (Programação), escolha se deseja adicionar a opção imediatamente ou na próxima janela de manutenção.

8. Escolha Add option (Adicionar opção).

CLI

Como adicionar a opção SSRS

1. Crie um arquivo JSON; por exemplo, `ssrs-option.json`.

a. Defina os seguintes parâmetros necessários:

- `OptionGroupName` – O nome do grupo de opções que você criou ou selecionou anteriormente (`ssrs-se-2017` no exemplo a seguir).
- `Port` – A porta de escuta do serviço SSRS. O padrão é 8443. Para obter uma lista de valores permitidos, consulte Limitações e recomendações.

- `VpcSecurityGroupMemberships` – Associações de grupos de segurança da VPC para a instância de banco de dados do RDS.
- `MAX_MEMORY` – O limite superior acima do qual não são concedidas novas solicitações de alocação de memória aos aplicativos do servidor de relatório. O número é uma porcentagem da memória total da instância de banco de dados. Os valores permitidos são de 10 a 80.

b. (Opcional) Defina os seguintes parâmetros para usar o SSRS Email:

- `SMTP_ENABLE_EMAIL`: defina como `true` para usar o SSRS Email. O padrão é `false`.
- `SMTP_SENDER_EMAIL_ADDRESS`: o endereço de e-mail a ser usado no campo From (De) das mensagens enviadas pelo SSRS Email. Especifique uma conta de usuário que tenha permissão para enviar e-mails do servidor SMTP.
- `SMTP_SERVER`: o gateway ou servidor SMTP a ser usado. Pode ser um endereço IP, o nome NetBIOS de um computador na intranet corporativa ou um nome de domínio totalmente qualificado.
- `SMTP_PORT`: a porta a ser usada para se conectar ao servidor de e-mail. O padrão é 25.
- `SMTP_USE_SSL`: defina como `true` para criptografar as mensagens de e-mail usando SSL. O padrão é `true`.
- `SMTP_EMAIL_CREDENTIALS_SECRET_ARN`: o ARN do Secrets Manager que contém as credenciais do usuário. Use o formato a seguir:

  `arn:aws:secretsmanager:`*`Region`*`:`*`AccountId`*`:secret:`*`SecretName-6RandomCharact`

  Para obter mais informações sobre como criar o segredo, consulte Como usar o SSRS Email para enviar relatórios.
- `SMTP_USE_ANONYMOUS_AUTHENTICATION`: defina como `true` e não inclua `SMTP_EMAIL_CREDENTIALS_SECRET_ARN` se não quiser usar a autenticação.

  O padrão é `false` quando `SMTP_ENABLE_EMAIL` é `true`.

O exemplo a seguir inclui os parâmetros de SSRS Email usando o ARN do segredo.

```
{
"OptionGroupName": "ssrs-se-2017",
"OptionsToInclude": [
 {
```

```
 "OptionName": "SSRS",
 "Port": 8443,
 "VpcSecurityGroupMemberships": ["sg-0abcdef123"],
 "OptionSettings": [
            {"Name": "MAX_MEMORY","Value": "60"},
            {"Name": "SMTP_ENABLE_EMAIL","Value": "true"}
            {"Name": "SMTP_SENDER_EMAIL_ADDRESS","Value": "nobody@example.com"},
            {"Name": "SMTP_SERVER","Value": "email-smtp.us-west-2.amazonaws.com"},
            {"Name": "SMTP_PORT","Value": "25"},
            {"Name": "SMTP_USE_SSL","Value": "true"},
            {"Name": "SMTP_EMAIL_CREDENTIALS_SECRET_ARN","Value":
 "arn:aws:secretsmanager:us-west-2:123456789012:secret:MySecret-a1b2c3"}
            ]
 }],
"ApplyImmediately": true
}
```

2. Adicione a opção SSRS ao grupo de opções.

   Example

   Para Linux, macOS ou Unix:

   ```
   aws rds add-option-to-option-group \
       --cli-input-json file://ssrs-option.json \
       --apply-immediately
   ```

   Para Windows:

   ```
   aws rds add-option-to-option-group ^
       --cli-input-json file://ssrs-option.json ^
       --apply-immediately
   ```

Associar o grupo de opções à instância de banco de dados

Use o AWS Management Console ou a AWS CLI para associar o grupo de opções à instância de banco de dados.

Se você usar uma instância de banco de dados existente, ela já deverá ter um domínio do Active Directory e uma função do AWS Identity and Access Management (IAM) associados a ela. Se você criar uma nova instância, especifique um domínio do Active Directory e uma função do IAM

existentes. Para ter mais informações, consulte Trabalhar com o Active Directory com o RDS para SQL Server.

Console

É possível associar o grupo de opções a uma instância de banco de dados nova ou existente:

- Para uma nova instância de banco de dados, associe o grupo de opções ao executar a instância. Para obter mais informações, consulte Criar uma instância de banco de dados do Amazon RDS.
- Para uma instância de banco de dados existente, modifique a instância e associe o novo grupo de opções. Para obter mais informações, consulteModificar uma instância de banco de dados do Amazon RDS

CLI

É possível associar o grupo de opções a uma instância de banco de dados nova ou existente.

Como criar uma instância de banco de dados que usa o grupo de opções

- Especifique o mesmo tipo de mecanismo de banco de dados e a versão principal que você usou ao criar o grupo de opções.

  Example

  Para Linux, macOS ou Unix:

```
aws rds create-db-instance \
    --db-instance-identifier myssrsinstance \
    --db-instance-class db.m5.2xlarge \
    --engine sqlserver-se \
    --engine-version 14.00.3223.3.v1 \
    --allocated-storage 100 \
    --manage-master-user-password  \
    --master-username admin \
    --storage-type gp2 \
    --license-model li \
    --domain-iam-role-name my-directory-iam-role \
    --domain my-domain-id \
    --option-group-name ssrs-se-2017
```

  Para Windows:

```
aws rds create-db-instance ^
    --db-instance-identifier myssrsinstance ^
    --db-instance-class db.m5.2xlarge ^
    --engine sqlserver-se ^
    --engine-version 14.00.3223.3.v1 ^
    --allocated-storage 100 ^
    --manage-master-user-password ^
    --master-username admin ^
    --storage-type gp2 ^
    --license-model li ^
    --domain-iam-role-name my-directory-iam-role ^
    --domain my-domain-id ^
    --option-group-name ssrs-se-2017
```

Como modificar uma instância de banco de dados para usar o grupo de opções

- Execute um dos seguintes comandos:

  Example

  Para Linux, macOS ou Unix:

  ```
  aws rds modify-db-instance \
      --db-instance-identifier myssrsinstance \
      --option-group-name ssrs-se-2017 \
      --apply-immediately
  ```

  Para Windows:

  ```
  aws rds modify-db-instance ^
      --db-instance-identifier myssrsinstance ^
      --option-group-name ssrs-se-2017 ^
      --apply-immediately
  ```

Como conceder acesso de entrada ao grupo de segurança da VPC

Para permitir o acesso de entrada ao grupo de segurança da VPC associado à instância de banco de dados, crie uma regra de entrada para a porta especificada do listener do SSRS. Para obter mais

informações sobre como configurar um grupo de segurança, consulte Fornecer acesso à instância de banco de dados na VPC criando um grupo de segurança.

Bancos de dados do servidor de relatórios

Quando a instância de banco de dados está associada à opção SSRS, dois novos bancos de dados são criados na instância de banco de dados:

- `rdsadmin_ReportServer`
- `rdsadmin_ReportServerTempDB`

Esses bancos de dados atuam como os bancos de dados ReportServer e ReportServerTempDB. O SSRS armazena os dados no banco de dados ReportServer e armazena em cache os dados no banco de dados ReportServerTempDB. Para obter mais informações, consulte Bancos de dados do servidor de relatórios.

O RDS possui e gerencia esses bancos de dados, portanto, não são permitidas operações de banco de dados neles, como ALTER e DROP. O acesso não é permitido no banco de dados `rdsadmin_ReportServerTempDB`. No entanto, é possível executar operações de leitura no banco de dados `rdsadmin_ReportServer`.

Arquivos de log do SSRS

É possível listar, visualizar e baixar arquivos de log do SSRS. Os arquivos de log do SSRS seguem uma convenção de nomenclatura de ReportServerService_*timestamp*.log. Esses logs do servidor de relatórios podem ser encontrados no diretório `D:\rdsdbdata\Log\SSRS`. (O diretório `D:\rdsdbdata\Log` também é o diretório pai dos logs de erros e dos logs do SQL Server Agent.) Para ter mais informações, consulte Como visualizar e listar arquivos de log do banco de dados.

Para instâncias SSRS existentes, a reinicialização do serviço SSRS pode ser necessária para acessar os logs do servidor de relatório. É possível reiniciar o serviço atualizando a opção `SSRS`.

Para obter mais informações, consulteTrabalhar com logs do Amazon RDS para Microsoft SQL Server

Acessar o portal da Web do SSRS

Use o seguinte processo para acessar o portal da web do SSRS:

1. Ative o Secure Sockets Layer (SSL).
2. Conceda acesso a usuários de domínio.
3. Acesse o portal da web usando um navegador e as credenciais do usuário do domínio.

Como usar SSL no RDS

O SSRS usa o protocolo SSL HTTPS para as conexões. Para trabalhar com este protocolo, importe um certificado SSL para o sistema operacional Microsoft Windows no computador cliente.

Para obter mais informações sobre certificados SSL, consulte Usar SSL/TLS para criptografar uma conexão com uma instância ou um cluster de banco de dados. Para obter mais informações sobre como usar o SSL com o SQL Server, consulte ???.

Conceder acesso a usuários de domínio

Em uma nova ativação do SSRS, não há atribuições de função no SSRS. Para conceder a um usuário de domínio ou um grupo de usuários acesso ao portal da web, o RDS fornece um procedimento armazenado.

Como conceder acesso a um usuário de domínio no portal da web

- Use o procedimento armazenado a seguir.

```
exec msdb.dbo.rds_msbi_task
@task_type='SSRS_GRANT_PORTAL_PERMISSION',
@ssrs_group_or_username=N'AD_domain\user';
```

O usuário de domínio ou o grupo de usuários recebe a função `RDS_SSRS_ROLE` do sistema. Esta função tem as seguintes tarefas no nível do sistema concedidas a ela:

- Executar relatórios
- Gerenciar trabalhos
- Gerenciar agendas compartilhadas
- Visualizar agendas compartilhadas

A função no nível do item de `Content Manager` na pasta raiz também é concedida.

Acessar o portal da Web

Depois que a tarefa `SSRS_GRANT_PORTAL_PERMISSION` for concluída com êxito, você terá acesso ao portal usando um navegador da web. O URL do portal da web tem o formato a seguir.

```
https://rds_endpoint:port/Reports
```

Neste formato, aplica-se o seguinte item:

- *`rds_endpoint`* – o endpoint para a instância de banco de dados do RDS que você está usando com o SSRS.

  É possível encontrar o endpoint na guia Conectividade e segurança da instância de banco de dados. Para obter mais informações, consulte Conectar-se à instância de banco de dados do Microsoft SQL Server.
- *`port`* – A porta do listener para o SSRS definida na opção `SSRS`.

Como acessar o portal da web

1. Insira o URL do portal da web no navegador.

   ```
   https://myssrsinstance.cg034itsfake.us-east-1.rds.amazonaws.com:8443/Reports
   ```
2. Faça login com as credenciais de um usuário de domínio ao qual você concedeu acesso com a tarefa `SSRS_GRANT_PORTAL_PERMISSION`.

Implantação de relatórios e configuração de fontes de dados de relatórios

Use os procedimentos a seguir para implantar relatórios no SSRS e configurar as fontes de dados de relatórios:

Tópicos

- Implantar relatórios no SSRS
- Configurar a fonte de dados de relatórios

Implantar relatórios no SSRS

Após ter acesso ao portal da web, você poderá implantar relatórios nele. É possível usar a ferramenta de upload no portal da web para fazer upload de relatórios ou implantar diretamente do SQL Server Data Tools (SSDT). Ao implantar pelo SSDT, verifique o seguinte:

- O usuário que executou o SSDT tem acesso ao portal da web do SSRS.
- O valor `TargetServerURL` nas propriedades do projeto do SSRS é definido como o endpoint HTTPS da instância de banco de dados do RDS com o sufixo `ReportServer`, por exemplo:

```
https://myssrsinstance.cg034itsfake.us-east-1.rds.amazonaws.com:8443/ReportServer
```

Configurar a fonte de dados de relatórios

Depois de implantar um relatório no SSRS, você deve configurar a fonte de dados do relatório. Ao configurar a fonte de dados do relatório, assegure-se do seguinte:

- Para instâncias de banco de dados do RDS para SQL Server associadas ao AWS Directory Service for Microsoft Active Directory, use o nome de domínio totalmente qualificado (FQDN) como o nome da fonte de dados da string de conexão. Um exemplo é *`myssrsinstance.corp-ad.example.com`*, onde *`myssrsinstance`* é o nome da instância de banco de dados e *`corp-ad.example.com`* é o nome de domínio totalmente qualificado.
- Para instâncias de banco de dados do RDS para SQL Server associadas ao Active Directory autogerenciado, use `.` ou *`LocalHost`* como o nome da fonte de dados da string de conexão.

Como usar o SSRS Email para enviar relatórios

O SSRS inclui a extensão SSRS Email que pode ser usada para enviar relatórios aos usuários.

Para configurar o SSRS Email, use as configurações de opções do SSRS. Para obter mais informações, consulte Adicionar a opção SSRS ao grupo de opções.

Depois de configurar o SSRS Email, você pode assinar relatórios no servidor de relatórios. Para obter mais informações, consulte Entrega de email no Reporting Services na documentação da Microsoft.

A integração com o AWS Secrets Manager é necessária para que o SSRS Email funcione no RDS. Para realizar a integração com o Secrets Manager, crie um segredo.

> **Note**
>
> Se você alterar o segredo posteriormente, também precisará atualizar a opção SSRS no grupo de opções.

Como criar um segredo para o SSRS Email

1. Siga as etapas em Criar um segredo no Guia do usuário do AWS Secrets Manager.

   a. Em Selecionar Tipo de Segredo, selecione Outro Tipo de Segredo.

   b. Para Key/value pairs (Pares de chave/valor), insira o seguinte:

      - **`SMTP_USERNAME`**: insira um usuário com permissão para enviar e-mails do servidor SMTP.
      - **`SMTP_PASSWORD`**: insira uma senha para o usuário do SMTP.

   c. Para Encryption key (Chave de criptografia), não use o padrão AWS KMS key. Use sua própria chave existente ou crie uma chave.

      A política de chaves do KMS deve permitir que a ação `kms:Decrypt`, por exemplo:

```
{
    "Sid": "Allow use of the key",
    "Effect": "Allow",
    "Principal": {
        "Service": [
            "rds.amazonaws.com"
        ]
    },
    "Action": [
        "kms:Decrypt"
    ],
    "Resource": "*"
}
```

2. Siga as etapas em Anexar uma política de permissões a um segredo no Guia do usuário do AWS Secrets Manager. A política de permissões fornece a ação `secretsmanager:GetSecretValue` à entidade principal do serviço `rds.amazonaws.com`.

Recomendamos que você use as condições `aws:sourceAccount` e `aws:sourceArn` na política para evitar o problema de representante confuso. Use sua Conta da AWS para `aws:sourceAccount` e o ARN do grupo de opções para `aws:sourceArn`. Para obter mais informações, consulte Prevenção do problema do substituto confuso entre serviços.

A seguir, um exemplo de uma política de permissões.

JSON

```
{
  "Version" : "2012-10-17",
  "Statement" : [ {
    "Effect" : "Allow",
    "Principal" : {
      "Service" : "rds.amazonaws.com"
    },
    "Action" : "secretsmanager:GetSecretValue",
    "Resource" : "*",
    "Condition" : {
      "StringEquals" : {
        "aws:sourceAccount" : "123456789012"
      },
      "ArnLike" : {
        "aws:sourceArn" : "arn:aws:rds:us-west-2:123456789012:og:ssrs-
se-2017"
      }
    }
  } ]
}
```

Para obter mais exemplos, consulte Exemplos de políticas de permissões para o AWS Secrets Manager no Guia do usuário do AWS Secrets Manager.

Revogar permissões no nível do sistema

A função `RDS_SSRS_ROLE` do sistema não tem permissões suficientes para excluir atribuições de função no nível do sistema. Para remover um usuário ou um grupo de usuários da `RDS_SSRS_ROLE`,

use o mesmo procedimento armazenado que você usou para conceder a função, mas use o tipo de tarefa `SSRS_REVOKE_PORTAL_PERMISSION`.

Como revogar o acesso de um usuário de domínio para o portal da web

- Use o procedimento armazenado a seguir.

```
exec msdb.dbo.rds_msbi_task
@task_type='SSRS_REVOKE_PORTAL_PERMISSION',
@ssrs_group_or_username=N'AD_domain\user';
```

Fazer isso exclui o usuário da função `RDS_SSRS_ROLE` do sistema. Se o usuário tiver uma, isso também o exclui da função `Content Manager` no nível do item.

Monitorar o status de uma tarefa

Para acompanhar o status da tarefa de concessão ou revogação, chame a função `rds_fn_task_status`. Ela leva dois parâmetros. O primeiro parâmetro sempre deve ser `NULL` porque ele não se aplica ao SSRS. O segundo parâmetro aceita um ID de tarefa.

Para ver uma lista de todas as tarefas, defina o primeiro parâmetro como `NULL` e o segundo parâmetro como `0`, conforme mostrado no seguinte exemplo.

```
SELECT * FROM msdb.dbo.rds_fn_task_status(NULL,0);
```

Para obter uma tarefa específica, defina o primeiro parâmetro como `NULL` e o segundo parâmetro como o ID da tarefa, conforme mostrado no exemplo a seguir.

```
SELECT * FROM msdb.dbo.rds_fn_task_status(NULL,42);
```

A função `rds_fn_task_status` retorna as informações a seguir.

Parâmetro de saída	Descrição
`task_id`	O ID da tarefa.
`task_type`	No SSRS, as tarefas podem ter os seguintes tipos de tarefa:

Parâmetro de saída	Descrição
	• SSRS_GRANT_PORTAL_PERMISSION • SSRS_REVOKE_PORTAL_PERMISSION
database_name	Não aplicável a tarefas do SSRS.
% complete	O progresso da tarefa em porcentagem.
duration (mins)	A quantidade de tempo gasta na tarefa, em minutos.

Parâmetro de saída	Descrição
`lifecycle`	O status da tarefa. Os possíveis status são os seguintes: • `CREATED` – Após chamar um dos procedimentos armazenados do SSRS, uma tarefa será criada, e o status será definido como `CREATED`. • `IN_PROGRESS` – Após iniciar uma tarefa, o status será definido como `IN_PROGRESS`. Pode demorar até 5 minutos para que o status mude de `CREATED` para `IN_PROGRESS`. • `SUCCESS` – Após concluir uma tarefa, o status será definido como `SUCCESS`. • `ERROR` – Se uma tarefa falhar, o status será definido como `ERROR`. Para obter mais informações sobre o erro, consulte a coluna `task_info`. • `CANCEL_REQUESTED` – Após chamar procedimento armazenado `rds_cancel_task`, o status da tarefa será definido como `CANCEL_REQUESTED`. • `CANCELLED` – após uma tarefa ter sido cancelada com sucesso, o status da tarefa será definido como `CANCELLED`.
`task_info`	Informações adicionais sobre a tarefa. Se um erro ocorrer durante o processamento, essa coluna conterá informações sobre o erro.

Parâmetro de saída	Descrição
`last_updated`	A data e hora em que o status da tarefa foi atualizado pela última vez.
`created_at`	A data e hora em que a tarefa foi criada.
`S3_object_arn`	Não aplicável a tarefas do SSRS.
`overwrite_S3_backup_file`	Não aplicável a tarefas do SSRS.
`KMS_master_key_arn`	Não aplicável a tarefas do SSRS.
`filepath`	Não aplicável a tarefas do SSRS.
`overwrite_file`	Não aplicável a tarefas do SSRS.
`task_metadata`	Metadados associados à tarefa do SSRS.

Desabilitar e excluir bancos de dados do SSRS

Use os seguintes procedimentos para desabilitar o SSRS e excluir bancos de dados do SSRS:

Tópicos

- Desativar o SSRS
- Excluir os bancos de dados do SSRS

Desativar o SSRS

Para desativar o SSRS, remova a opção `SSRS` do grupo de opções. Remover a opção não exclui os bancos de dados do SSRS. Para ter mais informações, consulte Excluir os bancos de dados do SSRS.

Você pode ativar o SSRS novamente adicionando a opção `SSRS` de volta. Se você também excluiu os bancos de dados do SSRS, adicionar novamente a opção na mesma instância de banco de dados criará novos bancos de dados do servidor de relatórios.

Console

Como remover a opção SSRS do grupo de opções

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Option groups (Grupos de opções).
3. Selecione o grupo de opções com a opção `SSRS` (`ssrs-se-2017` nos exemplos anteriores).
4. Selecione Delete option (Excluir opção).
5. Em Opções de exclusão, selecione SSRS em Opções a serem excluídas.
6. Em Apply immediately (Aplicar imediatamente), selecione Yes (Sim) para excluir a opção imediatamente ou No (Não) para excluí-la na próxima janela de manutenção.
7. Escolha Excluir.

CLI

Como remover a opção SSRS do grupo de opções

- Execute um dos seguintes comandos:

  Example

  Para Linux, macOS ou Unix:

```
aws rds remove-option-from-option-group \
    --option-group-name ssrs-se-2017 \
    --options SSRS \
    --apply-immediately
```

  Para Windows:

```
aws rds remove-option-from-option-group ^
    --option-group-name ssrs-se-2017 ^
    --options SSRS ^
    --apply-immediately
```

Excluir os bancos de dados do SSRS

Remover a opção SSRS não exclui os bancos de dados do servidor de relatório. Para excluí-los, use o procedimento armazenado a seguir.

Para excluir os bancos de dados do servidor de relatório, certifique-se de remover a opção SSRS primeiro.

Como excluir os bancos de dados do SSRS

- Use o procedimento armazenado a seguir.

```
exec msdb.dbo.rds_drop_ssrs_databases
```

Suporte para o Microsoft Distributed Transaction Coordinator no RDS for SQL Server

Uma transação distribuída é uma transação de banco de dados na qual dois ou mais hosts de rede estão envolvidos. O Amazon RDS for SQL Server oferece suporte a transações distribuídas entre hosts. Um host único pode ser o seguinte:

- Instância de banco de dados do RDS for SQL Server
- Host do SQL Server no local
- O host do Amazon EC2 com o SQL Server instalado
- Qualquer outro host do EC2 ou instância de banco de dados do RDS com um mecanismo de banco de dados que ofereça suporte a transações distribuídas

No RDS, a começar pelo SQL Server 2012 (versão 11.00.5058.0.v1 e posterior), todas as edições do RDS for SQL Server oferecem suporte a transações distribuídas. O suporte é fornecido usando o MSDTC (Microsoft Distributed Transaction Coordinator). Para obter informações detalhadas sobre o MSDTC, consulte Distributed Transaction Coordinator na documentação da Microsoft.

Sumário

- Limitações
- Habilitar o MSDTC
  - Criar o grupo de opções para o MSDTC
  - Adicionar a opção do MSDTC ao grupo de opções
  - Criar o grupo de parâmetros para o MSDTC
  - Modificar o parâmetro para o MSDTC
  - Associar o grupo de opções e o grupo de parâmetros à instância de banco de dados.
  - Modificar a opção do MSDTC
- Usar transações
  - Usar transações distribuídas
  - Como usar transações XA
  - Usar rastreamento de transações
- Desabilitar o MSDTC
- Solução de problemas do MSDTC para o RDS for SQL Server

Limitações

As seguintes limitações se aplicam ao uso do MSDTC no RDS for SQL Server:

- Não há suporte para o MSDTC em instâncias que usam o SQL Server Database Mirroring. Para obter mais informações, consulte Transações – grupos de disponibilidade e espelhamento de banco de dados.
- O parâmetro `in-doubt xact resolution` deve ser definido como 1 ou 2. Para ter mais informações, consulte Modificar o parâmetro para o MSDTC.
- O MSDTC requer que todos os hosts participantes em transações distribuídas sejam resolvidos usando os nomes de host. O RDS mantém automaticamente essa funcionalidade para instâncias associadas ao domínio. No entanto, para instâncias autônomas, configure o servidor DNS manualmente.
- As transações XA do Java Database Connectivity (JDBC) são compatíveis com a versão 14.00.3223.3 e superior do SQL Server 2017 e o SQL Server 2019.
- Não há suporte para transações distribuídas que dependem de bibliotecas de links dinâmicos (DLLs) do cliente em instâncias do RDS.
- Não há suporte para o uso de bibliotecas de links dinâmicos XA personalizadas.

Habilitar o MSDTC

Use o seguinte processo para habilitar o MSDTC para a instância de banco de dados:

1. Crie um grupo de opções ou selecione um existente.
2. Adicione a opção `MSDTC` ao grupo de opções.
3. Crie um novo grupo de parâmetros ou escolha um existente.
4. Modifique o grupo de parâmetros para definir o parâmetro `in-doubt xact resolution` como 1 ou 2.
5. Associe o grupo de opções e o grupo de parâmetros à instância de banco de dados.

Criar o grupo de opções para o MSDTC

Use o AWS Management Console ou a AWS CLI para criar um grupo de opções que corresponda à versão e ao mecanismo do SQL Server da instância de banco de dados.

> **Note**
>
> Também é possível usar um grupo de opções existente se ele for para a versão e o mecanismo e corretos do SQL Server.

Console

O procedimento a seguir cria um grupo de opções para o SQL Server Standard Edition 2016.

Como criar o grupo de opções

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Option groups (Grupos de opções).
3. Escolha Create group (Criar grupo).
4. No painel Create option group (Criar grupo de opções), faça o seguinte:
   a. Em Name (Nome), insira um nome para o grupo de opções que seja exclusivo na sua conta da AWS, como **`msdtc-se-2016`**. O nome pode conter somente letras, números e hifens.
   b. Em Descrição, insira uma breve descrição do grupo de opções, como **`MSDTC option group for SQL Server SE 2016`**. A descrição é usada para fins de exibição.
   c. Em Engine (Mecanismo), selecione sqlserver-se.
   d. Em Versão do mecanismo principal, selecione 13.00.
5. Escolha Criar.

CLI

O exemplo a seguir cria um grupo de opções para o SQL Server Standard Edition 2016.

Como criar o grupo de opções

- Use um dos seguintes comandos.

  Example

  Para Linux, macOS ou Unix:

```
aws rds create-option-group \
    --option-group-name msdtc-se-2016 \
    --engine-name sqlserver-se \
    --major-engine-version 13.00 \
    --option-group-description "MSDTC option group for SQL Server SE 2016"
```

Para Windows:

```
aws rds create-option-group ^
    --option-group-name msdtc-se-2016 ^
    --engine-name sqlserver-se ^
    --major-engine-version 13.00 ^
    --option-group-description "MSDTC option group for SQL Server SE 2016"
```

Adicionar a opção do MSDTC ao grupo de opções

Depois, use o AWS Management Console ou a AWS CLI para adicionar a opção `MSDTC` ao grupo de opções.

São necessárias as seguintes configurações de opção:

- Porta – a porta que você usa para acessar o MSDTC. Os valores permitidos são 1150–49151, exceto 1234, 1434, 3260, 3343, 3389 e 47001. O valor padrão é 5000.

  Verifique se a porta que pretende usar está habilitada nas regras de firewall. Além disso, conforme necessário, verifique se essa porta está habilitada nas regras de entrada e saída para o grupo de segurança associado à instância de banco de dados. Para ter mais informações, consulte Não é possível conectar-se à instância de banco de dados do Amazon RDS.
- Security groups (Grupos de segurança): as associações de grupo de segurança da VPC para a instância de banco de dados do RDS.
- Tipo de autenticação – o modo de autenticação entre hosts. Os seguintes tipos de autenticação são compatíveis:
  - Mútuo – as instâncias do RDS são autenticadas mutuamente entre si usando autenticação integrada. Se esta opção estiver selecionada, todas as instâncias associadas a este grupo de opções deverão estar associadas ao domínio.
  - Nenhum – nenhuma autenticação é realizada entre hosts. Não recomendamos usar esse modo em ambientes de produção.

- Tamanho do log de transações – o tamanho do log de transações do MSDTC. Os valores permitidos são de 4 a 1024 MB. O tamanho padrão é 4 MB.

As seguintes configurações de opção são opcionais:

- Habilitar conexões de entrada – se deseja permitir conexões de entrada do MSDTC para instâncias associadas a este grupo de opções.
- Habilitar conexões de saída – se deseja permitir conexões de saída do MSDTC em instâncias associadas a este grupo de opções.
- Habilitar XA – se deseja permitir transações XA. Para obter mais informações sobre o protocolo XA, consulte XA Specification.
- Habilitar SNA LU – se deseja permitir que o protocolo SNA LU seja usado para transações distribuídas. Para obter mais informações sobre o suporte ao protocolo SNA LU, consulte Managing IBM CICS LU 6.2 Transactions na documentação da Microsoft.

Console

Como adicionar a opção do MSDTC

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Option groups (Grupos de opções).
3. Selecione o grupo de opções que você acabou de criar.
4. Escolha Add option (Adicionar opção).
5. Em Detalhes da opção, selecione MSDTC em Nome da opção.
6. Em Configurações de opção:
    a. Em Porta, insira o número da porta para acessar o MSDTC. O padrão é 5000.
    b. Em Security groups (Grupos de segurança), selecione o grupo de segurança da VPC a ser associado à opção.
    c. Em Tipo de autenticação, escolha Mútuo ou Nenhum.
    d. Em Tamanho do log de transações, insira um valor de 4 a 1024. O padrão é 4.
7. Em Configuração adicional, faça o seguinte:

a. Em Conexões, conforme necessário, escolha Habilitar conexões de entrada e Habilitar conexões de saída.

b. Em Protocolos permitidos, conforme necessário, escolha Habilitar XA e Habilitar SNA LU.

8. Em Scheduling (Programação), escolha se deseja adicionar a opção imediatamente ou na próxima janela de manutenção.
9. Escolha Add option (Adicionar opção).

   Nenhuma reinicialização é necessária para adicionar essa opção.

CLI

Como adicionar a opção do MSDTC

1. Crie um arquivo JSON, por exemplo `msdtc-option.json`, com os seguintes parâmetros necessários:

```
{
"OptionGroupName":"msdtc-se-2016",
"OptionsToInclude": [
 {
 "OptionName":"MSDTC",
 "Port":5000,
 "VpcSecurityGroupMemberships":["sg-0abcdef123"],
 "OptionSettings":[{"Name":"AUTHENTICATION","Value":"MUTUAL"},
{"Name":"TRANSACTION_LOG_SIZE","Value":"4"}]
 }],
"ApplyImmediately": true
}
```

2. Adicione a opção `MSDTC` ao grupo de opções.

   Example

   Para Linux, macOS ou Unix:

```
aws rds add-option-to-option-group \
    --cli-input-json file://msdtc-option.json \
    --apply-immediately
```

Para Windows:

```
aws rds add-option-to-option-group ^
    --cli-input-json file://msdtc-option.json ^
    --apply-immediately
```

Nenhuma reinicialização é necessária.

Criar o grupo de parâmetros para o MSDTC

Crie ou modifique um grupo de parâmetros para o parâmetro `in-doubt xact resolution` que corresponde à edição e à versão do SQL Server da instância de banco de dados.

Console

O exemplo a seguir cria um grupo de parâmetros para o SQL Server Standard Edition 2016.

Como criar o grupo de parâmetros

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Parameter groups (Grupos de parâmetros).
3. Escolha Create parameter group (Criar parameter group).
4. No painelCreate parameter group (Criar parameter group), faça o seguinte:
   a. Em Família de grupos de parâmetros, escolha sqlserver-se-13.0.
   b. Em Nome do grupo, insira um identificador para o grupo de parâmetros, como **`msdtc-sqlserver-se-13`**.
   c. Em Descrição, insira **`in-doubt xact resolution`**.
5. Escolha Criar.

CLI

O exemplo a seguir cria um grupo de parâmetros para o SQL Server Standard Edition 2016.

Como criar o grupo de parâmetros

- Use um dos seguintes comandos.

Example

Para Linux, macOS ou Unix:

```
aws rds create-db-parameter-group \
    --db-parameter-group-name msdtc-sqlserver-se-13 \
    --db-parameter-group-family "sqlserver-se-13.0" \
    --description "in-doubt xact resolution"
```

Para Windows:

```
aws rds create-db-parameter-group ^
    --db-parameter-group-name msdtc-sqlserver-se-13 ^
    --db-parameter-group-family "sqlserver-se-13.0" ^
    --description "in-doubt xact resolution"
```

Modificar o parâmetro para o MSDTC

Modifique o parâmetro `in-doubt xact resolution` no grupo de parâmetros que corresponde à edição e à versão do SQL Server da instância de banco de dados.

Para o MSDTC, defina o parâmetro `in-doubt xact resolution` como um dos seguintes procedimentos:

- 1: `Presume commit`. Presume-se que as transações em dúvida do MSDTC foram confirmadas.
- 2: `Presume abort`. Presume-se que as transações em dúvida do MSDTC foram interrompidas.

Para obter mais informações, consulte in-doubt xact resolution server configuration option na documentação da Microsoft.

Console

O exemplo a seguir modifica o grupo de parâmetros que você criou para o SQL Server Standard Edition 2016.

Como modificar o grupo de parâmetros

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.

2. No painel de navegação, selecione Parameter groups.
3. Escolha o grupo de parâmetros, como msdtc-sqlserver-se-13.
4. Em Parâmetros, filtre a lista de parâmetros para **`xact`**.
5. Escolha resolução xact em dúvida.
6. Escolha Edit parameters.
7. Insira **1** ou **2**.
8. Escolha Salvar alterações.

CLI

O exemplo a seguir modifica o grupo de parâmetros que você criou para o SQL Server Standard Edition 2016.

Como modificar o grupo de parâmetros

- Use um dos seguintes comandos.

  Example

  Para Linux, macOS ou Unix:

```
aws rds modify-db-parameter-group \
    --db-parameter-group-name msdtc-sqlserver-se-13 \
    --parameters "ParameterName='in-doubt xact
 resolution',ParameterValue=1,ApplyMethod=immediate"
```

  Para Windows:

```
aws rds modify-db-parameter-group ^
    --db-parameter-group-name msdtc-sqlserver-se-13 ^
    --parameters "ParameterName='in-doubt xact
 resolution',ParameterValue=1,ApplyMethod=immediate"
```

Associar o grupo de opções e o grupo de parâmetros à instância de banco de dados.

É possível usar o AWS Management Console ou a AWS CLI para associar o grupo de opções e o grupo de parâmetros do MSDTC à instância de banco de dados.

Console

É possível associar o grupo de opções e o grupo de parâmetros do MSDTC a uma instância de banco de dados nova ou existente.

- Para uma nova instância de banco de dados, associe-os ao executar a instância. Para obter mais informações, consulte Criar uma instância de banco de dados do Amazon RDS.
- Para uma instância de banco de dados existente, associe-os modificando a instância. Para obter mais informações, consulte Modificar uma instância de banco de dados do Amazon RDS.

> **Note**
>
> Se você usar uma instância de banco de dados existente associada ao domínio, ela já deverá ter um domínio do Active Directory e uma função do AWS Identity and Access Management (IAM) associados a ela. Se você criar uma nova instância associada ao domínio, especifique um domínio do Active Directory e uma função do IAM existentes. Para ter mais informações, consulte Trabalhar com o AWS Managed Active Directory com o RDS para SQL Server.

CLI

É possível associar o grupo de opções e o grupo de parâmetros do MSDTC a uma instância de banco de dados nova ou existente.

> **Note**
>
> Se você usar uma instância de banco de dados existente associada ao domínio, ela já deverá ter um domínio do Active Directory e uma função do IAM associados a ela. Se você criar uma nova instância associada ao domínio, especifique um domínio do Active Directory e uma função do IAM existentes. Para obter mais informações, consulte Trabalhar com o AWS Managed Active Directory com o RDS para SQL Server.

Como criar uma instância de banco de dados com o grupo de opções e o grupo de parâmetros do MSDTC

- Especifique o mesmo tipo de mecanismo de banco de dados e a versão principal que você usou ao criar o grupo de opções.

Example

Para Linux, macOS ou Unix:

```
aws rds create-db-instance \
    --db-instance-identifier mydbinstance \
    --db-instance-class db.m5.2xlarge \
    --engine sqlserver-se \
    --engine-version 13.00.5426.0.v1 \
    --allocated-storage 100 \
    --manage-master-user-password \
    --master-username admin \
    --storage-type gp2 \
    --license-model li \
    --domain-iam-role-name my-directory-iam-role \
    --domain my-domain-id \
    --option-group-name msdtc-se-2016 \
    --db-parameter-group-name msdtc-sqlserver-se-13
```

Para Windows:

```
aws rds create-db-instance ^
    --db-instance-identifier mydbinstance ^
    --db-instance-class db.m5.2xlarge ^
    --engine sqlserver-se ^
    --engine-version 13.00.5426.0.v1 ^
    --allocated-storage 100 ^
    --manage-master-user-password ^
    --master-username admin ^
    --storage-type gp2 ^
    --license-model li ^
    --domain-iam-role-name my-directory-iam-role ^
    --domain my-domain-id ^
    --option-group-name msdtc-se-2016 ^
    --db-parameter-group-name msdtc-sqlserver-se-13
```

Como modificar uma instância de banco de dados e associar o grupo de opções e o grupo de parâmetros do MSDTC

- Use um dos seguintes comandos.

Example

Para Linux, macOS ou Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --option-group-name msdtc-se-2016 \
    --db-parameter-group-name msdtc-sqlserver-se-13 \
    --apply-immediately
```

Para Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --option-group-name msdtc-se-2016 ^
    --db-parameter-group-name msdtc-sqlserver-se-13 ^
    --apply-immediately
```

Modificar a opção do MSDTC

Depois de habilitar a opção `MSDTC`, é possível modificar as configurações. Para obter informações sobre como modificar as configurações da opção, consulte Modificar uma configuração de opção.

> **Note**
>
> Algumas alterações às configurações de opção do MSDTC exigem que o serviço MSDTC seja reiniciado. Esse requisito pode afetar a execução de transações distribuídas.

Usar transações

Usar transações distribuídas

No Amazon RDS for SQL Server, você executa transações distribuídas da mesma maneira que transações distribuídas em execução no local:

- Usando transações promovíveis `System.Transactions` do .NET framework, que otimiza transações distribuídas ao adiar as criações até que sejam necessárias.

Neste caso, a promoção é automática e não requer que você faça qualquer intervenção. Se houver apenas um gerenciador de recursos dentro da transação, nenhuma promoção será executada. Para obter mais informações sobre escopos de transação implícita, consulte Implementing an implicit transaction using transaction scope na documentação da Microsoft.

As transações promovíveis são compatíveis com estas implementações .NET:

- Começando com o ADO.NET 2.0, `System.Data.SqlClient` oferece suporte a transações promovíveis com o SQL Server. Para obter mais informações, consulte System.Transactions integration with SQL Server na documentação da Microsoft.
- O ODP.NET oferece suporte a `System.Transactions`. Uma transação local é criada para a primeira conexão aberta no escopo `TransactionsScope` para o Oracle Database 11g release 1 (versão 11.1) e posterior. Quando uma segunda conexão é aberta, essa transação é automaticamente promovida para uma transação distribuída. Para obter mais informações sobre suporte de transações distribuídas no ODP.NET, consulte Microsoft Distributed Transaction Coordinator integration na documentação da Microsoft.

- Usando a instrução `BEGIN DISTRIBUTED TRANSACTION`. Para obter mais informações, consulte BEGIN DISTRIBUTED TRANSACTION (Transact-SQL) na documentação da Microsoft.

Como usar transações XA

A partir da versão 14.00.3223.3 do RDS para SQL Server 2017, você pode controlar transações distribuídas usando o JDBC. Quando você define a opção `Enable XA` para `true` na opção `MSDTC`, o RDS habilita automaticamente as transações do JDBC e concede a função `SqlJDBCXAUser` ao usuário `guest`. Isso permite a execução de transações distribuídas por meio do JDBC. Para obter mais informações, incluindo um exemplo de código, consulte Noções básicas sobre transações XA na documentação da Microsoft.

Usar rastreamento de transações

O RDS oferece suporte ao controle de rastreamentos de transação do MSDTC e ao download deles na instância de banco de dados do RDS para solução de problemas. É possível controlar sessões de rastreamento de transação executando o procedimento armazenado do RDS a seguir.

```
exec msdb.dbo.rds_msdtc_transaction_tracing 'trace_action',
[@traceall='0|1'],
[@traceaborted='0|1'],
[@tracelong='0|1'];
```

O parâmetro a seguir é obrigatório.

- `trace_action` – A ação de rastreamento. Ela pode ser: `START`, `STOP` ou `STATUS`.

Os seguintes parâmetros são opcionais:

- `@traceall` – Defina como 1 para rastrear todas as transações distribuídas. O padrão é 0.
- `@traceaborted` – Defina como 1 para rastrear transações distribuídas canceladas. O padrão é 0.
- `@tracelong` – Defina como 1 para rastrear transações distribuídas de execução longa. O padrão é 0.

Example de ação de rastreamento START

Para iniciar uma nova sessão de rastreamento de transação, execute o exemplo de instrução a seguir.

```
exec msdb.dbo.rds_msdtc_transaction_tracing 'START',
@traceall='0',
@traceaborted='1',
@tracelong='1';
```

> **Note**
>
> Apenas uma sessão de rastreamento de transação pode estar ativa ao mesmo tempo. Se um novo comando de sessão de rastreamento `START` for emitido enquanto uma sessão de rastreamento estiver ativa, um erro será retornado e a sessão de rastreamento ativa permanecerá inalterada.

Example de ação de rastreamento STOP

Para interromper uma sessão de rastreamento de transação, execute a instrução a seguir.

```
exec msdb.dbo.rds_msdtc_transaction_tracing 'STOP'
```

Esta instrução interrompe a sessão de rastreamento de transação ativa e salva os dados de rastreamento de transação no diretório de log na instância de banco de dados do RDS. A primeira linha da saída contém o resultado geral e as linhas a seguir indicam os detalhes da operação.

Veja a seguir um exemplo de uma interrupção de sessão de rastreamento bem-sucedida.

```
OK: Trace session has been successfully stopped.
Setting log file to: D:\rdsdbdata\MSDTC\Trace\dtctrace.log
Examining D:\rdsdbdata\MSDTC\Trace\msdtctr.mof for message formats,  8 found.
Searching for TMF files on path: (null)
Logfile D:\rdsdbdata\MSDTC\Trace\dtctrace.log:
 OS version    10.0.14393  (Currently running on 6.2.9200)
 Start Time    <timestamp>
 End Time      <timestamp>
 Timezone is   @tzres.dll,-932 (Bias is 0mins)
 BufferSize            16384 B
 Maximum File Size     10 MB
 Buffers  Written      Not set (Logger may not have been stopped).
 Logger Mode Settings (11000002) ( circular paged
 ProcessorCount         1
Processing completed   Buffers: 1, Events: 3, EventsLost: 0 :: Format Errors: 0,
 Unknowns: 3
Event traces dumped to d:\rdsdbdata\Log\msdtc_<timestamp>.log
```

É possível usar as informações detalhadas para consultar o nome do arquivo de log gerado. Para obter mais informações sobre como baixar arquivos de log na instância de banco de dados do RDS, consulte Monitorar arquivos de log do Amazon RDS.

Os logs de sessão de rastreamento permanecem na instância por 35 dias. Todos os logs de sessão de rastreamento mais antigos são excluídos automaticamente.

Example de ação de rastreamento STATUS

Para rastrear o status de uma sessão de rastreamento de transação, execute a instrução a seguir.

```
exec msdb.dbo.rds_msdtc_transaction_tracing 'STATUS'
```

Esta instrução gera o seguinte como linhas separadas do conjunto de resultados.

```
OK
SessionStatus: <Started|Stopped>
TraceAll: <True|False>
TraceAborted: <True|False>
TraceLongLived: <True|False>
```

A primeira linha indica o resultado geral da operação: `OK` ou `ERROR` com detalhes, se aplicável. As linhas subsequentes indicam detalhes sobre o status da sessão de rastreamento:

- `SessionStatus` pode ser um dos seguintes:
  - `Started` se uma sessão de rastreamento estiver sendo executada.
  - `Stopped` se nenhuma sessão de rastreamento estiver sendo executada.
- Os sinalizadores de sessão de rastreamento podem ser `True` ou `False` dependendo de como foram definidos no comando `START`.

Desabilitar o MSDTC

Para desabilitar o MSDTC, remova a opção `MSDTC` do grupo de opções.

Console

Como remover a opção do MSDTC do grupo de opções

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Option groups (Grupos de opções).
3. Selecione o grupo de opções com a opção `MSDTC` (`msdtc-se-2016` nos exemplos anteriores).
4. Selecione Delete option (Excluir opção).
5. Em Opções de exclusão, selecione MSDTC em Opções a serem excluídas.
6. Em Apply immediately (Aplicar imediatamente), selecione Yes (Sim) para excluir a opção imediatamente ou No (Não) para excluí-la na próxima janela de manutenção.
7. Escolha Excluir.

CLI

Como remover a opção do MSDTC do grupo de opções

- Use um dos seguintes comandos.

  Example

  Para Linux, macOS ou Unix:

```
aws rds remove-option-from-option-group \
    --option-group-name msdtc-se-2016 \
    --options MSDTC \
    --apply-immediately
```

Para Windows:

```
aws rds remove-option-from-option-group ^
    --option-group-name msdtc-se-2016 ^
    --options MSDTC ^
    --apply-immediately
```

Solução de problemas do MSDTC para o RDS for SQL Server

Em alguns casos, talvez você tenha problemas para estabelecer uma conexão entre o MSDTC em execução em um computador cliente e o serviço do MSDTC em execução em uma instância de banco de dados do RDS for SQL Server. Em caso afirmativo, verifique o seguinte:

- As regras de entrada do grupo de segurança associado à instância de banco de dados estão configuradas corretamente. Para obter mais informações, consulte Não é possível conectar-se à instância de banco de dados do Amazon RDS.
- O computador cliente está configurado corretamente.
- As regras de firewall do MSDTC no computador cliente estão habilitadas.

Como configurar o computador cliente

1. Abra os Serviços de componentes.

   Ou, no Gerenciador de servidores, escolha Ferramentas e Serviços de componentes.
2. Expanda Serviços de componentes, Computadores, Meu computador e Distributed Transaction Coordinator.
3. Abra o menu de contexto (clique com o botão direito do mouse) do DTC local e escolha Propriedades.
4. Escolha a guia Segurança.
5. Escolha todas as opções a seguir:

- Acesso ao DTC de rede
- Permitir entrada
- Permitir Saída

6. Certifique-se de que o modo de autenticação correto seja escolhido:

   - Autenticação mútua necessária – a máquina cliente é associada ao mesmo domínio que outros nós que participam da transação distribuída ou existe uma relação de confiança configurada entre domínios.
   - Nenhuma autenticação necessária – todos os outros casos.

7. Escolha OK para salvar as alterações.
8. Se solicitado a reiniciar o serviço, escolha Sim.

Como habilitar regras de firewall do MSDTC

1. Abra o Firewall do Windows e escolha Configurações avançadas.

   Ou, no Gerenciador de servidores, escolha Ferramentas e selecione Firewall do Windows com segurança avançada.

   > **Note**
   >
   > Dependendo do sistema operacional, o Firewall do Windows pode ser chamado de Firewall do Windows Defender.

2. Escolha Regras de entrada no painel esquerdo.
3. Habilite as seguintes regras de firewall, se elas ainda não estiverem habilitadas:

   - Distributed Transaction Coordinator (RPC)
   - Distributed Transaction Coordinator (RPC)-EPMAP
   - Distributed Transaction Coordinator (TCP-In)

4. Feche o Firewall do Windows.

Tarefas comuns de DBA no Amazon RDS para Microsoft SQL Server

Esta seção descreve implantações específicas ao Amazon RDS de algumas tarefas comuns de DBA para instâncias de bancos de dados que executam o mecanismo de banco de dados do Microsoft SQL Server. A fim de oferecer uma experiência de serviço gerenciado, o Amazon RDS não fornece acesso ao shell para instâncias de bancos de dados e restringe o acesso a determinados procedimentos e tabelas do sistema que exigem privilégios avançados.

> **Note**
>
> Ao trabalhar com uma instância de banco de dados SQL Server, é possível executar scripts para modificar um banco de dados recém-criado. No entanto, não é possível modificar o banco de dados [model], pois ele é usado como o modelo para novos bancos de dados.

Tópicos

- Acessar o banco de dados tempdb em instâncias de banco de dados Microsoft SQL Server no Amazon RDS
- Análise da workload do banco de dados em uma instância de banco de dados do Amazon RDS para SQL Server com o Orientador de Otimização do Mecanismo de Banco de Dados
- Alterar db_owner para a conta rdsa do banco de dados do Amazon RDS para SQL Server
- Gerenciar agrupamentos e conjuntos de caracteres para o Amazon RDS para Microsoft SQL Server
- Criar um usuário de banco de dados para o Amazon RDS para SQL Server
- Determinar um modelo de recuperação para o banco de dados do Amazon RDS para SQL Server
- Determinar o horário do último failover do Amazon RDS para SQL Server
- Solução de problemas de falhas de recuperação para um ponto no tempo devido a uma descontinuidade no número de sequência de log
- Negue ou permita a visualização de nomes de banco de dados no Amazon RDS para SQL Server
- Desabilitar inserções rápidas durante o carregamento em massa para o Amazon RDS para SQL Server
- Eliminar um banco de dados em uma instância de banco de dados do Amazon RDS para Microsoft SQL Server

- Renomear um banco de dados do Amazon RDS para Microsoft SQL Server em uma implantação multi-AZ
- Redefinir a associação ao perfil db_owner para o usuário principal no Amazon RDS para SQL Server
- Restaurar instâncias de banco de dados com licenças encerradas no Amazon RDS para SQL Server
- Transição de um banco de dados do Amazon RDS para SQL Server de OFFLINE para ONLINE
- Usar a captura de dados de alteração para o Amazon RDS para SQL Server
- Usar o agente do SQL Server para o Amazon RDS
- Trabalhar com logs do Amazon RDS para Microsoft SQL Server
- Trabalhar com arquivos de rastreamento e despejo do Amazon RDS para SQL Server

Acessar o banco de dados tempdb em instâncias de banco de dados Microsoft SQL Server no Amazon RDS

Você pode acessar o banco de dados `tempdb` nas instâncias de bancos de dados Microsoft SQL Server no Amazon RDS. É possível executar o código em `tempdb` usando Transact-SQL pelo Microsoft SQL Server Management Studio (SSMS) ou de qualquer outro aplicativo cliente SQL padrão. Para obter mais informações sobre como se conectar à sua instância de banco de dados, consulte Conectar-se à instância de banco de dados do Microsoft SQL Server.

O usuário mestre para a instância de banco de dados tem acesso `CONTROL` a `tempdb` para que esse usuário possa modificar as opções do banco de dados `tempdb`. O usuário mestre não é o proprietário do banco de dados `tempdb`. Se necessário, o usuário mestre poderá conceder acesso a `CONTROL` a outros usuários, para que eles também possam modificar as opções do banco de dados `tempdb`.

> **Note**
>
> Não é possível executar Database Console Commands (DBCC) no banco de dados `tempdb`.

Modificar opções do banco de dados tempdb

Você pode modificar as opções no banco de dados `tempdb` nas instâncias de bancos de dados do Amazon RDS. Para obter mais informações sobre quais opções podem ser modificadas, consulte Banco de dados tempdb na documentação da Microsoft.

As opções do banco de dados, como as opções de tamanho máximo do arquivo, são persistentes depois que você reinicia sua instância de banco de dados. Você pode modificar as opções do banco de dados para otimizar a performance ao importar dados e para não ficar sem espaço de armazenamento.

Otimizar a performance ao importar dados

Para otimizar a performance ao importar grandes quantidades de dados na sua instância de banco de dados, defina as propriedades `SIZE` e `FILEGROWTH` do banco de dados tempdb para grandes quantidades. Para obter mais informações sobre como otimizar `tempdb`, consulte Otimizando a performance do tempdb na documentação da Microsoft.

O exemplo a seguir demonstra a configuração do tamanho para 100 GB e o crescimento de arquivos para 10%.

```
alter database[tempdb] modify file (NAME = N'templog', SIZE=100GB, FILEGROWTH = 10%)
```

Prevenir problemas de armazenamento

Para evitar que o banco de dados `tempdb` use todo o espaço em disco disponível, defina a propriedade `MAXSIZE`. O exemplo a seguir demonstra a configuração da propriedade para 2048 MB.

```
alter database [tempdb] modify file (NAME = N'templog', MAXSIZE = 2048MB)
```

Recolher o banco de dados tempdb

Existem duas maneiras de reduzir o banco de dados `tempdb` na instância de banco de dados do Amazon RDS. Você pode usar o procedimento `rds_shrink_tempdbfile` ou pode definir a propriedade `SIZE`,

Usar o procedimento rds_shrink_tempdbfile

Você pode usar o procedimento `msdb.dbo.rds_shrink_tempdbfile` do Amazon RDS para reduzir o banco de dados `tempdb`. Só será possível chamar `rds_shrink_tempdbfile` se você tiver acesso `CONTROL` a `tempdb`. Quando você chama `rds_shrink_tempdbfile`, não há tempo de inatividade para sua instância de banco de dados.

O procedimento `rds_shrink_tempdbfile` tem os seguintes parâmetros.

Nome do parâmetro	Tipo de dados	Padrão	Obrigatório	Descrição
`@temp_filename`	SYSNAME	—	obrigatório	O nome lógico do arquivo para encolher.
`@target_size`	int	nulo	opcional	O novo tamanho para o arquivo, em megabytes.

O exemplo a seguir obtém os nomes dos arquivos para o banco de dados `tempdb`.

```
use tempdb;
GO

select name, * from sys.sysfiles;
GO
```

O exemplo a seguir reduz um arquivo do banco de dados `tempdb` chamado `test_file` e solicita um novo tamanho de `10` megabytes:

```
exec msdb.dbo.rds_shrink_tempdbfile @temp_filename = N'test_file', @target_size = 10;
```

Definir a propriedade SIZE

Você também pode reduzir o banco de dados `tempdb` definindo a propriedade `SIZE` e reiniciando a instância de banco de dados. Para obter mais informações sobre como reiniciar sua instância de banco de dados, consulte Reinicializar uma instância de banco de dados .

O exemplo a seguir demonstra a configuração da propriedade `SIZE` para 1024 MB.

```
alter database [tempdb] modify file (NAME = N'templog', SIZE = 1024MB)
```

Configuração TempDB para implantações multi-AZ

Se a instância de banco de dados do RDS para o SQL Server estiver em uma implantação multi-AZ usando o Database Mirroring (DBM) ou grupos de disponibilidade (AGs) Always On, lembre-se das considerações a seguir para o uso do banco de dados `tempdb`.

Não é possível replicar dados do `tempdb` da instância de banco de dados primária para a instância de banco de dados secundária. Ao fazer failover para uma instância de banco de dados secundária, o `tempdb` nessa instância de banco de dados secundária estará vazia.

É possível sincronizar a configuração das opções do banco de dados `tempdb`, incluindo o tamanho do arquivo e as configurações de crescimento automático, da instância de banco de dados primária para a instância de banco de dados secundária. A sincronização da configuração `tempDB` é aceita em todas as versões do RDS para SQL Server. É possível ativar a sincronização automática da configuração `tempdb` usando o seguinte procedimento armazenado:

```
EXECUTE msdb.dbo.rds_set_system_database_sync_objects @object_types = 'TempDbFile';
```

> **Important**
>
> Antes de usar o procedimento armazenado `rds_set_system_database_sync_objects`, defina a configuração `tempdb` preferencial na instância de banco de dados primária, e não na instância de banco de dados secundária. Se você tiver feito a alteração na instância de banco de dados secundária, a configuração `tempdb` preferencial poderá ser excluída ao ativar a sincronização automática.

É possível usar a seguinte função para confirmar se a sincronização automática da configuração `tempdb` está ativada:

```
SELECT * from msdb.dbo.rds_fn_get_system_database_sync_objects();
```

Quando a sincronização automática da configuração `tempdb` estiver ativada, haverá um valor exibido para o campo `object_class`. Quando está desativada, nenhum valor é exibido.

É possível usar a seguinte função para descobrir a última vez que os objetos foram sincronizados no horário UTC:

```
SELECT * from msdb.dbo.rds_fn_server_object_last_sync_time();
```

Por exemplo, se você modificou a configuração `tempdb` às 01:00 e depois executou a função `rds_fn_server_object_last_sync_time`, o valor exibido para `last_sync_time` deve ser posterior à 01:00, indicando que ocorreu uma sincronização automática.

Se você também estiver usando a replicação do SQL Server Agent, poderá habilitar a replicação tanto para os trabalhos do SQL Agent quanto para a configuração `tempdb` fornecendo-os no parâmetro `@object_type`:

```
EXECUTE msdb.dbo.rds_set_system_database_sync_objects @object_types =
 'SQLAgentJob,TempDbFile';
```

Para ter mais informações sobre a replicação de trabalhos do SQL Server Agent, consulte Ativar a replicação de trabalhos do SQL Server Agent.

Como alternativa ao uso do procedimento armazenado `rds_set_system_database_sync_objects` para garantir que as alterações da configuração `tempdb` sejam sincronizadas automaticamente, é possível usar um dos seguintes métodos manuais:

> **Note**
>
> Recomendamos ativar a sincronização automática da configuração `tempdb` usando o procedimento armazenado `rds_set_system_database_sync_objects`: O uso da sincronização automática evita a necessidade de realizar essas tarefas manuais sempre que você altera a configuração `tempdb`.

- Primeiro modifique sua instância de banco de dados e desative o Multi-AZ e depois modifique tempdb e finalmente volte a ativar o Multi-AZ. Esse método não envolve qualquer tempo de inatividade.

  Para obter mais informações, consulte Modificar uma instância de banco de dados do Amazon RDS.
- Modifique `tempdb` na instância primária original, faça failover manualmente e, por fim, modifique `tempdb` na nova instância primária. Esse método envolve tempo de inatividade.

  Para ter mais informações, consulte Reinicializar uma instância de banco de dados .

Análise da workload do banco de dados em uma instância de banco de dados do Amazon RDS para SQL Server com o Orientador de Otimização do Mecanismo de Banco de Dados

O Orientador de Otimização do Mecanismo de Banco de Dados é uma aplicação de cliente fornecida pela Microsoft que analisa a workload do banco de dados e recomenda um conjunto ideal de índices aos bancos de dados do Microsoft SQL Server, com base nos tipos de consultas que você executa. Assim como no SQL Server Management Studio, você executa o Orientador de Otimização em um computador cliente conectado à instância de banco de dados do Amazon RDS que está executando o SQL Server. O computador cliente pode ser um computador local que você executa nas instalações dentro de sua própria rede, ou pode ser uma instância do Amazon EC2 para Windows que está sendo executada na mesma região que a sua instância de banco de dados do Amazon RDS.

Esta seção mostra como capturar uma workload para análise do Orientador de Otimização. Este é o processo ideal para capturar uma workload, pois o Amazon RDS restringe o acesso do host à instância do SQL Server. Para obter mais informações, consulte Database Engine Tuning Advisor (Orientador de Otimização do Mecanismo de Banco de dados) na documentação da Microsoft.

Para usar Orientador de Otimização, é necessário fornecer uma workload ao aplicativo. Uma workload é um conjunto de instruções Transact-SQL executadas em um banco de dados ou nos bancos de dados que você deseja ajustar. Orientador de Otimização do Mecanismo de Banco de Dados usa arquivos de rastreamento, tabelas de rastreamento, scripts Transact-SQL ou arquivos XML como entrada de workload ao ajustar bancos de dados. Ao trabalhar com o Amazon RDS, uma workload pode ser um arquivo em um computador cliente ou uma tabela de banco de dados em um banco de dados do Amazon RDS para SQL Server que pode ser acessada pelo computador cliente. O arquivo ou a tabela deve conter consultas feitas nos bancos de dados que você deseja ajustar e em um formato adequado para repetição.

Para que o Orientador de Otimização seja mais eficiente, uma workload deve ser o mais realista possível. É possível gerar um arquivo ou uma tabela de workload executando rastreamento na instância de banco de dados. Durante a execução do rastreamento, é possível simular uma carga na instância de banco de dados ou executar os aplicativos com uma carga normal.

Há dois tipos de rastreamento: no lado do cliente e no lado do servidor. A configuração de um rastreamento no lado do cliente é mais fácil e permite acompanhar a captura dos eventos de rastreamento em tempo real no SQL Server Profiler. A configuração de um rastreamento no lado do servidor é mais complexa e requer o uso de scripts Transact-SQL. Além disso, como o rastreamento é gravado em um arquivo na instância de banco de dados do Amazon RDS, o espaço de armazenamento é consumido pelo rastreamento. É importante controlar o espaço de armazenamento usado pelo rastreamento em execução no lado do servidor, pois a instância de banco de dados não ficará mais disponível se não houver espaço de armazenamento.

No caso de um rastreamento no lado do cliente, quando uma quantidade suficiente de dados de rastreamento é capturada no SQL Server Profiler, é possível gerar o arquivo de workload. Para isso, basta salvar o rastreamento em um arquivo no computador local ou em uma tabela de banco de dados em uma instância de banco de dados disponível para o seu computador cliente. A principal desvantagem de usar um rastreamento no lado do cliente é que talvez nem todas as consultas sejam capturadas quando sob cargas pesadas. Isso pode diminuir a eficácia da análise realizada pelo Orientador de Otimização do Mecanismo de Banco de Dados. Se você precisar executar um rastreamento sob cargas pesadas e desejar garantir que sejam capturadas todas as consultas durante uma sessão de rastreamento, será necessário usar um rastreamento no lado do servidor.

Para realizar um rastreamento do lado do servidor, é necessário que os arquivos de rastreamento na instância de banco de dados estejam em um arquivo de workload adequado. Também é possível salvar o rastreamento em uma tabela na instância de banco de dados após a conclusão do rastreamento. Você pode usar SQL Server Profiler para salvar o rastreamento em um arquivo

no computador local ou usar Orientador de Otimização, de modo que ele faça a leitura na tabela de rastreamento na instância de banco de dados.

Execução de um rastreamento no lado do cliente em uma instância de banco de dados SQL Server

Para executar um rastreamento no lado do cliente em uma instância de banco de dados SQL Server

1. Inicie SQL Server Profiler. Ele é instalado na pasta Performance Tools da instância do SQL Server. É necessário carregar ou especificar um modelo de definição de rastreamento para iniciar um rastreamento no lado do cliente.
2. No menu Arquivo do SQL Server Profiler, escolha New Trace (Novo rastreamento). Na caixa de diálogo Connect to Server (Conectar-se ao servidor), insira o endpoint da instância de banco de dados, a porta, o nome de usuário principal e a senha do banco de dados no qual deseja executar um rastreamento.
3. Na caixa de diálogo Trace Properties (Propriedades de rastreamento), insira um nome de rastreamento e escolha um modelo de definição de rastreamento. O aplicativo vem com o modelo padrão TSQL_Replay. É possível editar este modelo para definir o rastreamento. Edite eventos e as informações sobre eles na guia Events Selection (Seleção de eventos) da caixa de diálogo Trace Properties (Propriedades de rastreamento).

   Para obter mais informações sobre os modelos de definição de rastreamento e sobre o uso do SQL Server Profiler para especificar um rastreamento no lado do cliente, consulte Database Engine Tuning Advisor (Orientador de Otimização do Mecanismo de Banco de dados) na documentação da Microsoft.
4. Inicie o rastreamento no lado do cliente e acompanhe as consultas do SQL em tempo real enquanto elas são executadas na instância de banco de dados.
5. Selecione Stop Trace (Parar rastreamento) no menu File (Arquivo) quando você tiver concluído o rastreamento. Salve os resultados como um arquivo ou como uma tabela de rastreamento na instância de banco de dados.

Execução de rastreamentos no lado do servidor em uma instância de banco de dados SQL Server

A criação de scripts para fazer um rastreamento no lado do servidor pode ser uma tarefa complexa e está além do escopo deste documento. Esta seção contém amostras de scripts que é possível usar como exemplos. Assim como em um rastreamento no lado do cliente, o objetivo é criar um arquivo

de workload ou uma tabela de rastreamento que você pode abrir usando o Orientador de Otimização do Mecanismo de Banco de Dados.

Veja a seguir um script de exemplo resumido iniciado no lado do servidor e que captura detalhes de rastreamento para um arquivo de workload. Inicialmente, o rastreamento é salvo no arquivo, que fica no diretório D:\RDSDBDATA\Log RDSTrace.trc, e é renovado a cada 100 MB. Assim, os arquivos de rastreamento subsequentes são chamados RDSTrace_1.trc, RDSTrace_2.trc etc.

```
DECLARE @file_name NVARCHAR(245) = 'D:\RDSDBDATA\Log\RDSTrace';
DECLARE @max_file_size BIGINT = 100;
DECLARE @on BIT = 1
DECLARE @rc INT
DECLARE @traceid INT

EXEC @rc = sp_trace_create @traceid OUTPUT, 2, @file_name, @max_file_size
IF (@rc = 0) BEGIN
   EXEC sp_trace_setevent @traceid, 10, 1, @on
   EXEC sp_trace_setevent @traceid, 10, 2, @on
   EXEC sp_trace_setevent @traceid, 10, 3, @on
 . . .
   EXEC sp_trace_setfilter @traceid, 10, 0, 7, N'SQL Profiler'
   EXEC sp_trace_setstatus @traceid, 1
   END
```

O exemplo a seguir é um script que interrompe um rastreamento. Observe que um rastreamento criado pelo script anterior continuará a ser executado até que seja explicitamente interrompido ou até que o processo fique sem espaço em disco.

```
DECLARE @traceid INT
SELECT @traceid = traceid FROM ::fn_trace_getinfo(default)
WHERE property = 5 AND value = 1 AND traceid <> 1

IF @traceid IS NOT NULL BEGIN
   EXEC sp_trace_setstatus @traceid, 0
   EXEC sp_trace_setstatus @traceid, 2
END
```

É possível salvar os resultados do rastreamento no lado do servidor em uma tabela de banco de dados e usar essa como a workload do Orientador de Otimização com a função fn_trace_gettable. Os comandos a seguir carregam os resultados de todos os arquivos chamados RDSTrace.trc

no diretório D:\rdsdbdata\Log RDSTrace.trc, inclusive todos os arquivos subsequentes como RDSTrace_1.trc, em uma tabela chamada RDSTrace no banco de dados atual.

```
SELECT * INTO RDSTrace
FROM fn_trace_gettable('D:\rdsdbdata\Log\RDSTrace.trc', default);
```

Para salvar um arquivo subsequente específico em uma tabela (por exemplo o RDSTrace_1.trc), especifique o nome do arquivo e substitua default pelo número 1 como o último parâmetro para fn_trace_gettable.

```
SELECT * INTO RDSTrace_1
FROM fn_trace_gettable('D:\rdsdbdata\Log\RDSTrace_1.trc', 1);
```

Execução do Orientador de Otimização com um rastreamento

Após a criação de um rastreamento, seja como um arquivo local ou uma tabela de banco de dados, é possível executar Orientador de Otimização na instância de banco de dados. Usar o Orientador de Otimização com o Amazon RDS é como trabalhar com uma instância autônoma e remota do SQL Server. É possível usar a interface de usuário do Orientador de Otimização na máquina cliente ou o utilitário dta.exe na linha de comando. Em ambos os casos, ao usar o Orientador de Otimização, você deve se conectar à instância de banco de dados do Amazon RDS usando o endpoint para a instância de banco de dados e fornecer seu nome de usuário mestre e senha de usuário mestre.

O exemplo de código a seguir demonstra como usar o utilitário de linha de comando dta.exe em uma instância de banco de dados do Amazon RDS com um endpoint de **`dta.cnazcmklsdei.us-east-1.rds.amazonaws.com`**. O exemplo inclui o nome de usuário mestre **`admin`** e a senha do usuário mestre **`test`**. O banco de dados de exemplo a ser ajustado é chamado de máquina **`C:\RDSTrace.trc`**. O código de linha de comando do exemplo também especifica uma sessão de rastreamento chamada **`RDSTrace1`** e especifica os arquivos de saída para a máquina local chamados **`RDSTrace.sql`** para o script de saída do SQL, **`RDSTrace.txt`** para um arquivo de resultado e **`RDSTrace.xml`** para um arquivo XML da análise. Há também uma tabela de erros especificada no banco de dados do RDSDTA chamada **`RDSTraceErrors`**.

```
dta -S dta.cnazcmklsdei.us-east-1.rds.amazonaws.com -U admin -P test -D RDSDTA -
if C:\RDSTrace.trc -s RDSTrace1 -of C:\ RDSTrace.sql -or C:\ RDSTrace.txt -ox C:\
 RDSTrace.xml -e RDSDTA.dbo.RDSTraceErrors
```

Veja o mesmo exemplo de código de linha de comando, exceto que a workload de entrada é uma tabela na instância remota do Amazon RDS denominada **`RDSTrace`** que está no banco de dados **`RDSDTA`**.

```
dta -S dta.cnazcmklsdei.us-east-1.rds.amazonaws.com -U admin -P test -D RDSDTA -it
 RDSDTA.dbo.RDSTrace -s RDSTrace1 -of C:\ RDSTrace.sql -or C:\ RDSTrace.txt -ox C:\
 RDSTrace.xml -e RDSDTA.dbo.RDSTraceErrors
```

Para obter uma lista completa dos parâmetros da linha de comando do utilitário dta, consulte dta Utility (Utilitário dta) na documentação da Microsoft.

Alterar **`db_owner`** para a conta **`rdsa`** do banco de dados do Amazon RDS para SQL Server

Quando você cria ou restaura um banco de dados em uma instância de banco de dados do RDS para SQL Server, o Amazon RDS define o proprietário do banco de dados como. `rdsa`. Se você tiver uma implantação multi-AZ usando o espelhamento de banco de dados (DBM) ou grupos de disponibilidade AlwaysOn (AGs) do SQL Server, o Amazon RDS definirá o proprietário do banco de dados na instância de banco de dados secundária como `NT AUTHORITY\SYSTEM`. O proprietário do banco de dados secundário não poderá ser alterado enquanto a instância de banco de dados secundária não for promovida à função principal. Na maioria dos casos, definir o proprietário do banco de dados como `NT AUTHORITY\SYSTEM` não é um problema ao executar consultas, mas pode gerar erros ao executar procedimentos armazenados no sistema, como `sys.sp_updatestats` que exigem permissões elevadas para execução.

Você pode usar a seguinte consulta para identificar o proprietário dos bancos de dados de propriedade de `NT AUTHORITY\SYSTEM`:

```
SELECT name FROM sys.databases WHERE SUSER_SNAME(owner_sid) = 'NT AUTHORITY\SYSTEM';
```

É possível usar o procedimento armazenado `rds_changedbowner_to_rdsa` do Amazon RDS para alterar o proprietário do banco de dados para `rdsa`. Os seguintes bancos de dados não podem ser usados com `rds_changedbowner_to_rdsa`: `master, model, msdb, rdsadmin, rdsadmin_ReportServer, rdsadmin_ReportServerTempDB, SSISDB`.

Para alterar o proprietário do banco de dados para `rdsa`, chame o procedimento armazenado `rds_changedbowner_to_rdsa` e forneça o nome do banco de dados.

Example uso:

```
exec msdb.dbo.rds_changedbowner_to_rdsa 'TestDB1';
```

O seguinte parâmetro é obrigatório:

- `@db_name`: o nome do banco de dados para o qual alterar o proprietário do banco de dados `rdsa`.

> **Important**
>
> Não é possível usar `rds_changedbowner_to_rdsa` para alterar a propriedade de um banco de dados para um login diferente de `rdsa`. Por exemplo, não é possível alterar a propriedade do login com o qual você criou o banco de dados. Para restaurar a associação perdida no perfil `db_owner` de seu usuário mestre quando nenhum outro usuário do banco de dados puder ser usado para conceder a associação, redefina a senha do usuário mestre para ter a associação ao perfil `db_owner`. Para ter mais informações, consulte Redefinir a associação ao perfil db_owner para o usuário principal no Amazon RDS para SQL Server.

Gerenciar agrupamentos e conjuntos de caracteres para o Amazon RDS para Microsoft SQL Server

Este tópico fornece orientações sobre como gerenciar agrupamentos e conjuntos de caracteres do Microsoft SQL Server no Amazon RDS. Ele explica como configurar agrupamentos durante a criação do banco de dados e modificá-los posteriormente, garantindo o tratamento adequado dos dados de texto com base nos requisitos de idioma e localidade. Além disso, ele aborda as práticas recomendadas para manter a compatibilidade e o desempenho em ambientes do SQL Server no Amazon RDS.

O SQL Server é compatível com agrupamentos em vários níveis. Você define o agrupamento de servidores padrão quando cria a instância de banco de dados. É possível substituir o agrupamento no nível de banco de dados, tabela ou coluna.

Tópicos

- Agrupamento no nível do servidor para o Microsoft SQL Server
- Agrupamento no nível do banco de dados para o Microsoft SQL Server

Agrupamento no nível do servidor para o Microsoft SQL Server

Ao criar uma instância de banco de dados do Microsoft SQL Server, você pode definir o agrupamento de servidor que você deseja usar. Se você não escolher um agrupamento diferente, o agrupamento no nível do servidor será padronizado como SQL_Latin1_General_CP1_CI_AS. O agrupamento do servidor é aplicado por padrão a todos os bancos de dados e objetos de banco de dados.

> **Note**
>
> Não é possível alterar o agrupamento ao fazer a restauração a partir de um snapshot de banco de dados.

Atualmente, o Amazon RDS é compatível com os seguintes agrupamentos de servidores:

Collation (Agrupamento)	Descrição
Árabe_CI_AS	Arabic, não diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura
Chinese_PRC_BIN2	Chinese-PRC, ordem de classificação por pontos de código binário
Chinese_PRC_CI_AS	Chinese-PRC, não diferencia letras maiúscula s de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura
Chinese_Taiwan_Stroke_CI_AS	Chinese-Taiwan-Stroke, não diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura
Danish_Norwegian_CI_AS	Danish-Norwegian, não diferencia letras maiúsculas de minúsculas, diferencia acentos,

Collation (Agrupamento)	Descrição
	não diferencia o tipo de kana, não diferencia a largura
Danish_Norwegian_CI_AS_KS	Danish-Norwegian, não diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, não diferencia a largura
Danish_Norwegian_CI_AS_KS_WS	Danish-Norwegian, não diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, não diferencia a largura
Danish_Norwegian_CI_AS_WS	Danish-Norwegian, não diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, diferencia a largura
Danish_Norwegian_CS_AI	Danish-Norwegian, diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, não diferencia a largura
Danish_Norwegian_CS_AI_KS	Danish-Norwegian, diferencia letras maiúsculas de minúsculas, não diferencia acentos, diferencia o tipo de kana, não diferencia a largura
Finnish_Swedish_100_BIN	Finnish-Swedish-100, classificação binária
Finnish_Swedish_100_BIN2	Finnish-Swedish-100, classificação binária de comparação de ponto de código
Finnish_Swedish_100_CI_AI	Finnish-Swedish 100, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, não diferencia a largura

Collation (Agrupamento)	Descrição
Finnish_Swedish_100_CI_AS	Finnish-Swedish 100, não diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura
Finnish_Swedish_CI_AS	Finlandês, sueco e sueco (Finlândia) – não diferenciam maiúsculas de minúsculas, diferenciação de acentos, indiferentes ao tipo de kana, indiferentes à largura
French_CI_AS	French, não diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura
Hebrew_BIN	Hebraico, classificação binária
Hebrew_CI_AS	Hebrew, não diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura
Japanese_BIN	Japonês, tipo binário
Japanese_CI_AS	Japanese, não diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura
Japanese_CS_AS	Japanese, diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura
Japanese_XJIS_140_CI_AS	Japanese, não diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferenci a o tipo de kana, não diferencia a largura, caracteres complementares, não diferencia o seletor de variação

Collation (Agrupamento)	Descrição
Japanese_XJIS_140_CI_AS_KS_VSS	Japanese, não diferencia letras maiúsculas de minúsculas, diferencia acentos, diferenci a o tipo de kana, não diferencia a largura, caracteres complementares, diferencia o seletor de variação
Japanese_XJIS_140_CI_AS_VSS	Japanese, não diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferenci a o tipo de kana, não diferencia a largura, caracteres complementares, diferencia o seletor de variação
japonês_XJIS_140_CS_AS_KS_WS	Japanese, diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, diferencia a largura, caracteres complementares, não diferencia o seletor de variação
Korean_Wansung_CI_AS	Korean-Wansung, não diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura
Latin1_General_100_BIN	Latin1-General-100, tipo binário
Latin1_General_100_BIN2	Latin1-General-100, ordem de classificação por pontos de código binário
Latin1_General_100_BIN2_UTF8	Latin1-General-100, ordem de classificação por pontos de código binário, codificado em UTF-8
Latin1_General_100_CI_AS	Latin1-General-100, não diferencia maiúscula s de minúsculas, diferenciação de acentos, indiferente ao tipo de kana, indiferente à largura

Collation (Agrupamento)	Descrição
Latin1_General_100_CI_AS_SC_UTF8	Latin1-General-100, não diferencia letras maiúsculas de minúsculas, diferencia acentos, caracteres complementares, codificação em UTF-8
Latin1_General_BIN	Latin1-General, tipo binário
Latin1_General_BIN2	Latin1-General, ordem de classificação por pontos de código binário
Latin1_General_CI_AI	Latin1-General, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, não diferencia a largura
Latin1_General_CI_AS	Latin1-General, não diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura
Latin1_General_CI_AS_KS	Latin1-General, não diferencia letras maiúsculas de minúsculas, diferencia acentos, diferencia o tipo de kana, não diferencia a largura
Latin1_General_CS_AS	Latin1-Geral, diferencia maiúsculas de minúsculas, diferenciação de acentos, indiferente ao tipo de kana, indiferente à largura
Modern_Spanish_CI_AS	Modern-Spanish, não diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura
polish_ci_as	Polish, não diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura

Collation (Agrupamento)	Descrição
SQL_1xCompat_CP850_CI_AS	Latin1-Geral, não diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura para dados do Unicode, ordem de classificação do SQL Server 49 na página de código 850 para dados que não são do Unicode
SQL_Latin1_General_CP1_CI_AI	Latin1-Geral, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, não diferencia a largura para dados do Unicode, ordem de classificação do SQL Server 54 na página de código 1252 para dados que não são do Unicode
SQL_Latin1_General_CP1_CI_AS (padrão)	Latin1-Geral, não diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura para dados do Unicode, ordem de classificação do SQL Server 52 na página de código 1252 para dados que não são do Unicode
SQL_Latin1_General_CP1_CS_AS	Latin1-Geral, diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura para dados do Unicode, ordem de classificação do SQL Server 51 na página de código 1252 para dados que não são do Unicode

Collation (Agrupamento)	Descrição
SQL_Latin1_General_CP437_CI_AI	Latin1-Geral, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, não diferencia a largura para dados do Unicode, ordem de classificação do SQL Server 34 na página de código 437 para dados que não são do Unicode
SQL_Latin1_General_CP850_bin	Latin1-General, ordem de classificação por pontos de código binário para dados do Unicode, ordem de classificação do SQL Server 40 na página de código 850 para dados que não são do Unicode
SQL_Latin1_General_CP850_BIN2	Latin1-General, ordem de classificação por pontos de código binário para dados do Unicode, ordem de classificação do SQL Server 40 na página de código 850 para dados que não são do Unicode
SQL_Latin1_General_CP850_CI_AI	Latin1-Geral, não diferencia letras maiúsculas de minúsculas, não diferencia acentos, não diferencia o tipo de kana, não diferencia a largura para dados do Unicode, ordem de classificação do SQL Server 44 na página de código 850 para dados que não são do Unicode
SQL_Latin1_General_CP850_CI_AS	Latin1-Geral, não diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura para dados do Unicode, ordem de classificação do SQL Server 42 na página de código 850 para dados que não são do Unicode

Collation (Agrupamento)	Descrição
SQL_Latin1_General_CP1256_CI_AS	Latin1-Geral, não diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura para dados do Unicode, ordem de classificação do SQL Server 146 na página de código 1256 para dados que não são do Unicode
SQL_Latin1_General_CP1255_CS_AS	Latin1-Geral, diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura para dados do Unicode, ordem de classificação do SQL Server 137 na página de código 1255 para dados que não são do Unicode
Thai_CI_AS	Thai, não diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura
turkish_ci_as	Turkish, não diferencia letras maiúsculas de minúsculas, diferencia acentos, não diferencia o tipo de kana, não diferencia a largura

Para escolher o agrupamento:

- Se você estiver usando o console do Amazon RDS, ao criar uma instância de banco de dados, escolha Additional configuration (Configuração adicional) e insira o agrupamento no campo Collation (Agrupamento). Para obter mais informações, consulte Criar uma instância de banco de dados do Amazon RDS.
- Se você estiver usando a AWS CLI, use a opção `--character-set-name` com o comando `create-db-instance`. Para obter mais informações, consulte create-db-instance.
- Se você estiver usando a API do Amazon RDS, use o parâmetro `CharacterSetName` com a operação `CreateDBInstance`. Para obter mais informações, consulte CreateDBInstance.

Agrupamento no nível do banco de dados para o Microsoft SQL Server

Para alterar o agrupamento padrão no banco de dados, na tabela ou na coluna, anule o nível do agrupamento ao criar um novo banco de dados ou um objeto de banco de dados. Por exemplo, se o agrupamento de servidores padrão for SQL_Latin1_General_CP1_CI_AS, você poderá alterá-lo para Mohawk_100_CI_AS para suporte ao agrupamento Mohawk. Se necessário, é possível até mesmo converter o tipo de argumentos em uma consulta, de modo que eles usem um tipo de agrupamento diferente.

Por exemplo, a consulta abaixo altera o agrupamento padrão da coluna AccountName para Mohawk_100_CI_AS

```
CREATE TABLE [dbo].[Account]
 (
     [AccountID] [nvarchar](10) NOT NULL,
     [AccountName] [nvarchar](100) COLLATE Mohawk_100_CI_AS NOT NULL
 ) ON [PRIMARY];
```

O mecanismo de banco de dados Microsoft SQL Server é compatível com Unicode por meio dos tipos de dados NCHAR, NVARCHAR e NTEXT. Por exemplo, se precisar de suporte para CJK, use esses tipos de dados Unicode para armazenamento de caracteres e substitua o agrupamento padrão de servidor ao criar seus bancos de dados e tabelas. Veja a seguir alguns links da Microsoft sobre agrupamentos e compatibilidade com o Unicode do SQL Server:

- Trabalhando com agrupamentos
- Collation and international terminology
- Using SQL Server collations
- Considerações internacionais para aplicativos de banco de dados e aplicativos do Mecanismo de Banco de Dados

Criar um usuário de banco de dados para o Amazon RDS para SQL Server

Você pode criar um usuário de banco de dados para a instância de banco de dados Amazon RDS para Microsoft SQL Server executando um script T-SQL como o exemplo a seguir. Use uma aplicação como o SQL Server Management Suite (SSMS). Você faz login na instância de banco de dados como o usuário mestre que foi criado ao criar a instância de banco de dados.

```
--Initially set context to master database
```

```
USE [master];
GO
--Create a server-level login named theirname with password theirpassword
CREATE LOGIN [theirname] WITH PASSWORD = 'theirpassword';
GO
--Set context to msdb database
USE [msdb];
GO
--Create a database user named theirname and link it to server-level login theirname
CREATE USER [theirname] FOR LOGIN [theirname];
GO
```

Para obter um exemplo da adição de um usuário de banco de dados a uma função, consulte Adicionar um usuário à função SQLAgentUser.

> **Note**
>
> Se você receber erros de permissão ao adicionar um usuário, poderá restaurar privilégios modificando a senha de usuário primário da instância de banco de dados. Para ter mais informações, consulte Redefinir a associação ao perfil db_owner para o usuário principal no Amazon RDS para SQL Server.
> Não é uma prática recomendada clonar as permissões do usuário principal em suas aplicações. Para ter mais informações, consulte How to clone master user permissions in Amazon RDS for SQL Server.

Determinar um modelo de recuperação para o banco de dados do Amazon RDS para SQL Server

No Amazon RDS, o modelo de recuperação, o período de retenção e o status do banco de dados estão vinculados.

É importante entender as consequências antes de fazer uma alteração em uma dessas configurações. Cada configuração pode afetar as outras. Por exemplo:

- Se você alterar o modelo de recuperação de um banco de dados para SIMPLE ou BULK_LOGGED enquanto a retenção de backup estiver habilitada, o Amazon RDS redefinirá o modelo de recuperação para FULL em cinco minutos. Isso também fará com que o RDS tire um snapshot da instância de banco de dados.

- Se você definir a retenção de backup como `0` dias, o RDS definirá o modo de recuperação como SIMPLE.
- Se você alterar o modelo de recuperação de um banco de dados de SIMPLE para qualquer outra opção enquanto a retenção de backup estiver definida como `0` dias, o RDS redefinirá o modelo de recuperação de volta para SIMPLE.

> **Important**
>
> Nunca altere o modelo de recuperação em instâncias multi-AZ, mesmo que pareça que você pode fazer isso—por exemplo, usando ALTER DATABASE. A retenção de backup e, portanto, o modo de recuperação FULL, é necessária para o multi-AZ. Se você alterar o modelo de recuperação, o RDS o alterará imediatamente para FULL.
> Esta reinicialização automática força o RDS a reconstruir completamente o espelhamento. Durante essa reconstrução, a disponibilidade do banco de dados é degradada por cerca de 30-90 a 90 minutos até que o espelhamento esteja pronto para failover. A instância de banco de dados também enfrenta a degradação da performance da mesma maneira que durante uma conversão de single-AZ para multi-AZ. O tempo durante o qual a performance é degradada depende do tamanho do armazenamento de banco de dados — quanto maior o banco de dados armazenado, mais longa será a degradação.

Para obter mais informações sobre modelos de recuperação do SQL Server, consulte Modelos de recuperação (SQL Server) na documentação da Microsoft.

Determinar o horário do último failover do Amazon RDS para SQL Server

Para determinar o último tempo de failover, use o seguinte procedimento armazenado:

```
execute msdb.dbo.rds_failover_time;
```

Este procedimento retorna as informações a seguir.

Parâmetro de saída	Descrição
errorlog_available_from	Mostra o tempo em que os logs de erro estão disponíveis no diretório de log.

Parâmetro de saída	Descrição
recent_failover_time	Mostra o último tempo de failover se estiver disponível nos logs de erro. Caso contrário, será exibido `null`.

> **Note**
>
> O procedimento armazenado pesquisa todos os logs de erro disponíveis do SQL Server no diretório de log para recuperar o tempo de failover mais recente. Se as mensagens de failover foram substituídas pelo SQL Server, o procedimento não recuperará o tempo de failover.

Example de nenhum failover recente

Este exemplo mostra a saída quando não há nenhum failover recente nos logs de erros. Nenhum failover aconteceu desde 29-04-2020 23:59:00.01.

errorlog_available_from	recent_failover_time
29-04-2020 23:59:00.0100000	nulo

Example de failover recente

Este exemplo mostra a saída quando há um failover nos logs de erros. O failover mais recente foi em 05-05-2020 18:57:51.89.

errorlog_available_from	recent_failover_time
29-04-2020 23:59:00.0100000	05-05-2020 18:57:51.8900000

Solução de problemas de falhas de recuperação para um ponto no tempo devido a uma descontinuidade no número de sequência de log

Ao tentar a recuperação para um ponto no tempo (PITR) no RDS para SQL Server, você pode encontrar falhas devido a descontinuidades nos números de sequência de log (LSNs). Essas descontinuidade impedem que o RDS restaure o banco de dados no horário solicitado, e o RDS coloca a instância de restauração no estado `incompatible-restore`.

Algumas causas comuns desse problema são:

- Alterações manuais no modelo de recuperação do banco de dados.
- Alterações automáticas no modelo de recuperação pelo RDS devido à insuficiência de recursos para concluir o backup dos logs de transações.

Para identificar descontinuidades no LSN em seu banco de dados, execute esta consulta:

```
SELECT * FROM msdb.dbo.rds_fn_list_tlog_backup_metadata(database_name)
ORDER BY backup_file_time_utc desc;
```

Se você descobrir uma descontinuidade no LSN, você pode:

- Escolher um ponto de restauração antes da descontinuidade do LSN.
- Aguardar e restaurar para um ponto após a conclusão do próximo backup da instância.

Para evitar esse problema, recomendamos que você não altere manualmente o modelo de recuperação dos bancos de dados do RDS para SQL Server, pois isso interrompe a durabilidade da instância. Também recomendamos que você escolha um tipo de instância com recursos suficientes para sua workload, a fim de garantir backups regulares do log de transações.

Para ter mais informações sobre o gerenciamento de logs de transações, consulte SQL Server transaction log architecture and management guide na documentação do Microsoft SQL Server.

Negue ou permita a visualização de nomes de banco de dados no Amazon RDS para SQL Server

O usuário principal não pode definir `DENY VIEW ANY DATABASE TO LOGIN` para ocultar bancos de dados de um usuário.

Para alterar essa permissão, use o seguinte procedimento armazenado:

- Negar acesso de visualização do banco de dados ao *LOGIN*:

```
EXEC msdb.dbo.rds_manage_view_db_permission @permission='DENY',
 @server_principal='LOGIN'

go
```

- Permitir acesso de visualização do banco de dados ao *LOGIN*:

```
EXEC msdb.dbo.rds_manage_view_db_permission @permission='GRANT',
 @server_principal='LOGIN'

go
```

Pense no seguinte ao usar esse procedimento armazenado:

- Os nomes dos bancos de dados estão ocultos do SSMS e do DMV interno (visualizações de gerenciamento dinâmico). No entanto, os nomes dos bancos de dados ainda são visíveis nas tabelas de auditoria, de logs e de metadados. Essas são permissões protegidas `VIEW ANY DATABASE` do servidor. Para ter mais informações, consulte Permissões do servidor DENY.
- Depois que a permissão for revertida para `GRANT` (permitido), o *LOGIN* poderá visualizar todos os bancos de dados.
- Se você excluir e recriar *LOGIN*, a permissão de visualização relacionada ao LOGIN será redefinida como `ALLOW`.
- Para instâncias multi-AZ, defina a permissão `DENY` ou `GRANT` somente para o *LOGIN* no host primário. As alterações são propagadas automaticamente para o host secundário.
- Essa permissão só altera se um login pode visualizar os nomes do banco de dados. No entanto, o acesso aos bancos de dados e objetos internos é gerenciado separadamente.

Desabilitar inserções rápidas durante o carregamento em massa para o Amazon RDS para SQL Server

A partir do SQL Server 2016, as inserções rápidas são habilitadas por padrão. As inserções rápidas aproveitam o registro em log mínimo que ocorre enquanto o banco de dados está no modelo de recuperação simples ou em massa registrada em log para otimizar a performance da inserção. Com

as inserções rápidas, cada lote de carregamento em massa adquire novas extensões, ignorando a pesquisa de alocação para extensões existentes com espaço livre disponível para otimizar a performance das inserções.

No entanto, com as inserções rápidas, os carregamentos em massa com tamanhos de lote pequenos podem levar ao aumento do espaço não utilizado consumido pelos objetos. Se aumentar o tamanho do lote não for viável, habilitar o sinalizador de rastreamento 692 poderá ajudar a reduzir o espaço reservado não utilizado, mas isso afetará a performance. Habilitar esse sinalizador de rastreamento desabilitará as inserções rápidas durante o carregamento em massa de dados no heap ou índices em cluster.

O sinalizador de rastreamento 692 é habilitado como um parâmetro de inicialização usando grupos de parâmetros de banco de dados. Para obter mais informações, consulte Grupos de parâmetros para Amazon RDS.

Há suporte ao sinalizador de rastreamento 692 para o Amazon RDS no SQL Server 2016 e posterior. Para obter mais informações sobre sinalizadores de rastreamento, consulte DBCC TRACEON – sinalizadores de rastreamento na documentação da Microsoft.

Eliminar um banco de dados em uma instância de banco de dados do Amazon RDS para Microsoft SQL Server

É possível eliminar um banco de dados em uma instância de banco de dados do Amazon RDS que executa o Microsoft SQL Server em uma implantação single-AZ ou multi-AZ. Para ignorar o banco de dados, use o seguinte comando:

```
--replace your-database-name with the name of the database you want to drop
EXECUTE msdb.dbo.rds_drop_database  N'your-database-name'
```

> **Note**
>
> Use aspas simples retas no comando. Aspas curvas causarão um erro.

Depois que você usa esse procedimento para ignorar o banco de dados, o Amazon RDS ignora todas as conexões existentes com o banco de dados e remove o histórico de backup do banco de dados.

Para conceder permissão de backup e restauração a outros usuários, siga este procedimento:

```
USE master
GO
CREATE LOGIN user1 WITH PASSWORD=N'changeThis', DEFAULT_DATABASE=master,
 CHECK_EXPIRATION=OFF, CHECK_POLICY=OFF
GO
USE msdb
GO
CREATE USER user1 FOR LOGIN user1
GO
use msdb
GO
GRANT EXECUTE ON msdb.dbo.rds_backup_database TO user1
GO
GRANT EXECUTE ON msdb.dbo.rds_restore_database TO user1
GO
```

Renomear um banco de dados do Amazon RDS para Microsoft SQL Server em uma implantação multi-AZ

Para renomear uma instância do banco de dados do Microsoft SQL Server que use o multi-AZ, use o seguinte procedimento:

1. Primeiramente, desative multi-AZ para a instância de banco de dados.
2. Renomeie o banco de dados executando `rdsadmin.dbo.rds_modify_db_name`.
3. Em seguida, ative o multi-AZ com Espelhamento ou os Grupos de disponibilidade AlwaysOn para a instância de banco de dados para retorná-la ao estado original.

Para obter mais informações, consulte Adicionar Multi-AZ a uma instância de banco de dados do Microsoft SQL Server.

> **Note**
>
> Se a instância não usar multi-AZ, não será necessário alterar nenhuma configuração antes ou depois de executar `rdsadmin.dbo.rds_modify_db_name`.
> Não é possível renomear um banco de dados em uma instância de origem de réplica de leitura.

Exemplo: No exemplo a seguir, o procedimento armazenado `rdsadmin.dbo.rds_modify_db_name` renomeia um banco de dados de **`MOO`** para **`ZAR`**. É semelhante a executar a instrução `DDL ALTER DATABASE [MOO] MODIFY NAME = [ZAR]`.

```
EXEC rdsadmin.dbo.rds_modify_db_name N'MOO', N'ZAR'
GO
```

Redefinir a associação ao perfil db_owner para o usuário principal no Amazon RDS para SQL Server

Se você bloquear seu usuário mestre da associação ao perfil `db_owner` em seu banco de dados do RDS para SQL Server e nenhum outro usuário do banco de dados puder conceder a associação, você poderá restaurar a associação perdida modificando a senha do usuário mestre da instância de banco de dados.

Ao alterar a senha do usuário mestre da instância de banco de dados, o RDS concede a associação `db_owner` aos bancos de dados na instância de banco de dados que pode ter sido revogada acidentalmente. É possível alterar a senha da instância de banco de dados usando o console do Amazon RDS, o comando da AWS CLI modify-db-instance ou a operação da API ModifyDBInstance. Para ter mais informações sobre a modificação de uma instância de banco de dados , consulte Modificar uma instância de banco de dados do Amazon RDS.

Restaurar instâncias de banco de dados com licenças encerradas no Amazon RDS para SQL Server

A Microsoft solicitou que alguns clientes do Amazon RDS que não informaram seus dados de Microsoft License Mobility encerrassem sua instância de banco de dados. O Amazon RDS faz snapshots dessas instâncias de banco de dados, e é possível restaurar do snapshot para uma nova instância de banco de dados que tenha o modelo de licença inclusa.

É possível restaurar a partir de um snapshot da Standard Edition para a própria Standard Edition ou para a Enterprise Edition.

É possível restaurar a partir de um snapshot da Enterprise Edition para a Standard Edition ou para a própria Enterprise Edition.

Como restaurar de um snapshot do SQL Server após o Amazon RDS criar um snapshot final da sua instância

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Snapshots.
3. Escolha o snapshot da instância de banco de dados do SQL Server. O Amazon RDS cria um snapshot final da instância de banco de dados. O nome do snapshot da instância encerrada está no formato *instance_name*`-final-snapshot`. Por exemplo, se o nome da instância de banco de dados for **`mytest.cdxgahslksma.us-east-1.rds.com`**, o snapshot final se chamará **`mytest-final-snapshot`** e estará na mesma região da AWS da instância de banco de dados original.
4. Em Actions (Ações), escolha Restore Snapshot (Restaurar snapshot).

   A janela Restore DB Instance (Restaurar instância de banco de dados) é exibida.
5. Em License Model (Modelo de licença), escolha license-included (licença incluída).
6. Escolha o mecanismo de banco de dados SQL Server que você deseja usar.
7. Em DB Instance Identifier (Identificador da instância de banco de dados), digite o nome da instância de banco de dados restaurada.
8. Escolha Restore DB Instance.

Para obter mais informações sobre a restauração a partir de um snapshot, consulte Restaurar uma instância de banco de dados.

Transição de um banco de dados do Amazon RDS para SQL Server de OFFLINE para ONLINE

Você pode mudar o estado de seu banco de dados do Microsoft SQL Server em uma instância de banco de dados do Amazon RDS de `OFFLINE` para `ONLINE`.

Método do SQL Server	Método do Amazon RDS
ALTER DATABASE *db_name* SET ONLINE;	EXEC rdsadmin.dbo.rds_set_database_online *db_name*

Usar a captura de dados de alteração para o Amazon RDS para SQL Server

O Amazon RDS comporta captura de dados de alteração (CDC) em instâncias de banco de dados que estejam executando o Microsoft SQL Server. A CDC captura alterações feitas nos dados de suas tabelas. Ela armazena metadados sobre cada alteração, que podem ser acessados posteriormente. Para obter mais informações sobre como a CDC funciona, consulte Captura de dados de alteração na documentação da Microsoft. Para usar a CDC com suas instâncias de banco de dados do Amazon RDS, habilite-a no banco de dados executando `msdb.dbo.rds_cdc_enable_db`. Após a habilitação da CDC, qualquer usuário que seja um `db_owner` desse banco de dados poderá habilitar ou desabilitar a CDC em tabelas desse banco de dados.

> **Important**
>
> Durante restaurações, a CDC será desabilitada. Todos os metadados relacionados são removidos automaticamente do banco de dados. Isso se aplica a restaurações de snapshot, a restaurações pontuais e a restaurações nativas do SQL Server no S3. Após a execução desses tipos de restauração, você pode habilitar novamente a CDC e especificar tabelas novamente para acompanhamento.

Para habilitar o CDC para uma instância de banco de dados, execute o procedimento `msdb.dbo.rds_cdc_enable_db` armazenado.

```
exec msdb.dbo.rds_cdc_enable_db 'database_name'
```

Para desabilitar o CDC para uma instância de banco de dados, execute o procedimento `msdb.dbo.rds_cdc_disable_db` armazenado.

```
exec msdb.dbo.rds_cdc_disable_db 'database_name'
```

Para conceder permissões de CDC a um usuário, use o seguinte procedimento:

```
go
  GRANT EXECUTE ON msdb.dbo.rds_cdc_enable_db TO User1
  GRANT EXECUTE ON msdb.dbo.rds_cdc_disable_db TO User1
```

Tópicos

- Acompanhamento de tabelas com a Captura de dados de alteração
- Trabalhos de captura de dados de alteração
- Change Data Capture para instâncias multi-AZ

Acompanhamento de tabelas com a Captura de dados de alteração

Após a CDC ser habilitada no banco de dados, você pode começar a acompanhar tabelas específicas. Você pode escolher as tabelas a serem acompanhadas executando sys.sp_cdc_enable_table.

```
--Begin tracking a table
exec sys.sp_cdc_enable_table
   @source_schema           = N'source_schema'
,  @source_name             = N'source_name'
,  @role_name               = N'role_name'

--The following parameters are optional:

--, @capture_instance       = 'capture_instance'
--, @supports_net_changes   = supports_net_changes
--, @index_name             = 'index_name'
--, @captured_column_list   = 'captured_column_list'
--, @filegroup_name         = 'filegroup_name'
--, @allow_partition_switch = 'allow_partition_switch'
;
```

Para visualizar a configuração de CDC para suas tabelas, execute sys.sp_cdc_help_change_data_capture.

```
--View CDC configuration
exec sys.sp_cdc_help_change_data_capture

--The following parameters are optional and must be used together.
--  'schema_name', 'table_name'
;
```

Para obter mais informações sobre tabelas, funções e procedimentos armazenados de CDC na documentação do SQL Server, consulte o seguinte:

- Procedimentos armazenados de captura de dados de alteração (Transact-SQL)
- Change data capture functions (Transact-SQL)
- tabelas Change Data Capture (Transact-SQL)

Trabalhos de captura de dados de alteração

Quando você habilita a CDC, o SQL Server cria os trabalhos de CDC. Os proprietários de bancos de dados (`db_owner`) podem visualizar, criar, modificar e excluir trabalhos de CDC. Contudo, os trabalhos são de propriedade da conta do sistema do RDS. Portanto, os trabalhos não estão visíveis em visualizações, procedimentos nativos ou no SQL Server Management Studio.

Para controlar o comportamento de CDC em um banco de dados, use procedimentos nativos do SQL Server, como sp_cdc_enable_table e sp_cdc_start_job . Para alterar os parâmetros de trabalhos de CDC, como `maxtrans` e `maxscans`, use sp_cdc_change_job..

Para obter mais informações sobre os trabalhos de CDC, você pode consultar as seguintes visualizações de gerenciamento dinâmico:

- sys.dm_cdc_errors
- sys.dm_cdc_log_scan_sessions
- sysjobs
- sysjobhistory

Change Data Capture para instâncias multi-AZ

Se você usar a CDC em uma instância multi-AZ, verifique se a configuração do trabalho de CDC do espelho corresponde a do principal. Os trabalhos de CDC são mapeados para o `database_id`. Se os IDs de bancos de dados no secundário forem diferentes do principal, os trabalhos não serão associados ao banco de dados correto. Para tentar evitar erros após um failover, o RDS descarta e recria os trabalhos no novo principal. Os trabalhos recriados usam os parâmetros que o principal registrou antes do failover.

Embora esse processo seja executado rapidamente, ainda é possível que os trabalhos de CDC possam ser executados antes que o RDS possa corrigi-los. Estas são três maneiras de forçar os parâmetros para que sejam consistentes entre as réplicas primária e secundária:

- Use os mesmos parâmetros de trabalho para todos os bancos de dados que têm a CDC habilitada.

- Antes de alterar a configuração do trabalho de CDC, converta a instância multi-AZ para single-AZ.
- Transfira os parâmetros manualmente sempre que alterá-los no principal.

Para visualizar e definir os parâmetros de CDC usados para recriar os trabalhos de CDC após um failover, use `rds_show_configuration` e `rds_set_configuration`.

O exemplo a seguir retorna o valor definido para `cdc_capture_maxtrans`. Para qualquer parâmetro definido como `RDS_DEFAULT`, o RDS configura o valor automaticamente.

```
-- Show configuration for each parameter on either primary and secondary replicas.
exec rdsadmin.dbo.rds_show_configuration 'cdc_capture_maxtrans';
```

Para definir a configuração na secundária, execute `rdsadmin.dbo.rds_set_configuration`. Esse procedimento define os valores dos parâmetros para todos os bancos de dados no servidor secundário. Essas configurações são usadas somente após um failover. Os exemplo a seguir define o `maxtrans` de todos os trabalhos de captura de CDC como *`1000`*:

```
--To set values on secondary. These are used after failover.
exec rdsadmin.dbo.rds_set_configuration 'cdc_capture_maxtrans', 1000;
```

Para definir os parâmetros de trabalhos de CDC no principal, use sys.sp_cdc_change_job.

Usar o agente do SQL Server para o Amazon RDS

Com o Amazon RDS, você pode usar o SQL Server Agent em uma instância de banco de dados que executa o Microsoft SQL Server Enterprise Edition, Standard Edition ou Web Edition. SQL Server Agent é um serviço do Microsoft Windows que executa tarefas administrativas programadas, chamadas de trabalhos. É possível usar SQL Server Agent para executar trabalhos T-SQL para recompilar índices, executar verificações de corrupção e agregar dados em uma instância de banco de dados SQL Server.

Ao criar uma instância de banco de dados do SQL Server, o nome de usuário principal é registrado na função `SQLAgentUserRole`.

SQL Server Agent pode executar um trabalho agendado em resposta a um evento específico ou sob demanda. Para ter mais informações, consulte a página SQL Server Agent na documentação do SQL Server.

> **Note**
>
> Evite programar trabalhos para execução durante as janelas de manutenção e backup da sua instância de banco de dados. Os processos de manutenção e backup que são iniciados pela AWS podem interromper um trabalho ou fazer com que ele seja cancelado.
> Em implantações multi-AZ, os trabalhos do SQL Server Agent são replicados do host primário para o host secundário quando o recurso de replicação de trabalhos é ativado. Para obter mais informações, consulte Ativar a replicação de trabalhos do SQL Server Agent.
> As implantações multi-AZ têm um limite de cem trabalhos do SQL Server Agent. Se um limite mais alto for necessário, solicite um aumento de cota entrando em contato com o Suporte. Abra a página do AWS Support Center, faça login, se necessário, e escolha Create case (Criar caso). Escolha Service limit increase (Aumento de limite do serviço). Preencha e envie o formulário.

Para visualizar o histórico de um trabalho específico do SQL Server Agent no SQL Server Management Studio (SSMS), abra o Object Explorer, clique com o botão direito no trabalho e, em seguida, escolha View History (Visualizar histórico).

Como o SQL Server Agent está sendo executado em um host gerenciado em uma instância de banco de dados, algumas ações não são aceitas:

- Não é possível executar trabalhos de replicação e de scripts da linha de comando usando o ActiveX, o shell de comando do Windows ou o Windows PowerShell.
- Não é possível iniciar, interromper ou reiniciar manualmente o SQL Server Agent.
- As notificações de e-mail por meio do SQL Server Agent não estão disponíveis a partir de uma instância de banco de dados.
- Alertas e operadores do SQL Server Agent não são aceitos.
- O uso do SQL Server Agent para criar backups não é aceito. Use o Amazon RDS para fazer backup de sua instância de banco de dados.
- Atualmente, o RDS para SQL Server não comporta o uso de tokens do SQL Server Agent.

Ativar a replicação de trabalhos do SQL Server Agent

Você pode ativar a replicação de trabalhos do SQL Server Agent usando o seguinte procedimento armazenado:

```
EXECUTE msdb.dbo.rds_set_system_database_sync_objects @object_types = 'SQLAgentJob';
```

É possível executar o procedimento armazenado em todas as versões do SQL Server compatíveis com o Amazon RDS para SQL Server. Os trabalhos nas seguintes categorias são replicados:

- [Sem categoria (local)]
- [Sem categoria (vários servidores)]
- [Sem categoria]
- Coletor de dados
- Database Engine Tuning Advisor
- Manutenção do banco de dados
- Texto completo

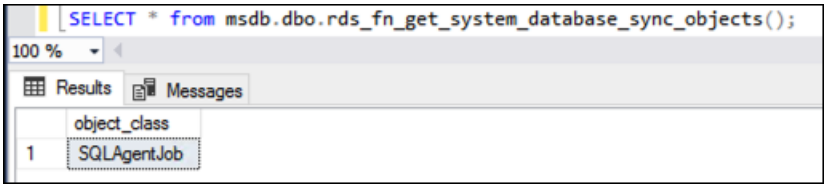
Somente trabalhos que usam etapas de trabalho T-SQL são replicados. Trabalhos com tipos de etapas, como o SQL Server Integration Services (SSIS), o SQL Server Reporting Services (SSRS), a Replicação e o PowerShell, não são replicados. Os trabalhos que usam o Database Mail e objetos no nível do servidor não são replicados.

> **Important**
>
> O host principal é a fonte da verdade para replicação. Antes de ativar a replicação do trabalho, verifique se os trabalhos do SQL Server Agent estão no host primário. Se você não fizer isso, poderá ocorrer a exclusão de seus trabalhos do SQL Server Agent se você ativar o recurso quando os trabalhos mais recentes estiverem no host secundário.

Você pode usar a função a seguir para confirmar se a replicação está ativada.

```
SELECT * from msdb.dbo.rds_fn_get_system_database_sync_objects();
```

A consulta T-SQL retornará o seguinte se os trabalhos do SQL Server Agent estiverem sendo replicados. Se eles não estiverem sendo replicados, não retornará nada para`object_class`.

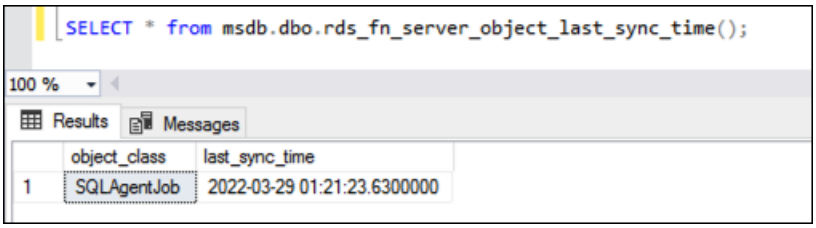


Você pode usar a função a seguir para descobrir a última vez que os objetos foram sincronizados no horário UTC.

```
SELECT * from msdb.dbo.rds_fn_server_object_last_sync_time();
```

Por exemplo, suponha que você modifique um trabalho do SQL Server Agent às 01:00. Você espera que o tempo de sincronização mais recente seja após a 01:00, indicando que a sincronização ocorreu.

Após a sincronização, espera-se que os valores retornados para `date_created` e `date_modified` no nó secundário coincidam.



Se você também estiver usando a replicação `tempdb`, poderá habilitar a replicação tanto para os trabalhos do SQL Agent quanto para a configuração `tempdb` fornecendo-os no parâmetro `@object_type`:

```
EXECUTE msdb.dbo.rds_set_system_database_sync_objects @object_types =
 'SQLAgentJob,TempDbFile';
```

Para ter mais informações sobre a replicação `tempdb`, consulte Configuração TempDB para implantações multi-AZ.

Perfis do SQL Server Agent

O RDS para SQL Server oferece suporte aos seguintes perfis do SQL Server Agent com diferentes níveis de permissões para gerenciar trabalhos:

- SQLAgentUserRole

  Permissões

  - Criar e gerenciar seus próprios trabalhos, agendamentos e operadores
  - Visualizar as propriedades de seus próprios trabalhos e agendamentos
  - Não é possível visualizar ou gerenciar trabalhos criados por outros usuários

Essa função é adequada para usuários que precisam criar e gerenciar seus próprios trabalhos, mas não requerem acesso aos trabalhos criados por outros usuários.

- SQLAgentReaderRole

  Permissões

  - Todas as permissões de SQLAgentUserRole
  - Visualizar uma lista de todos os trabalhos e agendamentos, incluindo aqueles criados por outros
  - Visualizar as propriedades de todos os trabalhos
  - Veja o histórico de trabalhos

  Essa função é adequada para usuários que precisam monitorar o status de todos os trabalhos, mas não precisam gerenciá-los.

- SQLAgentOperatorRole

  Permissões

  - Todas as permissões de SQLAgentUserRole e SQLAgentReaderRole
  - Executar, interromper ou iniciar trabalhos
  - Gerenciar o histórico de trabalhos
  - Ativar/desativar trabalhos e agendamentos
  - Visualizar operadores e proxies

  Esse perfil fornece as permissões mais abrangentes e é adequado para usuários que precisam ter controle total sobre todos os trabalhos.

Use o comando a seguir para atribuir as funções ao seu login do SQL Server:

```
USE msdb;
EXEC sp_addrolemember 'SQLAgentOperatorRole', 'username';
```

Gerenciamento do SQLAgentOperatorRole no RDS para SQL Server

Para visualizar os trabalhos atuais, você deve adicionar o SQLAgentOperatorRole ao seu login do SQL Server e removê-lo antes de se desconectar do banco de dados.

Para visualizar a árvore do SQL Server Agent no SQL Server Management Studio, siga estas instruções:

Visualizar o SQL Server Agent no SQL Server Management Studio (SSMS)

1. Usando as credenciais mestras do RDS, faça login na instância do SQL Server do RDS e conceda ao usuário desejado o SQLAgentUserRole.

```
USE msdb
GO
IF NOT EXISTS(SELECT name FROM sys.database_principals WHERE name = 'UserName')
BEGIN
CREATE USER UserName FROM LOGIN UserName
END
GO
ALTER ROLE SQLAgentUserRole ADD MEMBER UserName
GO
GRANT ALTER ON ROLE::[SQLAgentOperatorRole] to UserName
GO
```

   Esses comandos criam o usuário no banco de dados `msdb`, caso ele não exista. Ele também adiciona o usuário ao SQLAgentUserRole, para que a árvore do SQL Server Agent no SSMS possa ser visualizada. Por fim, ele concede permissões de alteração no SQLAgentOperatorRole ao usuário. Isso permite que o usuário adicione/remova a si mesmo dessa função.

2. Para se adicionar à função mencionada acima, conecte-se à instância do SQL Server do RDS com o usuário que precisa ver os trabalhos e execute o script a seguir.

```
use msdb
go
ALTER ROLE SQLAgentOperatorRole ADD MEMBER UserName
GO
```

   Depois disso, clique com o botão direito na pasta Jobs (Trabalhos) e escolha Refresh (Atualizar).

3. Quando você executa essa ação, a guia Jobs (Trabalhos) exibe um botão + (mais). Clique para expandir a lista de Jobs do SQL Server Agent.

4. **Important**

   Antes de se desconectar da instância do SQL Server do RDS, você precisa se remover da SQLAgentOperatorRole.

Para remover seu login do SQLAgentOperatorRole, execute a seguinte consulta antes de desconectar ou fechar o Management Studio:

```
USE msdb
GO
ALTER ROLE SQLAgentOperatorRole DROP MEMBER UserName
GO
```

Para ter mais informações, consulte Leveraging SQLAgentOperatorRole in RDS SQL Server.

Adicionar um usuário à função SQLAgentUser

Para permitir que um usuário ou login adicional use o SQL Server Agent, faça login com o usuário principal e realize as seguintes etapas:

1. Crie outro login no nível do servidor usando o comando `CREATE LOGIN`.
2. Crie um usuário no `msdb` usando o comando `CREATE USER` e, então, vincule esse usuário ao login criado na etapa anterior.
3. Adicione o usuário ao `SQLAgentUserRole` usando o procedimento armazenado pelo sistema `sp_addrolemember`.

Por exemplo, digamos que o nome de usuário principal seja **`admin`** e que você deseja conceder acesso ao SQL Server Agent para um usuário chamado **`theirname`** com uma senha **`theirpassword`**. Nesse caso, é possível ignorar o procedimento a seguir.

Para adicionar um usuário à função SQLAgentUser

1. Faça login como usuário mestre.
2. Execute os seguintes comandos:

```
--Initially set context to master database
USE [master];
GO
--Create a server-level login named theirname with password theirpassword
CREATE LOGIN [theirname] WITH PASSWORD = 'theirpassword';
GO
--Set context to msdb database
```

```
USE [msdb];
GO
--Create a database user named theirname and link it to server-level login
 theirname
CREATE USER [theirname] FOR LOGIN [theirname];
GO
--Added database user theirname in msdb to SQLAgentUserRole in msdb
EXEC sp_addrolemember [SQLAgentUserRole], [theirname];
```

Excluir um trabalho do SQL Server Agent

Você usa o procedimento armazenado `sp_delete_job` para excluir trabalhos do SQL Server Agent em Amazon RDS para Microsoft SQL Server.

Você não pode usar SSMS para excluir trabalhos do SQL Server Agent. Quando você tenta fazer isso, recebe uma mensagem de erro semelhante à seguinte:

```
The EXECUTE permission was denied on the object 'xp_regread', database
 'mssqlsystemresource', schema 'sys'.
```

Como um serviço gerenciado, o RDS é restrito na execução de procedimentos que acessam o Registro do Windows. Quando você usa o SSMS, ele tenta executar um processo (`xp_regread`) para o qual o RDS não está autorizado.

> **Note**
>
> No RDS para SQL Server, somente membros da função sysadmin podem atualizar ou excluir trabalhos pertencentes a um login diferente. Para ter mais informações, consulte Leveraging SQLAgentOperatorRole in RDS SQL Server.

Para excluir um trabalho do SQL Server Agent

- Execute a seguinte instrução T-SQL:

```
EXEC msdb..sp_delete_job @job_name = 'job_name';
```

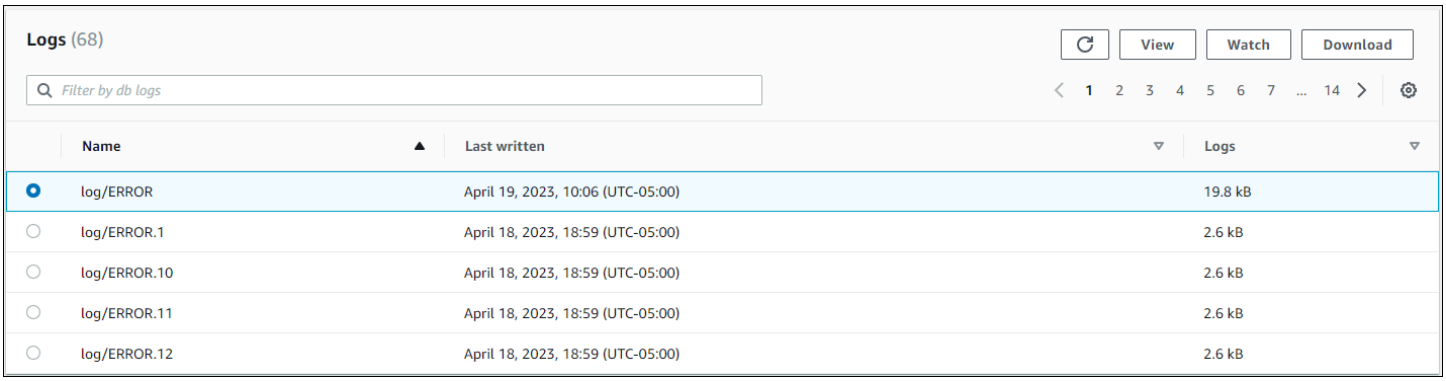
Trabalhar com logs do Amazon RDS para Microsoft SQL Server

É possível usar o console do Amazon RDS para visualizar, monitorar e baixar os logs do SQL Server Agent e os logs de erros do Microsoft SQL Server e os logs do SQL Server Reporting Services (SSRS)

Monitorar arquivos de log

Ao visualizar um log no console do Amazon RDS, você pode ver o conteúdo conforme ele é naquele momento. Ao observar um log no console, ele é aberto em um estado dinâmico para que você possa ver as atualizações feitas nele em tempo quase real.

Só é possível monitorar o log mais recente. Por exemplo, digamos que os seguintes logs mostrem:



Somente log/ERRO está sendo ativamente atualizado, por ser o log mais recente. É possível monitorar os demais, mas eles são estáticos e não serão atualizados.

Guardar arquivos de log

O console do Amazon RDS mostra os logs da última semana até o dia atual. É possível baixar os logs e arquivá-los se você deseja mantê-los para fins de referência após esse período. Uma maneira de arquivar logs é carregá-los em uma bucket do Amazon S3. Para obter instruções sobre como configurar um bucket do Amazon S3 e fazer upload de um arquivo, consulte Noções básicas do Amazon S3 no Guia de conceitos básicos do Amazon Simple Storage Service e clique em Get Started (Conceitos básicos).

Visualizar logs do agente ou erros

Para visualizar os logs de erro e do agente do Microsoft SQL Server, use o procedimento armazenado do Amazon RDS `rds_read_error_log` com os seguintes parâmetros:

- **`@index`** – a versão do log para recuperação. O valor padrão é 0, que recupera o log de erro atual. Especifique 1 para recuperar o registro anterior, especifique 2 para recuperar o anterior a ele, e assim por diante.
- **`@type`** – o tipo do log para recuperação. Especifique 1 para recuperar um log de erro. Especifique 2 para recuperar um log de agente.

Example

O exemplo a seguir solicita o log de erro atual.

```
EXEC rdsadmin.dbo.rds_read_error_log @index = 0, @type = 1;
```

Para obter mais informações sobre erros do SQL Server, consulte Erros do mecanismo de banco de dados na documentação da Microsoft.

Trabalhar com arquivos de rastreamento e despejo do Amazon RDS para SQL Server

Esta seção descreve como trabalhar com arquivos de rastreamento e arquivos de despejo para as instâncias de banco de dados do Amazon RDS que executam o Microsoft SQL Server.

Como gerar uma consulta de SQL de rastreamento

```
declare @rc int
declare @TraceID int
declare @maxfilesize bigint

set @maxfilesize = 5

exec @rc = sp_trace_create @TraceID output,  0, N'D:\rdsdbdata\log\rdstest',
 @maxfilesize, NULL
```

Como visualizar um rastreamento aberto

```
select * from ::fn_trace_getinfo(default)
```

Como visualizar o conteúdo de um rastreamento

```
select * from ::fn_trace_gettable('D:\rdsdbdata\log\rdstest.trc', default)
```

Como definir o período de retenção para arquivos de rastreamento e despejo

Os arquivos de rastreamento e despejo podem acumular e consumir espaço em disco. Por padrão, o Amazon RDS limpa os arquivos de rastreamento e despejo com mais de sete dias.

Para visualizar o período de retenção atual de arquivos de rastreamento e despejo, use o procedimento `rds_show_configuration`, conforme mostrado no exemplo a seguir.

```
exec rdsadmin..rds_show_configuration;
```

Para modificar o período de retenção dos arquivos de rastreamento, use o procedimento `rds_set_configuration` e defina a `tracefile retention` em minutos. O exemplo a seguir define o período de retenção para arquivos de rastreamento em 24 horas.

```
exec rdsadmin..rds_set_configuration 'tracefile retention', 1440;
```

Para modificar o período de retenção dos arquivos de despejo, use o procedimento `rds_set_configuration` e defina a `dumpfile retention` em minutos. O exemplo a seguir define o período de retenção para arquivos de despejo em 3 dias.

```
exec rdsadmin..rds_set_configuration 'dumpfile retention', 4320;
```

Por motivos de segurança, não é possível excluir um arquivo específico de rastreamento ou despejo em uma instância de banco de dados do SQL Server. Para excluir todos os arquivos de rastreamento ou despejo não utilizados, defina o período de retenção dos arquivos como 0.

Amazon RDS para MySQL

O Amazon RDS comporta várias versões do MySQL para instâncias de banco de dados. Para ter informações completas sobre as versões compatíveis, consulte MySQL em versões do Amazon RDS.

Para criar uma instância de banco de dados do Amazon RDS para MySQL, use as ferramentas de gerenciamento ou interfaces do Amazon RDS. Você pode, então, fazer o seguinte:

- Redimensionar sua instância de banco de dados
- Autorizar conexões com sua instância de banco de dados
- Criar e restaurar a partir de backups ou snapshots
- Criar secundários de multi-AZ
- Criar réplicas de leitura
- Monitorar a performance de sua instância de banco de dados

Para armazenar e acessar os dados em sua instância de banco de dados, use utilitários e aplicações padrão do MySQL.

O Amazon RDS para MySQL é compatível com vários padrões do setor. Por exemplo, você pode usar bancos de dados do RDS for MySQL para criar aplicações em conformidade com a norma HIPAA. É possível usar bancos de dados do RDS for MySQL para armazenar informações relacionadas à saúde, inclusive informações de saúde protegidas (PHI), segundo um Acordo de Associação Comercial (BAA) com a AWS. O Amazon RDS para MySQL também atende aos requisitos de segurança do Federal Risk and Authorization Management Program (FedRAMP). Além disso, o Amazon RDS para MySQL recebeu uma autoridade provisória da Joint Authorization Board (JAB) da FedRAMP para operar (P-ATO) na linha de base ALTA da FedRAMP nas regiões AWS GovCloud (US). Para obter mais informações sobre padrões de conformidade com suporte, consulte Conformidade da nuvem AWS.

Para obter informações sobre os recursos em cada versão do MySQL, consulte Principais recursos do MySQL na documentação do MySQL.

Antes de criar uma instância de banco de dados, conclua as etapas em Configuração do ambiente do Amazon RDS. Ao criar uma instância de banco de dados, o usuário principal do RDS obtém privilégios de DBA, com algumas limitações. Use essa conta para tarefas administrativas, como a criação de contas de banco de dados adicionais.

Você pode criar o seguinte:

- Instâncias de banco de dados
- DB snapshots
- Restaurações point-in-time
- Backups automatizados
- Backups manuais

Você pode usar instâncias de banco de dados executando o MySQL em uma nuvem privada virtual (VPC) com base na Amazon VPC. Também é possível adicionar recursos à sua instância de banco de dados do MySQL ativando várias opções. O Amazon RDS é compatível com implantações multi-AZ para MySQL como uma solução de failover de alta disponibilidade.

> **Important**
>
> Para oferecer uma experiência de serviço gerenciada, o Amazon RDS não fornece acesso ao shell para as instâncias de banco de dados. Também restringe o acesso a determinados procedimentos e tabelas do sistema que necessitam de privilégios avançados. É possível acessar seu banco de dados usando clientes SQL padrão, como o cliente mysql. No entanto, não é possível acessar o host diretamente usando Telnet ou Secure Shell (SSH).

Tópicos

- Compatibilidade de recursos do MySQL no Amazon RDS
- MySQL em versões do Amazon RDS
- Conectar-se à instância de banco de dados do MySQL
- Proteger as conexões de instância de banco de dados do MySQL
- Melhorar a performance das consultas para o RDS para MySQL com leituras otimizadas do Amazon RDS
- Melhorar a performance das gravações com gravações otimizadas pelo RDS para MySQL
- Atualizações do mecanismo de banco de dados do RDS para MySQL
- Atualizar uma versão do mecanismo de snapshot de banco de dados do MySQL
- Importar dados para uma instância de banco de dados do Amazon RDS para MySQL.
- Trabalhar com a replicação do MySQL no Amazon RDS

- Configurar clusters ativos-ativos para RDS para MySQL
- Exportar dados de uma instância de banco de dados MySQL usando replicação
- Opções para instâncias de bancos de dados MySQL
- Parâmetros do MySQL
- Tarefas comuns de DBA para instâncias de banco de dados MySQL
- Fuso horário local para instâncias de banco de dados MySQL
- Problemas conhecidos e limitações do Amazon RDS para MySQL
- Referência de procedimentos armazenados do RDS para MySQL

Compatibilidade de recursos do MySQL no Amazon RDS

O RDS para MySQL é compatível com a maioria dos recursos do MySQL. Alguns recursos podem ter suporte limitado ou privilégios restritos.

Você pode filtrar novos recursos do Amazon RDS na página What's New with Database? (Novidades sobre bancos de dados). Em Products (Produtos), escolha Amazon RDS. Em seguida, pesquise usando palavras-chave como **`MySQL 2022`**.

> **Note**
>
> As listas a seguir não são completas.

Tópicos

- Suporte a recursos do MySQL nas versões principais do Amazon RDS para MySQL
- Suporte aos mecanismos de armazenamento no RDS para MySQL
- Usar memcached e outras opções com o MySQL no Amazon RDS
- Aquecimento de cache do InnoDB para o MySQL no Amazon RDS
- Alterações de linguagem inclusiva para RDS para MySQL 8.4
- Recursos do MySQL sem suporte pelo Amazon RDS

Suporte a recursos do MySQL nas versões principais do Amazon RDS para MySQL

Nas seções a seguir, localize informações sobre compatibilidade de recursos do MySQL nas versões principais do Amazon RDS para MySQL:

Tópicos

- Suporte ao MySQL 8.4 no Amazon RDS

Para obter informações sobre versões secundárias compatíveis do Amazon RDS para MySQL, consulte Versões secundárias do MySQL compatíveis com o Amazon RDS.

Suporte ao MySQL 8.4 no Amazon RDS

O Amazon RDS oferece suporte aos novos recursos descritos a seguir para instâncias de banco de dados que executam o MySQL versão 8.4 ou posterior.

- Biblioteca criptográfica: o RDS para MySQL substituiu o OpenSSL pelo AWS Libcrypto (AWS-LC), que tem certificação da norma FIPS 140-3. Para obter mais informações, consulte o repositório do GitHub AWS-LC em https://github.com/aws/aws-lc.
- Alterações do TLS: o RDS para MySQL é compatível somente com o TLS 1.2 e o TLS 1.3. Para ter mais informações, consulte the section called “Suporte de SSL/TLS do MySQL”.
- Suporte ao memcached: a interface do memcached não está mais disponível no MySQL 8.4. Para ter mais informações, consulte Suporte a memcached MySQL.
- Plugin de autenticação padrão: o plugin de autenticação padrão é `caching_sha2_password`. Para ter mais informações, consulte the section called “Plug-in de autenticação padrão do MySQL”.
- Utilitário cliente **`mysqlpump`**: o utilitário cliente `mysqlpump` não está mais disponível no MySQL 8.4. Para obter mais informações, consulte Modelo de privilégios baseados em perfis para o RDS para MySQL e mysqldump e mysqlpump na Orientação prescritiva da AWS.
- Procedimentos armazenados de replicação gerenciada: ao usar procedimentos armazenados para gerenciar a replicação com um usuário de replicação configurado com `caching_sha2_password`, você deve configurar o TLS especificando `SOURCE_SSL=1`. `caching_sha2_password` é o plugin de autenticação padrão para o RDS para MySQL 8.4.
- Mudanças no comportamento dos parâmetros: os seguintes parâmetros foram alterados para o MySQL 8.4.
  - `innodb_dedicated_server`: por padrão, esse parâmetro está ativado. Para ter mais informações, consulte Configuração do tamanho do pool de buffer e a capacidade de log redo no MySQL 8.4.
  - `innodb_buffer_pool`: o mecanismo do banco de dados agora calcula esse parâmetro, mas você pode substituir essa configuração. Para ter mais informações, consulte Configuração do tamanho do pool de buffer e a capacidade de log redo no MySQL 8.4.
  - `innodb_redo_log_capacity`: esse parâmetro agora controla o tamanho dos arquivos de log de refazer. O mecanismo do banco de dados agora calcula esse parâmetro, mas você pode substituir essa configuração. Para ter mais informações, consulte Configuração do tamanho do pool de buffer e a capacidade de log redo no MySQL 8.4.

- Parâmetros obsoletos ou removidos: o RDS for MySQL removeu os seguintes parâmetros dos grupos de parâmetros para instâncias de banco de dados MySQL 8.4. O parâmetro `innodb_redo_log_capacity` agora controla o tamanho dos arquivos de log de refazer.
  - `innodb_log_file_size`
  - `innodb_log_files_in_group`
- Novos valores padrão para parâmetros: os seguintes parâmetros têm novos valores padrão para instâncias de banco de dados MySQL versão 8.4:
  - Vários parâmetros da comunidade MySQL relacionados ao desempenho foram alterados. Para obter mais informações, consulte O que há de novo no MySQL 8.4 desde o MySQL 8.0.

    Recomendamos que você teste o desempenho da aplicação no RDS for MySQL 8.4 antes de migrar uma instância de produção.
  - `innodb_purge_threads`: o valor padrão é definido como a fórmula `LEAST({DBInstanceVCPU/2},4)` para evitar que o comprimento da lista de histórico do InnoDB cresça demais.
  - `group_replication_exit_state_action`: o valor padrão é `OFFLINE_MODE`, que se alinha ao padrão no MySQL Community. Para ter mais informações, consulte Considerações e práticas recomendadas para cluster ativo-ativo do RDS para MySQL.
  - `binlog_format`: o valor padrão é `ROW`, que se alinha ao padrão no MySQL Community. Você pode modificar o parâmetro para instâncias de banco de dados single-AZ ou multi-AZ, mas não para clusters de banco de dados multi-AZ. Os clusters de banco de dados multi-AZ usam replicação semissíncrona e, quando `binlog_format` é configurado como `MIXED` ou `STATEMENT`, a replicação falha.
- Alterações de linguagem inclusiva: o RDS para MySQL 8.4 inclui alterações do RDS para MySQL 8.0 relacionadas a palavras-chave e esquemas de sistema para linguagem inclusiva. Para ter mais informações, consulte Alterações de linguagem inclusiva para RDS para MySQL 8.4.

Para obter uma lista de todos os recursos e alterações do MySQL 8.4, consulte O que há de novo no MySQL 8.4 desde o MySQL 8.0 na documentação do MySQL.

Para conferir uma lista de recursos não compatíveis, consulte the section called “Recursos não compatíveis”.

Suporte aos mecanismos de armazenamento no RDS para MySQL

Embora o MySQL ofereça suporte a vários mecanismos de armazenamento com recursos variados, nem todos eles são otimizados para durabilidade e recuperação de dados. O Amazon RDS oferece suporte total ao mecanismo de armazenamento do InnoDB para instâncias de banco de dados do MySQL. Os recursos de restauração point-in-time e de snapshot do Amazon RDS exigem um mecanismo de armazenamento recuperável e são compatíveis somente com o mecanismo de armazenamento do InnoDB. Para obter mais informações, consulteSuporte a memcached MySQL

O Federated Storage Engine atualmente não é compatível com o Amazon RDS para MySQL.

Para esquemas criados pelo usuário, o mecanismo de armazenamento MyISAM não oferece suporte para recuperação confiável e pode resultar em dados perdidos ou corrompidos quando o MySQL é reiniciado após uma recuperação, impedindo que uma restauração pontual ou uma restauração de snapshot funcione conforme pretendido. No entanto, se você ainda optar por usar o MyISAM com o Amazon RDS, snapshots podem ser úteis em algumas condições.

> **Note**
>
> As tabelas do sistema no esquema `mysql` podem estar no armazenamento MyISAM.

Se você deseja converter tabelas do MyISAM em tabelas do InnoDB, pode usar o comando `ALTER TABLE` (por exemplo, `alter table TABLE_NAME engine=innodb;`). Lembre-se de que o MyISAM e o InnoDB têm pontos fortes e fracos diferentes. Portanto, é necessário que você avalie totalmente o impacto que essas alterações terão em seus aplicativos antes de fazê-las.

Os MySQL 5.1, 5.5 e 5.6 não são mais compatíveis com o Amazon RDS. No entanto, é possível restaurar snapshots do MySQL 5.1, 5.5 e 5.6 existentes. Ao restaurar um snapshot do MySQL 5.1, 5.5 ou 5.6, a instância de banco de dados é automaticamente atualizada para o MySQL 5.7.

Usar memcached e outras opções com o MySQL no Amazon RDS

A maioria dos mecanismos de banco de dados do Amazon RDS oferece suporte a grupos de opções que permitem selecionar recursos adicionais para a sua instância de banco de dados. As instâncias de banco de dados do RDS para MySQL são compatíveis com a opção `memcached`, um cache simples baseado em chaves. Para obter mais informações sobre `memcached` e outras opções, consulte Opções para instâncias de bancos de dados MySQL. Para mais informações sobre como trabalhar com grupos de opções, consulte Trabalhar com grupos de opções.

Aquecimento de cache do InnoDB para o MySQL no Amazon RDS

O aquecimento de cache do InnoDB pode fornecer ganhos de performance para sua instância de banco de dados do MySQL salvando o estado atual do grupo de buffer quando a instância de banco de dados é desativada e, em seguida, recarregando o grupo de buffer a partir das informações salvas quando a instância do banco de dados for iniciada. Isso dispensa a necessidade de o grupo de buffer "aquecer" do uso normal de banco de dados e, em vez disso, pré-carrega o grupo de buffer com as páginas para consultas comuns conhecidas. O arquivo que armazena as informações do pool de buffer salvo armazena apenas os metadados para as páginas que estão no pool de buffer, e não as páginas propriamente ditas. Como resultado, o arquivo não requer muito espaço de armazenamento. O tamanho do arquivo é de aproximadamente 0,2 por cento do tamanho do cache. Por exemplo, para um cache de 64 GiB, o tamanho do arquivo de aquecimento do cache é de 128 MiB. Para obter mais informações sobre o aquecimento de cache do InnoDB, consulte Saving and restoring the buffer pool state na documentação do MySQL.

As instâncias de banco de dados do RDS para MySQL são compatíveis com o aquecimento de cache InnoDB. Para habilitar o aquecimento de cache do InnoDB, defina os parâmetros `innodb_buffer_pool_dump_at_shutdown` e `innodb_buffer_pool_load_at_startup` como 1 no grupo de parâmetros da sua instância de banco de dados. Mudar esses valores de parâmetro em um grupo de parâmetros afetará todas as instâncias de banco de dados MySQL que usam esse grupo de parâmetros. Para habilitar o aquecimento de cache do InnoDB para instâncias específicas de banco de dados MySQL, talvez você precise criar um grupo de parâmetros nessas instâncias. Para obter informações sobre grupos de parâmetros, consulte Grupos de parâmetros para Amazon RDS.

O aquecimento de cache do InnoDB fornece principalmente um benefício de performance para instâncias de banco de dados que usam armazenamento padrão. Se você usar o armazenamento PIOPS, não perceberá um benefício significativo de performance normalmente.

> **Important**
>
> Se sua instância de banco de dados MySQL não for encerrada normalmente, como durante um failover, o estado do pool de buffers não será salvo no disco. Nesse caso, o MySQL carrega qualquer arquivo de grupo de buffer disponível quando a instância de banco de dados for reiniciada. Não há problema, mas o grupo de buffer restaurado pode não refletir o estado mais recente do grupo de buffer antes do reinício. Para garantir que você tenha um estado recente do grupo de buffer disponível para aquecer o cache do InnoDB na

inicialização, recomendamos que você descarte periodicamente o grupo de buffer "sob demanda".
Você pode criar um evento para descartar o grupo de buffer automaticamente e em um intervalo regular. Por exemplo, a seguinte declaração cria um evento chamado `periodic_buffer_pool_dump` que descarta o grupo de buffer de hora em hora.

```
CREATE EVENT periodic_buffer_pool_dump
ON SCHEDULE EVERY 1 HOUR
DO CALL mysql.rds_innodb_buffer_pool_dump_now();
```

Para obter mais informações sobre eventos do MySQL, consulte Event syntax na documentação do MySQL.

Como descartar e carregar o grupo de buffer sob demanda

É possível salvar e carregar o cache InnoDB “sob demanda”.

- Para descartar o estado atual do grupo de buffer para o disco, chame o procedimento armazenado mysql.rds_innodb_buffer_pool_dump_now.
- Para carregar o estado salvo do grupo de buffer a partir do disco, chame o procedimento armazenado mysql.rds_innodb_buffer_pool_load_now.
- Para cancelar uma operação de carregamento em andamento, chame o procedimento armazenado mysql.rds_innodb_buffer_pool_load_abort.

Alterações de linguagem inclusiva para RDS para MySQL 8.4

O RDS para MySQL versão 8.4 também inclui alterações da edição da comunidade do MySQL 8.4 referentes a palavras-chave e esquemas de sistema para linguagem inclusiva. Por exemplo, o comando `SHOW REPLICA STATUS` substitui `SHOW SLAVE STATUS`.

Tópicos

- Alterações do nome do parâmetro de configuração
- Alterações do nome dos procedimentos armazenados

Alterações do nome do parâmetro de configuração

Os seguintes parâmetros de configuração têm nomes novos no RDS para MySQL 8.4.

Para fins de compatibilidade, é possível verificar os nomes dos parâmetros no cliente `mysql` utilizando qualquer nome. Você só pode usar os novos nomes ao modificar os valores em um grupo de parâmetros personalizado do MySQL 8.4. Para ter mais informações, consulte Grupos de parâmetros padrão e personalizados.

Nome a ser removido	Nome novo ou preferencial
`init_slave`	`init_replica`
`log_slave_updates`	`log_replica_updates`
`log_slow_slave_statements`	`log_slow_replica_statements`
`rpl_stop_slave_timeout`	`rpl_stop_replica_timeout`
`skip_slave_start`	`skip_replica_start`
`slave_checkpoint_group`	`replica_checkpoint_group`
`slave_checkpoint_period`	`replica_checkpoint_period`
`slave_compressed_protocol`	`replica_compressed_protocol`
`slave_exec_mode`	`replica_exec_mode`
`slave_load_tmpdir`	`replica_load_tmpdir`
`slave_max_allowed_packet`	`replica_max_allowed_packet`
`slave_net_timeout`	`replica_net_timeout`

Nome a ser removido	Nome novo ou preferencial
slave_parallel_type	replica_parallel_type
slave_parallel_workers	replica_parallel_workers
slave_pending_jobs_size_max	replica_pending_jobs_size_max
slave_preserve_commit_order	replica_preserve_commit_order
slave_skip_errors	replica_skip_errors
slave_sql_verify_checksum	replica_sql_verify_checksum
slave_transaction_retries	replica_transaction_retries
slave_type_conversions	replica_type_conversions
sql_slave_skip_counter	sql_replica_skip_counter

> **Note**
>
> O parâmetro `replica_allow_batching` não está disponível porque o Amazon RDS não é compatível com clusters do NDB.

Alterações do nome dos procedimentos armazenados

Os seguintes procedimentos armazenados têm novos nomes no RDS para MySQL 8.4.

Para compatibilidade, é possível utilizar qualquer um dos nomes na versão inicial do RDS para MySQL versão 8.4. Os nomes de procedimentos antigos serão removidos em um release futuro. Para ter mais informações, consulte Configurar, iniciar e interromper a replicação de logs binários (binlogs).

Nome a ser removido	Nome novo ou preferencial
`mysql.rds_next_master_log`	`mysql.rds_next_source_log`
`mysql.rds_reset_external_master`	`mysql.rds_reset_external_source`
`mysql.rds_set_external_master`	`mysql.rds_set_external_source`
`mysql.rds_set_external_master_with_auto_position`	`mysql.rds_set_external_source_with_auto_position`
`mysql.rds_set_external_master_with_delay`	`mysql.rds_set_external_source_with_delay`
`mysql.rds_set_master_auto_position`	`mysql.rds_set_source_auto_position`

Recursos do MySQL sem suporte pelo Amazon RDS

Atualmente, o Amazon RDS não oferece suporte aos seguintes recursos do MySQL:

- Plugin de autenticação
- Registro de erros no log do sistema
- Criptografia de espaço de tabela do InnoDB
- Clusters do NDB
- Plugin de intensidade da senha
- Variáveis de sistema mantidas

- Plugin de regravação de consulta de regravador
- Replicação semissíncrona, exceto para clusters de banco de dados multi-AZ
- Espaço de tabela transportável
- Plug-in X

Para oferecer uma experiência de serviço gerenciada, o Amazon RDS não fornece acesso ao shell para as instâncias de banco de dados. Ele também restringe o acesso a determinados procedimentos e tabelas do sistema que exigem privilégios avançados. O Amazon RDS oferece suporte a acesso a bancos de dados em uma instância de banco de dados que usa qualquer aplicação de cliente SQL padrão. O Amazon RDS não permite o acesso direto do host a uma instância de banco de dados usando Telnet, Secure Shell (SSH) ou Windows Remote Desktop Connection. Ao criar uma instância de banco de dados, você recebe a função db_owner para todos os bancos de dados nessa instância e tem todas as permissões em nível de banco de dados, exceto aquelas usadas para backups. O Amazon RDS gerencia backups para você.

MySQL em versões do Amazon RDS

No MySQL, os números de versão são organizados como versão = X.Y.Z. Na terminologia do Amazon RDS, X.Y representa a versão principal, e Z é o número da versão secundária. Para implementações do Amazon RDS, uma alteração da versão é considerada primária se o número da versão primária for alterado—, por exemplo, da versão 5.7 para a 8.0. Uma alteração da versão será considerada secundária se apenas o número da versão secundária for alterado; por exemplo, da versão 8.0.32 para 8.0.34.

Tópicos

- Versões secundárias do MySQL compatíveis com o Amazon RDS
- Versões principais do MySQL compatíveis com o Amazon RDS
- Versões do Suporte estendido do Amazon RDS para RDS para MySQL
- Trabalhar com o ambiente de visualização de banco de dados
- MySQL versão 8.4 no Ambiente de Pré-visualização do Banco de Dados
- MySQL versão 8.3 no Ambiente de Pré-visualização do Banco de Dados
- O MySQL versão 8.2 no Ambiente de Pré-visualização do Banco de Dados
- PostgreSQL versão 12.0 no ambiente de visualização de banco de dados
- Versões obsoletas do Amazon RDS para MySQL

Versões secundárias do MySQL compatíveis com o Amazon RDS

No momento, o Amazon RDS é compatível com as versões secundárias do MySQL a seguir.

> **Note**
>
> As datas com apenas mês e ano são aproximadas e são atualizadas com uma data exata quando ela é conhecida.
> O Suporte estendido do Amazon RDS não está disponível para versões secundárias.

No momento, a tabela a seguir mostra as versões secundárias do MySQL 8.0 compatíveis com o Amazon RDS.

Versão do mecanismo do MySQL	Data de lançamento para a comunidade	Data de lançamento do RDS	Data do fim do suporte padrão para o RDS
8.0.39	23 de julho de 2024	12 de agosto de 2024	Setembro de 2025
8.0.37	30 de abril de 2024	18 de junho de 2024	Setembro de 2025
8.0.36	16 de janeiro de 2018	12 de fevereiro de 2024	Março de 2024
8.0.35	4 de outubro de 2023	9 de novembro de 2022	Março de 2024
8.0.34	18 de julho de 2023	9 de agosto de 2023	Março de 2024
8.0.33	18 de abril de 2023	15 de junho de 2023	Março de 2024
8.0.32	17 de janeiro de 2023	7 de fevereiro de 2023	Março de 2024

No momento, a tabela a seguir mostra as versões secundárias do MySQL 5.7 compatíveis com o Amazon RDS Extended Support.

Versão do mecanismo do MySQL	Data de lançamento para a comunidade	Data de lançamento do RDS	Data de término do suporte estendido para o RDS
5.7.44-RDS.20240529*	Não aplicável	25 de junho de 2024	Setembro de 2025
5.7.44-RDS.20240408*	Não aplicável	17 de maio de 2024	Setembro de 2025
5.7.44	4 de outubro de 2023	2 de novembro de 2017	Março de 2024

* O MySQL Community retirou a versão principal 5.7 e não lançará novas versões secundárias. Essa é uma versão secundária que o Amazon RDS lançou com patches de segurança e correções de erros críticos para bancos de dados do MySQL 5.7 que são cobertos pelo Suporte estendido do RDS. Para ter mais informações sobre essas versões secundárias, consulte the section called "Versões do Suporte estendido do RDS para RDS para MySQL". Para ter mais informações sobre o Suporte estendido do RDS, consulte Suporte estendido do Amazon RDS com Amazon RDS.

As versões secundárias podem chegar ao fim do suporte padrão antes das versões principais. Por exemplo, a versão secundária 8.0.28 atingiu a data de fim do suporte padrão em 28 de março de 2024, enquanto a versão principal 8.0 alcançará essa data em 31 de julho de 2026. O RDS comportará versões secundárias 8.0.* adicionais que a comunidade do MySQL lançará entre essas datas.

Você pode especificar qualquer versão atualmente com suporte do MySQL ao criar uma nova instância de banco de dados. Você pode especificar a versão principal (como o MySQL 5.7) e qualquer versão secundária com suporte para a versão principal especificada. Se nenhuma versão for especificada, o Amazon RDS usará uma versão compatível por padrão, geralmente a versão mais recente. Se uma versão principal for especificada, mas uma versão secundária não, o Amazon RDS usará por padrão um lançamento recente da versão principal que você especificou. Para ver uma lista de versões com suporte, bem como os padrões para instâncias de banco de dados recém-criadas, use o comando da `describe-db-engine-versions` AWS CLI.

Por exemplo, para listar as versões de mecanismo compatíveis com o RDS para MySQL, execute o comando da CLI a seguir:

```
aws rds describe-db-engine-versions --engine mysql --query "*[].
{Engine:Engine,EngineVersion:EngineVersion}" --output text
```

A versão padrão do MySQL pode variar de acordo com a Região da AWS. Para criar uma instância de banco de dados com uma versão secundária específica, especifique a versão secundária durante a criação da instância de banco de dados. É possível determinar a versão secundária padrão para uma Região da AWS usando o seguinte comando da AWS CLI:

```
aws rds describe-db-engine-versions --default-only --engine mysql
 --engine-version major-engine-version --region region --query "*[].
{Engine:Engine,EngineVersion:EngineVersion}" --output text
```

Substitua *`major-engine-version`* pela versão principal do mecanismo e substitua *`region`* pela Região da AWS. Por exemplo, o comando da AWS CLI a seguir retorna a versão secundária padrão

do mecanismo do MySQL para a versão principal 5.7 e a Região da AWS Oeste dos EUA (Oregon) (us-west-2):

```
aws rds describe-db-engine-versions --default-only --engine mysql --engine-version 5.7
 --region us-west-2 --query "*[].{Engine:Engine,EngineVersion:EngineVersion}" --output
 text
```

Com o Amazon RDS, você controla quando vai atualizar a instância do MySQL para uma nova versão principal compatível com o Amazon RDS. É possível manter a compatibilidade com versões específicas do MySQL, testar novas versões com o aplicativo antes de implantar na produção e realizar atualizações da versão principal no período que melhor se adequar à sua programação.

Com a atualização automática de versão secundária habilitada, a instância de banco de dados é atualizada automaticamente para novas versões secundárias do MySQL, pois elas são compatíveis com o Amazon RDS. Essa correção ocorre durante a janela de manutenção programada. É possível modificar uma instância de banco de dados para habilitar ou desabilitar atualizações automáticas de versões secundárias.

Se você cancelar os upgrades automaticamente agendados, poderá fazer upgrade manualmente para um lançamento de versão secundária com suporte seguindo o mesmo procedimento de um upgrade de versão principal. Para ter mais informações, consulte Atualizar a versão de mecanismo de uma instância de banco de dados.

Atualmente, o Amazon RDS oferece suporte às atualizações de versão primária do MySQL versão 5.6 para a versão 5.7 e do MySQL versão 5.7 para a versão 8.0. Como as atualizações da versão primária envolvem um certo risco de compatibilidade, elas não ocorrem automaticamente; é necessário fazer uma solicitação para modificar a instância de banco de dados. Você deve testar completamente qualquer atualização antes de atualizar suas instâncias de produção. Para ter mais informações sobre como atualizar uma instância de banco de dados MySQL, consulte Atualizações do mecanismo de banco de dados do RDS para MySQL.

Você pode testar uma instância de banco de dados em relação a uma nova versão antes de atualizar, criando um snapshot de banco de dados da sua instância de banco de dados atual, restaurando do snapshot de banco de dados para criar uma nova instância de banco de dados e, depois, iniciando uma atualização de versão para a nova instância de banco de dados. Isso permite que você teste com segurança o clone atualizado de sua instância de banco de dados antes de decidir se deseja ou não atualizar sua instância de banco de dados original.

Versões secundárias do MySQL no Amazon RDS

Versões secundárias

- MySQL versão 8.0.39
- MySQL versão 8.0.37

MySQL versão 8.0.39

O MySQL versão 8.0.39 já está disponível no Amazon RDS. Essa versão contém correções e melhorias adicionadas pela comunidade do MySQL e pelo Amazon RDS.

Novos recursos e aprimoramentos

- Corrigimos um erro que impedia que `sql_log_off` funcionasse corretamente com o privilégio `SESSION_VARIABLES_ADMIN`.
- Corrigimos um erro que impedia o usuário principal de conceder o privilégio `SESSION_VARIABLE_ADMIN` a outros usuários do banco de dados.
- Corrigimos um erro que causava uma combinação ilegal de agrupamento ao executar procedimentos armazenados fornecidos pelo RDS.

MySQL versão 8.0.37

O MySQL versão 8.0.37 já está disponível no Amazon RDS. Essa versão contém correções e melhorias adicionadas pela comunidade do MySQL e pelo Amazon RDS.

Novos recursos e aprimoramentos

Correção de um bug referente à execução de uma instrução de DDL instantânea após uma ATUALIZAÇÃO que causava uma falha na declaração.

Versões principais do MySQL compatíveis com o Amazon RDS

As versões principais do RDS para MySQL estarão disponíveis pelo suporte padrão pelo menos até o fim da vida útil da comunidade para a versão da comunidade correspondente. Você pode continuar executando uma versão principal após a data de término do suporte padrão do RDS mediante o pagamento de uma taxa. Para obter mais informações, consulte Suporte estendido do Amazon RDS com Amazon RDSPreço do Amazon RDS para MySQL.

É possível usar as seguintes datas para planejar seus ciclos de teste e atualização.

> **Note**
>
> As datas com apenas mês e ano são aproximadas e são atualizadas com uma data exata quando ela é conhecida.

Versão principal do MySQL	Data de lançamento para a comunidade	Data de lançamento do RDS	Data do fim da vida útil para a comunidade	Data do fim do suporte padrão para o RDS	Data de definição de preço do início do suporte estendido para o RDS no primeiro ano	Data de definição de preço do início do suporte estendido para o RDS no terceiro ano	Data de término do suporte estendido para o RDS
MySQL 8.0	19 de abril de 2018	23 de outubro de 2018	Abril de 2026	31 de julho de 2026	1.º de agosto de 2026	1.º de agosto de 2028	31 de julho de 2029
MySQL 5.7*	21 de outubro de 2015	22 de fevereiro de 2016	Outubro de 2023	23 de fevereiro de 2024	1.º de março de 2024	1.º de março de 2026	28 de fevereiro de 2027

* O MySQL 5.7 agora só está disponível no Suporte estendido do RDS. Para ter mais informações, consulte Suporte estendido do Amazon RDS com Amazon RDS.

Versões do Suporte estendido do Amazon RDS para RDS para MySQL

O conteúdo a seguir lista todas as versões do Suporte estendido do RDS para as versões do RDS para MySQL.

Versões

- Suporte estendido do RDS para RDS para MySQL versão 5.7.44-RDS.20240529
- Suporte estendido do RDS para RDS para MySQL versão 5.7.44-RDS.20240408

Suporte estendido do RDS para RDS para MySQL versão 5.7.44-RDS.20240529

O Suporte estendido do RDS para RDS para MySQL versão 5.7.44-RDS.20240529 está disponível.

Erros corrigidos:

- Correção de uma falha na declaração de `field.cc` implementando `fix_after_pullout`.
- Correção de uma falha de ponteiro nulo ao retornar metadados ao cliente para determinadas consultas SQL. Essas consultas continham parâmetros dinâmicos e subconsultas em cláusulas `SELECT`.
- Correção de resultados incorretos ao usar `GROUP BY` para varreduras de índice soltas ou varreduras de intervalos não contíguos de um índice.
- Correção da perda de informações do GTID no MySQL durante a persistência.
- Correção de uma condição de corrida que poderia fazer com que uma transação do InnoDB fosse interrompida indefinidamente.
- Correção de uma condição de corrida na limpeza das informações de certificação do Group Replication.
- Correção do problema de verificação reversa do índice com operações simultâneas de página.
- Correção de um problema de estado de pesquisa de texto completo (FTS) inconsistente em cenários simultâneos.
- Correção do problema de declaração com o buffer de alteração na exclusão de tabelas.
- Comportamento unificado para chamar a função `deinit` em todos os tipos de plug-in.

CVEs corrigidos:

- CVE-2024-20963
- CVE-2024-20993
- CVE-2024-20998
- CVE-2024-21009
- CVE-2024-21054
- CVE-2024-21055

- CVE-2024-21057
- CVE-2024-21062
- CVE-2024-21008
- CVE-2024-21013
- CVE-2024-21047
- CVE-2024-21087
- CVE-2024-21096

Suporte estendido do RDS para RDS para MySQL versão 5.7.44-RDS.20240408

O Suporte estendido do RDS para RDS para MySQL versão 5.7.44-RDS.20240408 está disponível.

Esta versão contém patches para os seguintes CVEs:

- CVE-2024-20963

Trabalhar com o ambiente de visualização de banco de dados

Em julho de 2023, a Oracle anunciou um novo modelo de lançamento para o MySQL. Esse modelo inclui dois tipos de lançamentos: lançamentos de inovação e lançamentos de LTS. O Amazon RDS disponibiliza lançamentos inovadores do MySQL no ambiente RDS Preview. Para saber mais sobre as versões do MySQL Innovation, consulte Apresentando as versões MySQL Innovation e Long-Term Support (LTS).

As instâncias de banco de dados do RDS para PostgreSQL no Database Preview Environment são semelhantes às outras instâncias do RDS para PostgreSQL em termos de funcionalidade. No entanto, você não pode usar o ambiente Database Preview para cargas de trabalho de produção.

Os ambientes prévios têm as seguintes limitações:

- Todas as instâncias de banco de dados são excluídas 60 dias após serem criadas, juntamente com backups e snapshots.
- Você só pode usar o SSD de uso geral e o armazenamento em SSD IOPS provisionadas.
- Você não pode obter ajuda do Suporte para instâncias de banco de dados. Em vez disso, você pode postar suas perguntas na comunidade de perguntas e respostas gerenciadas do , AWSre:Post.

- Não é possível copiar um snapshot de uma instância de banco de dados para um ambiente de produção.

As opções a seguir são compatíveis com a visualização.

- Você pode criar instâncias de banco de dados usando as classes de instância de banco de dados db.m6i, db.r6i, db.m6g, db.m5, db.t3, db.r6g e db.r5. Para obter mais informações sobre as classes de instância do RDS, consulte Classes de instâncias de banco de dados .
- Você pode usar implantações single-AZ e multi-AZ.
- Você pode usar funções padrão de despejo e carregamento do PostgreSQL para exportar ou importar bancos de dados para o Database Preview Environment.

Atributos não compatíveis com o ambiente de visualização de banco de dados

Os atributos a seguir não estão disponíveis no ambiente de visualização de banco de dados:

- Cópia de snapshots entre regiões
- Réplicas de leitura entre regiões
- RDS Proxy

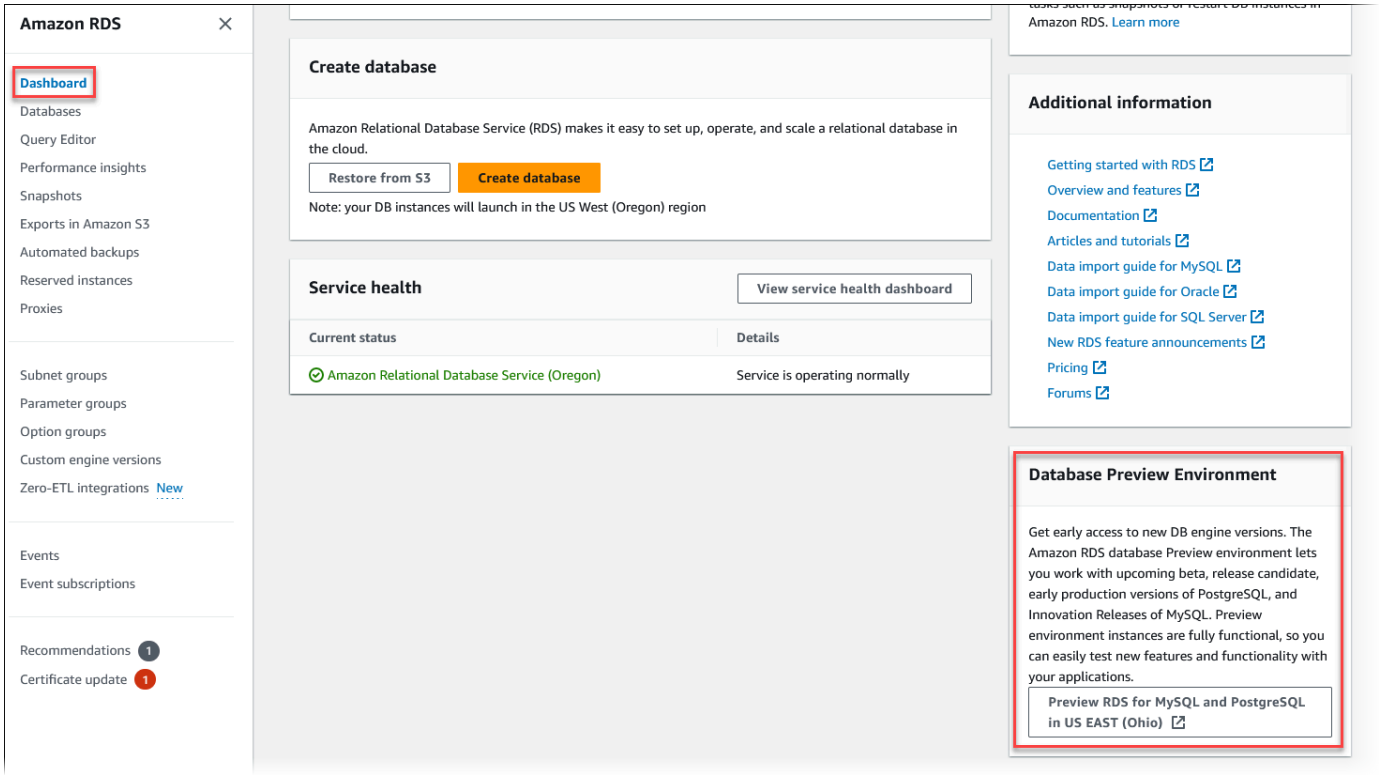
Criar uma nova instância de banco de dados no ambiente de visualização de banco de dados

Você pode criar uma instância de banco de dados no ambiente Database Preview usando o AWS Management ConsoleAWS CLI, a ou a API do RDS.

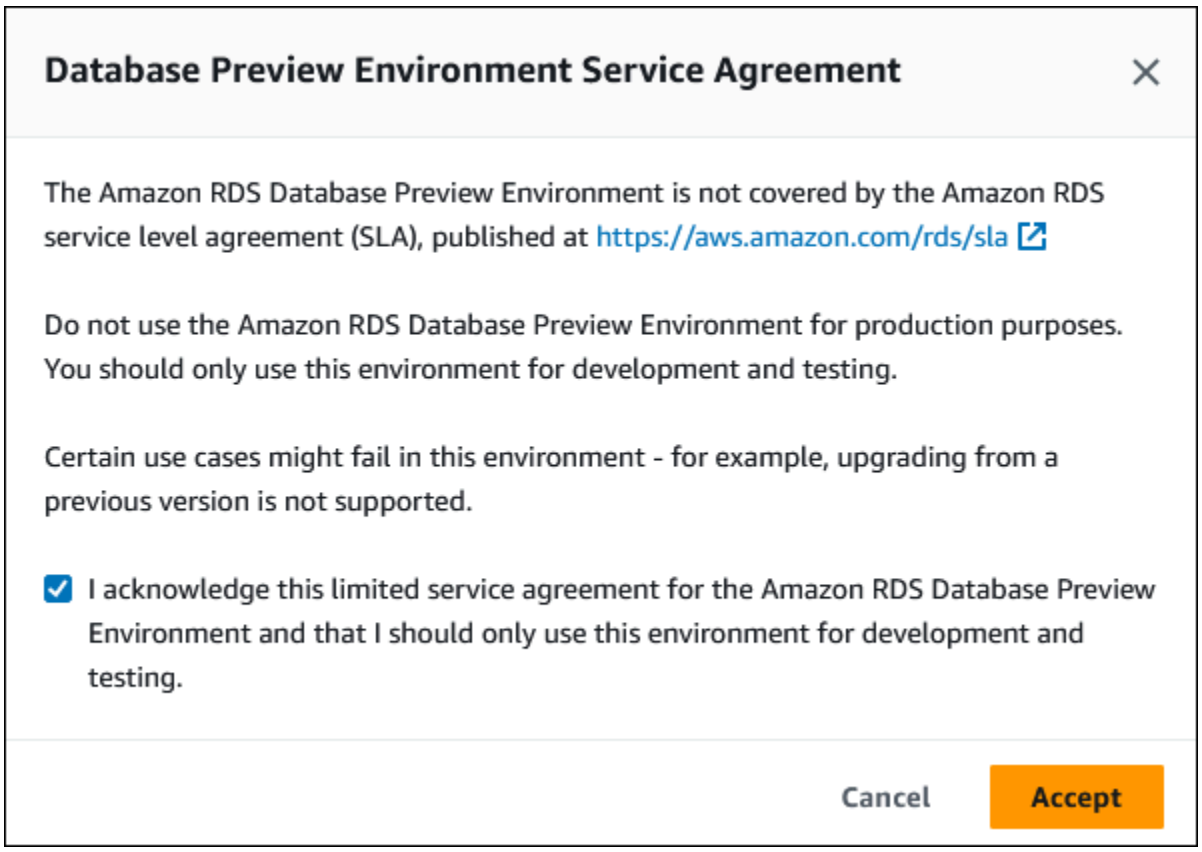
Console

Para criar uma instância de banco de dados no ambiente de visualização de banco de dados

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. Escolha Dashboard (Painel) no painel de navegação.
3. Na página Dashboard (Painel), localize a seção Database Preview Environment (Ambiente de visualização do banco de dados) na página Dashboard (Painel), conforme mostrado na imagem a seguir.



Você também pode navegar diretamente para o Ambiente de visualização do banco de dados. Antes de continuar, você deve reconhecer e aceitar as limitações.



4. Para criar a instância de banco de dados do RDS para PostgreSQL, siga o mesmo processo de criação de qualquer instância de banco de dados do Amazon RDS. Para obter mais informações, consulte o procedimento Console em Criar uma instância de banco de dados.

AWS CLI

Para criar uma instância no ambiente de visualização do banco de dados usando a API do RDS ou a AWS CLI, use o endpoint a seguir.

```
rds-preview.us-east-2.amazonaws.com
```

Para criar a instância de banco de dados do RDS para PostgreSQL, siga o mesmo processo de criação de qualquer instância de banco de dados do Amazon RDS. Para obter mais informações, consulte o procedimento AWS CLI em Criar uma instância de banco de dados.

API do RDS

Para criar uma instância no ambiente de visualização do banco de dados usando a API do RDS ou a , use o endpoint a seguir.

```
rds-preview.us-east-2.amazonaws.com
```

Para criar a instância de banco de dados do RDS para PostgreSQL, siga o mesmo processo de criação de qualquer instância de banco de dados do Amazon RDS. Para obter mais informações, consulte o procedimento API do RDS em Criar uma instância de banco de dados.

MySQL versão 8.4 no Ambiente de Pré-visualização do Banco de Dados

O MySQL versão 8.4 já está disponível no Ambiente de Pré-visualização do Banco de Dados do Amazon RDS. O MySQL versão 8.4 é a versão de LTS mais recente da comunidade e contém várias melhorias descritas em Changes in MySQL 8.4.0. É possível usar o ambiente de versão prévia do banco de dados para testar as workloads em relação a essa versão antes que ela esteja disponível em todas as Regiões da AWS para workloads de produção.

O MySQL versão 8.4 no Ambiente de Pré-visualização do Banco de Dados pode ser diferente da versão que o Amazon RDS lança em todas as Regiões da AWS para workloads de produção. A lista a seguir inclui os recursos que podem mudar. Essa lista não é exaustiva.

- A definição do grupo de parâmetros do RDS para MySQL 8.4. Por exemplo, o Amazon RDS pode adicionar, remover ou renomear parâmetros, bem como alterar os valores padrão dos parâmetros.
- O modelo de privilégio.
- A biblioteca TLS.

Para obter informações sobre o ambiente de visualização de banco de dados, consulte the section called “ Ambiente de visualização do banco de dados”. Para acessar o ambiente de visualização do console, selecione https://console.aws.amazon.com/rds-preview/.

MySQL versão 8.3 no Ambiente de Pré-visualização do Banco de Dados

O MySQL versão 8.3 já está disponível no Ambiente de Pré-visualização do Banco de Dados do Amazon RDS. O MySQL versão 8.3 contém várias melhorias descritas em Changes in MySQL 8.3.0.

Para obter informações sobre o ambiente de visualização de banco de dados, consulte the section called “ Ambiente de visualização do banco de dados”. Para acessar o ambiente de visualização do console, selecione https://console.aws.amazon.com/rds-preview/.

O MySQL versão 8.2 no Ambiente de Pré-visualização do Banco de Dados

O MySQL versão 8.2 já está disponível no Ambiente de Pré-visualização do Banco de Dados do Amazon RDS. O MySQL versão 8.2 contém várias melhorias descritas em Changes in MySQL 8.2.0.

Para obter informações sobre o ambiente de visualização de banco de dados, consulte the section called “ Ambiente de visualização do banco de dados”. Para acessar o ambiente de visualização do console, selecione https://console.aws.amazon.com/rds-preview/.

PostgreSQL versão 12.0 no ambiente de visualização de banco de dados

O PostgreSQL versão 16 já está disponível no ambiente de visualização de banco de dados. A versão 8.1 contém várias melhorias descritas em Alterações no MySQL 8.1.0.

Para obter informações sobre o ambiente de visualização de banco de dados, consulte the section called “ Ambiente de visualização do banco de dados”. Para acessar o ambiente de visualização do console, selecione https://console.aws.amazon.com/rds-preview/.

Versões obsoletas do Amazon RDS para MySQL

As versões 5.1, 5.5 e 5.6 do Amazon RDS para MySQL estão obsoletas.

Para obter informações sobre a política de defasagem do Amazon RDS para MySQL, consulte Perguntas frequentes sobre o Amazon RDS.

Conectar-se à instância de banco de dados do MySQL

Antes de poder se conectar a uma instância de banco de dados executando o mecanismo de banco de dados do MySQL, você deve criar uma instância de banco de dados. Para ter mais informações, consulte Criar uma instância de banco de dados do Amazon RDS. Depois que o Amazon RDS provisionar a sua instância de banco de dados, será possível usar qualquer aplicativo ou utilitário cliente MySQL padrão para se conectar à instância. Na string de conexão, você especifica o endereço DNS do endpoint da instância de banco de dados como o parâmetro de host e especifica o número da porta a partir desse endpoint como o parâmetro de porta.

Para autenticar-se na instância de banco de dados do RDS, você pode usar um dos métodos para autenticação do MySQL e a autenticação de banco de dados do AWS Identity and Access Management (IAM):

- Para saber como fazer a autenticação no MySQL usando um dos métodos de autenticação do MySQL, consulte Método de autenticação na documentação do MySQL.
- Para saber como fazer a autenticação no MySQL usando a autenticação de banco de dados do IAM, consulte Autenticação do banco de dados do IAM para MariaDB, MySQL e PostgreSQL.

Você pode se conectar a uma instância de banco de dados MySQL com ferramentas como o cliente MySQL de linha de comando do . Para ter mais informações sobre como usar o cliente MySQL de linha de comando, acesse mysql - the MySQL command-line client na documentação do MySQL. O MySQL Workbench é um aplicativo baseado em GUI que você pode usar para se conectar. Para ter mais informações, consulte a página Download MySQL Workbench. Para obter informações sobre como instalar o MySQL (incluindo o cliente MySQL de linha de comando), consulte Installing and upgrading MySQL (Instalar e atualizar o MySQL).

Para se conectar a uma instância de banco de dados de fora de sua Amazon VPC, a instância de banco de dados deve ser acessível ao público geral, o acesso deve ser concedido usando as regras de entrada do grupo de segurança da instância de banco de dados e outros requisitos devem ser atendidos. Para ter mais informações, consulte Não é possível conectar-se à instância de banco de dados do Amazon RDS.

Você pode usar a criptografia Secure Sockets Layer (SSL) ou Transport Layer Security (TLS) em conexões com uma instância de banco de dados do MySQL. Para ter mais informações, consulte Suporte do SSL/TLS para instâncias de banco de dados do MySQL no Amazon RDS. Se você estiver usando a autenticação de banco de dados do AWS Identity and Access Management (IAM),

use uma conexão SSL/TLS. Para ter mais informações, consulte Autenticação do banco de dados do IAM para MariaDB, MySQL e PostgreSQL.

Também é possível se conectar a uma instância de banco de dados a partir de um servidor web. Para ter mais informações, consulte Tutorial: crie um servidor Web e uma instância de banco de dados do Amazon RDS.

> **Note**
>
> Para obter informações sobre a conexão a uma instância de banco de dados MariaDB, consulte Conectar-se à instância de banco de dados do MariaDB.

Para encontrar uma instância de banco de dados do RDS para MySQL e conectar-se a ela, consulte os tópicos a seguir.

Tópicos

- Localizar as informações de conexão para uma instância de banco de dados do RDS para MySQL
- Instalar o cliente de linha de comando do MySQL
- Conectar-se pelo cliente de linha de comandos do MySQL (não criptografado)
- Conexão pelo MySQL Workbench
- Conectar-se ao RDS para MySQL com o driver JDBC da AWS, o driver Python da AWS e o driver ODBC da AWS para MySQL
- Solução de problemas de conexões com a instância de banco de dados MySQL

Localizar as informações de conexão para uma instância de banco de dados do RDS para MySQL

As informações de conexão de uma instância de banco de dados incluem seu endpoint, porta e um usuário de banco de dados válido, como o usuário mestre. Por exemplo, suponha que um valor de endpoint seja `mydb.123456789012.us-east-1.rds.amazonaws.com`. Nesse caso, o valor da porta é `3306`, e o usuário do banco de dados é `admin`. Com essas informações, você especifica os seguintes valores em uma string de conexão:

- Para o host, nome de host ou nome DNS, especifique `mydb.123456789012.us-east-1.rds.amazonaws.com`.

- Para a porta, especifique `3306`.
- Para usuário, especifique `admin`.

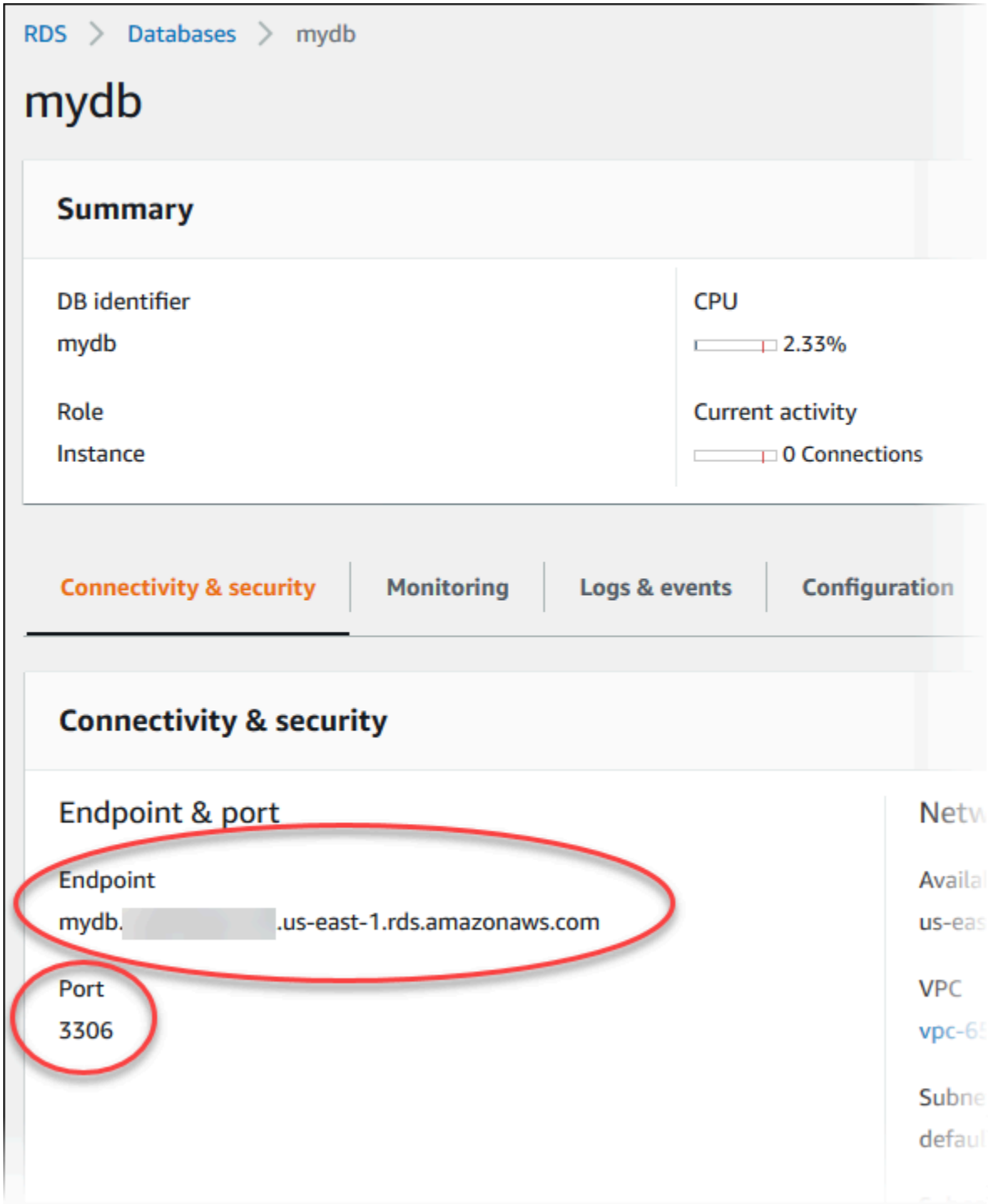
Para se conectar a uma instância de banco de dados, use qualquer cliente para um mecanismo de banco de dados do MySQL. Por exemplo, você pode usar o cliente MySQL de linha de comando ou o MySQL Workbench.

Para localizar as informações de conexão de uma instância de banco de dados, você pode usar o AWS Management Console, comando da AWS CLI describe-db-instances ou a operação da API do Amazon RDS DescribeDBInstances para listar seus detalhes.

Console

Para encontrar as informações de conexão de uma instância de banco de dados no AWS Management Console

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Bancos de dados para exibir uma lista de suas instâncias de banco de dados.
3. Escolha o nome da instância de banco de dados MySQL para exibir os detalhes.
4. Na guia Connectivity & security (Conectividade e segurança), copie o endpoint. Além disso, anote o número da porta. Você precisará do endpoint e do número da porta para conectar-se à instância de banco de dados.



5. Se você precisar encontrar o nome de usuário mestre, escolha a guia Configuração e visualize o valor do nome de usuário mestre .

AWS CLI

Para localizar as informações de conexão de uma instância de banco de dados MySQL usando a AWS CLI, execute o comando describe-db-instances. Na chamada, consulte o ID da instância de banco de dados, o endpoint, a porta e o nome de usuário mestre.

Para Linux, macOS ou Unix:

```
aws rds describe-db-instances \
  --filters "Name=engine,Values=mysql" \
  --query "*[].[DBInstanceIdentifier,Endpoint.Address,Endpoint.Port,MasterUsername]"
```

Para Windows:

```
aws rds describe-db-instances ^
  --filters "Name=engine,Values=mysql" ^
  --query "*[].[DBInstanceIdentifier,Endpoint.Address,Endpoint.Port,MasterUsername]"
```

Sua saída deve ser similar à seguinte.

```
[
    [
        "mydb1",
        "mydb1.123456789012.us-east-1.rds.amazonaws.com",
        3306,
        "admin"
    ],
    [
        "mydb2",
        "mydb2.123456789012.us-east-1.rds.amazonaws.com",
        3306,
        "admin"
    ]
]
```

API do RDS

Para localizar as informações de conexão de uma instância de banco de dados usando a Amazon RDS API, chame a operação DescribeDBInstances . Na saída, encontre os valores para o endereço do endpoint, a porta do endpoint e o nome do usuário mestre.

Instalar o cliente de linha de comando do MySQL

A maioria das distribuições do Linux inclui o cliente MariaDB em vez do cliente Oracle MySQL. Para instalar o cliente da linha de comando do MySQL no Amazon Linux 2023, execute o seguinte comando:

```
sudo dnf install mariadb105
```

Para instalar o cliente da linha de comando do MySQL no Amazon Linux 2, execute o seguinte comando:

```
sudo yum install mariadb
```

Para instalar o cliente MySQL de linha de comando na maioria das distribuições do Linux baseadas em DEB, execute o seguinte comando:

```
apt-get install mariadb-client
```

Para verificar a versão do cliente MySQL de linha de comando, execute o seguinte comando:

```
mysql --version
```

Para ler a documentação da versão atual do cliente MySQL, execute o seguinte comando:

```
man mysql
```

Conectar-se pelo cliente de linha de comandos do MySQL (não criptografado)

> **Important**
>
> Use uma conexão MySQL não criptografada apenas quando o cliente e o servidor estiverem na mesma VPC e a rede for confiável. Para obter informações sobre como usar conexões criptografadas, consulte Conectar-se à instância de banco de dados do MySQL no Amazon RDS com SSL/TLS por meio do cliente de linha de comandos do MySQL (criptografado).

Para se conectar a uma instância de banco de dados usando o cliente de linha de comando MySQL, insira o comando a seguir no prompt de comando. Para o parâmetro -h, substitua o nome do DNS (endpoint) da instância de banco de dados. Para o parâmetro -P, substitua a porta para sua instância de banco de dados. Para o parâmetro -u, substitua o nome de usuário de um usuário de banco de dados válido, como o usuário mestre. Digite a senha do usuário mestre quando lhe for solicitado.

```
mysql -h mysql-instance1.123456789012.us-east-1.rds.amazonaws.com -P 3306 -
u mymasteruser -p
```

Depois de inserir a senha do usuário, você deverá ver uma saída semelhante à seguinte.

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9738
Server version: 8.0.28 Source distribution

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql>
```

Conexão pelo MySQL Workbench

Para se conectar pelo MySQL Workbench

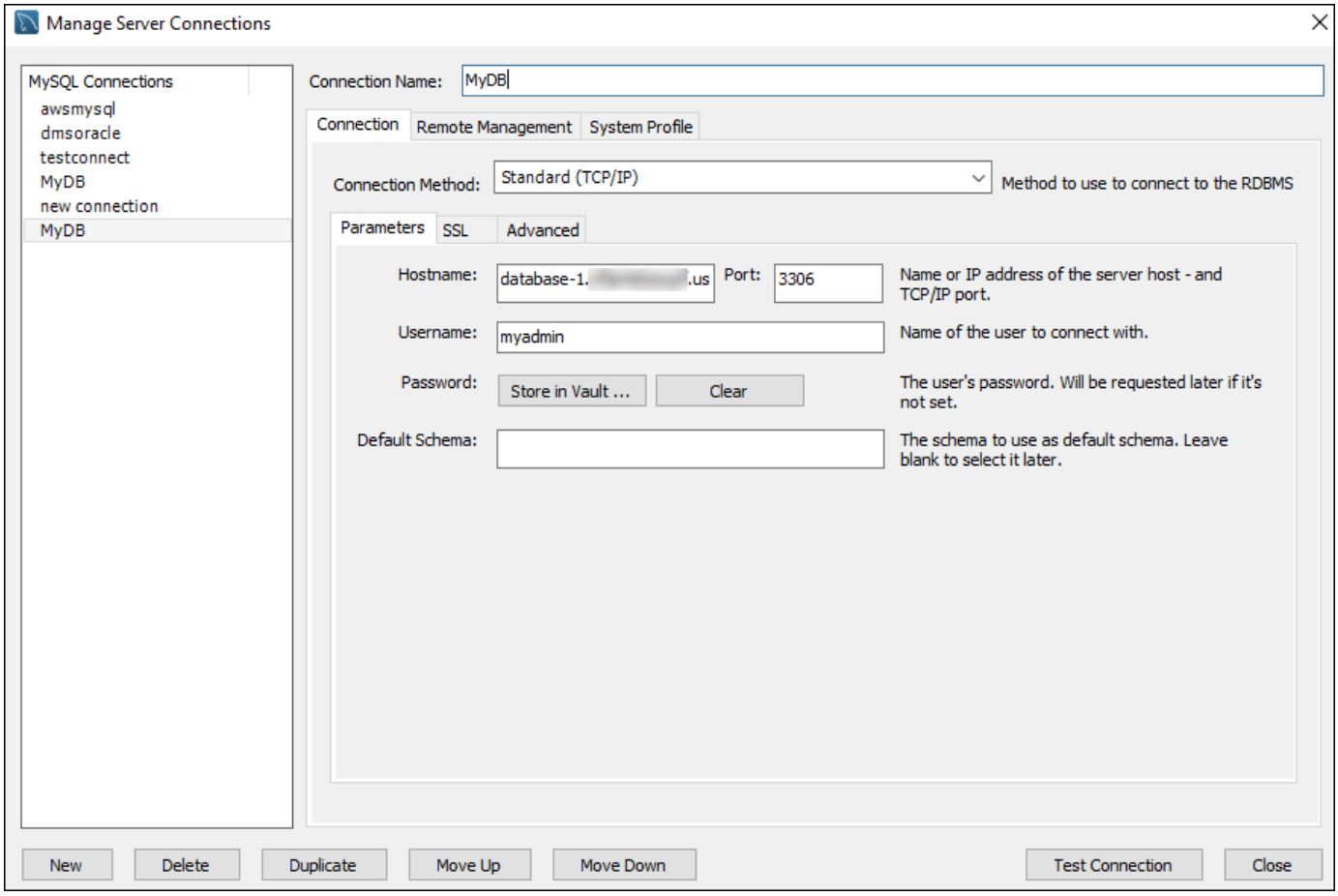
1. Faça download do MySQL Workbench em Download do MySQL Workbench e instale-o em seguida.
2. Abra o MySQL Workbench.



3. Em Banco de dados, escolha Gerenciar conexões.
4. Na janela Gerenciar conexões de servidor, escolha Novo.
5. Na janela Conectar-se ao banco de dados, insira as seguintes informações:

   - Conexão armazenada – insira um nome para a conexão, como **MyDB**.
   - Nome do host – insira o endpoint da instância de banco de dados.
   - Porta – insira a porta usada pela instância de banco de dados.
   - Nome de usuário – insira o nome de usuário de um usuário de banco de dados válido, como o usuário mestre.
   - Senha – opcionalmente, escolha Armazenar no cofre e depois insira e salve a senha do usuário.

   A janela é semelhante à seguinte:



É possível usar os recursos do MySQL Workbench para personalizar conexões. Por exemplo, é possível usar a guia SSL para configurar conexões SSL/TLS. Para obter informações sobre

como usar o MySQL Workbench, consulte a Documentação do MySQL Workbench. Para criptografar conexões de clientes com instâncias de banco de dados do MySQL com SSL/TLS, consulte Criptografar conexões de clientes com SSL/TLS para instâncias de banco de dados do MySQL no Amazon RDS.

6. Opcionalmente, escolha Testar conexão para confirmar se a conexão com a instância de banco de dados foi bem-sucedida.
7. Escolha Fechar.
8. Em Banco de dados, escolha Conectar-se ao banco de dados.
9. Em Conexão armazenada, escolha sua conexão.
10. Escolha OK.

Conectar-se ao RDS para MySQL com o driver JDBC da AWS, o driver Python da AWS e o driver ODBC da AWS para MySQL

Conecte-se a instâncias de banco de dados do RDS para MySQL com o driver JDBC da AWS, o driver Python da AWS e o driver ODBC da AWS para MySQL. Para obter mais informações, consulte os tópicos a seguir.

Tópicos

- Conectar-se ao RDS para MySQL com o driver JDBC da Amazon Web Services (AWS)
- Conectar-se ao RDS para MySQL com o driver Python da Amazon Web Services (AWS)
- Conectar-se ao RDS para MySQL com o Driver ODBC da Amazon Web Services (AWS) para MySQL

Conectar-se ao RDS para MySQL com o driver JDBC da Amazon Web Services (AWS)

O driver JDBC da Amazon Web Services (AWS) foi projetado como um wrapper JDBC avançado. Esse wrapper é complementar e amplia a funcionalidade do driver JDBC existente. O driver é compatível de forma intercambiável com os drivers MySQL Connector/J e MariaDB Connector/J da comunidade.

Para instalar o driver JDBC da AWS, anexe o arquivo .jar do driver JDBC da AWS (localizado na aplicação `CLASSPATH`) e mantenha referências ao respectivo driver da comunidade. Atualize o respectivo prefixo do URL de conexão da seguinte forma:

- `jdbc:mysql://` para `jdbc:aws-wrapper:mysql://`
- `jdbc:mariadb://` para `jdbc:aws-wrapper:mariadb://`

Consulte mais informações sobre o driver JDBC da AWS e siga as instruções para usá-lo em Amazon Web Services (AWS) JDBC Driver GitHub repository.

Conectar-se ao RDS para MySQL com o driver Python da Amazon Web Services (AWS)

O driver Python da Amazon Web Services (AWS) foi projetado como um wrapper Python avançado. Esse wrapper é complementar e amplia a funcionalidade do driver Psycopg de código aberto. O driver Python da AWS é compatível com as versões 3.8 e posterior do Python. É possível instalar o pacote `aws-advanced-python-wrapper` usando o comando `pip`, bem como os pacotes de código aberto `psycopg`.

Para ter mais informações sobre o driver Python da AWS e instruções completas para usá-lo, consulte Amazon Web Services (AWS) Python Driver GitHub repository.

Conectar-se ao RDS para MySQL com o Driver ODBC da Amazon Web Services (AWS) para MySQL

O Driver ODBC da AWS para MySQL é um driver de cliente projetado para a alta disponibilidade do RDS para MySQL. O driver pode existir junto com o driver MySQL Connector/ODBC e é compatível com os mesmos fluxos de trabalho.

Para ter mais informações sobre o driver ODBC da AWS para MySQL e instruções completas de uso e instalação, consulte o diretório do GitHub Amazon Web Services (AWS) ODBC Driver for MySQL.

Solução de problemas de conexões com a instância de banco de dados MySQL

Duas causas comuns de falhas de conexão a uma nova instância de banco de dados são:

- A instância de banco de dados foi criada usando um grupo de segurança que não autoriza conexões do dispositivo ou da instância do Amazon EC2 em que o aplicativo ou utilitário do MySQL está sendo executado. A instância de banco de dados deve ter um grupo de segurança de VPC que autorize as conexões. Para ter mais informações, consulte Amazon VPC e Amazon RDS.

É possível adicionar ou editar uma regra de entrada no grupo de segurança. Em Source (Origem), selecione My IP (Meu IP). Isso concede acesso à instância de banco de dados do endereço IP detectado no navegador.

- A instância de banco de dados foi criada usando a porta padrão 3306 e sua empresa emprega regras de firewall bloqueando conexões a essa porta a partir de dispositivos na rede de sua empresa. Para corrigir essa falha, recrie a instância com uma porta diferente.

Para ter mais informações sobre problemas de conexão, consulte Não é possível conectar-se à instância de banco de dados do Amazon RDS.

Proteger as conexões de instância de banco de dados do MySQL

Você pode implementar medidas robustas de segurança para proteger as instâncias de banco de dados do MySQL contra acesso não autorizado e possíveis ameaças. Grupos de segurança, criptografia SSL/TLS e autenticação de banco de dados do IAM funcionam juntos para criar várias camadas de segurança de conexão para instâncias de banco de dados do MySQL. Esses controles de segurança ajudam você a atender aos requisitos de conformidade, evitar violações de dados e manter canais de comunicação seguros entre aplicações e bancos de dados. Você pode proteger suas instâncias de banco de dados do MySQL criptografando dados em trânsito, restringindo o acesso a intervalos de IP específicos e gerenciando a autenticação de usuários por meio de perfis do IAM em vez de senhas de banco de dados.

A segurança de instâncias de bancos de dados MySQL é gerenciada em três níveis:

- O AWS Identity and Access Management controla quem pode executar ações de gerenciamento do Amazon RDS em instâncias de banco de dados. Ao se conectar à AWS usando credenciais do IAM, sua conta do IAM deve ter políticas do IAM que concedam as permissões necessárias para executar operações de gerenciamento do Amazon RDS. Para ter mais informações, consulte Gerenciamento de identidade e acesso no Amazon RDS.
- Quando você cria uma instância de banco de dados, é necessário usar um grupo de segurança da VPC para controlar quais dispositivos e instâncias do Amazon EC2 podem abrir conexões com o endpoint e a porta da instância de banco de dados. É possível estabelecer essas conexões usando o Transport Layer Security (TLS) e o Secure Sockets Layer (SSL). Além disso, as regras de firewall em sua empresa podem controlar se dispositivos sendo executados nela podem abrir conexões na instância de banco de dados.
- Para autenticar o login e as permissões para uma instância de banco de dados do MySQL, você pode adotar uma das seguintes abordagens ou uma combinação delas:
  - Você pode seguir a mesma abordagem de uma instância autônoma do MySQL. Comandos, como `CREATE USER`, `RENAME USER`, `GRANT`, `REVOKE` e `SET PASSWORD` funcionam exatamente como em bancos de dados locais, assim como modificando diretamente tabelas de esquema de banco de dados. No entanto, modificar diretamente as tabelas do esquema do banco de dados não é prática recomendada e, desde o RDS para MySQL versão 8.0.36, isso não é aceito. Para obter informações, consulte Access control and account management na documentação do MySQL.
  - Você também pode usar a autenticação de banco de dados do IAM. Com a autenticação de banco de dados do IAM, é possível autenticar sua instância de banco de dados usando um

usuário do IAM ou um perfil do IAM e um token de autenticação. Um token de autenticação é um valor exclusivo, gerado usando o processo de assinatura Signature Version 4. Ao usar a autenticação de banco de dados do IAM, você pode usar as mesmas credenciais para controlar o acesso aos seus recursos e bancos de dados da AWS. Para ter mais informações, consulte Autenticação do banco de dados do IAM para MariaDB, MySQL e PostgreSQL.

- Outra opção é a autenticação Kerberos para RDS para MySQL. A instância de banco de dados funciona com o AWS Directory Service for Microsoft Active Directory (AWS Managed Microsoft AD) para habilitar a autenticação Kerberos. Quando os usuários são autenticados por uma instância de banco de dados MySQL associada ao domínio de confiança, as solicitações de autenticação são encaminhadas. As solicitações encaminhadas vão para o diretório de domínio que você cria com o AWS Directory Service. Para ter mais informações, consulte Uso da autenticação do Kerberos para o Amazon RDS para MySQL.

Quando você cria uma instância de banco de dados do Amazon RDS, o usuário mestre apresenta os seguintes privilégios padrão:

Versão do mecanism	Privilégio do sistema	Função do banco de dados
RDS para MySQL versão 8.4.3 e posterior	`GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, RELOAD, PROCESS, REFERENCES ,INDEX, ALTER, SHOW DATABASES , CREATE TEMPORARY TABLES, LOCK TABLES, EXECUTE, REPLICATION SLAVE , REPLICATION CLIENT , CREATE VIEW, SHOW VIEW, CREATE ROUTINE, ALTER ROUTINE, CREATE USER, EVENT, TRIGGER, CREATE ROLE, DROP ROLE, APPLICATION_PASSWORD_ADMIN , FLUSH_OPTIMIZER_COSTS , FLUSH_PRIVILEGES , FLUSH_STATUS , FLUSH_TABLES , FLUSH_USER_RESOURCES , ROLE_ADMIN , SENSITIVE_VARIABLES_OBSERVER , SESSION_VARIABLES_ADMIN , SET_ANY_DEFINER , SHOW_ROUTINE , XA_RECOVER_ADMIN`	`rds_superuser_role` Para obter mais informações sobre o `rds_superuser_role` , consulte Modelo de privilégios baseados em perfis para o RDS para MySQL.

Versão do mecanism	Privilégio do sistema	Função do banco de dados
RDS para MySQL versão 8.0.36 e posterior	`SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, RELOAD, PROCESS, REFERENCES , INDEX, ALTER, SHOW DATABASES , CREATE TEMPORARY TABLES, LOCK TABLES, EXECUTE, REPLICATION SLAVE, REPLICATION CLIENT , CREATE VIEW, SHOW VIEW, CREATE ROUTINE, ALTER ROUTINE, CREATE USER, EVENT, TRIGGER, CREATE ROLE, DROP ROLE, APPLICATION_PASSWORD_ADMIN , ROLE_ADMIN , SET_USER_ID , XA_RECOVER_ADMIN`	`rds_superuser_role` Para obter mais informações sobre o `rds_superuser_role` , consulte Modelo de privilégios baseados em perfis para o RDS para MySQL.
Versões do RDS para MySQL anteriores à 8.0.36	`SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, RELOAD, PROCESS, REFERENCES , INDEX, ALTER, SHOW DATABASES , CREATE TEMPORARY TABLES, LOCK TABLES, EXECUTE, REPLICATION CLIENT , CREATE VIEW, SHOW VIEW, CREATE ROUTINE, ALTER ROUTINE, CREATE USER, EVENT, TRIGGER, REPLICATION SLAVE`	Nenhum

> **Note**
>
> Embora seja possível excluir o usuário mestre na instância de banco de dados, isso não é recomendado. Para recriar o usuário mestre, use a operação da API do RDS ModifyDBInstance ou execute o comando da AWS CLI modify-db-instance e especifique uma nova senha de usuário mestre com o parâmetro apropriado. Se o usuário mestre não existir na instância, ele será criado com a senha especificada.

Para fornecer serviços de gerenciamento para cada instância de banco de dados, o usuário `rdsadmin` é criado quando a instância de banco de dados é criada. Tentar descartar, renomear ou alterar a senha, ou alterar os privilégios da conta `rdsadmin` resultará em um erro.

Para permitir o gerenciamento da instância de banco de dados, os comandos `kill` e `kill_query` padrão foram restritos. Os comandos do Amazon RDS `rds_kill` e `rds_kill_query` são fornecidos para permitir encerrar sessões ou consultas de usuários em instâncias de banco de dados.

Validação de senhas do RDS para MySQL

O MySQL fornece o plugin `validate_password` para maior segurança. O plugin impõe políticas de senha usando parâmetros no grupo de parâmetro de banco de dados para sua instância de banco de dados do MySQL. O plugin tem suporte para instâncias de banco de dados que executam o MySQL versão 5.7, 8.0 e 8.4. Para obter mais informações sobre o plugin `validate_password`, consulte The Password Validation Plugin na documentação do MySQL.

Para habilitar o plugin **`validate_password`** para uma instância de banco de dados do MySQL

1. Conecte-se à instância de banco de dados do MySQL e execute o comando a seguir.

   ```
   INSTALL PLUGIN validate_password SONAME 'validate_password.so';
   ```

2. Configure os parâmetros para o plugin no grupo de parâmetros de banco de dados usado pela instância de banco de dados.

   Para obter mais informações sobre os parâmetros, consulte Opções e variáveis do plug-in de validação de senha na documentação do MySQL.

   Para obter informações sobre como modificar parâmetros de instâncias de banco de dados, consulte Modificar parâmetros em um grupo de parâmetros de banco de dados no Amazon RDS.
3. Reinicie a instância de banco de dados.

Depois de habilitar o plugin `validate_password`, redefina as senhas existentes para que elas estejam em conformidade com suas novas políticas de validação.

O Amazon RDS não valida senhas. A instância do banco de dados do MySQL executa a validação de senha. Se você definir uma senha de usuário com o AWS Management Console, o comando da `modify-db-instance` AWS CLI ou a operação da API do RDS `ModifyDBInstance`, a alteração poderá ser bem-sucedida mesmo que a nova senha não atenda às suas políticas de senha. No

entanto, uma nova senha será definida na instância de banco de dados do MySQL somente se atender às políticas de senha. Nesse caso, o Amazon RDS registra o seguinte evento.

```
   "RDS-EVENT-0067" - An attempt to reset the master password for the DB instance has
failed.
```

Para obter mais informações sobre eventos do Amazon RDS, consulte Trabalhar com a notificação de eventos do Amazon RDS.

Criptografar conexões de clientes com SSL/TLS para instâncias de banco de dados do MySQL no Amazon RDS

O Secure Sockets Layer (SSL) é um protocolo padrão do setor para proteger as conexões de rede entre o cliente e o servidor. Após o SSL versão 3.0, o nome mudou para Transport Layer Security (TLS). O Amazon RDS é compatível com criptografia SSL/TLS para instâncias de banco de dados do MySQL. Usando o SSL/TLS, você pode criptografar uma conexão entre seu cliente de aplicação e sua instância de banco de dados do MySQL. A compatibilidade com SSL/TLS está disponível em todas as Regiões da AWS para MySQL.

Com o Amazon RDS, é possível proteger dados em trânsito criptografando conexões de clientes com instâncias de banco de dados do MySQL por meio de SSL/TLS, exigindo SSL/TLS para todas as conexões com uma instância de banco de dados do MySQL e conectando-se pelo cliente da linha de comandos do MySQL com SSL/TLS (criptografado). As seções a seguir fornecem orientações sobre como configurar e utilizar a criptografia SSL/TLS para instâncias de banco de dados do MySQL no Amazon RDS.

Tópicos

- Suporte do SSL/TLS para instâncias de banco de dados do MySQL no Amazon RDS
- Exigir SSL/TLS para contas de usuário específicas em uma instância de banco de dados do MySQL no Amazon RDS
- Exigir SSL/TLS para todas as conexões com uma instância de banco de dados do MySQL no Amazon RDS
- Conectar-se à instância de banco de dados do MySQL no Amazon RDS com SSL/TLS por meio do cliente de linha de comandos do MySQL (criptografado)

Suporte do SSL/TLS para instâncias de banco de dados do MySQL no Amazon RDS

O Amazon RDS cria um certificado SSL/TLS e instala o certificado na instância de banco de dados quando o Amazon RDS provisionar a instância. Esses certificados são assinados por uma autoridade de certificado. O certificado SSL/TLS inclui o endpoint da instância de banco de dados como o nome comum (CN) do certificado SSL/TLS para se proteger contra ataques de falsificação.

Um certificado SSL/TLS criado pelo Amazon RDS é a entidade raiz confiável e deve funcionar na maioria dos casos, mas poderá falhar se a aplicação não aceitar cadeias de certificados. Se a aplicação não aceitar cadeias de certificados, tente usar um certificado intermediário para se conectar à Região da AWS. Por exemplo, é necessário usar um certificado intermediário para se conectar às regiões AWS GovCloud (US) com SSL/TLS.

Para obter informações sobre como baixar certificados, consulte Usar SSL/TLS para criptografar uma conexão com uma instância ou um cluster de banco de dados. Para obter mais informações sobre como usar o SSL/TLS com o MySQL, consulte Atualizar aplicações para conexão com instâncias de banco de dados MySQL usando novos certificados SSL/TLS.

Para o MySQL versão 8.0 e anteriores, o Amazon RDS para MySQL usa o OpenSSL para conexões seguras. Para o MySQL versão 8.4 e posterior, o Amazon RDS para MySQL usa a AWS-LC. O suporte ao TLS depende da versão do MySQL. A tabela a seguir mostra o suporte a TLS para as versões do MySQL.

Versão do MySQL	TLS 1.0	TLS 1.1	TLS 1.2	TLS 1.3
MySQL 8.4	Sem compatibilidade	Não suportado	Compatível	Compatível
MySQL 8.0	Sem compatibilidade	Não suportado	Compatível	Compatível
MySQL 5.7	Compatível	Compatível	Compatível	Sem compatibilidade

Exigir SSL/TLS para contas de usuário específicas em uma instância de banco de dados do MySQL no Amazon RDS

É possível exigir criptografia SSL/TLS para conexões de contas de usuário especificadas com as instâncias de banco de dados do MySQL no Amazon RDS. Proteger informações confidenciais contra acesso não autorizado ou interceptação é crucial para aplicar políticas de segurança nas quais a confidencialidade dos dados é uma questão importante.

Se deseja exigir conexões SSL/TLS para contas de usuários específicos, siga uma das instruções abaixo (dependendo da versão do MySQL) para exigir conexões SSL/TLS na conta de usuário `encrypted_user`.

Para fazer isso, use a instrução a seguir.

```
ALTER USER 'encrypted_user'@'%' REQUIRE SSL;
```

Para obter mais informações sobre conexões SSL/TLS com MySQL, consulte Using encrypted connections (Usar conexões criptografadas) na documentação do MySQL.

Exigir SSL/TLS para todas as conexões com uma instância de banco de dados do MySQL no Amazon RDS

Use o parâmetro `require_secure_transport` para exigir que todas as conexões de usuários à sua instância de banco de dados do MySQL usem SSL/TLS. Por padrão, o parâmetro `require_secure_transport` é definido como `OFF`. Você pode definir o parâmetro `require_secure_transport` como `ON` para exigir SSL/TLS para conexões com a sua instância de banco de dados.

Você pode definir o valor de parâmetro `require_secure_transport` atualizando o grupo de parâmetros do banco de dados de sua instância de banco de dados. Você não precisa reinicializar sua instância de banco de dados para que a alteração entre em vigor.

Quando o parâmetro `require_secure_transport` é definido como `ON` para uma instância de banco de dados, um cliente de banco de dados poderá se conectar a ela se puder estabelecer uma conexão criptografada. Caso contrário, uma mensagem de erro semelhante à seguinte é retornada para o cliente:

```
MySQL Error 3159 (HY000): Connections using insecure transport are prohibited while --
require_secure_transport=ON.
```

Para obter informações sobre como configurar parâmetros, consulte Modificar parâmetros em um grupo de parâmetros de banco de dados no Amazon RDS.

Para obter mais informações sobre o parâmetro `require_secure_transport`, consulte a documentação do MySQL.

Conectar-se à instância de banco de dados do MySQL no Amazon RDS com SSL/TLS por meio do cliente de linha de comandos do MySQL (criptografado)

Os parâmetros do programa cliente `mysql` são ligeiramente diferentes, dependendo da versão do MySQL ou do MariaDB que você está usando.

Para descobrir qual versão você tem, execute o comando `mysql` com a opção `--version`. No exemplo a seguir, a saída mostra que o programa cliente é do MariaDB.

```
$ mysql --version
mysql  Ver 15.1 Distrib 10.5.15-MariaDB, for osx10.15 (x86_64) using readline 5.1
```

A maioria das distribuições Linux, como o Amazon Linux, CentOS, SUSE e Debian, substituiu o MySQL pelo MariaDB, e a versão do `mysql` neles é de MariaDB.

Para conectar-se à sua instância de banco de dados usando o SSL/TLS, siga estas etapas:

Como conectar-se a uma instância de banco de dados com o SSL/TLS usando o cliente de linha de comando do MySQL

1. Você pode baixar um certificado raiz que funciona para todas as Regiões da AWS.

   Para obter informações sobre como baixar certificados, consulte Usar SSL/TLS para criptografar uma conexão com uma instância ou um cluster de banco de dados.
2. Use um cliente da linha de comando do MySQL para conectar-se a uma instância de banco de dados com criptografia SSL/TLS. Para o parâmetro `-h`, substitua o nome de DNS (endpoint) da instância de banco de dados. Para o parâmetro `--ssl-ca`, substitua o nome do arquivo de certificado SSL/TLS. Para o parâmetro `-P`, substitua a porta da sua instância de banco de dados. Para o parâmetro `-u`, substitua o nome de usuário de um usuário de banco de dados válido, como o usuário principal. Digite a senha do usuário mestre quando lhe for solicitado.

   O exemplo a seguir mostra como iniciar o cliente usando o parâmetro `--ssl-ca` com o MySQL 5.7 e versões posteriores.

```
mysql -h mysql-instance1.123456789012.us-east-1.rds.amazonaws.com --ssl-ca=global-bundle.pem --ssl-mode=REQUIRED -P 3306 -u myadmin -p
```

Para exigir que a conexão SSL/TLS verifique o endpoint da instância de banco de dados em relação ao endpoint no certificado SSL/TLS, insira o seguinte comando:

```
mysql -h mysql-instance1.123456789012.us-east-1.rds.amazonaws.com --ssl-ca=global-bundle.pem --ssl-mode=VERIFY_IDENTITY -P 3306 -u myadmin -p
```

O exemplo a seguir mostra como iniciar o cliente usando o parâmetro `--ssl-ca` usando o cliente MariaDB:

```
mysql -h mysql-instance1.123456789012.us-east-1.rds.amazonaws.com --ssl-ca=global-bundle.pem --ssl -P 3306 -u myadmin -p
```

3. Digite a senha do usuário mestre quando lhe for solicitado.

Você verá uma saída semelhante à seguinte.

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9738
Server version: 8.0.28 Source distribution

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql>
```

Atualizar aplicações para conexão com instâncias de banco de dados MySQL usando novos certificados SSL/TLS

Em 13 de janeiro de 2023, o Amazon RDS publicou novos certificados de autoridade de certificação (CA) para se conectar às suas instâncias de banco de dados do RDS via Secure Socket Layer ou Transport Layer Security (SSL/TLS). A seguir, você pode encontrar informações sobre como atualizar seus aplicativos para usar os novos certificados.

Este tópico pode ajudá-lo a determinar se algum aplicativo cliente usa SSL/TLS para conectar-se às suas instâncias de banco de dados. Em caso positivo, examine mais a fundo se esse aplicativo exige verificação de certificado para se conectar.

> **Note**
>
> Alguns aplicativos são configurados para se conectar a instâncias de banco de dados MySQL apenas quando podem verificar com sucesso o certificado no servidor. Para esses aplicativos, você deve atualizar os repositórios confiáveis de aplicativos cliente para incluir os novos certificados de CA.
>
> É possível especificar os seguintes modos SSL: `disabled`, `preferred` e `required`. Ao usar o modo `preferred` do SSL, e o certificado de CA não existe ou não está atualizado, a conexão volta a não usar SSL e se conecta com êxito.
>
> Recomendamos evitar o modo `preferred` No modo `preferred`, se a conexão encontrar um certificado inválido, ele para de usar criptografia e continua sem criptografia.

Depois de atualizar seus certificados de CA nos armazenamentos confiáveis do aplicativo cliente, você pode fazer o rodízio dos certificados nas suas instâncias de banco de dados. É altamente recomendável testar esses procedimentos em um ambiente de desenvolvimento ou teste antes de implementá-los em seus ambientes de produção.

Para obter mais informações sobre a mudança de certificados, consulte Alternar o certificado SSL/TLS. Para obter mais informações sobre como fazer download de certificados, consulte Usar SSL/TLS para criptografar uma conexão com uma instância ou um cluster de banco de dados. Para obter informações sobre o uso de SSL/TLS com instâncias de bancos de dados MySQL, consulte Suporte do SSL/TLS para instâncias de banco de dados do MySQL no Amazon RDS.

Tópicos

- Determinar se alguma aplicação está se conectando à instância de banco de dados MySQL usando SSL
- Determinar se um cliente requer verificação de certificado para se conectar
- Atualizar o armazenamento confiável de aplicações
- Exemplo de código Java para estabelecer conexões SSL

Determinar se alguma aplicação está se conectando à instância de banco de dados MySQL usando SSL

Se você estiver usando o Amazon RDS para MySQL versão 5.7, 8.0 ou 8.4 e o Esquema de desempenho estiver habilitado, execute a seguinte consulta para verificar se as conexões estão

usando SSL/TLS. Para obter informações sobre como habilitar o Esquema de performance, consulte Performance Schema quick start na documentação do MySQL.

```
mysql> SELECT id, user, host, connection_type
       FROM performance_schema.threads pst
       INNER JOIN information_schema.processlist isp
       ON pst.processlist_id = isp.id;
```

Nesta saída de exemplo, é possível ver que a sua própria sessão (`admin`) e um aplicativo conectado como `webapp1` estão usando SSL.

```
+----+-----------------+------------------+-----------------+
| id | user            | host             | connection_type |
+----+-----------------+------------------+-----------------+
|  8 | admin           | 10.0.4.249:42590 | SSL/TLS         |
|  4 | event_scheduler | localhost        | NULL            |
| 10 | webapp1         | 159.28.1.1:42189 | SSL/TLS         |
+----+-----------------+------------------+-----------------+
3 rows in set (0.00 sec)
```

Determinar se um cliente requer verificação de certificado para se conectar

É possível verificar se os clientes JDBC e MySQL exigem verificação de certificado para se conectarem.

JDBC

O exemplo a seguir com o MySQL Connector/J 8.0 mostra uma maneira de verificar as propriedades da conexão JDBC de um aplicativo para determinar se conexões bem-sucedidas exigem um certificado válido. Para obter mais informações sobre todas as opções de conexão JDBC para MySQL, consulte Propriedades da configuração na documentação do MySQL.

Ao usar o MySQL Connector/J 8.0, uma conexão SSL exigirá verificação com base no certificado do servidor de banco de dados se as propriedades da sua conexão tiverem `sslMode` definido como `VERIFY_CA` ou `VERIFY_IDENTITY`, como no exemplo a seguir.

```
Properties properties = new Properties();
properties.setProperty("sslMode", "VERIFY_IDENTITY");
properties.put("user", DB_USER);
properties.put("password", DB_PASSWORD);
```

> **Note**
>
> Se você usar o MySQL Java Connector v5.1.38 ou posterior, ou o MySQL Java Connector v8.0.9 ou posterior para se conectar a seus bancos de dados, mesmo que você não tenha configurado explicitamente seus aplicativos para usar SSL/TLS ao se conectar a seus bancos de dados, esses drivers cliente usam SSL/TLS como padrão. Além disso, ao usar SSL/TLS, eles executam a verificação parcial do certificado e haverá falha na conexão se o certificado do servidor de banco de dados tiver expirado.

MySQL

Os exemplos a seguir com o cliente MySQL mostram duas maneiras de verificar a conexão MySQL de um script para determinar se conexões bem-sucedidas exigem um certificado válido. Para obter mais informações sobre todas as opções de conexão com o cliente MySQL, consulte Configuração no lado do cliente para conexões criptografadas na documentação do MySQL.

Ao usar o Cliente MySQL 5.7 e posterior, uma conexão SSL requer verificação com base no certificado de CA de servidor se, para a opção `--ssl-mode`, você especificar `VERIFY_CA` ou `VERIFY_IDENTITY`, como no exemplo a seguir.

```
mysql -h mysql-database.rds.amazonaws.com -uadmin -ppassword --ssl-ca=/tmp/ssl-cert.pem
 --ssl-mode=VERIFY_CA
```

Atualizar o armazenamento confiável de aplicações

Para obter informações sobre como atualizar o armazenamento confiável para aplicações MySQL, consulte Installing SSL certificates na documentação do MySQL.

Para obter informações sobre como baixar o certificado raiz, consulte Usar SSL/TLS para criptografar uma conexão com uma instância ou um cluster de banco de dados.

Para obter exemplos de scripts que importam certificados, consulte Script de exemplo para importar certificados para o seu armazenamento confiável.

> **Note**
>
> Ao atualizar o armazenamento confiável, é possível reter certificados mais antigos, além de adicionar os novos certificados.

Se você estiver usando o driver JDBC mysql em um aplicativo, defina as seguintes propriedades nesse aplicativo.

```
System.setProperty("javax.net.ssl.trustStore", certs);
System.setProperty("javax.net.ssl.trustStorePassword", "password");
```

Ao iniciar o aplicativo, defina as seguintes propriedades.

```
java -Djavax.net.ssl.trustStore=/path_to_trust_store/MyTruststore.jks -
Djavax.net.ssl.trustStorePassword=my_trust_store_password com.companyName.MyApplication
```

> **Note**
>
> Especifique uma senha diferente do prompt mostrado aqui como prática recomendada de segurança.

Exemplo de código Java para estabelecer conexões SSL

O exemplo de código a seguir mostra como configurar a conexão SSL que valida o certificado de servidor usando JDBC.

```
public class MySQLSSLTest {

        private static final String DB_USER = "username";
```

```
        private static final String DB_PASSWORD = "password";
        // This trust store has only the prod root ca.
        private static final String TRUST_STORE_FILE_PATH = "file-path-to-trust-store";
        private static final String TRUST_STORE_PASS = "trust-store-password";

        public static void test(String[] args) throws Exception {
            Class.forName("com.mysql.jdbc.Driver");

            System.setProperty("javax.net.ssl.trustStore", TRUST_STORE_FILE_PATH);
            System.setProperty("javax.net.ssl.trustStorePassword", TRUST_STORE_PASS);

            Properties properties = new Properties();
            properties.setProperty("sslMode", "VERIFY_IDENTITY");
            properties.put("user", DB_USER);
            properties.put("password", DB_PASSWORD);


            Connection connection = null;
            Statement stmt = null;
            ResultSet rs = null;
            try {
                connection =
 DriverManager.getConnection("jdbc:mysql://mydatabase.123456789012.us-
east-1.rds.amazonaws.com:3306",properties);
                stmt = connection.createStatement();
                rs=stmt.executeQuery("SELECT 1 from dual");
            } finally {
                if (rs != null) {
                    try {
                        rs.close();
                    } catch (SQLException e) {
                    }
                }
                if (stmt != null) {
                   try {
                        stmt.close();
                    } catch (SQLException e) {
                   }
                }
                if (connection != null) {
                    try {
                        connection.close();
                    } catch (SQLException e) {
                        e.printStackTrace();
```

```
                    }
                }
            }
            return;
        }
    }
```

> **Important**
>
> Depois de determinar que suas conexões com o banco de dados usam SSL/TLS e ter atualizado o armazenamento confiável de aplicações, você poderá atualizar o banco de dados para usar os certificados rds-ca-rsa2048-g1. Para obter instruções, consulte a etapa 3 em Atualizar o certificado CA modificando a instância ou o cluster de banco de dados. Especifique uma senha diferente do prompt mostrado aqui como prática recomendada de segurança.

Uso da autenticação do Kerberos para o Amazon RDS para MySQL

É possível usar a autenticação Kerberos para autenticar usuários quando se conectam a sua instância de banco de dados do MySQL. A instância de banco de dados funciona com o AWS Directory Service for Microsoft Active Directory (AWS Managed Microsoft AD) para habilitar a autenticação Kerberos. Quando os usuários são autenticados por uma instância de banco de dados MySQL associada ao domínio de confiança, as solicitações de autenticação são encaminhadas. As solicitações encaminhadas vão para o diretório de domínio que você cria com o AWS Directory Service.

Manter todas as suas credenciais no mesmo diretório pode economizar tempo e esforço. Com essa abordagem, você tem um lugar centralizado para armazenar e gerenciar credenciais para várias instâncias de banco de dados. O uso de um diretório também pode melhorar o perfil de segurança geral.

Disponibilidade de região e versão

A disponibilidade e a compatibilidade de recursos varia entre versões específicas de cada mecanismo de banco de dados e entre Regiões da AWS. Para ter mais informações sobre a disponibilidade de versões e regiões do Amazon RDS com autenticação de Kerberos, consulte

Regiões e mecanismos de banco de dados compatíveis com a autenticação Kerberos no Amazon RDS.

Visão geral da configuração da autenticação Kerberos para instâncias de banco de dados MySQL

Para configurar a autenticação Kerberos para uma instância de banco de dados MySQL, conclua as seguintes etapas gerais, descritas em mais detalhes posteriormente:

1. Use AWS Managed Microsoft AD para criar um diretório do AWS Managed Microsoft AD. É possível usar o AWS Management Console, o AWS CLI ou o AWS Directory Service para criar o diretório. Para obter detalhes sobre como fazer isso, consulte o tópico sobre como Criar seu diretório AWS Managed Microsoft AD, no Guia de administração do AWS Directory Service.
2. Crie uma função do AWS Identity and Access Management (IAM) que use a politica gerenciada `AmazonRDSDirectoryServiceAccess` do IAM. A função permite ao Amazon RDS fazer chamadas para seu diretório.

   Para o perfil permitir o acesso, o endpoint do AWS Security Token Service (AWS STS) deve estar ativado na Região da AWS de sua conta da AWS. Os endpoints do AWS STS são ativados por padrão em todas as Regiões da AWS e você pode usá-los sem precisar tomar medidas adicionais. Para ter mais informações, consulte Ativar e desativar o AWS STS em uma Região da AWS no Guia do usuário do IAM.
3. Crie e configure usuários no diretório AWS Managed Microsoft AD usando as ferramentas do Microsoft Active Directory. Para ter mais informações sobre como criar usuários em seu Active Directory, consulte Gerenciar usuários e grupos no Microsoft AD gerenciado pela AWS no Guia de administração doAWS Directory Service.
4. Crie ou modifique uma instância de banco de dados MySQL. Se você usar a CLI ou a API do RDS na solicitação de criação, especifique um identificador de domínio com o parâmetro `Domain`. Use o identificador `d-*` que foi gerado quando você criou o diretório e o nome da função que você criou.

   Se você modificar uma instância de banco de dados MySQL existente para usar a autenticação Kerberos, defina os parâmetros de domínio e função do IAM para a instância de banco de dados. Localize a instância de banco de dados na mesma VPC que o diretório de domínio.
5. Use as credenciais de usuário mestre do Amazon RDS para conectar-se à instância de banco de dados MySQL. Crie o usuário no MySQL usando a cláusula `CREATE USER IDENTIFIED WITH`

`'auth_pam'`. Os usuários que você cria dessa maneira podem fazer login na instância de banco de dados MySQL usando a autenticação Kerberos.

Configurar a autenticação Kerberos para instâncias de banco de dados MySQL

Use o AWS Managed Microsoft AD para configurar a autenticação Kerberos para uma instância de banco de dados MySQL. Para configurar a autenticação Kerberos, você executa as seguintes etapas.

Etapa 1: Criar um diretório usando o AWS Managed Microsoft AD

O AWS Directory Service cria um Active Directory totalmente gerenciado na Nuvem AWS. Ao criar um diretório do AWS Managed Microsoft AD, o AWS Directory Service cria dois controladores de domínio e servidores do Domain Name System (DNS) em seu nome. Os servidores do diretório são criados em sub-redes diferentes em uma VPC. Essa redundância ajuda a garantir que o diretório permaneça acessível mesmo se ocorrer uma falha.

Ao criar um diretório do AWS Managed Microsoft AD, o AWS Directory Service executa as seguintes tarefas em seu nome:

- Configura um Active Directory dentro da VPC.
- Cria uma conta de administrador do diretório com o nome de usuário Admin e a senha especificada. Use essa conta para gerenciar seu diretório.

  > **Note**
  >
  > Salve essa senha. O AWS Directory Service não a armazena. Você pode redefini-la, mas não recuperá-la.

- Cria um grupo de segurança para os controladores do diretório.

Quando você inicia o AWS Managed Microsoft AD, o AWS cria uma Unidade organizacional (UO) que contém todos os objetos do diretório. Essa OU, que tem o nome NetBIOS que você digitou quando criou o diretório, está localizada na raiz do domínio. A raiz do domínio é controlada e de propriedade da AWS.

A conta admin que foi criada com o diretório do AWS Managed Microsoft AD tem permissões para as atividades administrativas mais comuns de sua OU:

- Criar, atualizar ou excluir usuários
- Adicionar recursos ao domínio, como servidores de arquivos ou de impressão, e atribuir permissões para esses recursos aos usuários na UO
- Criar OUs adicionais e contêineres
- Delegar autoridade
- Restaurar objetos excluídos da Lixeira do Active Directory
- Execute os módulos AD e DNS do Windows PowerShell no Active Directory Web Service

A conta admin também tem direitos para executar as seguintes atividades em todo o domínio:

- Gerenciar configurações de DNS (adicionar, remover ou atualizar registros, zonas e encaminhadores)
- Visualizar logs de eventos de DNS
- Visualizar logs de eventos de segurança

Como criar um diretório com AWS Managed Microsoft AD

1. Faça login no AWS Management Console e abra o console do AWS Directory Service em https://console.aws.amazon.com/directoryservicev2/.
2. No painel de navegação, escolha Directories (Diretórios) e escolha Set up directory (Configurar diretório).
3. Escolha AWS Managed Microsoft AD. O AWS Managed Microsoft AD é a única opção que você pode usar atualmente com o Amazon RDS.
4. Insira as seguintes informações:

   Nome do DNS do diretório

   O nome completo do diretório, como **`corp.example.com`**.

   Nome de NetBIOS do diretório

   O nome curto do diretório, como **`CORP`**.

   Descrição do diretório

   (Opcional) Uma descrição do diretório.

Senha do Admin

A senha do administrador do diretório. O processo de criação do diretório cria uma conta de administrador com o nome de usuário Admin e essa senha.

A senha do administrador do diretório e não pode incluir a palavra "admin". A senha diferencia letras maiúsculas de minúsculas e deve ter entre 8 e 64 caracteres. Ela também precisa conter pelo menos um caractere de três das quatro categorias a seguir:

- Letras minúsculas (a–z)
- Letras maiúsculas (A–Z)
- Números (0–9)
- Caracteres não alfanuméricos (~!@#$%^&*_-+=`|\(){}[]:;"'<>,.?/)

Confirmar senha

A senha do administrador digitada novamente.

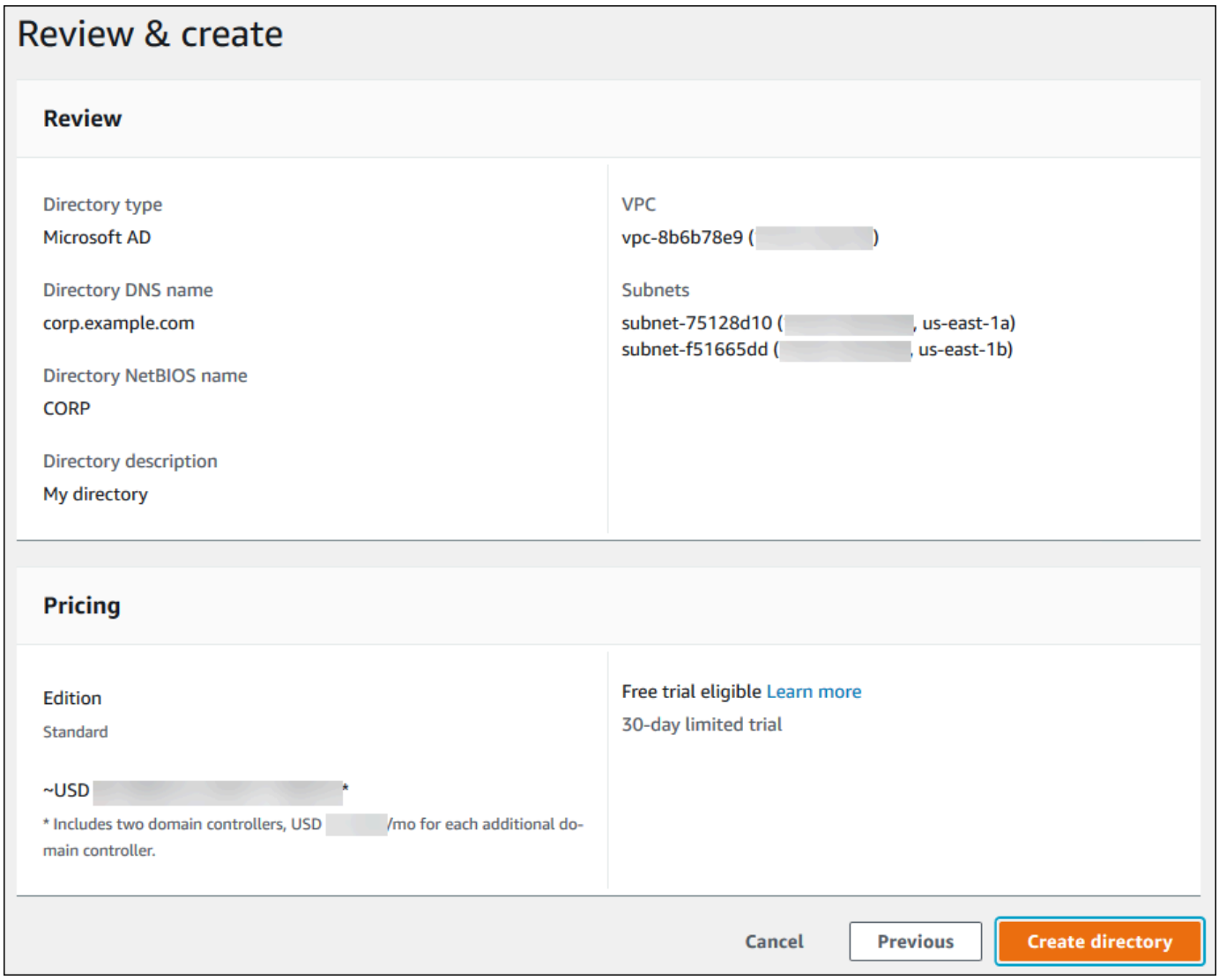
5. Escolha Next (Próximo).
6. Insira as seguintes informações na seção Networking (Rede) e escolha Next (Próximo):

   VPC

   A VPC do diretório. Crie a instância de banco de dados MySQL nesta mesma VPC.

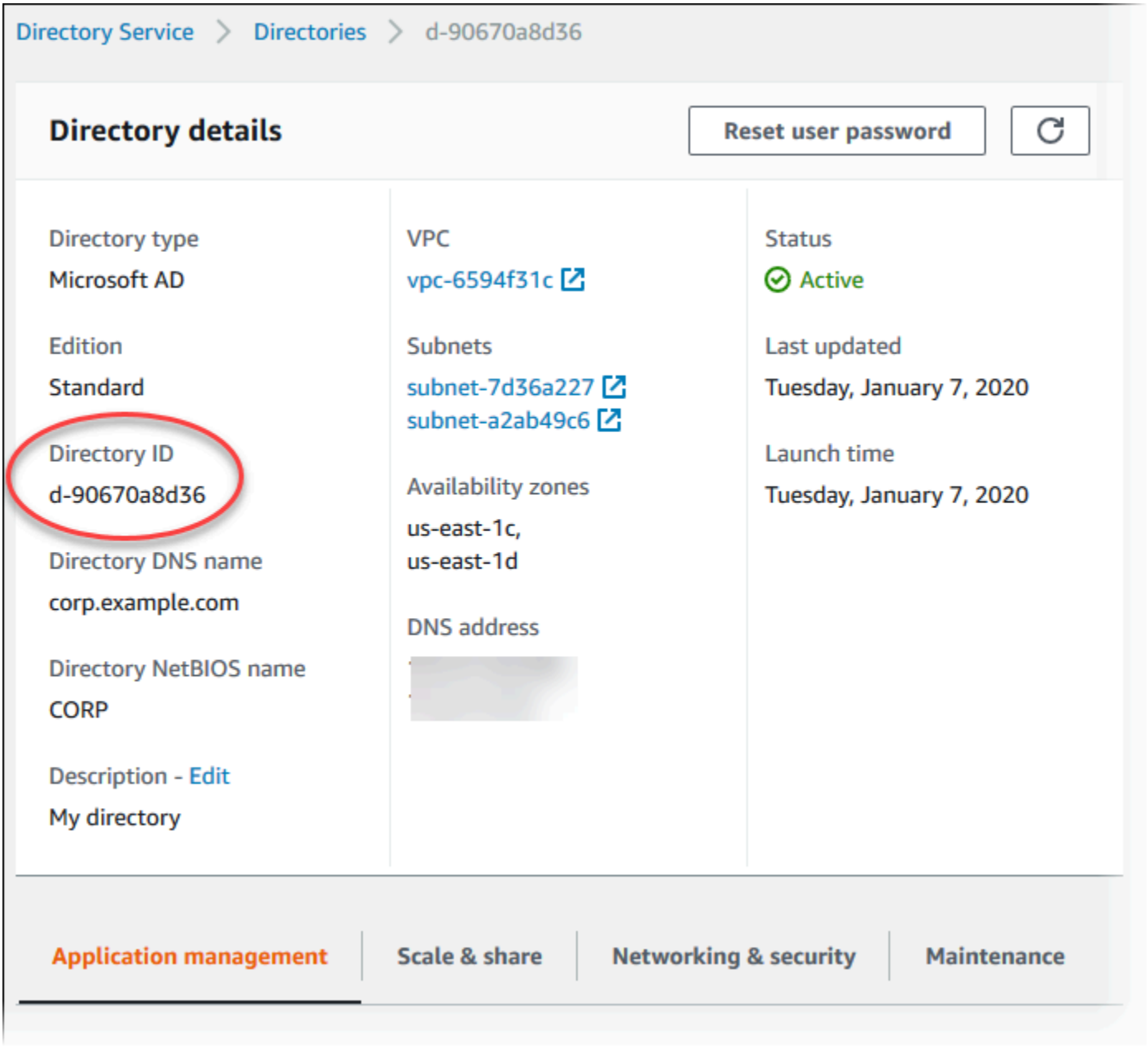
   Sub-redes

   Sub-redes para os servidores do diretório. As duas sub-redes deve estar em diferentes zonas de disponibilidade.
7. Revise as informações do diretório e faça as alterações necessárias. Quando as informações estiverem corretas, selecione Create directory (Criar diretório).



A criação do diretório leva vários minutos. Depois que o diretório tiver sido criado com sucesso, o valor de Status muda para Active (Ativo).

Para ver informações sobre o diretório, selecione o nome do diretório na listagem de diretórios. Anote o valor do Directory ID (ID do diretório) porque você precisará desse valor ao criar ou modificar sua instância de banco de dados MySQL.



Etapa 2: Criar a função do IAM para ser usada pelo Amazon RDS

Para que o Amazon RDS chame o AWS Directory Service para você, é necessária uma função do IAM que use a política gerenciada do IAM `AmazonRDSDirectoryServiceAccess`. Esta função permite que o Amazon RDS faça chamadas ao AWS Directory Service.

Quando uma instância de banco de dados é criada usando o AWS Management Console e o usuário do console tem a permissão `iam:CreateRole`, o console cria essa função automaticamente. Nesse caso, o nome da função é `rds-directoryservice-kerberos-access-role`. Caso contrário, é

necessário criar a função do IAM manualmente. Ao criar essa função do IAM, escolha `Directory Service` e associe a AWS política gerenciada da `AmazonRDSDirectoryServiceAccess` a ela.

Para ter mais informações sobre como criar funções do IAM para um serviço, consulte o tópico sobre como Criar uma função para delegar permissões a um serviço da AWS, no Guia do usuário do IAM.

> **Note**
>
> O perfil do IAM usado para a autenticação do Windows para RDS para o SQL Server não pode ser usado para o RDS para MySQL.

Se preferir, você poderá criar políticas com as permissões exigidas em vez de usar a política gerenciada do IAM `AmazonRDSDirectoryServiceAccess`. Nesse caso, o perfil do IAM deve ter a política de confiança do IAM a seguir.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "directoryservice.rds.amazonaws.com",
          "rds.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

A função também deve ter a política de função do IAM a seguir.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "ds:DescribeDirectories",
        "ds:AuthorizeApplication",
        "ds:UnauthorizeApplication",
        "ds:GetAuthorizedApplicationDetails"
      ],
    "Effect": "Allow",
    "Resource": "*"
    }
  ]
}
```

Etapa 3: Criar e configurar usuários

Você pode criar usuários com a ferramenta Usuários e computadores do Active Directory. Essa ferramenta faz parte das ferramentas do Active Directory Domain Services e do Active Directory Lightweight Directory Services. Os usuários representam pessoas ou entidades individuais que têm acesso ao seu diretório.

Para criar usuários em um diretório do AWS Directory Service, você deve estar conectado a uma instância do Amazon EC2 baseada no Microsoft Windows. Essa instância deve ser membro do diretório do AWS Directory Service e estar conectada como um usuário com privilégios para criar usuários. Para ter mais informações, consulte o tópico sobre como Gerenciar usuários e grupos AWS Managed Microsoft AD, no Guia de administração do AWS Directory Service.

Etapa 4: Criar ou modificar uma instância de banco de dados MySQL

Crie ou modifique uma instância de banco de dados MySQL para usar com o diretório. É possível usar o console, a CLI ou a API do RDS para associar uma instância de banco de dados a um diretório. Você pode fazer isso por meio de uma das seguintes maneiras:

- Crie uma instância de banco de dados MySQL usando o console, o comando create-db-instance da CLI ou a operação da API CreateDBInstance do RDS.

Para obter instruções, consulte Criar uma instância de banco de dados do Amazon RDS.

- Modifique uma instância de banco de dados existente do MySQL usando o console, o comando modify-db-instance da CLI ou a operação da API ModifyDBInstance do RDS.

  Para obter instruções, consulte Modificar uma instância de banco de dados do Amazon RDS.

- Restaure uma instância de banco de dados MySQL de um snapshot de banco de dados usando o console, o comando restore-db-instance-from-db-snapshot da CLI ou a operação da API RestoreDBInstanceFromDBSnapshot do RDS.

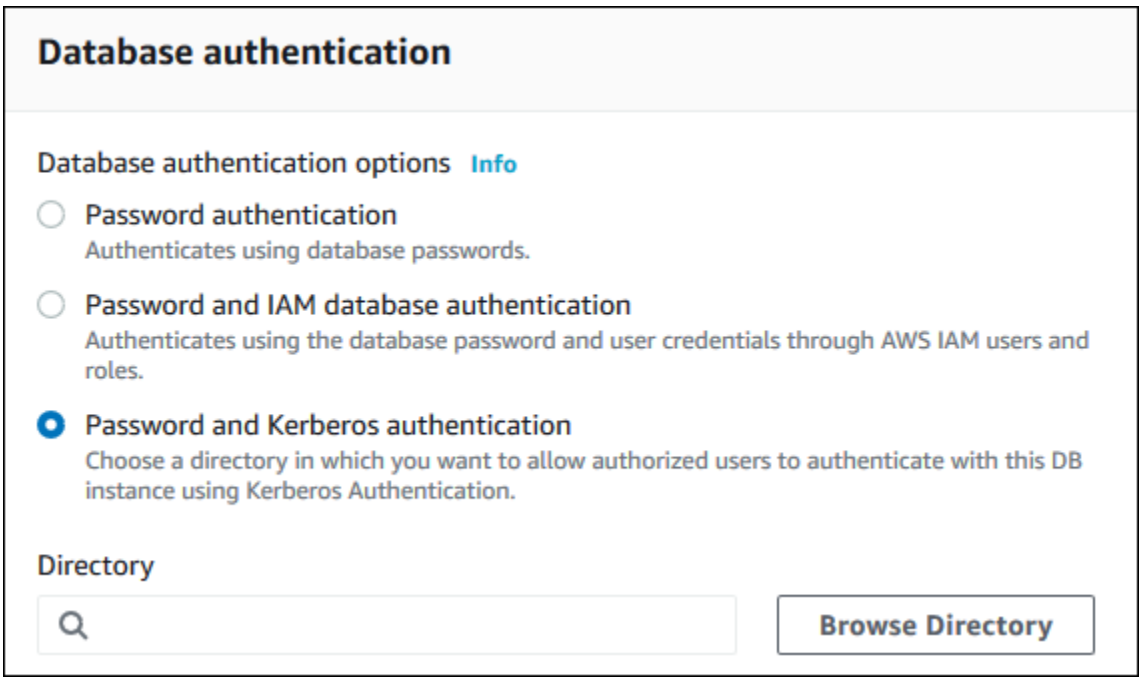
  Para obter instruções, consulte Restaurar uma instância de banco de dados.

- Restaure uma instância de banco de dados MySQL em um determinado momento usando o console, o comando restore-db-instance-to-point-in-time da CLI ou a operação da API RestoreDBInstanceToPointInTime do RDS.

  Para obter instruções, consulte Restaurar uma instância de banco de dados para um momento especificado no Amazon RDS.

A autenticação Kerberos só é compatível com instâncias de banco de dados MySQL em uma VPC. A instância de banco de dados pode estar na mesma VPC do diretório ou em uma VPC diferente. A instância de banco de dados que deve ser usada em um grupo de segurança que permita a saída na VPC do diretório para que a instância de banco de dados possa se comunicar com o diretório.

Ao usar o console para criar, modificar ou restaurar uma instância de banco de dados, escolha Password and Kerberos authentication (Senha e autenticação Kerberos) na seção Database authentication (Autenticação de banco de dados). Escolha Browse Directory (Procurar diretório) e selecione o diretório ou escolha Create a new directory (Criar um diretório).



Ao utilizar a AWS CLI ou a API do RDS, associe uma instância de banco de dados a um diretório. Os seguintes parâmetros são necessários para que a instância de banco de dados utilize o diretório de domínio que você criou:

- Para o parâmetro `--domain`, use o identificador de domínio (identificador "d-*") gerado quando o diretório foi criado.
- Para o parâmetro `--domain-iam-role-name`, use a função criada que usa a política gerenciada `AmazonRDSDirectoryServiceAccess` do IAM.

Por exemplo, o comando da CLI a seguir modifica uma instância de banco de dados para usar um diretório.

Para Linux, macOS ou Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --domain d-ID \
    --domain-iam-role-name role-name
```

Para Windows:

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

```
    --db-instance-identifier mydbinstance ^
    --domain d-ID ^
    --domain-iam-role-name role-name
```

> **Important**
>
> Se você modificar uma instância de banco de dados para permitir a autenticação Kerberos, reinicialize a instância de banco de dados após fazer a alteração.

Etapa 5: Criar logins do MySQL da autenticação Kerberos

Use as credenciais de usuário mestre do Amazon RDS para se conectar à instância de banco de dados do MySQL como você faz com qualquer outra instância de banco de dados. A instância de banco de dados é ingressada no domínio AWS Managed Microsoft AD. Assim, é possível provisionar logins e usuários do MySQL de usuários do Active Directory em seu domínio. As permissões de banco de dados são gerenciadas por meio de permissões padrão do MySQL que são concedidas e revogadas desses logins.

Você pode permitir que um usuário do Active Directory seja autenticado pelo MySQL. Para fazer isso, primeiro use as credenciais do usuário mestre do Amazon RDS para se conectar à instância de banco de dados MySQL como faz com qualquer outra instância de banco de dados. Depois de fazer login, crie um usuário autenticado externamente com PAM (Pluggable Authentication Modules) no MySQL executando o comando a seguir. Substitua *testuser* pelo nome de usuário.

```
CREATE USER 'testuser'@'%' IDENTIFIED WITH 'auth_pam';
```

Os usuários (humanos e aplicativos) de seu domínio agora podem se conectar à instância de banco de dados em uma máquina cliente conectada ao domínio usando a autenticação Kerberos.

> **Important**
>
> Recomendamos vivamente que os clientes usem conexões SSL/TLS ao usar a autenticação PAM. Se eles não usarem conexões SSL/TLS, a senha poderá ser enviada como texto não criptografado em alguns casos. Para exigir uma conexão criptografada SSL/TLS para o usuário do AD, execute o seguinte comando e substitua *testuser* pelo nome do usuário:
>
> ```
> ALTER USER 'testuser'@'%' REQUIRE SSL;
> ```

Para obter mais informações, consulte Suporte do SSL/TLS para instâncias de banco de dados do MySQL no Amazon RDS.

Gerenciamento de uma instância de banco de dados em um domínio

É possível usar a CLI ou a API do RDS para gerenciar a instância de banco de dados e suas relações com o Active Directory. Por exemplo, você pode associar um Active Directory para autenticação Kerberos e desassociar um Active Directory para desabilitar a autenticação Kerberos. Também é possível mover uma instância de banco de dados para ser autenticada externamente por um Microsoft Active Directory para outro.

Por exemplo, usando a API do Amazon RDS, você pode fazer o seguinte:

- Para tentar habilitar novamente a autenticação Kerberos para uma associação com falha, use a operação `ModifyDBInstance` da API e especifique o ID do diretório da associação atual.
- Para atualizar o nome da função do IAM para a associação, use a operação `ModifyDBInstance` da API e especifique o ID do diretório da associação atual e a nova função do IAM.
- Para desabilitar a autenticação Kerberos em uma instância de banco de dados, use a operação `ModifyDBInstance` da API e especifique `none` como o parâmetro de domínio.
- Para mover uma instância de banco de dados de um domínio para outro, use a operação `ModifyDBInstance` da API e especifique o identificador do novo domínio como o parâmetro do domínio.
- Para listar as associações de cada instância de banco de dados, use a operação `DescribeDBInstances` da API.

Compreensão da associação de domínio

Depois que você cria ou modifica a instância de banco de dados, ela se torna membro do domínio. É possível visualizar o status da associação do domínio da instância de banco de dados executando o comando describe-db-instances da CLI. O status da instância de banco de dados pode ser um dos seguintes:

- `kerberos-enabled` – a instância de banco de dados que tem a autenticação Kerberos habilitada.

- `enabling-kerberos`: a AWS está no processo de habilitar a autenticação Kerberos nessa instância de bancos de dados.
- `pending-enable-kerberos` – a habilitação da autenticação Kerberos está pendente nesta instância de banco de dados.
- `pending-maintenance-enable-kerberos`: a AWS tentará habilitar a autenticação Kerberos na instância de bancos de dados durante a próxima janela de manutenção programada.
- `pending-disable-kerberos` – a desabilitação da autenticação Kerberos está pendente nesta instância de banco de dados.
- `pending-maintenance-disable-kerberos`: a AWS tentará desabilitar a autenticação Kerberos na instância de banco de dados durante a próxima janela de manutenção programada.
- `enable-kerberos-failed`: um problema de configuração impediu que a AWS habilitasse a autenticação Kerberos na instância de banco de dados. Verifique e corrija sua configuração antes de emitir novamente o comando de modificação da instância de banco de dados.
- `disabling-kerberos`: a AWS está no processo de desabilitar a autenticação Kerberos nessa instância de bancos de dados.

Uma solicitação para habilitar a autenticação Kerberos pode falhar por conta de um novo problema de conectividade de rede ou de uma função do IAM incorreta. Por exemplo, suponha que você crie uma instância de banco de dados ou modifique uma instância de banco de dados existente e a tentativa de habilitar a autenticação Kerberos falhe. Se isso acontecer, emita o comando modify novamente ou modifique a instância de banco de dados recém-criada para ingressar no domínio.

Conectar-se ao MySQL com a autenticação Kerberos

Para conectar-se ao MySQL com a autenticação Kerberos, faça login usando o tipo de autenticação Kerberos.

Para criar um usuário de banco de dados ao qual você pode se conectar usando a autenticação Kerberos, use uma cláusula `IDENTIFIED WITH` na instrução `CREATE USER`. Para obter instruções, consulte Etapa 5: Criar logins do MySQL da autenticação Kerberos.

Para evitar erros, use o cliente `mysql` do MariaDB. É possível baixar o software MariaDB em https://downloads.mariadb.org/.

No prompt de comando, conecte-se a um dos endpoints associados à sua instância de banco de dados MySQL. Siga os procedimentos gerais em Conectar-se à instância de banco de dados do

MySQL. Quando a senha for solicitada, insira a senha do Kerberos associada a esse nome de usuário.

Restaurar uma instância de banco de dados MySQL e adicioná-la a um domínio

Você pode restaurar um snapshot de banco de dados ou fazer uma restauração point-in-time de uma instância de banco de dados MySQL e adicioná-la a um domínio. Depois que a instância de banco de dados tiver sido restaurada, modifique a instância de banco de dados usando o processo explicado em Etapa 4: Criar ou modificar uma instância de banco de dados MySQL para adicionar a instância de banco de dados a um domínio.

Limitações da autenticação Kerberos para MySQL

As seguintes limitações se aplicam à autenticação Kerberos para MySQL:

- Há suporte somente para AWS Managed Microsoft AD. Contudo, você pode associar as instâncias de banco de dados do RDS para MySQL a domínios gerenciados do Microsoft AD compartilhados de propriedade de contas diferentes na mesma Região da AWS.
- Você deve reinicializar a instância de banco de dados depois de habilitar o recurso.
- O tamanho do nome do domínio não pode ser maior que 61 caracteres.
- Você não pode habilitar a autenticação Kerberos e a autenticação do IAM ao mesmo tempo. Escolha um método de autenticação ou outro para sua instância de banco de dados MySQL.
- Não modifique a porta da instância de banco de dados depois de habilitar o recurso.
- Não use a autenticação Kerberos com réplicas de leitura.
- Se você tiver o upgrade automático de versão secundária ativado para uma instância de banco de dados MySQL que esteja utilizando a autenticação Kerberos, deverá desativar a autenticação Kerberos e depois ativá-la novamente após um upgrade automático. Para ter mais informações sobre upgrades de versões secundárias, consulte Atualizações automáticas da versão secundária do RDS para MySQL.
- Para excluir uma instância de banco de dados com esse recurso habilitado, primeiro desabilite o recurso. Para isso, execute o comando `modify-db-instance` da CLI para a instância de banco de dados e especifique `none` para o parâmetro `--domain`.

  Se você usar a CLI ou a API do RDS para excluir uma instância de banco de dados com esse recurso habilitado, espere um atraso.
- O RDS para MySQL não comporta a autenticação Kerberos em uma floresta de confiança entre seu AD on-premises ou auto-hospedado e o AWS Managed Microsoft AD.

Melhorar a performance das consultas para o RDS para MySQL com leituras otimizadas do Amazon RDS

É possível acelerar o processamento de consultas para o RDS para MySQL com leituras otimizadas do Amazon RDS. Uma instância de banco de dados ou um cluster de banco de dados multi-AZ do RDS para MySQL que utiliza o recurso Leituras otimizadas pelo RDS pode obter um processamento de consultas até duas vezes mais rápido em comparação com uma instância ou um cluster de banco de dados que não utiliza o recurso.

Tópicos

- Visão geral das leituras otimizadas do RDS
- Casos de uso para leituras otimizadas do RDS
- Práticas recomendadas para leituras otimizadas do RDS
- Utilizar leituras otimizadas do RDS
- Monitorar instâncias de banco de dados que utilizam leituras otimizadas do RDS
- Limitações das leituras otimizadas do RDS

Visão geral das leituras otimizadas do RDS

Quando você usa uma instância de banco de dados ou um cluster de banco de dados multi-AZ do RDS para MySQL com o recurso Leituras otimizadas pelo RDS ativado, essa instância ou cluster de banco de dados obtém uma performance de consulta mais rápida por meio do uso de um armazenamento de instância. Um armazenamento de instância fornece armazenamento temporário em nível de bloco para a instância de banco de dados ou o cluster de banco de dados multi-AZ. O armazenamento está localizado em unidades de estado sólido (SSDs) Non-Volatile Memory Express (NVMe) anexadas fisicamente ao servidor host. Esse armazenamento é otimizado para baixa latência, performance de E/S aleatória muito alta e alto throughput de leitura de sequencial.

O recurso Leituras otimizadas pelo RDS é ativado por padrão quando uma instância de banco de dados ou um cluster de banco de dados multi-AZ utiliza uma classe de instância de banco de dados com um armazenamento de instância, como db.m5d ou db.m6gd. Com leituras otimizadas do RDS, alguns objetos temporários são armazenados no armazenamento de instâncias. Esses objetos temporários incluem arquivos temporários internos, tabelas temporárias internas em disco, arquivos de mapa de memória e arquivos de cache de log binário. Para ter mais informações sobre

o armazenamento de instância, consulte Armazenamento de instância do Amazon EC2 no Amazon Elastic Compute Cloud Manual do usuário para instâncias do Linux.

As workloads que geram objetos temporários no MySQL para processamento de consultas podem aproveitar o armazenamento de instâncias para acelerar o processamento das consultas. Esse tipo de workload inclui consultas que envolvem classificações, agregações de hash, uniões de alta carga, expressões de tabela comuns (CTEs) e consultas em colunas não indexadas. Esses volumes de armazenamento de instâncias oferecem um nível superior de IOPS e performance, independentemente das configurações de armazenamento utilizadas para armazenamento persistente do Amazon EBS. Como as leituras otimizadas do RDS transferem operações em objetos temporários para o armazenamento de instâncias, as operações de entrada e saída por segundo (IOPS) ou o throughput do armazenamento persistente (Amazon EBS) agora podem ser utilizadas para operações em objetos persistentes. Essas operações incluem leituras e gravações regulares de arquivos de dados e operações do mecanismo em segundo plano, como limpar e inserir mesclagens de buffer.

> **Note**
>
> Os snapshots manuais e automatizados do RDS contêm apenas arquivos de mecanismo para objetos persistentes. Os objetos temporários criados no armazenamento de instâncias não estão incluídos nos snapshots do RDS.

Casos de uso para leituras otimizadas do RDS

Se você tem workloads que dependem muito de objetos temporários, como tabelas ou arquivos internos, para a execução de consultas, você pode se beneficiar da ativação de leituras otimizadas do RDS. Os seguintes casos de uso são candidatos para leituras otimizadas do RDS:

- Aplicações que executam consultas analíticas com expressões de tabela comuns (CTEs) complexas, tabelas derivadas e operações de agrupamento
- Réplicas de leitura que oferecem tráfego de leitura intenso com consultas não otimizadas
- Aplicações que executam consultas de relatórios dinâmicos ou sob demanda que envolvem operações complexas, como consultas com as cláusulas `GROUP BY` e `ORDER BY`
- Workloads que utilizam tabelas temporárias internas para processamento de consultas

Você pode monitorar a variável de status do mecanismo `created_tmp_disk_tables` para determinar o número de tabelas temporárias baseadas em disco criadas em sua instância de banco de dados.

- Aplicações que criam grandes tabelas temporárias, diretamente ou em procedimentos, para armazenar resultados intermediários
- Consultas de banco de dados que realizam agrupamento ou classificação em colunas não indexadas

Práticas recomendadas para leituras otimizadas do RDS

Use as práticas recomendadas a seguir para leituras otimizadas do RDS:

- Adicione uma lógica de repetição para consultas somente leitura caso elas falhem devido ao armazenamento de instância estar cheio durante a execução.
- Monitore o espaço de armazenamento disponível no armazenamento de instâncias com a métrica do CloudWatch `FreeLocalStorage`. Se o armazenamento de instâncias estiver atingindo seu limite devido à workload na instância de banco de dados, modifique a instância de banco de dados para utilizar uma classe de instância de banco de dados maior.
- Quando sua instância de banco de dados ou cluster de banco de dados multi-AZ tiver memória suficiente, mas ainda estiver atingindo o limite de armazenamento no armazenamento de instância, aumente o valor de `binlog_cache_size` para manter as entradas de log binário específicas da sessão na memória. Essa configuração impede a gravação das entradas de log binário em arquivos temporários de cache de log binário no disco.

  O parâmetro `binlog_cache_size` é específico da sessão. É possível alterar o valor de cada nova sessão. A configuração desse parâmetro pode aumentar a utilização da memória na instância de banco de dados durante a workload de pico. Portanto, considere aumentar o valor do parâmetro com base no padrão de workload de sua aplicação e na memória disponível na instância de banco de dados.
- Para o MySQL versão 8.0 e anterior, use o valor padrão de `MIXED` para o parâmetro `binlog_format`. Dependendo do tamanho das transações, definir `binlog_format` como `ROW` pode ocasionar arquivos de cache de log binário grandes no armazenamento de instâncias. Para o MySQL 8.4 e versões posteriores, use o valor padrão de `ROW` para o parâmetro `binlog_format`.

- Defina o parâmetro internal_tmp_mem_storage_engine como `TempTable` e defina o parâmetro temptable_max_mmap para corresponder ao tamanho do armazenamento disponível no armazenamento de instâncias.
- Evite realizar alterações em massa em uma única transação. Esses tipos de transações podem gerar arquivos de cache de log binário grandes no armazenamento de instâncias e causar problemas quando o armazenamento de instâncias está cheio. Considere dividir as gravações em várias transações pequenas para minimizar o uso de armazenamento para arquivos de cache de log binário.
- Utilize o valor padrão de `ABORT_SERVER` para o parâmetro `binlog_error_action`. Isso evita problemas com o registro em log binário em instâncias de banco de dados com backups ativados.

Utilizar leituras otimizadas do RDS

Quando você provisiona uma instância de banco de dados do RDS para MySQL com uma das classes de instância de banco de dados a seguir em uma implantação de instância de banco de dados single-AZ, uma implantação de instância de banco de dados multi-AZ ou uma implantação de cluster de banco de dados multi-AZ, a instância de banco de dados utiliza automaticamente o recurso Leituras otimizadas pelo RDS.

Para ativar as leituras otimizadas do RDS, execute um destes procedimentos:

- Crie uma instância de banco de dados ou um cluster de banco de dados multi-AZ do RDS para MySQL utilizando uma dessas classes de instância de banco de dados. Para ter mais informações, consulte Criar uma instância de banco de dados do Amazon RDS.
- Modifique uma instância de banco de dados ou um cluster de banco de dados multi-AZ existente do RDS para MySQL para utilizar uma dessas classes de instância de banco de dados. Para ter mais informações, consulte Modificar uma instância de banco de dados do Amazon RDS.

O recurso Leituras otimizadas pelo RDS está disponível em todas as Regiões da AWS onde há suporte para uma ou mais dessas classes de instância de banco de dados com SSD de NVMe local. Para obter informações sobre classes de instância de banco de dados, consulte the section called “Classes da instância de banco de dados”.

A disponibilidade da classe de instância de banco de dados difere em Regiões da AWS. Para determinar se uma classe de instância de banco de dados é compatível com uma Região da AWS específica, consulte the section called “Determinar o suporte para classes de instância de bancos de dados nas Regiões da AWS”.

Se você não quiser utilizar o recurso Leituras otimizadas pelo RDS, modifique sua instância de banco de dados ou cluster de banco de dados multi-AZ para que não use uma classe de instância de banco de dados compatível com o atributo.

Monitorar instâncias de banco de dados que utilizam leituras otimizadas do RDS

Você pode monitorar instâncias de banco de dados que utilizam leituras otimizadas do RDS com as seguintes métricas do CloudWatch:

- `FreeLocalStorage`
- `ReadIOPSLocalStorage`
- `ReadLatencyLocalStorage`
- `ReadThroughputLocalStorage`
- `WriteIOPSLocalStorage`
- `WriteLatencyLocalStorage`
- `WriteThroughputLocalStorage`

Essas métricas fornecem dados sobre armazenamento de instâncias, IOPS e throughput. Para ter mais informações sobre essas métricas, consulte Métricas específicas da instância do Amazon CloudWatch para Amazon RDS.

Limitações das leituras otimizadas do RDS

As seguintes limitações se aplicam às leituras otimizadas do RDS:

- O recurso de leituras otimizadas do RDS é compatível com as seguintes versões:
  - RDS para MySQL 8.0.28 e versões principais e secundárias posteriores

  Para obter informações sobre as versões do RDS para MySQL, consulte MySQL em versões do Amazon RDS.
- Você não pode alterar a localização de objetos temporários para armazenamento persistente (Amazon EBS) nas classes de instância de banco de dados compatíveis com leituras otimizadas do RDS.
- Quando o registro em log binário está ativado em uma instância de banco de dados, o tamanho máximo da transação é limitado pelo tamanho do armazenamento de instância. No MySQL,

qualquer sessão que exija mais armazenamento do que o valor de `binlog_cache_size` grava alterações de transações em arquivos temporários de cache de log binário, que são criados no armazenamento de instância.

- As transações podem falhar quando o armazenamento de instância está cheio.

Melhorar a performance das gravações com gravações otimizadas pelo RDS para MySQL

Você pode melhorar a performance das transações de gravação com as gravações otimizadas pelo Amazon RDS para MySQL. Quando seu banco de dados do RDS para MySQL utiliza gravações otimizadas pelo RDS, ele pode atingir um throughput de transações de gravação até duas vezes maior.

Tópicos

- Visão geral das gravações otimizadas do RDS
- Utilizar gravações otimizadas pelo RDS
- Habilitando gravações otimizadas para RDS em um banco de dados existente
- Limitações das gravações otimizadas pelo RDS

Visão geral das gravações otimizadas do RDS

Quando você ativa as gravações otimizadas pelo Amazon RDS, seus bancos de dados do RDS para MySQL gravam apenas uma vez ao liberar dados em um armazenamento durável sem a necessidade do buffer de gravação dupla. Os bancos de dados continuam fornecendo proteções de propriedade ACID para transações de banco de dados confiáveis, além de desempenho aprimorado.

Bancos de dados relacionais, como o MySQL, fornecem as propriedades ACID de atomicidade, consistência, isolamento e durabilidade para transações de banco de dados confiáveis. Para ajudar a fornecer essas propriedades, o MySQL utiliza uma área de armazenamento de dados denominada buffer de gravação dupla que evita erros parciais de gravação de página. Esses erros ocorrem quando há uma falha de hardware enquanto o banco de dados está atualizando uma página, como no caso de uma queda de energia. Um banco de dados MySQL pode detectar gravações parciais de páginas e recuperá-las com uma cópia da página no buffer de gravação dupla. Embora essa técnica ofereça proteção, ela também ocasiona operações extras de gravação. Para ter mais informações sobre o buffer de gravação dupla do MySQL, consulte Doublewrite Buffer (Buffer de gravação dupla) na documentação do MySQL.

Com as gravações otimizadas pelo RDS ativadas, os bancos de dados do RDS para MySQL gravam apenas uma vez ao liberar dados em um armazenamento durável sem a necessidade do buffer de gravação dupla. As gravações otimizadas pelo RDS são úteis se você executa workloads de uso intenso de gravação em seus bancos de dados do RDS para MySQL. Exemplos de bancos de

dados com workloads de uso intenso de gravação incluem os compatíveis com pagamentos digitais, negociação financeira e aplicações de jogos.

Esses bancos de dados são executados em classes de instância de banco de dados que utilizam o AWS Nitro System. Devido à configuração de hardware nesses sistemas, o banco de dados pode gravar páginas de 16 KiB diretamente nos arquivos de dados de forma confiável e durável em uma única etapa. O sistema AWS Nitro possibilita gravações otimizadas pelo RDS.

Você pode definir o novo parâmetro de banco de dados `rds.optimized_writes` para controlar o recurso de gravações otimizadas pelo RDS para bancos de dados do RDS para MySQL. Acesse esse parâmetro nos grupos de parâmetros de banco de dados do RDS para MySQL versão 8.0 e RDS para MySQL versão 8.4. Defina o parâmetro com os seguintes valores:

- `AUTO`: ative as gravações otimizadas pelo RDS se o banco de dados for compatível com o recurso. Desative as gravações otimizadas pelo RDS se o banco de dados não for compatível com o recurso. Essa é a configuração padrão.
- `OFF`: desative as gravações otimizadas pelo RDS mesmo se o banco de dados for compatível com o recurso.

Se você tem um banco de dados existente com uma versão de mecanismo, classe de instância de banco de dados e/ou formato de sistema de arquivos que não suporta gravações otimizadas para RDS, você pode ativar o recurso criando uma implantação azul/verde. Para ter mais informações, consulte the section called “Habilitação de ativação em um banco de dados existente”.

Se você migrar um banco de dados do RDS para MySQL configurado para usar gravações otimizadas pelo RDS para uma classe de instância de banco de dados que não oferece suporte ao recurso, o RDS desativará automaticamente as gravações otimizadas pelo RDS para o banco de dados.

Quando as gravações otimizadas pelo RDS estão desativadas, o banco de dados utiliza o buffer de gravação dupla do MySQL.

Para determinar se um banco de dados do RDS para MySQL está usando gravações otimizadas pelo RDS, visualize o valor atual do parâmetro `innodb_doublewrite` para o banco de dados. Se o banco de dados estiver usando gravações otimizadas pelo RDS, esse parâmetro será definido como `FALSE` (`0`).

Utilizar gravações otimizadas pelo RDS

Você pode ativar as gravações otimizadas pelo RDS ao criar um banco de dados do RDS para MySQL com o console do RDS, a AWS CLI o ou a API do RDS. As gravações otimizadas pelo RDS são ativadas automaticamente quando as duas condições a seguir se aplicam durante a criação do banco de dados:

- Você especifica uma versão de mecanismo de banco de dados e uma classe de instância de banco de dados compatíveis com as gravações otimizadas pelo RDS.
  - As gravações otimizadas pelo RDS são compatíveis com a versão 8.0.30 e posteriores do RDS para MySQL. Para obter informações sobre as versões do RDS para MySQL, consulte MySQL em versões do Amazon RDS.
  - As gravações otimizadas pelo RDS são compatíveis com bancos de dados do RDS para MySQL que usam as seguintes classes de instância de banco de dados:
    - db.m7i
    - db.m7g
    - db.m6g
    - db.m6gd
    - db.m6i
    - db.m5
    - db.m5d
    - db.r7i
    - db.r7g
    - db.r6g
    - db.r6gd
    - db.r6i
    - db.r5
    - db.r5b
    - db.r5d
    - db.x2idn
    - db.x2iedn

Para obter informações sobre classes de instância de banco de dados, consulte the section called “Classes da instância de banco de dados”.

A disponibilidade da classe de instância de banco de dados difere em Regiões da AWS. Para determinar se uma classe de instância de banco de dados é compatível com uma Região da AWS específica, consulte the section called “Determinar o suporte para classes de instância de bancos de dados nas Regiões da AWS”.

Para atualizar seu banco de dados para uma classe de instância de banco de dados compatível com as gravações otimizadas do RDS, você pode criar uma implantação azul/verde. Para ter mais informações, consulte the section called “Habilitação de ativação em um banco de dados existente”.

- No grupo de parâmetros associado ao banco de dados, o parâmetro `rds.optimized_writes` é definido como `AUTO`. Nos grupos de parâmetros padrão, esse parâmetro é sempre definido como `AUTO`.

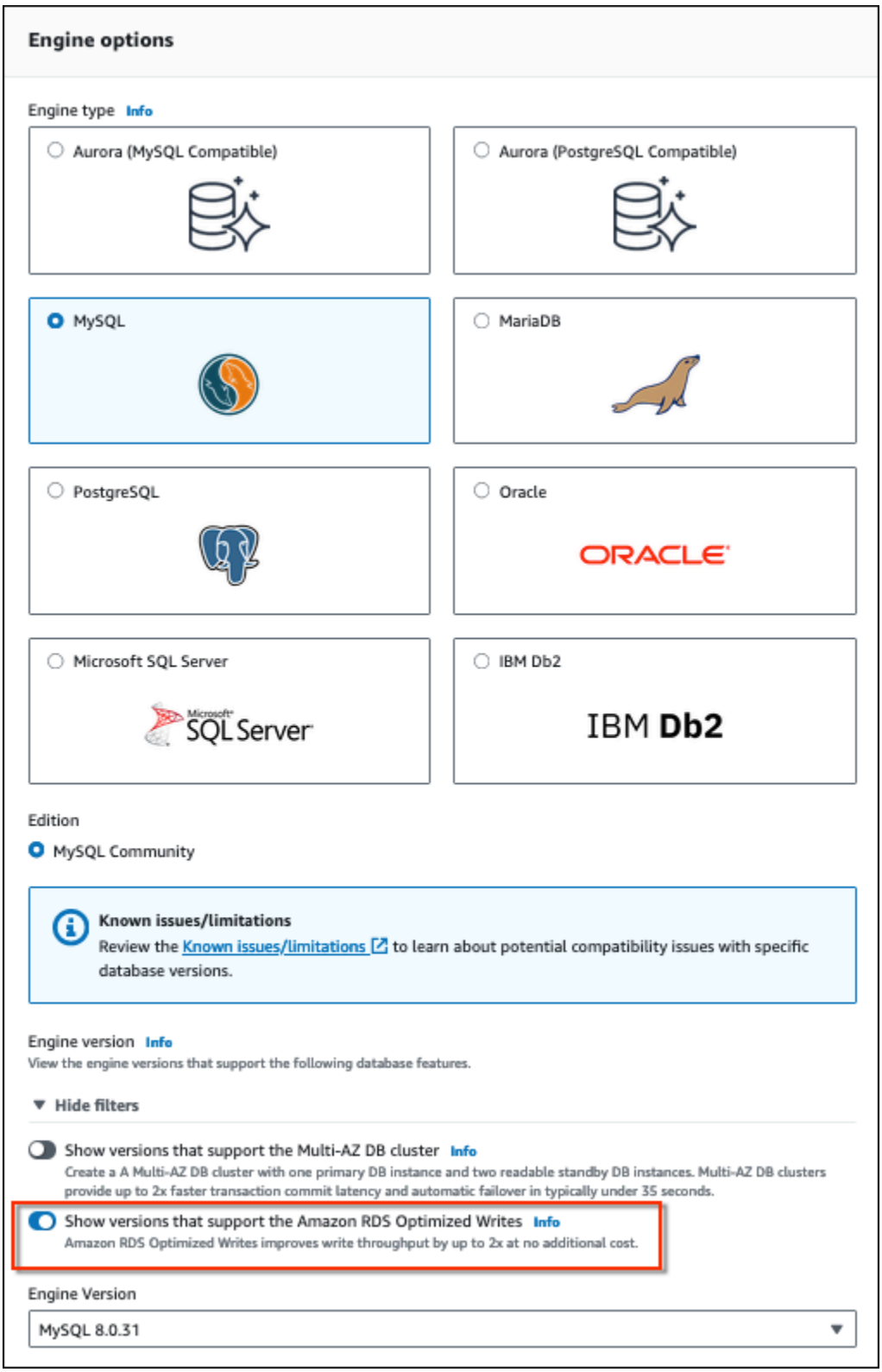
Se você quiser usar uma versão do mecanismo de banco de dados e uma classe de instância de banco de dados que sejam compatíveis com gravações otimizadas pelo RDS, mas não quer usar esse recurso, especifique um grupo de parâmetros personalizado ao criar o banco de dados. Nesse grupo de parâmetros, defina o parâmetro `rds.optimized_writes` como `OFF`. Se você quiser que o banco de dados use gravações otimizadas pelo RDS posteriormente, você pode definir o parâmetro `AUTO` para ativá-lo. Para obter informações sobre como criar grupos de parâmetros personalizados e definir parâmetros, consulte Grupos de parâmetros para Amazon RDS.

Para obter informações sobre como criar uma instância de banco de dados, consulte Criar uma instância de banco de dados do Amazon RDS.

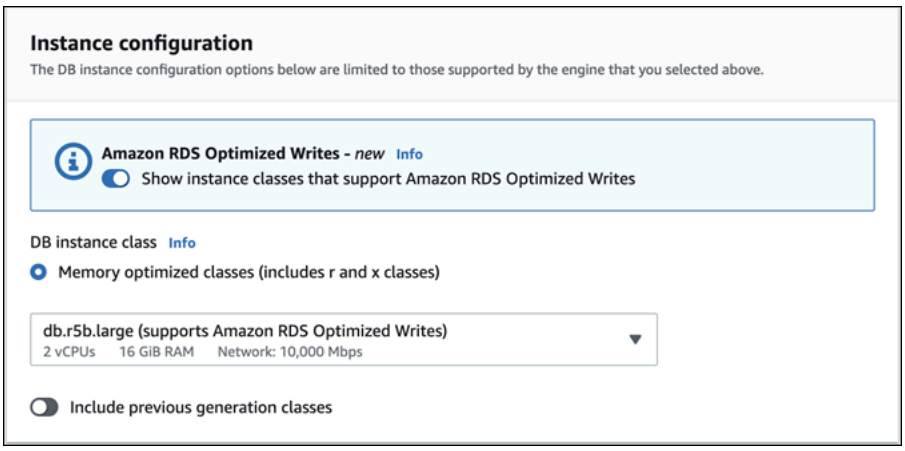
Console

Ao usar o console do RDS para criar um banco de dados do RDS para MySQL, você pode filtrar as versões do mecanismo de banco de dados e as classes de instância de banco de dados compatíveis com gravações otimizadas pelo RDS. Depois de ativar os filtros, é possível selecionar entre as versões disponíveis do mecanismo de banco de dados e classes de instância de banco de dados.

Para selecionar uma versão do mecanismo de banco de dados que seja compatível com gravações otimizadas pelo RDS, filtre as versões do mecanismo de banco de dados do RDS para MySQL compatíveis com o recurso em Engine version (Versão do mecanismo) e, depois, selecione uma versão.



Na seção Instance configuration (Configuração da instância), filtre as classes de instância de banco de dados que são compatíveis com gravações otimizadas pelo RDS e selecione uma classe de instância de banco de dados.



Depois de fazer essas seleções, você pode selecionar outras configurações que atendam aos seus requisitos e concluir a criação do banco de dados do RDS para MySQL com o console.

AWS CLI

Para criar uma instância de banco de dados utilizando o comando da AWS CLI, execute o comando create-db-instance. Os valores `--engine-version` e `--db-instance-class` devem ser compatíveis com gravações otimizadas pelo RDS. Além disso, verifique se o grupo de parâmetros associado a uma instância de banco de dados tem o parâmetro `rds.optimized_writes` definido como AUT0. O exemplo associa o grupo de parâmetros padrão à instância de banco de dados.

Example Criar uma instância de banco de dados que utilize gravações otimizadas pelo RDS

Para Linux, macOS ou Unix:

```
aws rds create-db-instance \
    --db-instance-identifier mydbinstance \
    --engine mysql \
    --engine-version 8.0.30 \
    --db-instance-class db.r5b.large \
    --manage-master-user-password \
    --master-username admin \
    --allocated-storage 200
```

Para Windows:

```
aws rds create-db-instance ^
    --db-instance-identifier mydbinstance ^
    --engine mysql ^
```

```
    --engine-version 8.0.30 ^
    --db-instance-class db.r5b.large ^
    --manage-master-user-password ^
    --master-username admin ^
    --allocated-storage 200
```

API do RDS

Você pode criar uma instância de banco de dados utilizando a operação CreateDBInstance. Ao utilizar essa operação, os valores `EngineVersion` e `DBInstanceClass` devem ser compatíveis com gravações otimizadas pelo RDS. Além disso, verifique se o grupo de parâmetros associado a uma instância de banco de dados tem o parâmetro `rds.optimized_writes` definido como `AUTO`.

Habilitando gravações otimizadas para RDS em um banco de dados existente

Para modificar um banco de dados existente do RDS para MySQL para ativar as gravações otimizadas do RDS, o banco de dados deve ter sido criado com uma versão do mecanismo de banco de dados e uma classe de instância de banco de dados compatíveis. Além disso, o banco de dados deve ter sido criado após o lançamento do RDS Optimized Writes em 27 de novembro de 2022, pois a configuração necessária do sistema de arquivos subjacente é incompatível com a dos bancos de dados criados antes do lançamento. Se essas condições forem atendidas, você poderá ativar as gravações otimizadas para RDS definindo o `rds.optimized_writes` parâmetro como. `AUTO`

Se seu banco de dados não foi criado com uma versão de mecanismo, classe de instância ou configuração de sistema de arquivos compatível, você pode usar o RDS Blue/Green Deployments para migrar para uma configuração compatível. Ao criar a implantação azul/verde, faça o seguinte:

- Selecione Habilitação de gravações otimizadas no banco de dados verde e, em seguida, especifique uma versão do mecanismo e uma classe de instância de banco de dados compatíveis com as gravações otimizadas do RDS. Para obter uma lista das versões dos mecanismos compatíveis, consulte Utilizar gravações otimizadas pelo RDS.
- Em Armazenamento, escolha Atualizar a configuração do sistema de arquivos de armazenamento. Essa opção atualiza o banco de dados para uma configuração de sistema de arquivos subjacente compatível.

Ao criar a implantação azul/verde, se o `rds.optimized_writes` parâmetro estiver definido para`AUTO`, as gravações otimizadas do RDS serão ativadas automaticamente no ambiente verde.

Você pode então fazer a transição azul/verde, o que promove o ambiente verde para o novo ambiente de produção.

Para ter mais informações, consulte the section called “Criar uma implantação azul/verde”.

Limitações das gravações otimizadas pelo RDS

Ao restaurar um banco de dados do RDS para MySQL a partir de um snapshot, você só pode ativar as gravações otimizadas pelo RDS no banco de dados se todas as condições a seguir se aplicarem:

- O snapshot foi criado a partir de um banco de dados compatível com gravações otimizadas pelo RDS.
- O snapshot foi criado por meio de um banco de dados criado após o lançamento de gravações otimizadas pelo RDS.
- O snapshot foi restaurado a partir de um banco de dados compatível com gravações otimizadas pelo RDS.
- O banco de dados restaurado é associado a um grupo de parâmetros que tenha o parâmetro `rds.optimized_writes` definido como `AUTO`.

Atualizações do mecanismo de banco de dados do RDS para MySQL

Quando o Amazon RDS oferece suporte a uma nova versão de um mecanismo de banco de dados, você pode atualizar suas instâncias de banco de dados para essa nova versão. Há dois tipos de atualizações para bancos de dados do MySQL: de versão principal e de versão secundária.

Atualizações da versão principal

As atualizações da versão principal podem conter as alterações de banco de dados incompatíveis com os aplicativos existentes. Como resultado, você deve realizar manualmente as atualizações de versões principais das suas instâncias de banco de dados. Você pode iniciar uma atualização de versão principal modificando manualmente sua instância de banco de dados. Antes de realizar uma atualização de versão principal, recomendamos seguir as instruções descritas em Atualizações da versão principal do RDS para MySQL.

Para atualizações de versão principal das implantações de instância de banco de dados multi-AZ, o Amazon RDS faz upgrade simultaneamente das réplicas de espera e primária. A instância de banco de dados não estará disponível até que a atualização seja concluída. Para upgrades de versão principal das implantações de cluster de banco de dados multi-AZ, o Amazon RDS faz upgrade das instâncias membro do cluster uma por vez.

> **Tip**
>
> É possível minimizar o tempo de inatividade necessário para a atualização da versão principal utilizando uma implantação azul/verde. Para ter mais informações, consulte Usar implantações azul/verde do Amazon RDS para atualizações de banco de dados.

Atualizações de versões secundárias

Atualizações de versões secundárias incluem somente alterações compatíveis com versões anteriores das aplicações existentes. Você pode iniciar uma atualização de versão secundária manualmente modificando sua instância de banco de dados. Ou é possível habilitar a opção Upgrade automático de versões secundárias ao criar ou modificar uma instância de banco de dados. Isso significa que o Amazon RDS atualiza automaticamente a instância de banco de dados depois de testar e aprovar a nova versão. Para obter informações sobre como realizar uma atualização, consulte Atualizar a versão de mecanismo de uma instância de banco de dados.

Quando você realiza uma atualização de versão secundária de um cluster de banco de dados multi-AZ, o Amazon RDS atualiza as instâncias de banco de dados de leitor, uma por vez. Depois, uma das instâncias de banco de dados de leitor passa a ser a nova instância de banco de dados de gravador. Depois, o Amazon RDS atualiza a antiga instância de gravador (que agora é uma instância de leitor).

> **Note**
>
> O tempo de inatividade da atualização de uma versão secundária de uma implantação de instância de banco de dados multi-AZ pode ser de alguns minutos. Os clusters de banco de dados multi-AZ normalmente reduzem o tempo de inatividade de atualizações de versões secundárias para aproximadamente 35 segundos. Quando usado com o RDS Proxy, é possível reduzir ainda mais o tempo de inatividade para um segundo ou menos. Para ter mais informações, consulte Amazon RDS Proxy. Como alternativa, é possível usar um proxy de banco de dados de código aberto, como ProxySQL, PgBouncer ou driver avançado de encapsulamento JDBC da AWS.

Se a instância de banco de dados do MySQL usar réplicas de leitura, atualize todas as réplicas de leitura antes de atualizar a instância de origem.

Tópicos

- Considerações sobre as atualizações do MySQL
- Encontrar destinos de atualização válidos
- Números de versão do MySQL
- Números de versão do RDS no RDS para MySQL
- Atualizações da versão principal do RDS para MySQL
- Testar uma atualização do RDS para MySQL
- Atualizar uma instância de banco de dados MySQL
- Atualizações automáticas da versão secundária do RDS para MySQL
- Usar uma réplica de leitura para reduzir o tempo de inatividade ao atualizar um banco de dados do RDS para MySQL

Considerações sobre as atualizações do MySQL

O Amazon RDS usa dois ou mais snapshots de banco de dados durante o processo de upgrade. O Amazon RDS gera até dois snapshots da instância de banco de dados antes de fazer qualquer alteração de upgrade. Se o upgrade não funcionar para seus bancos de dados, você poderá restaurar um desses snapshots para criar uma instância de banco de dados executando a versão antiga. O Amazon RDS gera outro snapshot da instância de banco de dados quando o upgrade é concluído. O Amazon RDS obtém esses snapshots independentemente de o AWS Backup gerenciar ou não os backups da instância de banco de dados.

> **Note**
>
> O Amazon RDS só tirará snapshot de banco de dadoss se você tiver definido o período de retenção de backup para sua instância de banco de dados como um número maior que 0. Para alterar o período de retenção de backup, consulte Modificar uma instância de banco de dados do Amazon RDS.

Após a conclusão da atualização, você não pode reverter para a versão anterior do mecanismo de banco de dados. Se quiser retornar à versão anterior, restaure o primeiro DB snapshot tirado para criar outra instância de banco de dados.

Você controla quando atualizar sua instância do banco de dados para uma nova versão compatível com o Amazon RDS. Esse nível de controle ajuda você a manter a compatibilidade com versões específicas de banco de dados e testar novas versões com seu aplicativo antes de implantá-lo na produção. Quando você estiver pronto, poderá executar as atualizações de versão quando achar melhor.

Se a instância de banco de dados usar replicação, atualize todas as réplicas de leitura antes de atualizar a instância de origem.

Encontrar destinos de atualização válidos

Ao usar o AWS Management Console para atualizar uma instância de banco de dados, ele exibe os destinos de atualização válidos para a instância de banco de dados. Também é possível executar o seguinte comando da AWS CLI para identificar os destinos de atualizações válidos para uma instância de banco de dados:

Para Linux, macOS ou Unix:

```
aws rds describe-db-engine-versions \
  --engine mysql \
  --engine-version version_number \
  --query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" --
output text
```

Para Windows:

```
aws rds describe-db-engine-versions ^
  --engine mysql ^
  --engine-version version_number ^
  --query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" --
output text
```

Por exemplo, para identificar os destinos de upgrades válidos para uma instância de banco de dados do MySQL versão 8.0.28, execute o seguinte comando da AWS CLI:

Para Linux, macOS ou Unix:

```
aws rds describe-db-engine-versions \
  --engine mysql \
  --engine-version 8.0.28 \
  --query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" --
output text
```

Para Windows:

```
aws rds describe-db-engine-versions ^
  --engine mysql ^
  --engine-version 8.0.28 ^
  --query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" --
output text
```

Números de versão do MySQL

A sequência de numeração de versões do mecanismo de banco de dados do RDS para MySQL está no formato major.minor.patch.YYYYMMDD ou major.minor.patch, por exemplo, 8.0.33.R2.20231201 ou 5.7.44. O formato usado depende da versão do mecanismo do MySQL. Consulte informações sobre a numeração de versão do Suporte estendido do RDS em Nomenclatura da versão do Suporte estendido do Amazon RDS.

principal

O número da versão principal é o número inteiro e a primeira parte fracionária do número da versão, por exemplo, 8.0. Uma atualização de versão principal aumenta a parte principal do número da versão. Por exemplo, uma atualização de 5.7.44 para 8.0.33 é uma atualização de versão principal, em que 5.7 e 8.0 são os números da versão principal.

secundária

O número da versão secundária é a terceira parte do número da versão, por exemplo, 33 em 8.0.33.

patch

O patch é a quarta parte do número da versão, por exemplo, o R2 em 8.0.33.R2. Uma versão do patch do RDS inclui correções de bugs importantes adicionadas a uma versão secundária após o lançamento.

YYYYMMDD

O patch é a quinta parte do número da versão, por exemplo, 20231201 em 8.0.33.R2.20231201. Uma versão da data do RDS é um patch de segurança que inclui correções de segurança importantes adicionadas a uma versão secundária após o lançamento. Ela não inclui nenhuma correção que possa mudar o comportamento de um mecanismo.

A tabela a seguir explica o esquema de nomenclatura do RDS para MySQL versão 8.4.

Versão secundária 8.4	Esquema da nomenclatura
≥ 3	Novas instâncias de banco de dados usam major.minor.patch.YYMMDD, por exemplo, 8.4.3.R2.20241201. As instâncias de banco de dados existentes podem usar major.minor.patch, por exemplo, 8.4.3.R2, até a próxima atualização de versão principal ou secundária.

A tabela a seguir explica o esquema de nomenclatura do RDS para MySQL versão 8.0.

8.0 versão secundária	Esquema da nomenclatura
≥ 33	Novas instâncias de banco de dados usam major.minor.patch.YYMMDD, por exemplo, 8.0.33.R2.20231201. As instâncias de banco de dados existentes podem usar major.minor.patch, por exemplo, 8.0.33.R2, até a próxima atualização de versão principal ou secundária.
< 33	As instâncias de banco de dados existentes usam major.minor.patch, por exemplo, 8.0.32.R2.

A tabela a seguir explica o esquema de nomenclatura do RDS para MySQL versão 5.7.

5.7 versão secundária	Esquema da nomenclatura
≥ 42	Novas instâncias de banco de dados usam major.minor.patch.YYMMDD, por exemplo, 5.7.42.R2.20231201. As instâncias de banco de dados existentes podem usar major.minor.patch, por exemplo, 5.7.42.R2, até a próxima atualização de versão principal ou secundária.

Números de versão do RDS no RDS para MySQL

Os números de versão do RDS usam o esquema de nomenclatura *major*.*minor*.*patch* ou *major*.*minor*.*patch*.*YYYYMMDD*. As versões do Suporte estendido do Amazon RDS usam o esquema de nomenclatura de versões secundárias *minor*-*RDS*.*AAAAMMDD*.

Uma versão do patch do RDS inclui correções de bugs importantes adicionadas a uma versão secundária após o lançamento. Uma versão de data do RDS (*AAAAMMDD*) é um patch de segurança. Um patch de segurança não inclui nenhuma correção que possa mudar o comportamento do mecanismo. Consulte informações sobre a numeração de versão do Suporte estendido do RDS em Nomenclatura da versão do Suporte estendido do Amazon RDS.

É possível descobrir o número da versão do RDS do banco de dados do RDS para MySQL com a seguinte consulta SQL:

```
mysql> select mysql.rds_version();
```

Por exemplo, consultar um banco de dados do RDS para MySQL 8.0.34 exibe a seguinte saída:

```
+---------------------+
| mysql.rds_version() |
+---------------------+
| 8.0.34.R2.20231201  |
+---------------------+
1 row in set (0.01 sec)
```

Atualizações da versão principal do RDS para MySQL

O Amazon RDS oferece suporte para as seguintes atualizações in-loco para versões principais do mecanismo de banco de dados MySQL:

- MySQL 5.7 a MySQL 8.0
- MySQL 8.0 a MySQL 8.4

> **Note**
>
> Você só pode criar instâncias de banco de dados MySQL versões 5.7, 8.0 e 8.4 com classes de instância de banco de dados de última geração e da geração atual.
> Em alguns casos, você deseja atualizar uma instância de banco de dados em execução em uma classe de instância de banco de dados de geração anterior com uma versão do mecanismo MySQL posterior. Nesses casos, primeiro modifique a instância de banco de dados para usar uma classe de instância de última geração ou de geração atual. Depois de fazer isso, você poderá modificar a instância de banco de dados para usar a versão de mecanismo de banco de dados MySQL posterior. Para obter informações sobre classes de instâncias de banco de dados do Amazon RDS, consulte Classes de instâncias de banco de dados .

Tópicos

- Visão geral de atualizações de versões principais do MySQL
- Pré-verificações de atualizações

- Reversão após falha na atualização

Visão geral de atualizações de versões principais do MySQL

As atualizações da versão principal podem conter as alterações de banco de dados incompatíveis com os aplicativos existentes. Como resultado, o Amazon RDS não aplica atualizações da versão principal automaticamente. Você deve modificar manualmente sua instância de banco de dados. Convém testar completamente qualquer atualização antes de aplicá-la às suas instâncias de produção.

Para realizar uma atualização de versão principal, primeiro execute todas as atualizações de sistema operacional disponíveis. Após a conclusão das atualizações do sistema operacional, atualize para as versões principais, por exemplo, de 5.7 para 8.0 e, depois, para 8.4. Para obter informações sobre a atualização de um cluster de banco de dados multi-AZ do RDS para MySQL, consulte Atualização da versão de mecanismo de um cluster de banco de dados multi-AZ para o Amazon RDS. As instâncias de banco de dados MySQL criadas antes de 24 de abril de 2014 mostrarão uma atualização do sistema operacional disponível enquanto a atualização não for aplicada. Para obter mais informações sobre atualizações do sistema operacional, consulte Aplicar atualizações para uma instância de banco de dados.

Durante uma atualização de versão principal do MySQL, o Amazon RDS executa o binário do MySQL `mysql_upgrade` para atualizar tabelas, se necessário. Além disso, o Amazon RDS esvazia as tabelas `slow_log` e `general_log` durante uma atualização de versão principal. Para preservar as informações do log, salve o conteúdo do log antes da atualização da versão principal.

Em geral, as atualizações da versão principal do MySQL são concluídas em cerca de 10 minutos. Algumas atualizações podem demorar mais por causa do tamanho da classe da instância de banco de dados ou porque a instância não segue certas diretrizes operacionais em Práticas recomendadas do Amazon RDS. Se você atualizar uma instância de banco de dados do console do Amazon RDS, o status dessa instância indicará quando a atualização estiver concluída. Se você atualizar usando a AWS Command Line Interface (AWS CLI), use o comando describe-db-instances e verifique o valor de `Status`.

Pré-verificações de atualizações

O Amazon RDS executa verificações prévias antes da atualização para verificar se há incompatibilidades. Essas incompatibilidades variam de acordo com a versão do MySQL para a qual está sendo feita a atualização.

As pré-verificações incluem algumas que estão incluídas com o MySQL e outras que foram criadas especificamente pela equipe do Amazon RDS. Para obter informações sobre as pré-verificações fornecidas pelo MySQL, consulte Utilitário verificador de atualização.

As pré-verificações são executadas antes que a instância de banco de dados seja interrompida para a atualização, o que significa que elas não causam nenhum tempo de inatividade quando são executadas. Se as pré-verificações encontrarem uma incompatibilidade, o Amazon RDS cancelará automaticamente a atualização antes que a instância de banco de dados seja interrompida. O Amazon RDS também gera um evento para a incompatibilidade. Para obter mais informações sobre eventos do Amazon RDS, consulte Trabalhar com a notificação de eventos do Amazon RDS.

O Amazon RDS registra informações detalhadas sobre cada incompatibilidade no arquivo de log `PrePatchCompatibility.log`. Na maioria dos casos, a entrada de log inclui um link para a documentação do MySQL para corrigir a incompatibilidade. Para obter mais informações sobre como exibir arquivos de log, consulte Como visualizar e listar arquivos de log do banco de dados.

Devido à natureza das pré-verificações, eles analisam os objetos do seu banco de dados. Essa análise resulta no consumo do recurso e aumenta o tempo para que a atualização seja concluída.

Tópicos

- Pré-verificações de atualizações do MySQL 8.0 para 8.4
- Pré-verificações de upgrades do MySQL 5.7 a 8.0

Pré-verificações de atualizações do MySQL 8.0 para 8.4

O MySQL 8.4 inclui várias incompatibilidades com o MySQL 8.0. Essas incompatibilidades podem causar problemas durante uma atualização do MySQL 8.0 para o MySQL 8.4. Portanto, pode ser necessária uma certa preparação no seu banco de dados para que a atualização seja bem-sucedida. Veja a seguir uma lista geral dessas incompatibilidades:

- Não deve haver tabelas que usam funções ou tipos de dados obsoletos.
- Os gatilhos não devem ter um definidor vazio ou ausente ou um contexto de criação inválido.
- Não deve haver violações de palavra-chave ou palavra reservada. Algumas palavras-chave podem ser reservadas no MySQL 8.4 que não eram reservadas anteriormente.

  Para obter mais informações, consulte Palavras-chave e palavras reservadas na documentação do MySQL.

- Não deve haver tabelas no banco de dados do sistema `mysql` do MySQL 8.0 com o mesmo nome de uma tabela usada pelo dicionário de dados do MySQL 8.4.
- Não deve haver modos obsoletos do SQL definidos na configuração de variável do sistema `sql_mode`.
- Não deve haver tabelas ou procedimentos armazenados com elementos de coluna `ENUM` ou `SET` individuais que excedam 255 caracteres ou 1020 bytes.
- A instalação do MySQL 8.0 não deve usar recursos incompatíveis com o MySQL 8.4.

  Para obter mais informações, consulte Recursos removidos no MySQL 8.4 na documentação do MySQL.
- Não deve haver nomes de restrição de chave externa maiores que 64 caracteres.
- Para obter suporte aprimorado ao Unicode, revise as seguintes informações:
  - Considere a conversão de objetos que usam o conjunto de caracteres `utf8mb3` para que usem o conjunto de caracteres `utf8mb4`. O conjunto de caracteres `utf8mb3` está obsoleto.
  - Considere o uso de `utf8mb4` para referências de conjuntos de caracteres em vez de `utf8`, pois, no momento, `utf8` é um alias para o conjunto de caracteres `utf8mb3`. Se possível, mude `utf8` para `utf8mb4` primeiro e, depois, faça upgrade do banco de dados.
  - Como clientes mais antigos podem receber um erro de conjunto de caracteres desconhecido para `utf8mb3`, faça upgrade dos clientes de banco de dados antes de fazer upgrade do banco de dados.

  Para obter mais informações, consulte The utf8mb3 character set (3-byte UTF-8 unicode encoding) na documentação do MySQL.

  Para alterar os conjuntos de caracteres, é possível executar manualmente o processo de backup, restauração e replicação do banco de dados. Ou você pode usar implantações azul/verde do Amazon RDS. Para ter mais informações, consulte Usar implantações azul/verde do Amazon RDS para atualizações de banco de dados.

Quando você inicia uma atualização do MySQL 8.0 para 8.4, o Amazon RDS executa pré-verificações automaticamente para detectar essas incompatibilidades. Para obter informações sobre como atualizar para o MySQL 8.4, consulte Atualização do MySQL na documentação do MySQL.

Essas pré-verificações são obrigatórias. Você não pode optar por ignorá-las. As pré-verificações fornecem os seguintes benefícios:

- Elas permitem evitar o tempo de inatividade não planejado durante a atualização.
- Se houver incompatibilidades, o Amazon RDS impedirá a atualização e fornecerá um log para que você saiba sobre elas. Dessa forma, você poderá usar o log para preparar seu banco de dados para a atualização para o MySQL 8.4 ao reduzir essas incompatibilidades. Consulte informações detalhadas sobre como remover incompatibilidades em Preparação da instalação para atualização na documentação do MySQL.

Pré-verificações de upgrades do MySQL 5.7 a 8.0

O MySQL 8.0 inclui várias incompatibilidades com o MySQL 5.7. Essas incompatibilidades podem causar problemas durante uma atualização do MySQL 5.7 para o MySQL 8.0. Portanto, pode ser necessária uma certa preparação no seu banco de dados para que a atualização seja bem-sucedida. Veja a seguir uma lista geral dessas incompatibilidades:

- Não deve haver tabelas que usam funções ou tipos de dados obsoletos.
- Não deve haver arquivos *.frm órfãos.
- Os gatilhos não devem ter um definidor vazio ou ausente ou um contexto de criação inválido.
- Não deve haver tabela particionada que usa um mecanismo de armazenamento que não tem suporte para particionamento nativo,
- Não deve haver violações de palavra-chave ou palavra reservada. Algumas palavras-chave podem ser reservadas no MySQL 8.0 que não eram reservadas anteriormente.

  Para obter mais informações, consulte Palavras-chave e palavras reservadas na documentação do MySQL.
- Não deve haver tabelas no banco de dados do sistema `mysql` do MySQL 5.7 com o mesmo nome de uma tabela usada pelo dicionário de dados do MySQL 8.0.
- Não deve haver modos obsoletos do SQL definidos na configuração de variável do sistema `sql_mode`.
- Não deve haver tabelas ou procedimentos armazenados com elementos de coluna `ENUM` ou `SET` individuais que excedam 255 caracteres ou 1020 bytes.
- Antes de atualizar para o MySQ 8.0.13 ou posterior, não deve haver partições de tabela que residam em espaços de tabela compartilhados do InnoDB.
- Não deve haver consultas e definições de programa armazenadas do MySQL 8.0.12 ou anterior que usem qualificadores `ASC` ou `DESC` para causas `GROUP BY`.
- A instalação do MySQL 5.7 não deve usar recursos incompatíveis com o MySQL 8.0.

Para obter mais informações, consulte Recursos removidos no MySQL 8.0 na documentação do MySQL.

- Não deve haver nomes de restrição de chave externa maiores que 64 caracteres.
- Para obter suporte aprimorado ao Unicode, revise as seguintes informações:
  - Considere a conversão de objetos que usam o conjunto de caracteres `utf8mb3` para que usem o conjunto de caracteres `utf8mb4`. O conjunto de caracteres `utf8mb3` está obsoleto.
  - Considere o uso de `utf8mb4` para referências de conjuntos de caracteres em vez de `utf8`, pois, no momento, `utf8` é um alias para o conjunto de caracteres `utf8mb3`. Se possível, mude `utf8` para `utf8mb4` primeiro e, depois, faça upgrade do banco de dados.
  - Como clientes mais antigos podem receber um erro de conjunto de caracteres desconhecido para `utf8mb3`, faça upgrade dos clientes de banco de dados antes de fazer upgrade do banco de dados.

  Para obter mais informações, consulte The utf8mb3 character set (3-byte UTF-8 unicode encoding) na documentação do MySQL.

  Para alterar os conjuntos de caracteres, é possível executar manualmente o processo de backup, restauração e replicação do banco de dados. Ou você pode usar implantações azul/verde do Amazon RDS. Para ter mais informações, consulte Usar implantações azul/verde do Amazon RDS para atualizações de banco de dados.

Quando você inicia uma atualização do MySQL 5,7 para 8,0, o Amazon RDS executa pré-verificações automaticamente para detectar essas incompatibilidades. Para obter informações sobre como atualizar para o MySQL 8.0, consulte Upgrading MySQL (em inglês) na documentação do MySQL.

Essas pré-verificações são obrigatórias. Você não pode optar por ignorá-las. As pré-verificações fornecem os seguintes benefícios:

- Elas permitem evitar o tempo de inatividade não planejado durante a atualização.
- Se houver incompatibilidades, o Amazon RDS impedirá a atualização e fornecerá um log para que você saiba sobre elas. Dessa forma, você poderá usar o log para preparar seu banco de dados para a atualização para o MySQL 8.0 ao reduzir essas incompatibilidades. Para obter informações detalhadas sobre como remover incompatibilidades, consulte Preparing your installation for upgrade na documentação do MySQL e Upgrading to MySQL 8.0? Here is what you need to know... no blog do MySQL Server.

Reversão após falha na atualização

Quando você atualiza uma instância de banco de dados do MySQL versão 5.7 para o MySQL versão 8.0 ou do MySQL versão 8.0 para 8.4, a atualização pode falhar. Em particular, ela pode falhar se o dicionário de dados contiver incompatibilidades que não foram capturadas pelas pré-verificações. Nesse caso, não é possível iniciar o banco de dados com sucesso na nova versão do MySQL, a 8.0 ou 8.4. Nesse ponto, o Amazon RDS reverte as alterações realizadas para a atualização. Após a reversão, a instância de banco de dados MySQL executa a versão original:

- MySQL versão 8.0 (para uma reversão do MySQL 8.4)
- MySQL versão 5.7 (para uma reversão do MySQL 8.0)

Quando uma atualização falha e é revertida, o Amazon RDS gera um evento com o ID de evento RDS-EVENT-0188.

Normalmente, uma atualização falha porque há incompatibilidades nos metadados entre os bancos de dados em sua instância de banco de dados e a versão de destino do MySQL. Quando uma atualização falha, você pode visualizar os detalhes sobre essas incompatibilidades no arquivo `upgradeFailure.log`. Resolva as incompatibilidades antes de tentar atualizar novamente.

Durante uma tentativa de atualização e reversão malsucedidas, sua instância de banco de dados é reiniciada. Quaisquer alterações de parâmetro pendentes são aplicadas durante a reinicialização e persistem após a reversão.

Para obter mais informações sobre como atualizar para o MySQL 8.0, consulte os seguintes tópicos na documentação do MySQL:

- Preparing Your Installation for Upgrade
- Upgrading to MySQL 8.0? Here is what you need to know…

Para obter mais informações sobre como atualizar para o MySQL 8.4, consulte Preparação da instalação para atualização na documentação do MySQL.

Testar uma atualização do RDS para MySQL

Antes de executar uma atualização de versão principal na sua instância de banco de dados, você deve testar completamente seu banco de dados para verificar a compatibilidade com a nova

versão. Além disso, teste exaustivamente todos os aplicativos que acessam o banco de dados para compatibilidade com a nova versão. Recomendamos que você use o procedimento a seguir.

Para testar um upgrade de versão principal

1. Reveja a documentação de upgrade para a nova versão do mecanismo de banco de dados para ver se existem problemas de compatibilidade que podem afetar seu banco de dados ou seus aplicativos:
   - Alterações feitas no MySQL 5.7
   - Alterações feitas no MySQL 8.0
   - Alterações feitas no MySQL 8.4
2. Se a sua instância de banco de dados for um membro de um grupo de parâmetros de banco de dados personalizado, crie um novo grupo de parâmetros de banco de dados com as configurações existentes que seja compatível com a nova versão principal. Especifique o novo grupo de parâmetros de banco de dados quando você atualizar sua instância de teste, de modo que seu teste de atualização garanta que ela funcione corretamente. Para obter mais informações sobre como criar um parameter group de banco de dados, consulte Grupos de parâmetros para Amazon RDS.
3. Crie um snapshot de banco de dados da instância de banco de dados a ser atualizada. Para ter mais informações, consulte Criar um snapshot de banco de dados para uma instância de banco de dados single-AZ para o Amazon RDS.
4. Restaure o snapshot de banco de dados para criar uma nova instância de banco de dados de teste. Para obter mais informações, consulte Restaurar uma instância de banco de dados.
5. Modifique essa nova instância de banco de dados de teste para atualizá-la para a nova versão, usando um dos métodos detalhados a seguir. Se você criou um novo parameter group na etapa 2, especifique-o.
6. Avalie o armazenamento usado pela instância atualizada para determinar se o upgrade requer armazenamento adicional.
7. Execute quantos testes de garantia de qualidade na instância de banco de dados atualizada forem necessários para garantir que o seu banco de dados e o seu aplicativo funcionem corretamente com a nova versão. Implemente os novos testes necessários para avaliar o impacto de problemas de compatibilidade que você identificou na etapa 1. Teste todos os procedimentos armazenados e funções. Direcione versões de teste dos seus aplicativos à instância de banco de dados atualizada.

8. Se todos os testes forem aprovados, realize o upgrade na sua instância de banco de dados de produção. Recomendamos que você não permita operações de gravação na instância de banco de dados até confirmar que tudo está funcionando corretamente.

Atualizar uma instância de banco de dados MySQL

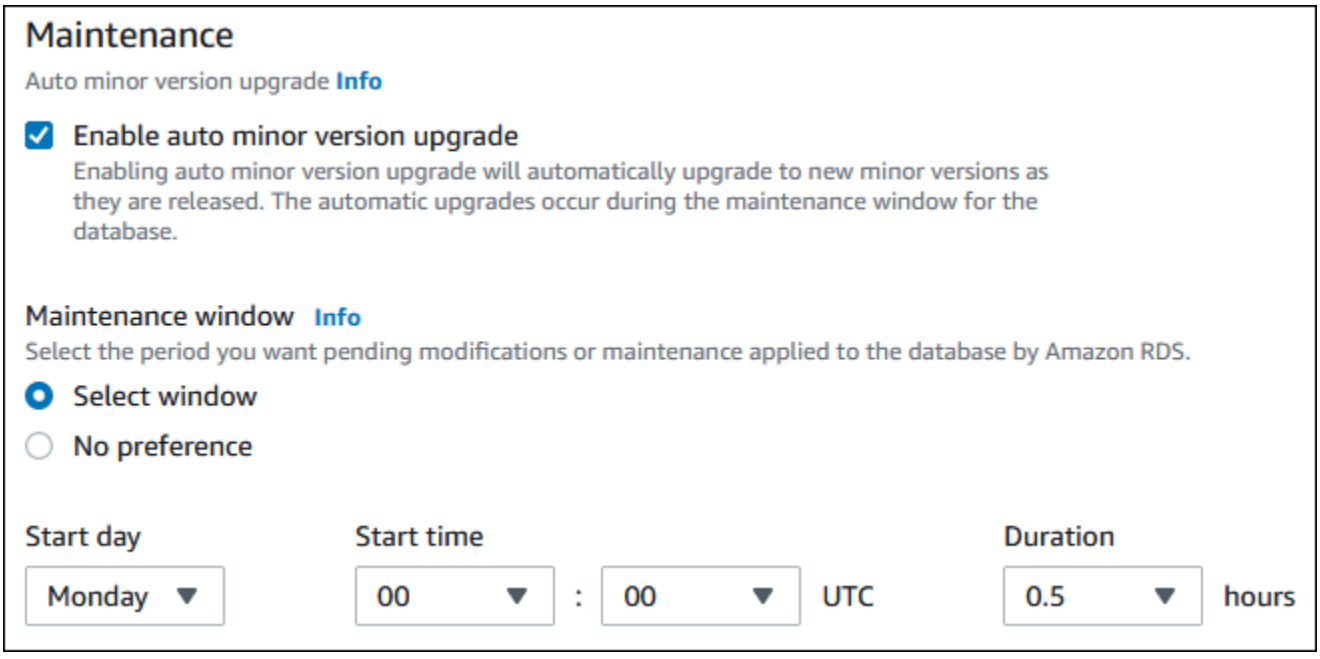
Para obter informações sobre como atualizar manual ou automaticamente uma instância de banco de dados MySQL, consulte Atualizar a versão de mecanismo de uma instância de banco de dados.

Atualizações automáticas da versão secundária do RDS para MySQL

Se você especificar as seguintes configurações ao criar ou modificar uma instância de banco de dados, poderá fazer com que sua instância de banco de dados seja atualizada automaticamente.

- A configuração Auto minor version upgrade (Atualização automática de versão secundária) está habilitada.
- A configuração Backup retention period (Período de retenção de backup) é maior que 0.

No AWS Management Console, essas configurações estão em Additional configuration (Configuração adicional). A imagem a seguir mostra a configuração Auto minor version upgrade (Atualização automática para versão secundária).



Para ter mais informações sobre essas configurações, consulte Configurações para instâncias de banco de dados.

Para algumas versões principais do RDS para MySQL em algumas Regiões da AWS, uma versão secundária é designada pelo RDS como a versão de atualização automática. Depois que uma versão secundária foi testada e aprovada pelo Amazon RDS, a atualização de versão secundária ocorre automaticamente durante a janela de manutenção. O RDS não define automaticamente versões secundárias lançadas mais recentemente como a versão de atualização automática. Antes de o RDS designar uma versão de atualização automática, diversos critérios são considerados, como estes:

- Problemas de segurança conhecidos
- Bugs na versão da comunidade do MySQL
- Estabilidade geral de frota desde que a versão secundária foi lançada.

É possível executar o comando da AWS CLI a seguir para determinar a versão atual secundária de destino de atualização automática para uma versão secundária do MySQL especificada em uma Região da AWS específica.

Para Linux, macOS ou Unix:

```
aws rds describe-db-engine-versions \
--engine mysql \
--engine-version minor_version \
--region region \
--query "DBEngineVersions[*].ValidUpgradeTarget[*].
{AutoUpgrade:AutoUpgrade,EngineVersion:EngineVersion}" \
--output text
```

Para Windows:

```
aws rds describe-db-engine-versions ^
--engine mysql ^
--engine-version minor_version ^
--region region ^
--query "DBEngineVersions[*].ValidUpgradeTarget[*].
{AutoUpgrade:AutoUpgrade,EngineVersion:EngineVersion}" ^
--output text
```

Por exemplo, o comando da AWS CLI a seguir determina o destino de atualização secundária automática para a versão 8.0.11 secundária do MySQL na Região da AWS Leste dos EUA (Ohio) (us-east-2).

Para Linux, macOS ou Unix:

```
aws rds describe-db-engine-versions \
--engine mysql \
--engine-version 8.0.11 \
--region us-east-2 \
--query "DBEngineVersions[*].ValidUpgradeTarget[*].
{AutoUpgrade:AutoUpgrade,EngineVersion:EngineVersion}" \
--output table
```

Para Windows:

```
aws rds describe-db-engine-versions ^
--engine mysql ^
--engine-version 8.0.11 ^
--region us-east-2 ^
--query "DBEngineVersions[*].ValidUpgradeTarget[*].
{AutoUpgrade:AutoUpgrade,EngineVersion:EngineVersion}" ^
--output table
```

A saída é semelhante à seguinte.

```
---------------------------------
|    DescribeDBEngineVersions    |
+--------------+-----------------+
|  AutoUpgrade |  EngineVersion  |
+--------------+-----------------+
|  False       |  8.0.15         |
|  False       |  8.0.16         |
|  False       |  8.0.17         |
|  False       |  8.0.19         |
|  False       |  8.0.20         |
|  False       |  8.0.21         |
|  True        |  8.0.23         |
|  False       |  8.0.25         |
+--------------+-----------------+
```

Neste exemplo, o valor de `AutoUpgrade` é `True` para o MySQL versão 8.0.23. Portanto, o destino de upgrade secundário automático é o MySQL versão 8.0.23, que está realçado na saída.

Uma instância de banco de dados MySQL é atualizada automaticamente durante a janela de manutenção, caso os seguintes critérios sejam atendidos:

- A configuração Auto minor version upgrade (Atualização automática de versão secundária) está habilitada.
- A configuração Backup retention period (Período de retenção de backup) é maior que 0.
- A instância de banco de dados está executando uma versão de mecanismo de banco de dados secundária que é menor que a versão secundária de atualização automática.

Para obter mais informações, consulte Atualizar automaticamente a versão do mecanismo espelho.

Usar uma réplica de leitura para reduzir o tempo de inatividade ao atualizar um banco de dados do RDS para MySQL

Na maioria dos casos, uma implantação azul/verde é a melhor opção para reduzir o tempo de inatividade ao atualizar uma instância de banco de dados MySQL. Para ter mais informações, consulte Usar implantações azul/verde do Amazon RDS para atualizações de banco de dados.

Se não for possível usar uma implantação azul/verde e se sua instância de banco de dados MySQL estiver em uso com uma aplicação de produção, você poderá usar o procedimento a seguir para atualizar a versão de banco de dados da sua instância de banco de dados. Esse procedimento pode reduzir o tempo de inatividade do seu aplicativo.

Usando uma réplica de leitura, você pode executar a maioria das etapas de manutenção antes do tempo e minimizar as alterações necessárias durante a interrupção real. Com essa técnica, você pode testar e preparar a nova instância de banco de dados sem fazer alterações em sua instância de banco de dados existente.

O procedimento a seguir mostra um exemplo de atualização do MySQL versão 5.7 para o MySQL versão 8.0. Você pode usar as mesmas etapas gerais nas atualizações para outras versões principais. Você pode usar as mesmas etapas gerais nas atualizações para outras versões principais.

> **Note**
>
> Quando você estiver atualizando do MySQL versão 5.7 para o MySQL versão 8.0 ou do MySQL versão 8.0 para o MySQL versão 8.4, conclua as pré-verificações antes de executar a atualização. Para ter mais informações, consulte Pré-verificações de upgrades do MySQL 5.7 a 8.0 e Pré-verificações de atualizações do MySQL 8.0 para 8.4.

Como atualizar um banco de dados MySQL enquanto uma instância de banco de dados está em uso

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. Crie uma réplica de leitura da instância de banco de dados MySQL 5.7. Esse processo cria uma cópia atualizável do seu banco de dados. Outras réplicas de leitura da instância de banco de dados também podem existir.

   a. No console, escolha Databases (Bancos de dados) e depois escolha a instância de banco de dados que você deseja atualizar.

   b. Em Actions (Ações), escolha Create read replica (Criar réplica de leitura).

   c. Forneça o valor do DB instance identifier (Identificador de instância de banco de dados) para a réplica de leitura e verifique se a DB instance class (Classe da instância de banco de dados) e as outras configurações correspondem à instância de banco de dados MySQL 5.7.

   d. Escolha Create read replica (Criar réplica de leitura).
3. (Opcional) Quando a réplica de leitura for criada e Status for exibida Available (Disponível), converta a réplica de leitura em uma implantação multi-AZ e ative backups.

   Por padrão, uma réplica de leitura é criada com backups desabilitados. Como a réplica de leitura se torna a instância de banco de dados de produção, uma prática recomendada é configurar uma implantação multi-AZ e habilitar backups.

   a. No console, selecione Databases (Bancos de dados) e a réplica de leitura que você acabou de criar.

   b. Selecione Modify.

   c. Para Multi-AZ deployment (Implantação multi-AZ), escolha Create a standby instance (Criar uma instância em espera).

d. Em Backup Retention Period (Período de retenção de backup), escolha um valor positivo diferente de zero, como 3 dias, depois selecione Continue (Continuar).

e. Em Scheduling of modifications (Programação de modificações), selecione Apply immediately (Aplicar imediatamente).

f. Selecione Modify DB instance (Modificar instância de banco de dados).

4. Quando o Status da réplica de leitura mostrar Available (Disponível), atualize a réplica de leitura para o MySQL 8.0:

   a. No console, selecione Databases (Bancos de dados) e a réplica de leitura que você acabou de criar.

   b. Selecione Modify.

   c. Em DB engine version (Versão do mecanismo de banco de dados), selecione a versão do MySQL 8.0 para a qual atualizar e, em seguida, selecione Continue (Continuar).

   d. Em Scheduling of modifications (Programação de modificações), selecione Apply immediately (Aplicar imediatamente).

   e. Selecione Modify DB instance (Modificar instância de banco de dados) para iniciar a atualização.

5. Quando a atualização for concluída e o Status mostrar Disponível, verifique se a réplica de leitura foi atualizada com a instância de banco de dados MySQL 5.7 de origem. Para verificar, conecte-se à réplica de leitura e execute o comando `SHOW REPLICA STATUS`. Se o campo `Seconds_Behind_Master` for `0`, a replicação está atualizada.

   > **Note**
   >
   > As versões anteriores do MySQL usavam `SHOW SLAVE STATUS` em vez de `SHOW REPLICA STATUS`. Se você estiver usando uma versão do MySQL anterior à 8.0.23, use `SHOW SLAVE STATUS`.

6. (Opcional) Crie uma réplica de leitura da réplica de leitura.

   Se você quiser que a instância de banco de dados tenha uma réplica de leitura depois que ela for promovida para uma instância de banco de dados autônoma, você pode criar a réplica de leitura agora.

   a. No console, selecione Databases (Bancos de dados) e selecione a réplica de leitura que você acabou de atualizar.

b. Em Actions (Ações), escolha Create read replica (Criar réplica de leitura).

c. Forneça o valor do DB instance identifier (Identificador de instância de banco de dados) para a réplica de leitura e verifique se a DB instance class (Classe da instância de banco de dados) e as outras configurações correspondem à instância de banco de dados MySQL 5.7.

d. Escolha Create read replica (Criar réplica de leitura).

7. (Opcional) Configure um grupo de parâmetros de banco de dados personalizado para a réplica de leitura.

   Se você quiser que a instância de banco de dados use um grupo de parâmetros personalizado depois de ser promovida para uma instância de banco de dados autônoma, poderá criar o grupo de parâmetros de banco de dados agora pode associá-lo à réplica de leitura.

   a. Crie um grupo de parâmetros de banco de dados personalizado para o MySQL 8.0. Para obter instruções, consulte Criar um grupo de parâmetros de banco de dados no Amazon RDS.

   b. Modifique os parâmetros que você deseja alterar no grupo de parâmetros de banco de dados que você acabou de criar. Para obter instruções, consulte Modificar parâmetros em um grupo de parâmetros de banco de dados no Amazon RDS.

   c. No console, selecione Databases (Bancos de dados) e a réplica de leitura.

   d. Selecione Modify.

   e. Para o DB parameter group (Grupo de parâmetros do banco de dados), escolha o grupo de parâmetros do banco de dados MySQL 8.0 que você acabou de criar e, em seguida, escolha Continue (Continuar).

   f. Em Scheduling of modifications (Programação de modificações), selecione Apply immediately (Aplicar imediatamente).

   g. Selecione Modify DB instance (Modificar instância de banco de dados) para iniciar a atualização.

8. Torne sua réplica de leitura MySQL 8.0 uma instância de banco de dados autônoma.

   > **Important**
   >
   > Quando você promove a réplica de leitura do MySQL 8.0 para uma instância de banco de dados autônoma, ela deixa de ser uma réplica da instância de banco de dados MySQL 5.7. Recomendamos que você promova a replica de leitura do MySQL 8.0 durante uma janela de manutenção, quando a instância de banco de dados MySQL

5.7 estiver no modo somente leitura e todas as operações de gravação estiverem suspensas. Quando a promoção terminar, você poderá direcionar suas operações de gravação para a instância de banco de dados MySQL 8.0 atualizada, para garantir que nenhuma operação de gravação seja perdida.
Além disso, recomendamos que, antes de promover a réplica de leitura do MySQL 8.0, você realize todas as operações de DDL (linguagem de definição de dados) necessárias na réplica de leitura do MySQL 8.0. Um exemplo é a criação de índices. Essa abordagem evita efeitos negativos sobre a performance da réplica de leitura do MySQL 8.0 após a sua promoção. Para promover uma réplica de leitura, use o procedimento a seguir.

a. No console, selecione Databases (Bancos de dados) e selecione a réplica de leitura que você acabou de atualizar.
b. Em Actions (Ações), selecione Promote (Promover).
c. Selecione Yes (Sim) para habilitar backups automatizados da instância da réplica de leitura. Para obter mais informações, consulte Introdução aos backups.
d. Escolha Continue.
e. Escolha Promote Read Replica (Promover réplica de leitura).

9. Agora, você possui uma versão atualizada do seu banco de dados MySQL. Neste ponto, você pode direcionar as aplicações para a nova instância de banco de dados MySQL 8.0.

Atualizar uma versão do mecanismo de snapshot de banco de dados do MySQL

Com o Amazon RDS, você pode criar um snapshot de banco de dados do volume de armazenamento de sua instância de banco de dados MySQL. Ao criar um snapshot de banco de dados, ele é baseado na versão do mecanismo usada pela instância de banco de dados. É possível fazer a atualização da versão do mecanismo para os snapshots de banco de dados.

Para o RDS para MySQL, é possível atualizar um snapshot versão 5.7 para a versão 8.0 ou um snapshot versão 8.0 para a versão 8.4. É possível atualizar snapshots de banco de dados criptografados ou não criptografados.

Para visualizar as versões de mecanismo disponíveis para o snapshot de banco de dados do RDS para MySQL, use o exemplo da AWS CLI a seguir.

```
aws rds describe-db-engine-versions --engine mysql --include-all --engine-
version example-engine-version --query "DBEngineVersions[*].ValidUpgradeTarget[*].
{EngineVersion:EngineVersion}" --output text
```

Se você não vir resultados para o snapshot, a versão do seu mecanismo pode estar obsoleta. Se a versão do mecanismo tiver sido descontinuada, recomendamos que você atualize para o destino de atualização da versão principal mais recente ou para um dos outros destinos de atualização disponíveis para essa versão. Para ter mais informações, consulte Opções de atualização para snapshots de banco de dados com versões de mecanismo incompatíveis para o RDS para MySQL.

Após restaurar um snapshot de banco de dados atualizado para uma nova versão do mecanismo, certifique-se de testar se a atualização foi bem-sucedida. Para obter mais informações sobre uma atualização de versão principal, consulte the section called “Atualizações do mecanismo de banco de dados do MySQL”. Para saber como restaurar um snapshot de banco de dados, consulte the section called “Restaurar uma instância de banco de dados”.

> **Note**
>
> Não é possível atualizar snapshots de banco de dados automatizados criados durante o processo de backup automatizado.

É possível atualizar um snapshot de banco de dados usando o AWS Management Console, a AWS CLI ou a API do RDS.

Console

Para atualizar uma versão do mecanismo de snapshot de banco de dados usando o AWS Management Console, use o procedimento a seguir.

Para atualizar um snapshot de banco de dados

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, selecione Snapshots.
3. Escolha o snapshot que você deseja atualizar.
4. Em Actions (Ações), selecione Upgrade snapshot (Atualizar snapshot). A página Upgrade snapshot (Atualizar snapshot) é exibida.
5. Escolha a opção New engine version (Nova versão de mecanismo) para a qual atualizar.
6. Escolha Save changes (Salvar alterações) para atualizar o snapshot.

   Durante o processo de atualização, todas as ações do snapshot serão desabilitadas para esse snapshot de banco de dados. Além disso, o status do snapshot de banco de dados muda de Disponível para Fazendo upgrade e, depois, muda para Ativo após a conclusão. Se não for possível atualizar o snapshot de banco de dados devido a problemas de corrupção do snapshot, o status mudará para Indisponível. Não é possível recuperar o snapshot desse estado.

   > **Note**
   >
   > Se a atualização do snapshot de banco de dados falhar, o snapshot será revertido para o estado original com a versão original.

AWS CLI

Para atualizar um snapshot de banco de dados para uma nova versão do mecanismo de banco de dados, execute o comando modify-db-snapshot da AWS CLI.

Opções

- `--db-snapshot-identifier` – o identificador do snapshot de banco de dados a ser atualizado. O identificador deve ser um nome de recurso da Amazon (ARN) exclusivo. Para obter mais informações, consulte Nomes de recurso da Amazon (ARNs) no Amazon RDS.

- `--engine-version` – a versão do mecanismo para a qual será feita a atualização do snapshot de banco de dados.

Example

Para Linux, macOS ou Unix:

```
aws rds modify-db-snapshot \

    --db-snapshot-identifier my_db_snapshot \
    --engine-version new_version
```

Para Windows:

```
aws rds modify-db-snapshot ^
    --db-snapshot-identifier my_db_snapshot ^
    --engine-version new_version
```

Amazon RDS API

Para atualizar um snapshot de banco de dados para uma nova versão do mecanismo de banco de dados, chame a operação ModifyDBSnapshot da API do RDS.

Parâmetros

- `DBSnapshotIdentifier` – o identificador do snapshot de banco de dados a ser atualizado. O identificador deve ser um nome de recurso da Amazon (ARN) exclusivo. Para obter mais informações, consulte Nomes de recurso da Amazon (ARNs) no Amazon RDS.
- `EngineVersion` – a versão do mecanismo para a qual será feita a atualização do snapshot de banco de dados.

Opções de atualização para snapshots de banco de dados com versões de mecanismo incompatíveis para o RDS para MySQL

A tabela a seguir mostra para quais versões do mecanismo é possível fazer a atualização de uma versão de mecanismo incompatível com snapshots de banco de dados do RDS para MySQL.

> **Note**
>
> Talvez você precise atualizar o snapshot de banco de dados mais de uma vez para fazer a atualização para a versão do mecanismo escolhida.

Versão do mecanismo do snapshot de banco de dados	Versões do mecanismo disponíveis para atualização
5.5.8	5.5.62, 5.6.51
5.5.12	5.5.62, 5.6.51
5.5.20	5.5.62, 5.6.51
5.5.23	5.5.62, 5.6.51
5.5.25a	5.5.62, 5.6.51
5.5.27	5.5.62, 5.6.51
5.5.31	5.5.62, 5.6.51
5.5.33	5.5.62, 5.6.51
5.5.37	5.5.62, 5.6.51
5.5.38	5.5.62, 5.6.51
5.5.40	5.5.62, 5.6.51
5.5.40a	5.5.62, 5.6.51
5.5.40b	5.5.62, 5.6.51
5.5.41	5.5.62, 5.6.51
5.5.42	5.5.62, 5.6.51
5.5.59	5.5.62, 5.6.51

Versão do mecanismo do snapshot de banco de dados	Versões do mecanismo disponíveis para atualização
5.6.12	5.6.51, 5.7.44
5.6.13	5.6.51, 5.7.44
5.6.17	5.6.51, 5.7.44
5.6.19	5.6.51, 5.7.44
5.6.19a	5.6.51, 5.7.44
5.6.19b	5.6.51, 5.7.44
5.6.21	5.6.51, 5.7.44
5.6.21b	5.6.51, 5.7.44
5.6.22	5.6.51, 5.7.44
5.6.23	5.6.51, 5.7.44
5.6.27	5.6.51, 5.7.44
5.6.27a	5.6.51, 5.7.44
5.7.10	5.7.44, 5.7.44-rds.20240408, 5.7.44-rds.20240529, 5.7.44-rds.20250103, 5.7.44-rds.20250213, 8.0.32, 8.0.33, 8.0.34, 8.0.35, 8.0.36, 8.0.37, 8.0.39, 8.0.40, 8.0.41
5.7.11	5.7.44, 5.7.44-rds.20240408, 5.7.44-rds.20240529, 5.7.44-rds.20250103, 5.7.44-rds.20250213, 8.0.32, 8.0.33, 8.0.34, 8.0.35, 8.0.36, 8.0.37, 8.0.39, 8.0.40, 8.0.41
5.7.12	5.7.44, 5.7.44-rds.20240408, 5.7.44-rds.20240529, 5.7.44-rds.20250103, 5.7.44-rds.20250213, 8.0.32, 8.0.33, 8.0.34, 8.0.35, 8.0.36, 8.0.37, 8.0.39, 8.0.40, 8.0.41

Importar dados para uma instância de banco de dados do Amazon RDS para MySQL.

Você pode usar várias técnicas diferentes para importar os dados para uma instância de banco de dados do RDS para MySQL. A melhor abordagem depende de inúmeros fatores:

- origem dos dados;
- Quantidade de dados
- importação única ou contínua;
- quantidade de tempo de inatividade.

Se você também estiver migrando uma aplicação com os dados, é importante considerar a quantidade de tempo de inatividade.

A seguinte tabela lista técnicas para importar dados em uma instância de banco de dados do RDS para MySQL:

Origem	Quantidade de dados	Uma vez ou contínuo	Tempo de inatividade da aplicação	Técnica	Mais informações
Banco de dados MySQL existente no local ou no Amazon EC2	Quaisquer	Uma vez	Alguns	Crie um backup de seu banco de dados no local, armazene-o no Amazon S3 e, em seguida, restaure o arquivo de backup para uma nova instância de banco de dados do Amazon RDS executando MySQL.	Restauração de um backup em uma instância de banco de dados MySQL

Origem	Quantidade de dados	Uma vez ou contínuo	Tempo de inatividade da aplicação	Técnica	Mais informações
Banco de dados MySQL existente no local ou no Amazon EC2	Quaisquer	Contínuo	Mínimo	Configure a replicação com um banco de dados MySQL existente como origem de replicação.	Configurar a replicação da posição do arquivo de log binário com uma instância de origem externa Importar dados para um banco de dados do Amazon RDS para MySQL com tempo de inatividade

Origem	Quantidade de dados	Uma vez ou contínuo	Tempo de inatividade da aplicação	Técnica	Mais informações
					de reduzido

Origem	Quantidade de dados	Uma vez ou contínuo	Tempo de inatividade da aplicação	Técnica	Mais informações
Qualquer banco de dados existente	Quaisquer	Uma vez ou contínuo	Mínimo	Use o AWS Database Migration Service para migrar o banco de dados com o mínimo de tempo de inatividade e, para diversos mecanismos de banco de dados (DB), continuar a replicação contínua.	What is AWS Database Migration Service (O que é o) e Using a MySQL-compatible database as a target for AWS DMS (Usar um banco de dados compatível com MySQL como destino para o) no Guia do

Origem	Quantidade de dados	Uma vez ou contínuo	Tempo de inatividade da aplicação	Técnica	Mais informações
					usuário do AWS Database Migration Service
Instância de banco de dados MySQL existente	Quaisquer	Uma vez ou contínuo	Mínimo	Crie uma réplica de leitura para a replicação contínua. Promova a réplica de leitura para a criação única de uma instância de banco de dados.	Trabalhar com réplicas de leitura de instância de banco de dados

Origem	Quantidade de dados	Uma vez ou contínuo	Tempo de inatividade da aplicação	Técnica	Mais informações
Banco de dados existente do MySQL	Pequeno	Uma vez	Alguns	Copie os dados diretamente para sua instância de banco de dados MySQL usando um utilitário de linha de comando.	Importar dados de um banco de dados externo do MySQL para uma instância de banco de dados do Amazon RDS para MySQL.

Origem	Quantidade de dados	Uma vez ou contínuo	Tempo de inatividade da aplicação	Técnica	Mais informações
Dados não armazenados em um banco de dados existente	Médio	Uma vez	Alguns	Crie arquivos simples e importe-os usando instruções MySQL `LOAD DATA LOCAL INFILE`.	Importar dados de qualquer fonte para uma instância de banco de dados do Amazon RDS para MySQL

> **Note**
>
> O banco de dados do sistema `mysql` contém informações de autenticação e autorização necessárias para fazer login na instância de banco de dados e acessar os dados. Soltar, alterar, renomear ou truncar tabelas, dados ou outros conteúdos do banco de dados `mysql` em sua instância de banco de dados pode resultar em erros e pode tornar a instância de banco de dados e seus dados inacessíveis. Se isso ocorrer, você pode restaurar a instância de banco de dados a partir de um snapshot usando o comando AWS CLI da restore-db-instance-from-db-snapshot. Você pode recuperar a instância de banco de dados usando o comando AWS CLI da restore-db-instance-to-point-in-time.

Considerações sobre importação de dados para o MySQL

O conteúdo a seguir apresenta informações técnicas relacionadas ao carregamento de dados no MySQL. Este conteúdo é destinado a usuários que estão familiarizados com a arquitetura do servidor MySQL.

registro em log binário

A habilitação do registro em log binário reduz a performance do carregamento de dados e requer até quatro vezes mais espaço em disco em comparação com o registro em log desabilitado. O tamanho da transação usada para carregar os dados afeta diretamente a performance do sistema e as necessidades de espaço em disco. As transações maiores exigem mais recursos.

Tamanho da transação

O tamanho da transação influencia os seguintes aspectos dos carregamentos de dados do MySQL:

- Consumo de recursos
- Utilização de espaço em disco
- Processo de retomada
- Tempo de recuperação
- Formato de entrada (arquivos simples ou SQL)

Esta seção descreve como o tamanho da transação afeta o registro em log binário e justifica a desabilitação do registro em log binário durante grandes cargas de dados. Você pode habilitar e desabilitar o registro em log binário configurando o período de retenção de backup automatizado do Amazon RDS. Valores diferentes de zero habilitam o registro em log binário, enquanto um valor de zero o desabilita. Para ter mais informações, consulte Backup retention period (Período de retenção de backup).

Esta seção descreve também o impacto de grandes transações sobre o InnoDB e por que é importante que o tamanho das transações seja pequeno.

Transações pequenas

Para pequenas transações, o registro em log binário duplica o número de gravações em disco necessárias para carregar os dados. Esse efeito pode degradar severamente a performance de outras sessões de banco de dados e aumentar o tempo necessário para carregar os dados. A degradação experimentada depende em parte dos seguintes fatores:

- Taxa de upload
- Outra atividade do banco de dados que ocorre durante o carregamento
- Capacidade da instância de banco de dados do Amazon RDS

Os logs binários também consomem um espaço em disco aproximadamente igual à quantidade de dados carregados enquanto os logs são copiados e removidos. O Amazon RDS minimiza isso fazendo backup e removendo os logs binários com frequência.

Transações grandes

Quanto a transações grandes, o registro em log binário triplica as IOPS e o uso do disco pelos seguintes motivos:

- O cache de log binário armazena os dados da transação temporariamente no disco.
- Esse cache se amplia de acordo com o tamanho da transação, o que consome espaço em disco.
- Quando a transação (de confirmação ou reversão) é concluída, o sistema copia o cache para o log binário.

Esse processo cria três cópias dos dados:

- os dados originais;
- o cache no disco;
- a entrada final do log binário.

Cada operação de gravação incorre em E/S adicional, afetando ainda mais a performance.

Por isso, o registro em log binário requer o triplo de espaço em disco em comparação com o registro desabilitado. Por exemplo, carregar 10 GiB de dados como uma única transação cria três cópias:

- 10 GiB para os dados da tabela;
- 10 GiB para o cache de log binário;
- 10 GiB para o arquivo de log binário.

O total de espaço em disco temporário necessário é 30 GiB.

Considerações importantes sobre o espaço em disco:

- O arquivo de cache persiste até que a sessão termine ou uma nova transação crie outro cache.
- O log binário permanece até o momento em que ele é copiado, possivelmente mantendo 20 GiB (cache e log) por um período prolongado.

Se você usar `LOAD DATA LOCAL INFILE` para carregar os dados, a recuperação de dados criará uma quarta cópia caso o banco de dados tenha que ser recuperado de um backup feito antes do carregamento. Durante a recuperação, o MySQL extrai dados de log binário em um arquivo simples. Em seguida, o MySQL executa `LOAD DATA LOCAL INFILE`. Com base no exemplo anterior, essa recuperação requer um total de espaço em disco temporário de 40 GiB ou 10 GiB cada para tabela, cache, log e arquivo local. Sem pelo menos 40 GiB de espaço livre em disco, a recuperação falhará.

Otimizar grandes carregamentos de dados

No caso de grandes carregamentos de dados, desabilite o registro em log binário para reduzir os custos indiretos e os requisitos de espaço em disco. Você pode desabilitar o registro em log binário definindo o período de retenção de backup como 0. Após o término do carregamento, substitua o período de retenção de backup por um valor apropriado diferente de zero. Para ter mais informações, consulte Modificar uma instância de banco de dados do Amazon RDS e Período de retenção de backup na tabela de configurações.

> **Note**
>
> Se a instância de banco de dados for uma instância de banco de dados de origem para réplicas de leitura, não será possível definir o período de retenção de backup como 0.

Antes de carregar os dados, recomendamos criar um snapshot do banco de dados. Para ter mais informações, consulte Gerenciar backups manuais.

InnoDB

As informações a seguir sobre o registro em log de desfazer e as opções de recuperação ajudam a manter as transações do InnoDB pequenas para otimizar a performance do banco de dados.

Conceitos básicos sobre o registro em log de desfazer no InnoDB

Desfazer é um mecanismo de registro em log que permite a reversão de transações e comporta o controle de simultaneidade de várias versões (MVCC).

No caso do MySQL 5.7 e versões anteriores, os logs de desfazer são armazenados no espaço de tabela do sistema InnoDB (geralmente ibdata1) e são retidos até que o thread de limpeza os remova. Por esse motivo, grandes transações de carregamento de dados podem fazer com que o espaço de tabela do sistema se torne muito grande e consuma espaço em disco que não pode ser recuperado sem recriar o banco de dados.

Para todas as versões do MySQL, o thread de limpeza só deve remover quaisquer logs de desfazer depois que a transação ativa mais antiga for confirmada ou revertida. Se o banco de dados estiver processando outras transações durante o carregamento, os logs de desfazer também se acumularão e não poderão ser removidos, mesmo que as transações sejam confirmadas e nenhuma outra precise dos logs de desfazer para o MVCC. Nessa situação, todas as transações, incluindo as transações somente leitura, ficam mais lentas. Essa desaceleração ocorre porque todas as transações acessam todas as linhas que qualquer transação, não apenas a transação de carregamento, altera. Na prática, as transações devem examinar os logs de desfazer cuja eliminação foi impedida pelas transações de carregamento de longa duração durante uma limpeza de logs de desfazer. Isso afeta a performance de qualquer operação que acesse linhas modificadas.

Opções de recuperação de transações do InnoDB

Embora o InnoDB otimize as operações de confirmação, a reversão de grandes transações é lenta. Para uma recuperação mais rápida, execute uma recuperação para um ponto no tempo ou restaure um snapshot do banco de dados. Para obter mais informações, consulte Recuperação para um ponto no tempo e Restaurar uma instância de banco de dados.

Formatos de importação de dados

O MySQL aceita dois formatos de importação de dados: arquivos simples e SQL. Analise as informações sobre cada formato para determinar a melhor opção para suas necessidades.

Arquivos simples

Para transações pequenas, carregue arquivos simples com `LOAD DATA LOCAL INFILE`. Esse formato de importação de dados pode oferecer os seguintes benefícios em relação ao uso do SQL:

- regras de tráfego de rede;
- custos de transmissão de dados mais baixos;
- redução dos custos indireto de processamento do banco de dados;
- processamento de fluxos.

`LOAD DATA LOCAL INFILE` carrega todo o arquivo simples como uma única transação. Mantenha os arquivos individuais com um tamanho pequeno para obter as seguintes vantagens:

- Capacidade de retomar: é fácil manter o controle de quais arquivos foram carregados. Se surgir um problema durante o carregamento, você poderá continuar de onde parou. Você pode precisar retransmitir alguns dados ao Amazon RDS, mas, com arquivos pequenos, a quantidade retransmitida é mínima.
- Carregamento de dados em paralelo: se você tiver IOPS e largura de banda da rede suficientes para carregamento de um único arquivo, o carregamento em paralelo pode economizar tempo.
- Controle da taxa de carregamento: se o carregamento de dados tiver um impacto negativo em outros processos, você poderá controlar a taxa de carregamento aumentando o intervalo entre os arquivos.

Grandes transações reduzem os benefícios de usar `LOAD DATA LOCAL INFILE` para importar dados. Quando você não conseguir dividir uma grande quantidade de dados em arquivos menores, considere a possibilidade de usar o SQL.

SQL

O SQL tem uma vantagem fundamental em relação aos arquivos simples: é fácil manter as transações com um tamanho pequeno. Contudo, o tempo de carregamento do SQL pode ser significativamente mais longo do que o dos arquivos simples. Além disso, após uma falha, pode ser difícil determinar onde retomar. Não é possível reiniciar os arquivos mysqldump. Se ocorrer uma falha ao carregar um arquivo mysqldump, você deve modificá-lo ou substituí-lo para que o carregamento possa continuar. Ou, alternativamente, depois de corrigir a causa da falha, você pode restaurar para o ponto no tempo anterior ao carregamento e reenviar o arquivo. Para ter mais informações, consulte Recuperação para um ponto no tempo.

Usar snapshots de banco de dados do Amazon RDS para pontos de verificação de banco de dados

Se você carrega dados por longos períodos, como horas ou dias, sem registro em log binário, use snapshots de banco de dados para oferecer pontos de verificação periódicos em prol da segurança dos dados. Cada snapshot de banco de dados cria uma cópia consistente da instância de banco de dados, que serve como ponto de recuperação durante falhas do sistema ou eventos de corrupção de dados. Como os snapshots de banco de dados são rápidos, os pontos de verificação frequentes têm um impacto mínimo na performance do carregamento. Você pode excluir snapshots de banco de

dados anteriores sem afetar a durabilidade ou os recursos de recuperação do banco de dados. Para ter mais informações sobre snapshots de banco de dados, consulte Gerenciar backups manuais.

Reduzir os tempos de carregamento do banco de dados

Os seguintes itens são dicas adicionais para reduzir os tempos de carregamento:

- Crie todos os índices secundários antes de carregar dados nos bancos de dados MySQL. Ao contrário de outros sistemas de banco de dados, o MySQL reconstrói a tabela inteira ao adicionar ou modificar índices secundários. Esse processo cria outra tabela com alterações no índice, copia todos os dados e descarta a tabela original.
- Carregue os dados na ordem da chave primária. Quanto a tabelas do InnoDB, isso pode reduzir os tempos de carregamento em 75% a 80% e diminuir o tamanho do arquivo de dados em 50%.
- Desabilite as restrições de chave estrangeira definindo `foreign_key_checks` como `0`. Isso com frequência é necessário para arquivos simples carregados com `LOAD DATA LOCAL INFILE`. Com relação a qualquer carregamento, desabilitar as verificações de chave estrangeira acelera o carregamento de dados. Após a conclusão do carregamento, habilite novamente as restrições configurando `foreign_key_checks` como 1 e verifique os dados.
- Carregue os dados em paralelo, a menos que esteja se aproximando de um limite de recursos. Para habilitar o carregamento simultâneo em vários segmentos da tabela, use tabelas particionadas quando apropriado.
- Para reduzir os custos indiretos de execução do SQL, combine várias instruções `INSERT` em operações `INSERT` únicas de vários valores. `mysqldump` implementa essa otimização automaticamente.
- Reduza as operações de E/S de log do InnoDB configurando `innodb_flush_log_at_trx_commit` como `0`. Após a conclusão do carregamento, restaure `innodb_flush_log_at_trx_commit` para 1.

> **Warning**
>
> Ao definir `innodb_flush_log_at_trx_commit` como `0`, o InnoDB descarrega os respectivos logs a cada segundo em vez de a cada confirmação. Essa configuração aumenta a performance, mas pode gerar risco de perda de transações durante falhas no sistema.

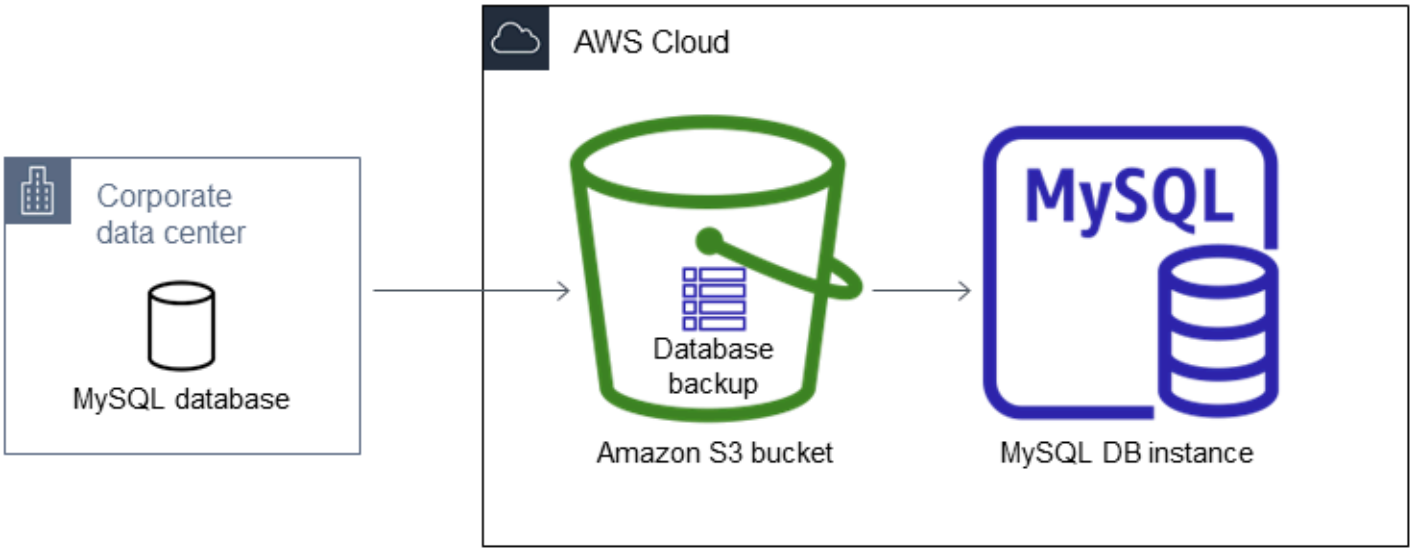
- Se você estiver carregando dados em uma instância de banco de dados que não tem réplicas de leitura, defina `sync_binlog` como `0`. Após a conclusão do carregamento, restaure `sync_binlog parameter` para `1`.
- Carregue os dados antes de converter a instância de banco de dados em uma implantação multi-AZ. Se a instância de banco de dados já utiliza uma implantação multi-AZ, não recomendamos mudar para uma implantação single-AZ para carregamento de dados. Se fizer isso, obterá apenas pequenas melhorias.

Restauração de um backup em uma instância de banco de dados MySQL

O Amazon RDS oferece suporte à importação de bancos de dados MySQL usando arquivos de backup. Você pode criar um backup do banco de dados, armazená-lo no Amazon S3 e, depois, restaurar o arquivo de backup em uma nova instância de banco de dados do Amazon RDS que executa o MySQL.

O cenário descrito nesta seção restaura um backup de um banco de dados on-premises. Você pode usar essa técnica para bancos de dados em outros locais, como o Amazon EC2 ou serviços de nuvem que não são da AWS, desde que o banco de dados esteja acessível.

Você pode localizar o cenário compatível no diagrama a seguir.



A importação de arquivos de backup do Amazon S3 é compatível com o MySQL em todas as Regiões da AWS.

Recomendamos que você importe o banco de dados para o Amazon RDS usando arquivos de backup se o banco de dados on-premises estiver offline enquanto o arquivo de backup for criado,

copiado e restaurado. Se não for possível deixar o banco de dados offline, use a replicação de log binário (binlog) para atualizar o banco de dados depois de migrar para o Amazon RDS por meio do Amazon S3, conforme explicado neste tópico. Para ter mais informações, consulte Configurar a replicação da posição do arquivo de log binário com uma instância de origem externa. Também é possível usar o AWS Database Migration Service para migrar seu banco de dados para o Amazon RDS. Para ter mais informações, consulte O que é o AWS Database Migration Service?

Limitações e recomendações para importar arquivos de backup do Amazon S3 para o Amazon RDS

Veja a seguir algumas limitações e recomendações para importar arquivos de backup do Amazon S3:

- Você só pode importar os dados para uma nova instância de banco de dados, não para uma instância existente.
- É necessário usar o Percona XtraBackup para criar o backup de seu banco de dados no local.
- Você não pode importar dados de uma exportação de snapshot de banco de dados para o Amazon S3.
- Não é possível migrar de um banco de dados de origem que tenha tabelas definidas fora do diretório de dados MySQL padrão.
- O Percona Server para MySQL não é aceito como banco de dados de origem porque ele pode conter tabelas `compression_dictionary*` no esquema `mysql`.
- Você deve importar seus dados para a versão secundária padrão da versão principal do MySQL em sua Região da AWS. Por exemplo, se a versão principal for MySQL 8.0, e a versão secundária padrão de sua Região da AWS for 8.0.28, importe os dados para uma instância de banco de dados do MySQL versão 8.0.28. Você pode atualizar sua instância de banco de dados após a importação. Para obter informações sobre como determinar a versão secundária padrão, consulte MySQL em versões do Amazon RDS.
- Reverter migrações não é uma ação compatível com versões principais e secundárias. Por exemplo, você não pode migrar da versão 8.0 para a versão 5.7 e não pode migrar da versão 8.0.32 para a versão 8.0.31.
- Não é possível importar um banco de dados do MySQL 5.5 ou 5.6.
- Não é possível importar um banco de dados MySQL on-premises de uma versão principal para outra. Por exemplo, não é possível importar um banco de dados do MySQL 5.7 para um banco de dados do RDS para MySQL 8.0. Você pode atualizar sua instância de banco de dados depois de concluir a importação.

- Não é possível restaurar de um banco de dados de origem criptografado, mas você pode restaurar para uma instância de banco de dados do Amazon RDS.
- Não é possível fazer a restauração de um backup criptografado no bucket do Amazon S3.
- Você não pode restaurar de um bucket do Amazon S3 em uma Região da AWS diferente da instância de banco de dados do Amazon RDS.
- A importação do Amazon S3 não tem suporte na classe de Instância de Banco de Dados db.t2.micro. Contudo, é possível restaurar para outra classe de instância de banco de dados e alterar a instância de banco de dados posteriormente. Para mais informações sobre as classes da instância, consulte Especificações de hardware para classes de instância de banco de dados .
- O Amazon S3 limita o tamanho de um arquivo carregado para um bucket do Amazon S3 a 5 TB. Se um arquivo de backup exceder 5 TB, você deverá dividi o arquivo de backup em arquivos menores.
- Quando você restaura o banco de dados, o backup é copiado e extraído na instância de banco de dados. Portanto, provisione espaço de armazenamento para a instância de banco de dados igual ou maior que a soma do tamanho do backup, além do tamanho do banco de dados original no disco.
- O Amazon RDS limita a 1 milhão o número de arquivos carregados para um bucket do Amazon S3. Se os dados de backup do banco de dados, incluindo todos os backups completos e incrementais, exceder 1 milhão de arquivos, use um arquivo Gzip (.gz), tar (.tar.gz) ou Percona xbstream (.xbstream) para armazenar arquivos de backup completos e incrementais no bucket do Amazon S3. O Percona XtraBackup 8.0 oferece suporte apenas ao Percona xbstream para compactação.
- As contas de usuário não são importadas automaticamente. Salve suas contas de usuário de seu banco de dados de origem e adicione-as à sua nova instância de banco de dados posteriormente.
- As funções não são importadas automaticamente. Salve suas funções de seu banco de dados de origem e adicione-as à sua nova instância de banco de dados posteriormente.
- Os procedimentos armazenados não são importados automaticamente. Salve seus procedimentos armazenados de seu banco de dados de origem e adicione-o à sua nova instância de banco de dados posteriormente.
- As informações de fuso horário não são importadas automaticamente. Registre as informações de fuso horário de seu banco de dados de origem e defina o fuso horário de sua nova instância de banco de dados posteriormente. Para ter mais informações, consulte Fuso horário local para instâncias de banco de dados MySQL.

- O parâmetro `innodb_data_file_path` deve ser configurado com apenas um arquivo de dados que usa o nome de arquivo de dados padrão `"ibdata1:12M:autoextend"`. Bancos de dados com dois arquivos de dados ou com um arquivo de dados com um nome diferente não podem ser migrados usando esse método.

  Veja a seguir, exemplos de nomes de arquivos que não são permitidos: `"innodb_data_file_path=ibdata1:50M; ibdata2:50M:autoextend"` e `"innodb_data_file_path=ibdata01:50M:autoextend"`.
- O tamanho máximo do banco de dados restaurado é o tamanho máximo do banco de dados compatível menos o tamanho do backup. Portanto, se o tamanho máximo do banco de dados compatível for de 64 TiB e o tamanho do backup for de 30 TiB, o tamanho máximo do banco de dados restaurado será de 34 TiB, como no exemplo a seguir:

  ```
  64 TiB - 30 TiB = 34 TiB
  ```

  Para obter informações sobre o tamanho máximo do banco de dados compatível com o Amazon RDS para MySQL, consulte Armazenamento SSD de uso geral e Armazenamento SSD de IOPS provisionadas.

Visão geral da configuração para importar arquivos de backup do Amazon S3 para o Amazon RDS

Estes são os componentes necessários para configurar a importação de arquivos de backup do Amazon S3 para o Amazon RDS:

- Um bucket do Amazon S3 para armazenar seus arquivos de backup.
- Um backup de seu banco de dados no local criado pelo Percona XtraBackup.
- Uma função do AWS Identity and Access Management (IAM) para permitir que o Amazon RDS acesse o bucket.

Se você já tiver um bucket do Amazon S3, você pode usá-lo. Se você não tiver um bucket do Amazon S3, crie um novo. Se você quiser criar um novo bucket, consulte Criar um bucket.

Use a ferramenta Percona XtraBackup para criar seu backup. Para ter mais informações, consulte Criar o backup de banco de dados.

Se você já tiver uma função do IAM, você pode usá-la. Se você não tiver uma função do IAM, crie uma nova manualmente. Se preferir, crie uma nova função do IAM para você em sua conta por meio

do assistente ao restaurar o banco de dados usando o AWS Management Console. Se você quiser criar manualmente uma nova função do IAM ou anexar políticas de confiança e de permissões a uma função do IAM existente, consulte Criação manual de uma função do IAM. Caso queira ter uma nova função do IAM criada para você, siga o procedimento em Console.

Criar o backup de banco de dados

Use o software Percona XtraBackup para criar seu backup. É recomendável utilizar a versão mais recente do Percona XtraBackup. Você pode instalar o Percona XtraBackup no Download Percona XtraBackup.

> **Warning**
>
> Ao criar um backup de banco de dados, o XtraBackup pode salvar credenciais no arquivo xtrabackup_info. Certifique-se de examinar esse arquivo para que a configuração `tool_command` nele não contenha informações confidenciais.

> **Note**
>
> Você deve usar o Percona XtraBackup 8.0 para a migração do MySQL 8.0. O Percona XtraBackup 8.0.12 e versões posteriores oferecem suporte à migração de todas as versões do MySQL. Se você estiver migrando para o RDS para MySQL 8.0.20 ou posterior, use o Percona XtraBackup 8.0.12 ou posterior.
> Para migrações do MySQL 5.7, você também pode usar o Percona XtraBackup 2.4. Para migrações de versões anteriores do MySQL, você também pode usar o Percona XtraBackup 2.3 ou 2.4.

Você pode criar um backup completo de seus arquivos de banco de dados MySQL usando o Percona XtraBackup. Ou, se você já usa o Percona XtraBackup para fazer o backup dos arquivos do banco de dados MySQL, pode fazer upload dos arquivos e diretórios de backup completos e incrementais.

Para ter mais informações sobre como fazer backup de seu banco de dados com o Percona XtraBackup, consulte Percona XtraBackup - documentation e The xtrabackup binary no site da Percona.

Criar um backup completo com o Percona XtraBackup

Para criar um backup completo dos arquivos do banco de dados MySQL que podem ser restaurados a partir do Amazon S3, use o utilitário Percona XtraBackup (`xtrabackup`) para fazer backup do seu banco de dados.

Por exemplo, o seguinte comando cria um backup de um banco de dados MySQL e armazena os arquivos na pasta `/on-premises/s3-restore/backup`.

```
xtrabackup --backup --user=<myuser> --password=<password> --target-dir=</on-premises/
s3-restore/backup>
```

Se você deseja compactar o backup em um único arquivo (que pode ser dividido posteriormente, se necessário), salve o backup em um dos seguintes formatos:

- Gzip (.gz)
- tar (.tar)
- Percona xbstream (.xbstream)

> **Note**
>
> O Percona XtraBackup 8.0 oferece suporte apenas ao Percona xbstream para compactação.

O comando a seguir cria um backup do seu banco de dados MySQL dividido em vários arquivos Gzip.

```
xtrabackup --backup --user=<myuser> --password=<password> --stream=tar \
   --target-dir=</on-premises/s3-restore/backup> | gzip - | split -d --bytes=500MB \
   - </on-premises/s3-restore/backup/backup>.tar.gz
```

O comando a seguir cria um backup do seu banco de dados MySQL dividido em vários arquivos tar.

```
xtrabackup --backup --user=<myuser> --password=<password> --stream=tar \
   --target-dir=</on-premises/s3-restore/backup> | split -d --bytes=500MB \
   - </on-premises/s3-restore/backup/backup>.tar
```

O comando a seguir cria um backup do seu banco de dados MySQL dividido em vários arquivos xbstream.

```
xtrabackup --backup --user=<myuser> --password=<password> --stream=xbstream \
   --target-dir=</on-premises/s3-restore/backup> | split -d --bytes=500MB \
   - </on-premises/s3-restore/backup/backup>.xbstream
```

> **Note**
>
> Se você vir o erro a seguir, saiba que ele pode ser causado pela mistura de formatos de arquivo em seu comando:
>
> ```
> ERROR:/bin/tar: This does not look like a tar archive
> ```

Usar backups incrementais com o Percona XtraBackup

Se você já usa o Percona XtraBackup para fazer backups completos e incrementais de seus arquivos de banco de dados MySQL, não precisa criar um backup completo e fazer upload dos arquivos de backup no Amazon S3. Em vez disso, você pode economizar muito tempo copiando os diretórios e arquivos de backup existentes no bucket do Amazon S3. Para ter mais informações sobre como criar backups incrementais usando o Percona XtraBackup, consulte Backups incrementais.

Quando copiar os arquivos existentes de backup completo e incremental para um bucket do Amazon S3, copie recursivamente o conteúdo do diretório de base. Esse conteúdo inclui o backup completo e também todo o backup incremental dos diretórios e arquivos. Essa cópia deve preservar a estrutura de diretórios no bucket do Amazon S3. O Amazon RDS percorre todos os arquivos e diretórios. O Amazon RDS usa o arquivo `xtrabackup-checkpoints` incluído em cada backup incremental para identificar o diretório de base e ordenar os backups incrementais por intervalo de número de sequência de log (LSN).

Considerações sobre backup para o Percona XtraBackup

O Amazon RDS consome seus arquivos de backup com base no nome do arquivo. Nomeie seus arquivos de backup com a extensão de arquivo apropriada com base no formato do arquivo, como `.xbstream` para arquivos armazenados usando o formato Percona xbstream.

O Amazon RDS consome os arquivos de backup em ordem alfabética assim como na ordem numérica natural. Use a opção `split` ao emitir o comando `xtrabackup` para garantir que os arquivos de backup sejam gravados e nomeados na ordem apropriada.

O Amazon RDS não oferece suporte a backups parciais criados com o Percona XtraBackup. Você não pode usar as seguintes opções para criar um backup parcial quando faz backup dos arquivos de origem de seu banco de dados: `--tables`, `--tables-exclude`, `--tables-file`, `--databases`, `--databases-exclude` ou `--databases-file`.

O Amazon RDS oferece suporte a backups incrementais criados com o Percona XtraBackup. Para ter mais informações sobre como criar backups incrementais usando o Percona XtraBackup, consulte Backups incrementais.

Criação manual de uma função do IAM

Se você não tiver uma função do IAM, crie uma nova manualmente. No entanto, se você restaurar o banco de dados usando o AWS Management Console, recomendamos que siga o procedimento em Console e deixe que o RDS crie esse perfil do IAM para você.

Para criar manualmente uma nova função do IAM para importar seu banco de dados do Amazon S3, crie uma função para delegar permissões do Amazon RDS para seu bucket do Amazon S3. Quando você cria um perfil do IAM, você anexa as políticas de confiança e permissões. Para importar os arquivos de backup do Amazon S3, use políticas de confiança e de permissões semelhantes aos exemplos a seguir. Para ter mais informações sobre como criar a função, consulte Criar uma função para delegar permissões a um AWSserviço da .

As políticas de confiança e permissões exigem que você forneça um Nome do recurso da Amazon (ARN). Para ter mais informações sobre como formatar o ARN, consulte Nomes de recurso da Amazon (ARNs) e AWS namespaces de serviço da .

Example política de confiança para importar do Amazon S3

```
{
    "Version": "2012-10-17",
    "Statement":
    [{
        "Effect": "Allow",
        "Principal": {"Service":  "rds.amazonaws.com"},
        "Action": "sts:AssumeRole"
    }]
}
```

Example política de permissões para importar do Amazon S3: permissões de usuário do IAM

No exemplo a seguir, substitua *iam_user_id* por seu próprio valor.

```
{
    "Version":"2012-10-17",
    "Statement":
    [
        {
            "Sid":"AllowS3AccessRole",
            "Effect":"Allow",
            "Action":"iam:PassRole",
            "Resource":"arn:aws:iam::iam_user_id:role/S3Access"
        }
    ]
}
```

Example Política de permissões para importar do Amazon S3: permissões de perfil

No exemplo a seguir, substitua *amzn-s3-demo-bucket* e *prefix* por seus próprios valores.

```
{
    "Version": "2012-10-17",
    "Statement":
    [
        {
        "Effect": "Allow",
        "Action":
            [
                "s3:ListBucket",
                "s3:GetBucketLocation"
            ],
        "Resource": "arn:aws:s3:::amzn-s3-demo-bucket"
        },
        {
        "Effect": "Allow",
        "Action":
            [
                "s3:GetObject"
            ],
        "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/prefix*"
        },
        { // If your bucket is encrypted, include the following permission. This
 permission allows decryption of your AWS KMS key.
        "Effect": "Allow",
        "Action":
            [
```

```
                "kms:Decrypt"
            ],
        "Resource": [
            "arn:aws:kms:region:customer_id:key/key_id*"
            ]
        }
    ]
}
```

> **Note**
>
> Se você incluir um prefixo de nome de arquivo, inclua o asterisco (*) após o prefixo. Se não quiser especificar um prefixo, especifique apenas um asterisco.

Importar dados do Amazon S3 para uma nova instância de banco de dados MySQL

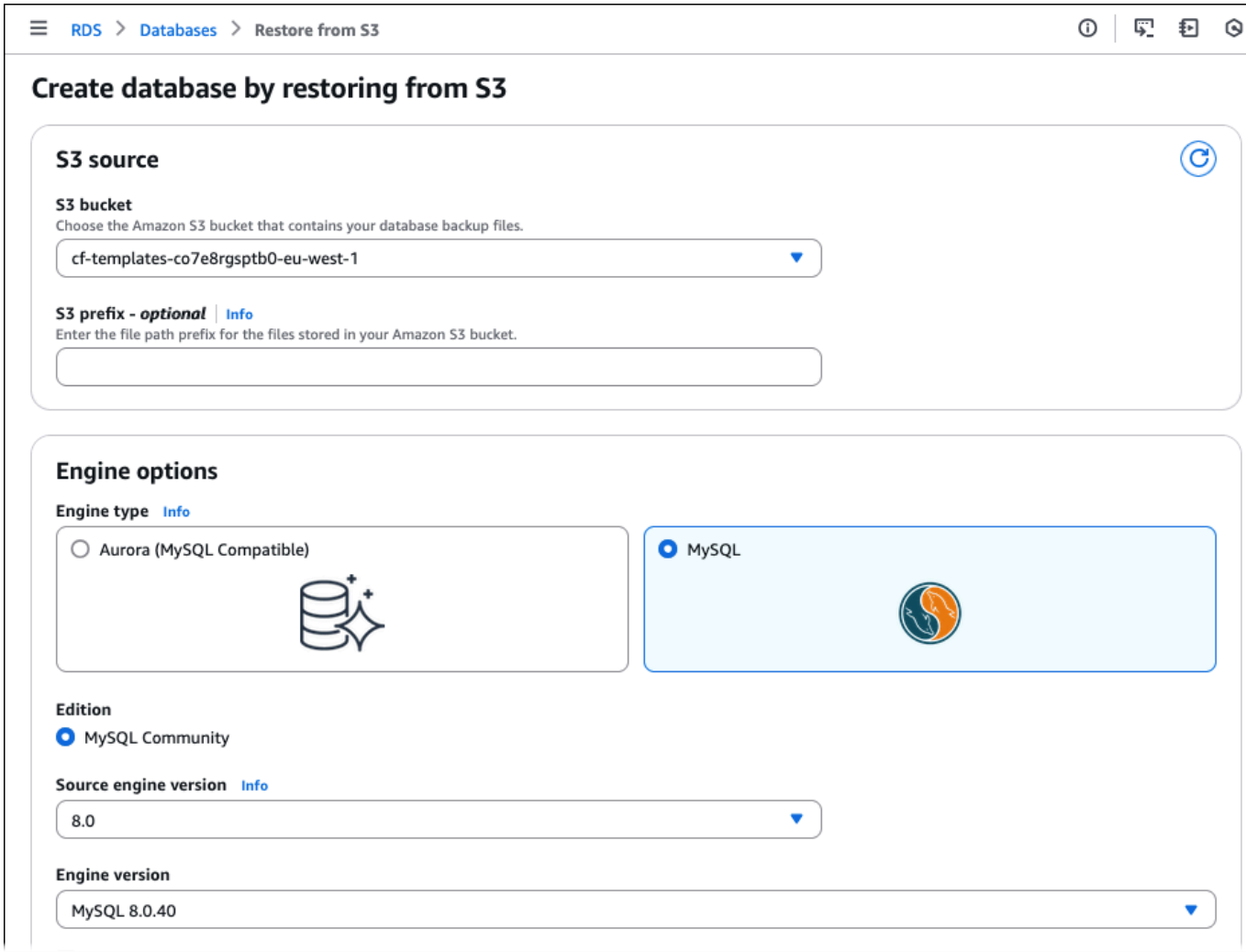
Você pode importar dados do Amazon S3 para uma nova instância de banco de dados do MySQL usando o AWS Management Console, a AWS CLI ou a API do RDS.

Console

Para importar dados do Amazon S3 para uma nova instância de banco de dados MySQL

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No canto superior direito do console do Amazon RDS, escolha a Região da AWS na qual deseja criar a instância de banco de dados. Escolha a mesma Região da AWS do bucket do Amazon S3 que contém o backup do banco de dados.
3. No painel de navegação, escolha Databases (Bancos de dados).
4. Escolha Restore from S3 (Restaurar do S3).

   A página Create database by restoring from S3 (Criar banco de dados restaurando a partir do S3) é exibida.



5. Em S3 destination (destino do S3):

   a. Selecione o S3 bucket (bucket do S3) que contém o backup.

   b. (Opcional) Em Prefixo do S3, insira o prefixo do caminho dos arquivos armazenados no bucket do Amazon S3.

      Se você não especificar um prefixo, o RDS criará a instância de banco de dados usando todos os arquivos e as pastas na pasta raiz do bucket do S3. Se você especificar um prefixo, o RDS criará a instância de banco de dados usando os arquivos e as pastas no bucket do S3 no qual o caminho para o arquivo começa com o prefixo especificado.

      Por exemplo, suponha que você armazene seus arquivos de backup no S3 em uma subpasta denominada backups e que você tenha vários conjuntos de arquivos de backup, cada um em seu próprio diretório (gzip_backup1, gzip_backup2 e assim por diante). Nesse

caso, especifique um prefixo de backups/gzip_backup1 para restaurar dos arquivos na pasta gzip_backup1.

6. Em Engine options (Opções de mecanismo):

   a. Em Engine type (Tipo de mecanismo), escolha MySQL.

   b. Para Source engine version (Versão do mecanismo de origem), escolha a versão principal do MySQL de seu banco de dados de origem.

   c. Em Versão do mecanismo, escolha a versão secundária padrão da versão principal do MySQL na Região da AWS.

      No AWS Management Console, apenas a versão secundária padrão está disponível. Você pode atualizar sua instância de banco de dados após a importação.

7. Em Perfil do IAM, crie ou escolha o perfil do IAM com a política de confiança e a política de permissões necessárias para que o Amazon RDS acesse o bucket do Amazon S3. Execute uma das seguintes ações:

   - (Recomendado) Escolha Criar um perfil e insira o Nome do perfil do IAM. Com essa opção, o RDS cria automaticamente o perfil com a política de confiança e a política de permissões para você.
   - Escolha um perfil do IAM existente. Verifique se esse perfil atende a todos os critérios em the section called “Criação manual de uma função do IAM”.

8. Especifique as informações da instância de banco de dados. Para obter informações sobre cada configuração, consulte Configurações para instâncias de banco de dados.

   > **Note**
   >
   > Certifique-se de alocar memória suficiente para sua nova instância de banco de dados para que a operação de restauração possa continuar.
   > Você também pode escolher Enable storage autoscaling (Habilitar autoscaling do armazenamento) para permitir o crescimento futuro automaticamente.

9. Escolha configurações adicionais conforme necessário.

10. Escolha Criar banco de dados.

AWS CLI

Para importar dados do Amazon S3 para uma nova instância de banco de dados MySQL usando a AWS CLI, chame o comando restore-db-instance-from-s3 com os parâmetros a seguir. Para obter informações sobre cada configuração, consulte Configurações para instâncias de banco de dados.

> **Note**
>
> Certifique-se de alocar memória suficiente para sua nova instância de banco de dados para que a operação de restauração possa continuar.
> Você também pode usar o parâmetro `--max-allocated-storage` para habilitar a autoescalabilidade de armazenamento e permitir o crescimento futuro automaticamente.

- `--allocated-storage`
- `--db-instance-identifier`
- `--db-instance-class`
- `--engine`
- `--master-username`
- `--manage-master-user-password`
- `--s3-bucket-name`
- `--s3-ingestion-role-arn`
- `--s3-prefix`
- `--source-engine`
- `--source-engine-version`

Example

Para Linux, macOS ou Unix:

```
aws rds restore-db-instance-from-s3 \
    --allocated-storage 250 \
    --db-instance-identifier myidentifier \
    --db-instance-class db.m5.large \
    --engine mysql \
    --master-username admin \
```

```
    --manage-master-user-password \
    --s3-bucket-name amzn-s3-demo-bucket \
    --s3-ingestion-role-arn arn:aws:iam::account-number:role/rolename \
    --s3-prefix bucketprefix \
    --source-engine mysql \
    --source-engine-version 8.0.32 \
    --max-allocated-storage 1000
```

Para Windows:

```
aws rds restore-db-instance-from-s3 ^
    --allocated-storage 250 ^
    --db-instance-identifier myidentifier ^
    --db-instance-class db.m5.large ^
    --engine mysql ^
    --master-username admin ^
    --manage-master-user-password ^
    --s3-bucket-name amzn-s3-demo-bucket ^
    --s3-ingestion-role-arn arn:aws:iam::account-number:role/rolename ^
    --s3-prefix bucketprefix ^
    --source-engine mysql ^
    --source-engine-version 8.0.32 ^
    --max-allocated-storage 1000
```

API do RDS

Para importar dados do Amazon S3 para uma nova instância de banco de dados MySQL usando a API do Amazon RDS, chame a operação RestoreDBInstanceFromS3.

Importar dados de um banco de dados externo do MySQL para uma instância de banco de dados do Amazon RDS para MySQL.

Também é possível importar dados de um banco de dados MySQL ou MariaDB existente para uma instância de banco de dados do RDS para MariaDB ou do RDS para MySQL. Para isso, copie o banco de dados com mysqldump e conecte-o diretamente à instância de banco de dados do RDS para MariaDB ou do RDS para MySQL. O utilitário de linha de comando `mysqldump` é comumente usado para fazer backups e transferir dados de um servidor MariaDB ou MySQL para outro. Ele está incluído no software cliente MySQL e MariaDB.

No MariaDB 10.5, o cliente é chamado de mariadb-dump. A partir do MariaDB 11.0.1, você deve usar `mariadb-dump` em vez de `mysqldump`.

> **Note**
>
> Se você estiver importando ou exportando grandes volumes de dados com uma instância de banco de dados do MySQL, será mais confiável e mais rápido mover dados para dentro e para fora do Amazon RDS usando arquivos de backup `xtrabackup` e o Amazon S3. Para ter mais informações, consulte Restauração de um backup em uma instância de banco de dados MySQL.
>
> O Amazon RDS não é compatível com `xtrabackup` para MariaDB ou `mariabackup`. O Amazon RDS é compatível somente com a importação do Amazon S3 para MySQL.

Um comando `mysqldump` típico para mover dados de um banco de dados externo para uma instância de banco de dados do Amazon RDS é semelhante ao exemplo a seguir. Substitua os valores por suas próprias informações. Para o MariaDB 11.0.1 e versões posteriores, substitua `mysqldump` por `mariadb-dump` e `mysql` por `mariadb`.

```
mysqldump -u local_user \
    --databases database_name \
    --single-transaction \
    --compress \
    --order-by-primary  \
    --routines=0 \
    --triggers=0 \
    --events=0 \
    -plocal_password | mysql -u RDS_user \
        --port=port_number \
        --host=host_name \
        -pRDS_password
```

> **Important**
>
> Não deixe um espaço entre a opção `-p` e a senha inserida.
>
> Faz parte das práticas recomendadas de segurança especificar credenciais diferentes dos prompts mostrados neste exemplo.

Analise as seguintes recomendações e considerações:

- Exclua os seguintes esquemas do arquivo de despejo:

  - `sys`
  - `performance_schema`
  - `information_schema`

  O utilitário `mysqldump` e `mariadb-dump` exclui esses esquemas por padrão.
- Para migrar usuários e privilégios, considere o uso de uma ferramenta que gera a linguagem de controle de dados (DCL) para recriá-los, como o utilitário pt-show-grants.
- Para realizar a importação, verifique se o usuário que está fazendo isso tem acesso à instância de banco de dados. Para obter mais informações, consulteControlar acesso com grupos de segurança

Os parâmetros utilizados são os seguintes:

- `-u` *`local_user`*: use para especificar um nome do usuário. Ao usar esse parâmetro pela primeira vez, especifique o nome de uma conta de usuário no banco de dados local MariaDB ou MySQL que você identifica com o parâmetro `--databases`.
- `--databases` *`database_name`*: use para especificar o nome do banco de dados na instância do MySQL ou MariaDB local que você deseja importar para o Amazon RDS.
- `--single-transaction` – use para garantir que todos os dados carregados do banco de dados local sejam consistentes com um único ponto no tempo. Se houver outros processos alterando os dados enquanto eles são lidos por `mysqldump`, usar esse parâmetro ajudará a manter a integridade dos dados.
- `--compress` – use para reduzir o consumo de largura de banda da rede, compactando os dados do banco de dados local antes de os enviar para o Amazon RDS.
- `--order-by-primary` – use para reduzir o tempo de carregamento, ordenando os dados de cada tabela de acordo com a chave primária.
- `--routines`: use se houver determinadas rotinas, como funções ou procedimentos armazenados, no banco de dados que você está copiando. Defina o parâmetro como `0`, o que exclui as rotinas durante o processo de importação. Depois, recrie manualmente as rotinas no banco de dados do Amazon RDS.
- `--triggers`: use se houver gatilhos no banco de dados que você está copiando. Defina o parâmetro como `0`, o que exclui os gatilhos durante o processo de importação. Depois, recrie manualmente os gatilhos no banco de dados do Amazon RDS.

- `--events`: use se houver eventos no banco de dados que você está copiando. Defina o parâmetro como `0`, o que exclui os eventos durante o processo de importação. Em seguida, recrie manualmente os eventos no banco de dados do Amazon RDS.
- `-p`*`local_password`*: use para especificar uma senha. Ao usar esse parâmetro pela primeira vez, especifique a senha da conta de usuário que você identifica com o primeiro parâmetro `-u`.
- `-u` *`RDS_user`*: use para especificar um nome do usuário. Ao usar esse parâmetro pela segunda vez, especifique o nome de uma conta de usuário no banco de dados padrão para a instância de banco de dados MariaDB ou MySQL que você identifica com o parâmetro `--host`.
- `--port` *`port_number`*: use para especificar a porta da sua instância de banco de dados MariaDB ou MySQL. Por padrão, essa porta é 3306, a menos que você tenha alterado o valor ao criar a instância de banco de dados.
- `--host` *`host_name`*: use para especificar o nome do Sistema de Nome de Domínio (DNS) do endpoint da instância de banco de dados Amazon RDS, por exemplo, `myinstance.123456789012.us-east-1.rds.amazonaws.com`. Você pode encontrar o valor do endpoint nos detalhes da instância do banco de dados no console do Amazon RDS.
- `-p`*`RDS_password`*: use para especificar uma senha. Na segunda utilização desse parâmetro, você especifica a senha da conta de usuário identificada pelo segundo parâmetro `-u`.

Certifique-se de criar procedimentos armazenados, acionadores, funções ou eventos manualmente no seu banco de dados Amazon RDS. Se algum desses objetos estiver no banco de dados que você está copiando, exclua-os quando executar `mysqldump` ou `mariadb-dump`. Para fazer isso, inclua os seguintes parâmetros com o comando `mysqldump` ou `mariadb-dump`:

- `--routines=0`
- `--triggers=0`
- `--events=0`

Exemplo

O exemplo a seguir copia o banco de dados de exemplo `world` no host local para uma instância de banco de dados do RDS para MySQL. Substitua os valores por suas próprias informações. Para copiar o banco de dados de exemplo para uma instância de banco de dados do RDS para MariaDB, substitua `mysqldump` por `mariadb-dump` e `mysql` por `mariadb`.

Para Linux, macOS ou Unix:

```
sudo mysqldump -u local_user \
    --databases world \
    --single-transaction \
    --compress \
    --order-by-primary  \
    --routines=0 \
    --triggers=0 \
    --events=0 \
    -plocal_password | mysql -u rds_user \
        --port=3306 \
        --host=my_instance.123456789012.us-east-1.rds.amazonaws.com \
        -pRDS_password
```

Para Windows:

Execute o comando a seguir em um prompt de comando que foi aberto clicando com o botão direito do mouse em Prompt de comando no menu de programas do Windows e escolhendo Executar como administrador. Substitua os valores por suas próprias informações. Para copiar o banco de dados de exemplo para uma instância de banco de dados do RDS para MariaDB, substitua `mysqldump` por `mariadb-dump` e `mysql` por `mariadb`.

```
mysqldump -u local_user ^
    --databases world ^
    --single-transaction ^
    --compress ^
    --order-by-primary  ^
    --routines=0 ^
    --triggers=0 ^
    --events=0 ^
    -plocal_password | mysql -u RDS_user ^
        --port=3306 ^
        --host=my_instance.123456789012.us-east-1.rds.amazonaws.com ^
        -pRDS_password
```

> **Note**
>
> Faz parte das práticas recomendadas de segurança especificar credenciais diferentes dos prompts mostrados no exemplo.

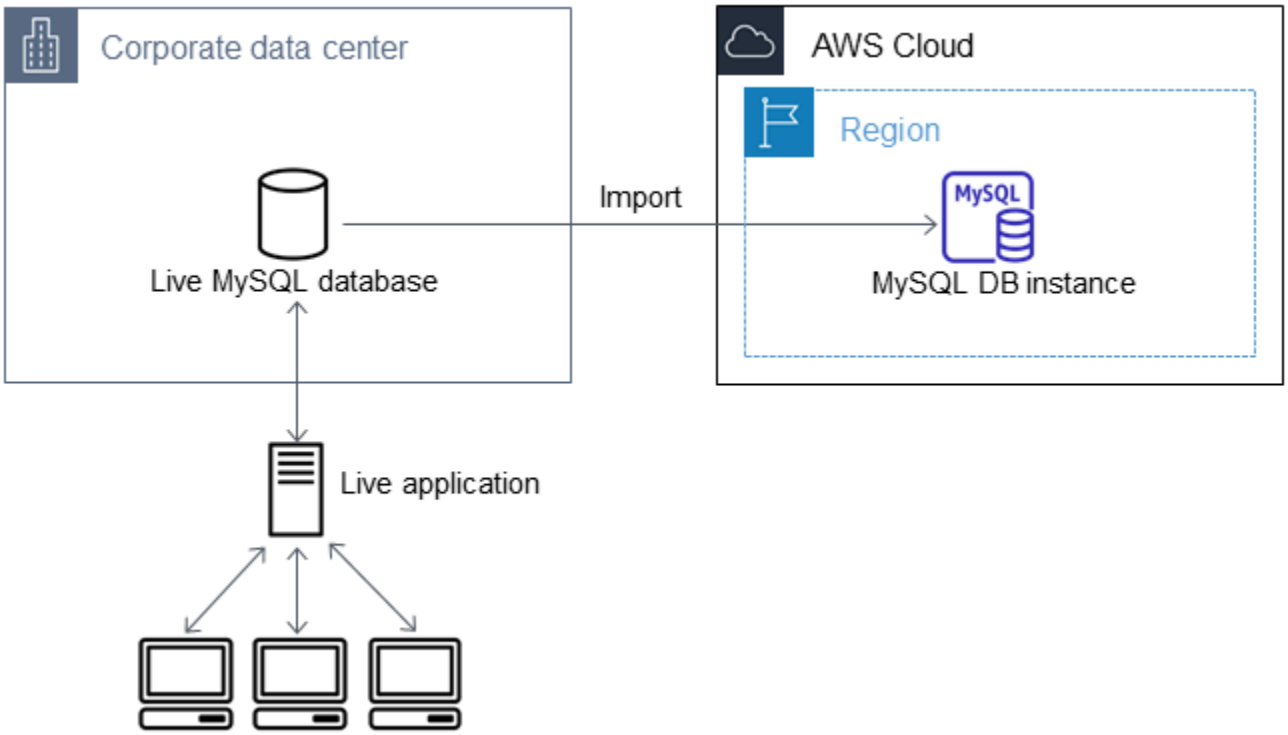
Importar dados para um banco de dados do Amazon RDS para MySQL com tempo de inatividade reduzido

Em alguns casos, pode ser preciso importar dados de um banco de dados MariaDB ou MySQL compatível com uma aplicação ativa para uma instância de banco de dados do RDS para MariaDB, uma instância de banco de dados do RDS para MySQL ou um cluster de banco de dados multi-AZ do RDS para MySQL. Use o procedimento a seguir para minimizar o impacto sobre a disponibilidade das aplicações. Este procedimento também pode ser útil se você está trabalhando com um banco de dados muito grande. Usando esse procedimento, você pode reduzir o custo da importação ao diminuir a quantidade de dados transmitidos pela rede para a AWS.

Neste procedimento, você transfere uma cópia dos dados do banco de dados para uma instância do Amazon EC2 e importa os dados em um novo banco de dados do Amazon RDS. Depois, use a replicação para atualizar o banco de dados do Amazon RDS em relação à instância externa dinâmica, antes de redirecionar sua aplicação ao banco de dados do Amazon RDS. Se a instância externa for do MariaDB 10.0.24 ou posterior e a instância de destino for do RDS para MariaDB, configure a replicação do MariaDB com base em identificadores de transação global (GTIDs). Caso contrário, configure a replicação com base nas coordenadas do log binário. Recomendamos a replicação baseada em GTID caso o banco de dados externo comporte esse tipo de replicação, já que é um método mais confiável. Para obter mais informações, consulte Global Transaction ID na documentação do MariaDB.

> **Note**
>
> Se você quiser importar dados para uma instância de banco de dados RDS para MySQL e o seu ambiente for compatível, recomendamos mover dados para dentro e para fora do Amazon RDS usando arquivos de backup e o Amazon S3. Para ter mais informações, consulte Restauração de um backup em uma instância de banco de dados MySQL.

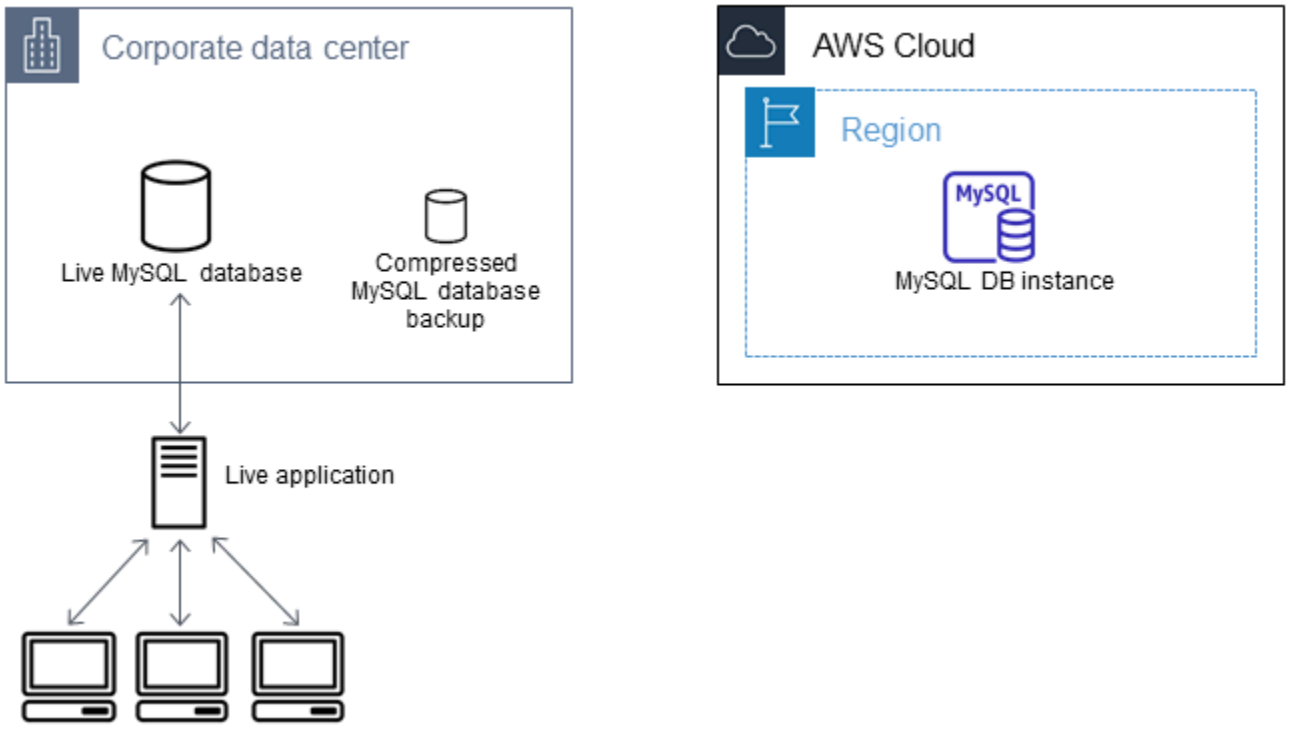
O diagrama a seguir mostra como importar um banco de dados MySQL externo para um banco de dados MySQL no Amazon RDS.



Tarefa 1: criar uma cópia do banco de dados existente

A primeira etapa no processo de migração de uma grande quantidade de dados para um banco de dados do RDS para MariaDB ou RDS para MySQL com o mínimo de tempo de inatividade é criar uma cópia dos dados de origem.

O diagrama a seguir mostra como criar um backup do banco de dados MySQL.



Você pode usar o utilitário `mysqldump` para criar um backup de banco de dados no formato SQL ou de texto delimitado. No MariaDB 10.5, o cliente é chamado de mariadb-dump. A partir do MariaDB 11.0.1, você deve usar `mariadb-dump` em vez de `mysqldump`. Recomendamos que você faça uma execução de teste com cada formato em um ambiente que não seja de produção para ver qual método minimiza o tempo de execução de mysqldump.

Recomendamos também que você pondere a performance de `mysqldump` em relação ao benefício oferecido usando o formato de texto delimitado para o carregamento. Um backup usando o formato de texto delimitado cria um arquivo de texto separado por tabulação para cada tabela que está sendo despejada. Você pode carregar esses arquivos em paralelo usando o comando `LOAD DATA LOCAL INFILE` para reduzir a quantidade de tempo necessária para importar seu banco de dados. Para ter mais informações, consulte Carregar os dados para o MariaDB ou Carregar os dados para o MySQL.

Antes de iniciar a operação de backup, certifique-se de definir as opções de replicação no banco de dados MariaDB ou MySQL que está copiando para o Amazon RDS. As opções de replicação incluem ativar o registro em log binário e definir um ID de servidor exclusivo. Configurar essas opções faz com que seu servidor comece a registrar em log transações de banco de dados e o prepara para ser um uma instância de replicação de origem mais adiante nesse processo.

Analise as seguintes recomendações e considerações:

- Use a opção `--single-transaction` com `mysqldump` porque ela despeja um estado consistente do banco de dados. Para garantir um arquivo de despejo válido, não execute instruções de linguagem de definição de dados (DDL) enquanto `mysqldump` estiver em execução. É possível programar uma janela de manutenção para essas operações.
- Exclua os seguintes esquemas do arquivo de despejo:
  - `sys`
  - `performance_schema`
  - `information_schema`

  O utilitário `mysqldump` e `mariadb-dump` exclui esses esquemas por padrão.
- Para migrar usuários e privilégios, considere o uso de uma ferramenta que gera a linguagem de controle de dados (DCL) para recriá-los, como o utilitário pt-show-grants.

Como definir opções de replicação

1. Edite o arquivo `my.cnf`. Esse arquivo geralmente está localizado em `/etc`.

   ```
   sudo vi /etc/my.cnf
   ```

   Adicione as opções `log_bin` e `server_id` à seção `[mysqld]`. A opção `log_bin` fornece um identificador de nome de arquivo para arquivos de log binário. A opção `server_id` fornece um identificador exclusivo para o servidor em relações entre origem e réplica.

   O seguinte exemplo mostra a seção `[mysqld]` atualizada de um arquivo `my.cnf`:

   ```
   [mysqld]
   log-bin=mysql-bin
   server-id=1
   ```

   Para ter mais informações, consulte Setting the Replication Source Configuration na documentação do MySQL.
2. Para replicação com um cluster de banco de dados multi-AZ, defina os parâmetros `ENFORCE_GTID_CONSISTENCY` e `GTID_MODE` como `ON`.

   ```
   mysql> SET @@GLOBAL.ENFORCE_GTID_CONSISTENCY = ON;
   ```

```
mysql> SET @@GLOBAL.GTID_MODE = ON;
```

Essas configurações não são necessárias para a replicação com uma instância de banco de dados.

3. Reinicie o serviço `mysql`.

```
sudo service mysqld restart
```

Como criar uma cópia de backup do banco de dados existente

1. Crie um backup dos dados por meio do utilitário `mysqldump` ou `mariadb-dump`, especificando o formato SQL ou de texto delimitado.

   Para o MySQL 8.0.25 e versões anteriores, especifique `--master-data=2` para criar um arquivo de backup que possa ser usado para iniciar a replicação entre servidores. Para o MySQL 8.0.26 e versões posteriores, especifique `--source-data=2` para criar um arquivo de backup que possa ser usado para iniciar a replicação entre servidores. Para ter mais informações, consulte mysqldump — A Database Backup Program na documentação do MySQL.

   Para melhorar a performance e garantir a integridade dos dados, use as opções `--order-by-primary` e `--single-transaction` para `mysqldump`.

   Para evitar a inclusão do banco de dados do sistema MySQL no backup, não use a opção `--all-databases` com `mysqldump`. Para ter mais informações, consulte Creating a Data Snapshot Using mysqldump na documentação do MySQL.

   Use `chmod`, se necessário, para garantir que o diretório no qual o arquivo de backup está sendo criado seja gravável.

   > **Important**
   >
   > No Windows, execute a janela de comando como um administrador.

   - Para produzir a saída SQL, use o seguinte comando:

     Para Linux, macOS ou Unix:

```
sudo mysqldump \
    --databases database_name \
    --master-data=2  \
    --single-transaction \
    --order-by-primary \
    -r backup.sql \
    -u local_user \
    -ppassword
```

> **Note**
>
> Faz parte das práticas recomendadas de segurança especificar credenciais diferentes dos prompts mostrados no exemplo.

Para Windows:

```
mysqldump ^
    --databases database_name ^
    --master-data=2  ^
    --single-transaction ^
    --order-by-primary ^
    -r backup.sql ^
    -u local_user ^
    -ppassword
```

> **Note**
>
> Faz parte das práticas recomendadas de segurança especificar credenciais diferentes dos prompts mostrados no exemplo.

- Para produzir uma saída de texto delimitado, use o seguinte comando:

  Para Linux, macOS ou Unix:

```
sudo mysqldump \
    --tab=target_directory \
    --fields-terminated-by ',' \
    --fields-enclosed-by '"' \
```

```
    --lines-terminated-by 0x0d0a \
    database_name \
    --master-data=2 \
    --single-transaction \
    --order-by-primary \
    -ppassword
```

Para Windows:

```
mysqldump ^
    --tab=target_directory ^
    --fields-terminated-by "," ^
    --fields-enclosed-by """ ^
    --lines-terminated-by 0x0d0a ^
    database_name ^
    --master-data=2 ^
    --single-transaction ^
    --order-by-primary ^
    -ppassword
```

> **Note**
>
> Faz parte das práticas recomendadas de segurança especificar credenciais diferentes dos prompts mostrados no exemplo.
> Certifique-se de criar procedimentos armazenados, acionadores, funções ou eventos manualmente no seu banco de dados Amazon RDS. Se algum desses objetos estiver no banco de dados que você está copiando, exclua-os quando executar `mysqldump`. Para fazer isso, inclua estes argumentos com o comando`mysqldump`:
>
> - `--routines=0`
> - `--triggers=0`
> - `--events=0`

Para o MySQL 8.0.22 e versões anteriores e versões do MariaDB, quando você executa `mysqldump` e especifica o formato de texto delimitado, um comentário `CHANGE MASTER TO` é exibido. Esse comentário contém o nome e a posição do arquivo de log mestre. Para o MySQL 8.0.23 e versões posteriores, quando você executa `mysqldump` usando o formato de texto delimitado, um comentário `CHANGE REPLICATION SOURCE TO` é exibido. Esse comentário

contém o nome e a posição do arquivo de log de origem. Se a versão da instância externa não for MariaDB 10.0.24 e versões posteriores ou MySQL 8.0.23 e versões posteriores, anote os valores referentes a `MASTER_LOG_FILE` e `MASTER_LOG_POS`. Esses valores são necessários ao configurar a replicação.

A seguinte saída é retornada para versões do MariaDB e para o MySQL 8.0.22 e versões anteriores:

```
-- Position to start replication or point-in-time recovery from
--
-- CHANGE MASTER TO MASTER_LOG_FILE='mysql-bin-changelog.000031',
 MASTER_LOG_POS=107;
```

A seguinte saída é retornada para o MySQL 8.0.23 e versões posteriores:

```
-- Position to start replication or point-in-time recovery from
--
-- CHANGE SOURCE TO SOURCE_LOG_FILE='mysql-bin-changelog.000031',
 SOURCE_LOG_POS=107;
```

Para o MySQL 8.0.22 e versões anteriores, se você estiver usando o formato SQL, poderá obter o nome e a posição do arquivo de log mestre no comentário `CHANGE MASTER TO` no arquivo de backup. Para o MySQL 8.0.23 e versões posteriores, se você estiver usando o formato SQL, poderá obter o nome e a posição do arquivo de log de origem no comentário `CHANGE REPLICATION SOURCE TO` no arquivo de backup. Se a instância externa for o MariaDB versão 10.0.24 ou posteriores, você poderá obter o GTID na próxima etapa.

2. Se a versão da instância externa que você está usando for MariaDB 10.0.24 ou posterior, use a replicação baseada em GTID. Execute `SHOW MASTER STATUS` na instância externa do MariaDB para obter o nome e a posição do arquivo de log binário. Em seguida, converta-o em um GTID executando `BINLOG_GTID_POS` na instância externa do MariaDB.

   ```
   SELECT BINLOG_GTID_POS('binary_log_file_name', binary_log_file_position);
   ```

   Anote o GTID retornado. Você precisará dele para configurar a replicação.

3. Compacte os dados copiados para reduzir a quantidade de recursos de rede necessários para copiar seus dados para o banco de dados do Amazon RDS. Anote o tamanho do arquivo de backup. Você precisará dessas informações ao determinar o tamanho de uma instância do

Amazon EC2 a ser criada. Quando terminar, compacte o arquivo de backup usando o GZIP ou seu utilitário de compactação preferido.

- Para compactar a saída SQL, use o seguinte comando:

```
gzip backup.sql
```

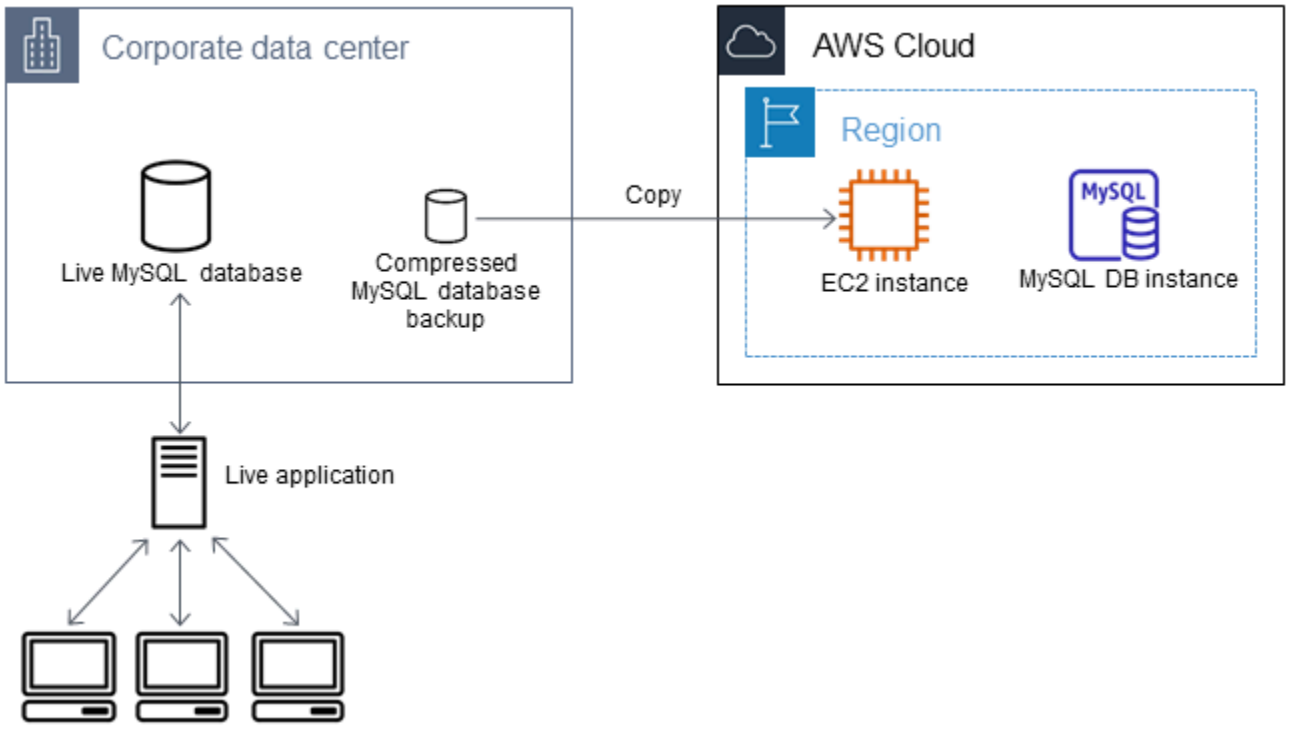
- Para compactar a saída de texto delimitado, use o seguinte comando:

```
tar -zcvf backup.tar.gz target_directory
```

Tarefa 2: criar uma instância do Amazon EC2 e copiar o banco de dados compactado

Copiar seu arquivo de backup de banco de dados compactado para uma instância do Amazon EC2 requer menos recursos de rede do que fazer uma cópia direta de dados não compactados entre instâncias de bancos de dados. Depois que seus dados estiverem no Amazon EC2, você poderá copiá-los de lá diretamente para o banco de dados MariaDB ou MySQL. Para reduzir o custo dos recursos de rede, a instância do Amazon EC2 deve estar na mesma Região da AWS que a instância de banco de dados do Amazon RDS. Ter a instância do Amazon EC2 na mesma Região da AWS que o banco de dados do Amazon RDS também reduz a latência da rede durante a importação.

O diagrama a seguir mostra como copiar o backup do banco de dados para uma instância do Amazon EC2.



Como criar uma instância do Amazon EC2 e copiar seus dados

1. Na Região da AWS em que você planeja criar o banco de dados do Amazon RDS, crie uma nuvem privada virtual (VPC), um grupo de segurança de VPC e uma sub-rede de VPC. Certifique-se de que as regras de entrada para o grupo de segurança da VPC permitam os endereços IP necessários para a aplicação se conectar ao AWS. Você pode especificar uma variedade de endereços IP (por exemplo, `203.0.113.0/24`) ou outro grupo de segurança de VPC. É possível usar o console da Amazon VPC para criar e gerenciar VPCs, sub-redes e grupos de segurança. Para ter mais informações, consulte Conceitos básicos da Amazon VPC no Manual do usuário da Amazon Virtual Private Cloud.
2. Abra o console do Amazon EC2 e escolha a Região da AWS para conter tanto a instância do Amazon EC2 como o banco de dados do Amazon RDS. Inicie uma instância do Amazon EC2 usando a VPC, a sub-rede e o grupo de segurança que você criou na Etapa 1. Certifique-se de selecionar um tipo de instância com armazenamento suficiente para o arquivo de backup de banco de dados quando ele não estiver compactado. Para ter detalhes sobre instâncias do Amazon EC2, consulte Comece a usar o Amazon EC2 no Guia do usuário do Amazon Elastic Compute Cloud.
3. Para se conectar ao banco de dados do Amazon RDS pela sua instância do Amazon EC2, edite seu grupo de segurança da VPC. Adicione uma regra de entrada especificando o endereço IP privado da sua instância do EC2. Você pode encontrar o endereço IP privado na guia Details

(Detalhes) do painel Instance (Instância) na janela do console do EC2. Para editar o grupo de segurança de VPC e adicionar uma regra de entrada, escolha Security Groups (Grupos de segurança) no painel de navegação do console do EC2, escolha o grupo de segurança e, em seguida, adicione uma regra de entrada para MySQL ouAurora especificando o endereço IP privado da sua instância do EC2. Para saber como adicionar uma regra de entrada a um grupo de segurança de VPC, consulte Regras de grupos de segurança no Manual do usuário da Amazon Virtual Private Cloud.

4. Copie o arquivo de backup de banco de dados compactado do seu sistema local para a sua instância do Amazon EC2. Use `chmod`, se necessário, para garantir que você tenha permissão de gravação para o diretório de destino da instância do Amazon EC2. Você pode usar `scp` ou um cliente Secure Shell (SSH) para copiar o arquivo. O seguinte comando é um exemplo de comando `scp`:

```
scp -r -i key pair.pem backup.sql.gz ec2-user@EC2 DNS:/target_directory/backup.sql.gz
```

> **Important**
>
> Ao copiar dados sensíveis, use um protocolo de transferência de rede segura.

5. Conecte-se à sua instância do Amazon EC2 e instale as atualizações e as ferramentas do cliente MySQL mais recentes usando os seguintes comandos:

```
sudo yum update -y
sudo yum install mysql -y
```

Para ter mais informações, consulte Conectar-se à instância no Guia do usuário do Amazon Elastic Compute Cloud para Linux.

> **Important**
>
> Este exemplo instala o cliente MySQL em uma distribuição de imagem de máquina da Amazon (AMI) do Amazon Linux. Este exemplo não instala o cliente MySQL em uma distribuição diferente, como Ubuntu ou Red Hat Enterprise Linux. Para ter informações sobre como instalar o MySQL, consulte Installing MySQL na documentação do MySQL.

6. Enquanto estiver conectado à sua instância do Amazon EC2, descompacte o arquivo de backup do banco de dados. Os comandos a seguir são exemplos.

- Para descompactar a saída SQL, use o seguinte comando:

```
gzip backup.sql.gz -d
```

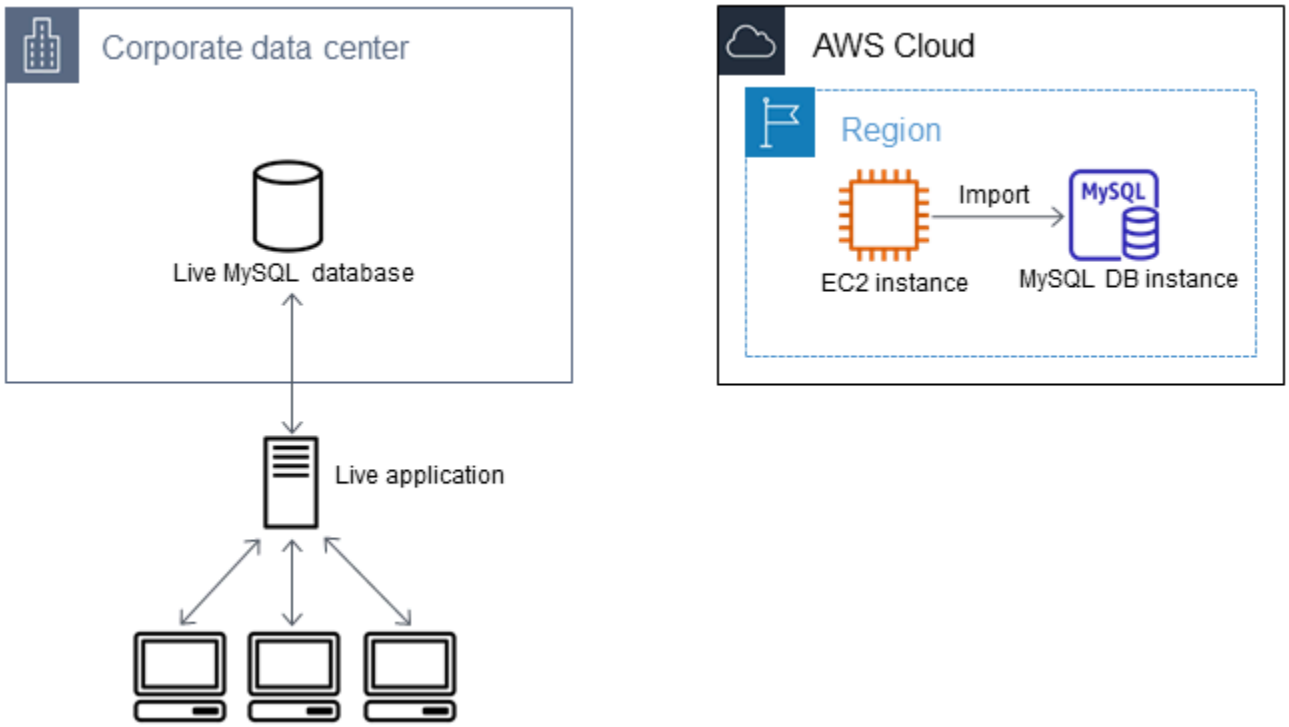
- Para descomprimir a saída de texto delimitada, use o seguinte comando:

```
tar xzvf backup.tar.gz
```

Tarefa 3: criar um banco de dados MariaDB ou MySQL e importar os dados da instância do Amazon EC2

Ao criar uma instância de banco de dados do RDS para MariaDB, uma instância de banco de dados do RDS para MySQL ou um cluster de banco de dados multi-AZ do RDS para MySQL na mesma Região da AWS que a instância do Amazon EC2, é possível importar o arquivo de backup do banco de dados do Amazon EC2 mais rapidamente do que pela internet.

O diagrama a seguir mostra como importar o backup de uma instância do Amazon EC2 para um banco de dados MySQL.



Como criar um banco de dados MariaDB ou MySQL e importar seus dados

1. Determine qual classe de instância de banco de dados e que quantidade de espaço de armazenamento são necessários para atender à workload esperada para esse banco de dados do Amazon RDS. Como parte desse processo, decida sobre o espaço suficiente e a capacidade de processamento para os seus procedimentos de carregamento de dados. Além disso, decida o que é necessário para lidar com a workload de produção. Você pode estimar isso com base no tamanho e nos recursos do banco de dados MySQL ou MariaDB de origem. Para ter mais informações, consulte Classes de instâncias de banco de dados .
2. Crie uma instância de banco de dados ou um cluster de banco de dados multi-AZ na Região da AWS que contém a instância do Amazon EC2.

   Para criar um cluster de banco de dados multi-AZ do RDS para MySQL, siga as instruções em Criar um cluster de banco de dados multi-AZ para o Amazon RDS.

   Para criar uma instância de banco de dados do RDS para MariaDB ou do RDS para MySQL, siga as instruções em Criar uma instância de banco de dados do Amazon RDS e use as seguintes diretrizes:

   - Especifique uma versão do mecanismo de banco de dados que seja compatível com a sua instância de banco de dados de origem.
     - Se a instância de origem for do MySQL 5.5.x, a instância de banco de dados do Amazon RDS deverá ser do MySQL.
     - Se a instância de origem for do MySQL 5.6.x ou 5.7.x, a instância de banco de dados do Amazon RDS deverá ser do MySQL ou do MariaDB.
     - Se a sua instância de origem for o MySQL 8.0.x, a instância de banco de dados do Amazon RDS deverá ser o MySQL 8.0.x.
     - Se a sua instância de origem for o MySQL 8.4.x, a instância de banco de dados do Amazon RDS deverá ser o MySQL 8.4.x.
     - Se a instância de origem for do MariaDB 5.5 ou superior, a instância de banco de dados do Amazon RDS deverá ser do MariaDB.
   - Especifique a mesma nuvem privada virtual (VPC) e o grupo de segurança da VPC para sua instância do Amazon EC2. Essa abordagem garante que sua instância do Amazon EC2 e sua instância do Amazon RDS sejam visíveis uma para a outra na rede. Verifique se sua instância de banco de dados pode ser acessada publicamente. Para configurar a replicação com o

banco de dados de origem, conforme descrito em uma seção mais adiante, a instância de banco de dados deve ser acessível ao público.

- Não configure várias zonas de disponibilidade, retenção de backup nem réplicas de leitura até depois de ter importado o backup do banco de dados. Quando a importação estiver concluída, você poderá configurar o recurso multi-AZ e a retenção de backup para a instância de produção.

3. Reveja as opções de configuração padrão para o banco de dados do Amazon RDS. Se o grupo de parâmetros padrão para o banco de dados não tiver as opções de configuração desejadas, encontre outro que as inclua ou crie um grupo de parâmetros. Para ter mais informações sobre como criar um grupo de parâmetros, consulte Grupos de parâmetros para Amazon RDS.
4. Conecte-se ao novo banco de dados do Amazon RDS como usuário principal. Crie os usuários necessários para dar suporte aos administradores, aplicações e serviços que precisam acessar a instância de banco de dados. O nome do host para o banco de dados do Amazon RDS é o valor do endpoint dessa instância, de banco de dados sem o número da porta; por exemplo, `mysampledb.123456789012.us-west-2.rds.amazonaws.com`. Você pode encontrar o valor do endpoint nos detalhes do banco de dados no console do Amazon RDS.
5. Conecte-se à sua instância Amazon EC2. Para ter mais informações, consulte Conectar-se à instância no Guia do usuário do Amazon Elastic Compute Cloud para Linux.
6. Conecte-se ao seu banco de dados do Amazon RDS como um host remoto pela sua instância do Amazon EC2 usando o comando `mysql`. O seguinte comando é um exemplo:

```
mysql -h host_name -P 3306 -u db_master_user -p
```

O *host_name* é o endpoint do banco de dados do Amazon RDS.

7. No prompt `mysql`, execute o comando `source` e passe a ele o nome do arquivo de despejo do banco de dados. Esse comando carrega os dados na instância de banco de dados do Amazon RDS.

- Para o formato SQL, use o seguinte comando:

```
mysql> source backup.sql;
```

- Para o formato de texto delimitado, primeiro crie o banco de dados, se ele não for o banco de dados padrão que você criou ao configurar o banco de dados do Amazon RDS.

```
mysql> create database database_name;
```

```
mysql> use database_name;
```

Em seguida, crie as tabelas.

```
mysql> source table1.sql
mysql> source table2.sql
etc...
```

Em seguida, importe os dados.

```
mysql> LOAD DATA LOCAL INFILE 'table1.txt' INTO TABLE table1 FIELDS TERMINATED BY
 ',' ENCLOSED BY '"' LINES TERMINATED BY '0x0d0a';
mysql> LOAD DATA LOCAL INFILE 'table2.txt' INTO TABLE table2 FIELDS TERMINATED BY
 ',' ENCLOSED BY '"' LINES TERMINATED BY '0x0d0a';
etc...
```

Para melhorar a performance, você pode realizar essas operações em paralelo de várias conexões, para que todas as tabelas sejam criadas e carregadas ao mesmo tempo.

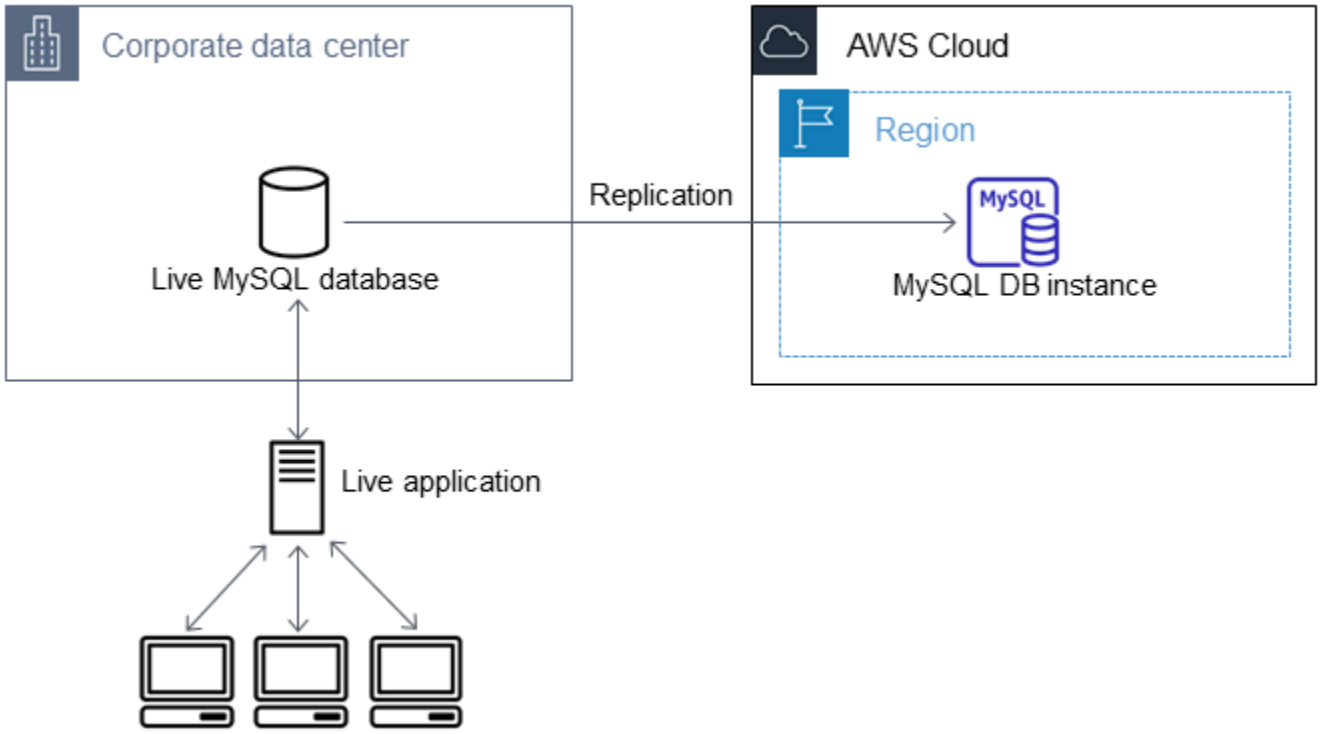
> **Note**
>
> Se você usou qualquer opção de formatação de dados com `mysqldump` quando despejou inicialmente a tabela, use as mesmas opções com `LOAD DATA LOCAL INFILE` para assegurar uma interpretação adequada do conteúdo do arquivo de dados.

8. Execute uma consulta `SELECT` simples em uma ou duas das tabelas no banco de dados importado para verificar se a importação foi bem-sucedida.

Se você não precisa mais da instância do Amazon EC2 usada neste procedimento, encerre-a para reduzir o uso de recursos de uso da AWS. Para encerrar uma instância do EC2, consulte Como encerrar uma instância no Guia do usuário do Amazon Elastic Compute Cloud.

Tarefa 4: replicar dados do banco de dados externo para o novo banco de dados do Amazon RDS

Seu banco de dados de origem provavelmente foi atualizado durante o tempo decorrido entre a cópia e a transferência dos dados para o banco de dados MariaDB ou MySQL. Assim, você pode usar a replicação para atualizar o banco de dados copiado com o banco de dados de origem.



As permissões necessárias para iniciar a replicação em um banco de dados do Amazon RDS são restritas e não estão disponíveis ao seu usuário principal do Amazon RDS. Por esse motivo, use o procedimento armazenado apropriado do Amazon RDS para a versão principal do mecanismo:

- mysql.rds_set_external_master (RDS para MariaDB e RDS para MySQL versões principais 8.0 e anteriores)
- mysql.rds_set_external_source (RDS para MySQL versões principais 8.4 e posteriores)
- mysql.rds_set_external_master_gtid para configurar a replicação e mysql.rds_start_replication para iniciar a replicação

Como iniciar a replicação

Na Tarefa 1, ao definir as opções de replicação, você ativou o registro em log binário e definiu um ID de servidor exclusivo para o banco de dados de origem. Agora, é possível configurar seu banco de

dados do Amazon RDS como uma réplica com seu banco de dados dinâmico como a instância de replicação de origem.

1. No console do Amazon RDS, adicione o endereço IP do servidor que hospeda o banco de dados de origem ao grupo de segurança da VPC para o banco de dados do Amazon RDS. Para ter mais informações sobre grupos de segurança da VPC, consulte Configurar regras de grupo de segurança no Manual do usuário da Amazon Virtual Private Cloud.

   Você também pode precisar configurar a rede local para permitir conexões com o endereço IP do banco de dados do Amazon RDS, para que ele possa se comunicar com a instância de origem. Para localizar o endereço IP do banco de dados do Amazon RDS, use o comando `host`:

   ```
   host host_name
   ```

   O *`host_name`* é o nome DNS do endpoint do banco de dados do Amazon RDS; por exemplo, `myinstance.123456789012.us-east-1.rds.amazonaws.com`. Você pode encontrar o valor do endpoint nos detalhes da instância do banco de dados no console do Amazon RDS.
2. Usando o cliente de sua preferência, conecte-se à instância de origem e crie um usuário a ser usado para a replicação. Esta conta é usada unicamente para replicação e deve estar restrita ao seu domínio para melhorar a segurança. O seguinte comando é um exemplo:

   ```
   CREATE USER 'repl_user'@'mydomain.com' IDENTIFIED BY 'password';
   ```

   > **Note**
   >
   > Especifique credenciais diferentes dos prompts mostrados aqui como prática recomendada de segurança.

3. Para a instância de origem, conceda os privilégios `REPLICATION CLIENT` e `REPLICATION SLAVE` para seu usuário de replicação. Por exemplo, para conceder os privilégios de `REPLICATION CLIENT` e `REPLICATION SLAVE` em todos os bancos de dados para o usuário '`repl_user`' de seu domínio, emita o seguinte comando:

   ```
   GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'mydomain.com';
   ```

4. Se você tiver usado o formato SQL para criar o arquivo de backup e a versão da instância externa não for MariaDB 10.0.24 ou posterior, veja o conteúdo desse arquivo executando o seguinte comando:

```
cat backup.sql
```

O arquivo inclui um comentário `CHANGE MASTER TO` que contém o nome e a posição do arquivo de log mestre. Esse comentário está incluído no arquivo de backup quando você usa a opção `--master-data` com `mysqldump`. Observe os valores para `MASTER_LOG_FILE` e `MASTER_LOG_POS`.

```
--
-- Position to start replication or point-in-time recovery from
--

-- CHANGE MASTER TO MASTER_LOG_FILE='mysql-bin-changelog.000031', MASTER_LOG_POS=107;
```

Se você usou o formato de texto delimitado para criar o arquivo de backup e a versão da instância externa não for MariaDB 10.0.24 ou posterior, você já deve ter coordenadas de logs binários da Etapa 1 do procedimento “Como criar uma cópia de backup do banco de dados existente” na Tarefa 1.

Se a versão da instância externa for MariaDB 10.0.24 ou posterior, você já deve ter o GTID com base no qual iniciar a replicação na Etapa 2 do procedimento “Como criar uma cópia de backup do banco de dados existente” na Tarefa 1.

5. Torne o banco de dados do Amazon RDS a réplica. Se a versão da instância externa não for MariaDB 10.0.24 ou posterior, conecte-se ao banco de dados do Amazon RDS como usuário principal e identifique o banco de dados de origem como a instância de replicação de origem usando o comando o procedimento armazenado apropriado do Amazon RDS.

   - mysql.rds_set_external_master (RDS para MariaDB e RDS para MySQL versões principais 8.0 e anteriores)
   - mysql.rds_set_external_source (RDS para MySQL versões principais 8.4 e posteriores)

   Se você tiver um arquivo de backup no formato SQL, use o nome do arquivo de log mestre e a posição do log mestre que você determinou na Etapa 4. Se você usou o formato de texto delimitado, use o nome e a posição que você determinou ao criar os arquivos de backup. Os seguintes comandos são exemplos:

   MySQL 8.4 e versões posteriores

```
CALL mysql.rds_set_external_source ('myserver.mydomain.com', 3306,
```

```
    'repl_user', 'password', 'mysql-bin-changelog.000031', 107, 1);
```

MariaDB e MySQL 8.0 e versões anteriores

```
CALL mysql.rds_set_external_master ('myserver.mydomain.com', 3306,
    'repl_user', 'password', 'mysql-bin-changelog.000031', 107, 1);
```

> **Note**
>
> Especifique credenciais diferentes dos prompts mostrados aqui como prática recomendada de segurança.

Se a instância externa for o MariaDB 10.0.24 ou posterior, conecte-se ao banco de dados do Amazon RDS como o usuário principal e identifique o banco de dados de origem como a instância de replicação de origem usando o comando mysql.rds_set_external_master_gtid. Use o GTID que você determinou na Etapa 2 do procedimento “Como criar uma cópia de backup do banco de dados existente” na Tarefa 1. O seguinte comando é um exemplo:

```
CALL mysql.rds_set_external_master_gtid ('source_server_ip_address', 3306,
 'ReplicationUser', 'password', 'GTID', 1);
```

O *source_server_ip_address* é o endereço IP da instância de replicação de origem. No momento, não é possível usar um endereço DNS privado do EC2.

> **Note**
>
> Especifique credenciais diferentes dos prompts mostrados aqui como prática recomendada de segurança.

6. No banco de dados do Amazon RDS, para iniciar a replicação, execute o comando mysql.rds_start_replication abaixo:

```
CALL mysql.rds_start_replication;
```

7. No banco de dados do Amazon RDS, para determinar quando a réplica está atualizada com a instância de replicação de origem, execute o comando SHOW REPLICA STATUS. Os resultados

do comando `SHOW REPLICA STATUS` incluem o campo `Seconds_Behind_Master`. Quando o campo `Seconds_Behind_Master` retorna 0, a réplica é atualizada com a instância de replicação de origem.

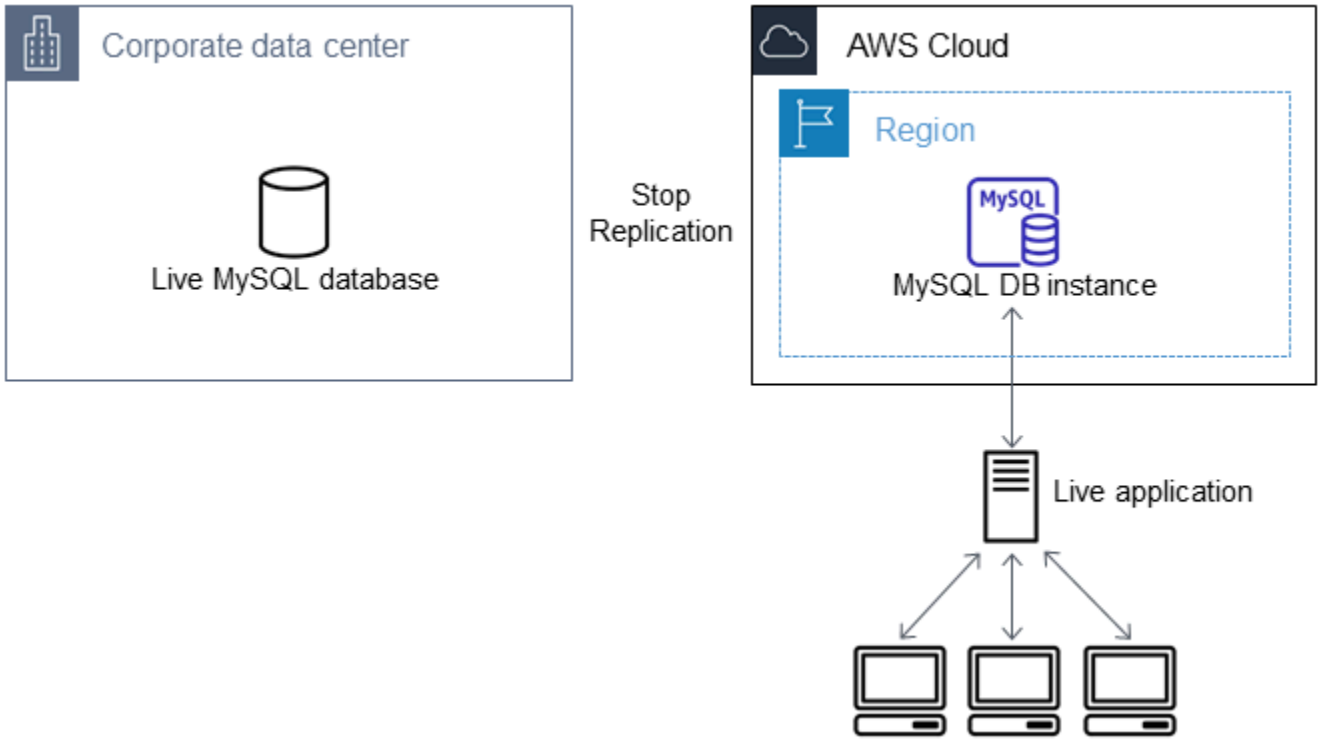
> **Note**
>
> As versões anteriores do MySQL usavam `SHOW SLAVE STATUS` em vez de `SHOW REPLICA STATUS`. Se você estiver usando uma versão do MySQL anterior à 8.0.23, use `SHOW SLAVE STATUS`.

Para uma instância de banco de dados MariaDB 10.5, 10.6 ou 10.11, execute o procedimento mysql.rds_replica_status em vez do comando MySQL.

8. Depois que o banco de dados do Amazon RDS estiver atualizado, ative backups automatizados para poder restaurar esse banco de dados, se necessário. Você pode ativar ou modificar backups automatizados para um banco de dados do Amazon RDS usando o console do Amazon RDS. Para ter mais informações, consulte Introdução aos backups.

Tarefa 5: redirecionar a aplicação ativa para a instância do Amazon RDS

Depois que o banco de dados MariaDB ou MySQL estiver atualizado com a instância de replicação de origem, será possível atualizar a aplicação ativa para usar a instância do Amazon RDS.



Como redirecionar a aplicação dinâmica para o banco de dados MariaDB ou MySQL e interromper a replicação

1. Para adicionar o grupo de segurança de VPC para o banco de dados do Amazon RDS, adicione o endereço IP do servidor que hospeda a aplicação. Para ter mais informações sobre como modificar um grupo de segurança de VPC, consulte Configurar regras de grupo de segurança no Manual do usuário da Amazon Virtual Private Cloud.
2. Verifique se o campo `Seconds_Behind_Master` nos resultados do comando SHOW REPLICA STATUS é 0, o que indica que a réplica está atualizada com a instância de replicação de origem.

```
SHOW REPLICA STATUS;
```

> **Note**
>
> As versões anteriores do MySQL usavam `SHOW SLAVE STATUS` em vez de `SHOW REPLICA STATUS`. Se você estiver usando uma versão do MySQL anterior à 8.0.23, use `SHOW SLAVE STATUS`.

Para uma instância de banco de dados MariaDB 10.5, 10.6 ou 10.11, execute o procedimento mysql.rds_replica_status em vez do comando MySQL.

3. Feche todas as conexões com a origem quando suas transações terminarem.
4. Atualize sua aplicação para usar o banco de dados do Amazon RDS. Normalmente, essa atualização envolve a alteração das configurações de conexão para identificar o nome do host e a porta do banco de dados do Amazon RDS, a conta de usuário e a senha para conexão e o banco de dados para uso.
5. Conecte-se à instância de banco de dados.

   No caso de um cluster de banco de dados multi-AZ, conecte-se a uma instância de banco de dados gravadora.
6. Interrompa a replicação da instância do Amazon RDS usando o comando mysql.rds_stop_replication abaixo:

   ```
   CALL mysql.rds_stop_replication;
   ```
7. Execute o procedimento armazenado apropriado do Amazon RDS no banco de dados do Amazon RDS para redefinir a configuração de replicação e fazer com que a instância não seja mais identificada como uma réplica.

   - mysql.rds_reset_external_master (RDS para MariaDB e RDS para MySQL versões principais 8.0 e anteriores)
   - mysql.rds_reset_external_source (RDS para MySQL versões principais 8.4 e posteriores)

   MySQL 8.4 e versões posteriores

   ```
   CALL mysql.rds_reset_external_source;
   ```

   MariaDB e MySQL 8.0 e versões anteriores

   ```
   CALL mysql.rds_reset_external_master;
   ```
8. Ative recursos adicionais do Amazon RDS, como o suporte a multi-AZ e réplicas de leitura. Para obter mais informações, consulte Configurar e gerenciar uma implantação multi-AZ para o Amazon RDS e Trabalhar com réplicas de leitura de instância de banco de dados.

Importar dados de qualquer fonte para uma instância de banco de dados do Amazon RDS para MySQL

As seguintes etapas explicam como importar dados para uma instância de banco de dados do Amazon RDS:

Etapa 1: Criar arquivos simples contendo os dados a serem carregados

Use um formato comum, como valores separados por vírgula (CSV), para armazenar os dados a serem carregados. Cada tabela deve ter um arquivo próprio. Não é possível combinar os dados de várias tabelas no mesmo arquivo. Dê a cada arquivo o mesmo nome que a tabela correspondente. A extensão do arquivo pode ser qualquer opção desejada. Por exemplo, se o nome da tabela for `sales`, o nome do arquivo poderá ser `sales.csv` ou `sales.txt`.

Se possível, ordene os dados pela chave primária da tabela que está sendo carregada. Fazer isso melhora drasticamente os tempos de carregamento e minimiza os requisitos de armazenamento em disco.

A velocidade e a eficiência desse procedimento dependem de manter o tamanho dos arquivos pequenos. Se o tamanho descompactado de qualquer arquivo individual for maior que 1 GiB, divida-o em vários arquivos e carregue cada um separadamente.

Em sistemas semelhantes ao Unix (incluindo o Linux), use o comando `split`. Por exemplo, o seguinte comando divide o arquivo `sales.csv` em vários arquivos com menos de 1 GiB, dividindo apenas em intervalos de linha (-C 1024m). Os nomes dos novos arquivos incluem sufixos numéricos ascendentes. O comando a seguir produz arquivos com nomes, como `sales.part_00` e `sales.part_01`.

```
split -C 1024m -d sales.csv sales.part_
```

Utilitários semelhantes estão disponíveis para outros sistemas operacionais.

Você pode armazenar os arquivos simples em qualquer lugar. No entanto, ao carregar os dados na Etapa 5, você deve invocar o shell `mysql` do mesmo local em que os arquivos estão ou usar o caminho absoluto para os arquivos ao executar `LOAD DATA LOCAL INFILE`.

Etapa 2: impedir que quaisquer aplicações acessem a instância de banco de dados de destino

Antes de iniciar um grande carregamento, impeça que qualquer atividade da aplicação acesse a instância de banco de dados de destino para a qual planeja fazer o carregamento. Recomendamos isso especialmente se outras sessões modificarem as tabelas que estão sendo carregadas ou as tabelas às quais elas fazem referência. Isso reduz o risco de violações de restrição durante o carregamento e melhora a performance do carregamento. Também possibilita restaurar a instância de banco de dados até o ponto antes do carregamento, sem perder as alterações feitas pelos processos não envolvidos no carregamento.

É que isso pode não ser possível ou prático. Se você não conseguir impedir que as aplicações acessem a instância de banco de dados antes do carregamento, tome medidas para garantir a disponibilidade e a integridade dos seus dados. As etapas específicas necessárias variam muito dependendo dos casos específicos de uso e dos requisitos do site.

Etapa 3: Criar um snapshot de banco de dados

Se você planeja carregar dados em uma nova instância de banco de dados que não contém dados, pode ignorar essa etapa. Do contrário, recomendamos criar snapshots de banco de dados da instância de banco de dados de destino do Amazon RDS antes e depois do carregamento dos dados. Os snapshots de banco de dados do Amazon RDS são backups completos da instância de banco de dados que podem ser usados para restaurá-la em um estado conhecido. Quando você inicia um snapshot de banco de dados, as operações de E/S na sua instância de banco de dados são momentaneamente suspensas enquanto é feito o backup do banco de dados.

Criar um snapshot de banco de dados imediatamente antes do carregamento possibilita a restauração do banco de dados para o estado anterior ao carregamento, se necessário. Um snapshot de banco de dados feito imediatamente após a carga evita que você tenha que carregar os dados novamente caso ocorra um problema. Você também pode usar snapshots de banco de dados após o carregamento para importar dados para novas instâncias de banco de dados.

O exemplo a seguir executa o comando create-db-snapshot da AWS CLI para criar um snapshot de banco de dados da instância `AcmeRDS` e fornece ao snapshot do banco de dados o identificador `"preload"`.

Para Linux, macOS ou Unix:

```
aws rds create-db-snapshot \
    --db-instance-identifier AcmeRDS \
```

```
    --db-snapshot-identifier preload
```

Para Windows:

```
aws rds create-db-snapshot ^
    --db-instance-identifier AcmeRDS ^
    --db-snapshot-identifier preload
```

Você também pode usar a restauração da funcionalidade de snapshot do banco de dados para criar instâncias de banco de dados de teste para execuções simuladas ou para desfazer alterações feitas durante o carregamento.

Tenha em mente que restaurar um banco de dados de um snapshot de banco de dados cria uma nova instância de banco de dados que, como todas as instâncias de banco de dados, possui um identificador exclusivo e um endpoint. Para restaurar a instância de banco de dados sem alterar o endpoint, primeiro exclua a instância de banco de dados para poder reutilizar o endpoint.

Por exemplo, para criar uma instância de banco de dados para execuções simuladas ou outros testes, forneça à instância de banco de dados um identificador próprio. Na exemplo, o identificador é `AcmeRDS-2`. O exemplo se conecta à instância de banco de dados usando o endpoint associado ao `AcmeRDS-2`. Para ter mais informações, consulte restore-db-instance-from-db-snapshot.

Para Linux, macOS ou Unix:

```
aws rds restore-db-instance-from-db-snapshot \
    --db-instance-identifier AcmeRDS-2 \
    --db-snapshot-identifier preload
```

Para Windows:

```
aws rds restore-db-instance-from-db-snapshot ^
    --db-instance-identifier AcmeRDS-2 ^
    --db-snapshot-identifier preload
```

Para reutilizar o endpoint existente, primeiro exclua a instância de banco de dados e forneça ao banco de dados restaurado o mesmo identificador. Para ter mais informações, consulte delete-db-instance.

O exemplo a seguir também cria um snapshot de banco de dados final da instância de banco de dados antes de excluí-la. Isso é opcional, porém é recomendado.

Para Linux, macOS ou Unix:

```
aws rds delete-db-instance \
    --db-instance-identifier AcmeRDS \
    --final-db-snapshot-identifier AcmeRDS-Final

aws rds restore-db-instance-from-db-snapshot \
    --db-instance-identifier AcmeRDS \
    --db-snapshot-identifier preload
```

Para Windows:

```
aws rds delete-db-instance ^
    --db-instance-identifier AcmeRDS ^
    --final-db-snapshot-identifier AcmeRDS-Final

aws rds restore-db-instance-from-db-snapshot ^
    --db-instance-identifier AcmeRDS ^
    --db-snapshot-identifier preload
```

Etapa 4 (opcional): desativar os backups automatizados do Amazon RDS

> **Warning**
>
> Não desative os backups automatizados se você precisa realizar uma recuperação para um ponto no tempo.

A desativação de backups automatizados ajuda a otimizar a performance e não é necessária para carregamentos de dados. Ao desativar os backups automatizados, todos os backups existentes são apagados. Consequentemente, após a desativação dos backups automatizados, não é possível realizar uma recuperação para um ponto no tempo. Os snapshots do banco de dados manuais não são afetados com a desativação dos backups automatizados. Todos os snapshots de banco de dados manuais existentes ainda estão disponíveis para restauração.

Desativar os backups automatizados reduz o tempo de carregamento em cerca de 25% e reduz a quantidade de espaço de armazenamento necessária durante o carregamento. Se você planeja carregar dados em uma nova instância de banco de dados que não contém dados, desativar os backups é uma maneira fácil de acelerar o carregamento e evitar o uso do armazenamento adicional

necessário para backups. No entanto, em alguns casos, você pode planejar fazer o carregamento em uma instância de banco de dados que já contenha dados. Se for o caso, avalie os benefícios da desativação de backups em relação ao impacto da perda da capacidade de realizar a recuperação em um ponto anterior no tempo.

As instâncias de bancos de dados têm backups automatizados desativados por padrão (com um período de retenção de um dia). Para desatiar os backups automáticos, defina o período de retenção de backup para zero. Após o carregamento, você pode reativar os backups definindo o período de retenção de backup como um valor diferente de zero. Para ativar ou desativar os backups, o Amazon RDS desliga a instância de banco de dados e a reinicia para habilitar ou desabilitar o registro em log do MariaDB ou do MySQL.

Execute o comando AWS CLI da `modify-db-instance` para definir a retenção do backup como zero e aplicar a alteração imediatamente. Definir o período de retenção como zero exige uma reinicialização da instância de banco de dados. Por isso, aguarde até que a reinicialização tenha sido concluída antes de prosseguir. Para obter mais informações, consulte modify-db-instance.

Para Linux, macOS ou Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier AcmeRDS \
    --apply-immediately \
    --backup-retention-period 0
```

Para Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier AcmeRDS ^
    --apply-immediately ^
    --backup-retention-period 0
```

Você pode verificar o status sua instância de banco de dados com o comando describe-db-instances. O seguinte exemplo mostra o status da instância de banco de dados `AcmeRDS`:

```
aws rds describe-db-instances --db-instance-identifier AcmeRDS --query "*[].
{DBInstanceStatus:DBInstanceStatus}"
```

Quando o status da instância de banco de dados for `available`, você poderá prosseguir para a próxima etapa.

Etapa 5: Carregar os dados

Para ler as linhas dos arquivos simples nas tabelas do banco de dados, use a instrução MySQL `LOAD DATA LOCAL INFILE`.

> **Note**
>
> Você deve invocar o shell `mysql` do mesmo local em que seus arquivos simples estão ou usar o caminho absoluto para os arquivos ao executar `LOAD DATA LOCAL INFILE`.

O seguinte exemplo mostra como carregar dados de um arquivo chamado `sales.txt` em uma tabela denominada `Sales` no banco de dados:

```
mysql> LOAD DATA LOCAL INFILE 'sales.txt' INTO TABLE Sales FIELDS TERMINATED BY ' '
 ENCLOSED BY '' ESCAPED BY '\\';
Query OK, 1 row affected (0.01 sec)
Records: 1  Deleted: 0  Skipped: 0  Warnings: 0
```

Para ter mais informações sobre a instrução `LOAD DATA`, consulte LOAD DATA Statement na documentação do MySQL.

Etapa 6: reativar os backups automatizados do Amazon RDS

Se você desativou os backups automatizados do Amazon RDS na Etapa 4, depois que o carregamento for concluído, ative os backups automatizados definindo o período de retenção de backup de volta para o valor anterior ao carregamento. Conforme mencionado na Etapa 4, como o Amazon RDS reinicia a instância de banco de dados, prepare-se para uma breve interrupção.

O seguinte exemplo executa o comando modify-db-instance da AWS CLI para ativar os backups automatizados da instância de banco de dados `AcmeRDS` e define o período de retenção para um dia:

Para Linux, macOS ou Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier AcmeRDS \
    --backup-retention-period 1 \
    --apply-immediately
```

Para Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier AcmeRDS ^
    --backup-retention-period 1 ^
    --apply-immediately
```

Trabalhar com a replicação do MySQL no Amazon RDS

Normalmente, você usa réplicas de leitura para configurar a replicação entre as instâncias de banco de dados do Amazon RDS. Para obter informações gerais sobre réplicas de leitura, consulte Trabalhar com réplicas de leitura de instância de banco de dados. Para obter informações específicas sobre como trabalhar com réplicas de leitura no Amazon RDS para MySQL, consulte Trabalhar com réplicas de leitura MySQL.

Você pode utilizar Global Transaction Identifiers (GTIDs – Identificadores de transação global) para replicação com o RDS para MySQL. Para obter mais informações, consulteUsar a replicação baseada em GTID

A replicação também pode ser configurada entre uma instância de banco de dados do RDS para MySQL e uma instância MySQL ou MariaDB externa ao Amazon RDS. Para obter informações sobre como configurar a replicação com uma origem externa, consulte Configurar a replicação da posição do arquivo de log binário com uma instância de origem externa.

Para todas essas opções de replicação, use replicação baseada em linha, em instrução ou mista. A replicação baseada em linha só replica as linhas alteradas resultantes de uma instrução SQL. A replicação baseada em instrução copia toda a instrução SQL. A replicação mista usa uma replicação baseada em instrução quando possível, mas alterna para replicação baseada em linha quando as instruções SQL não seguras para replicação baseada em instrução são executadas. Na maioria dos casos, a replicação mista é recomendada. O formato de log binário da instância de banco de dados determina se a replicação se baseia na linha, na instrução ou mista. Para obter informações sobre como definir o formato de log binário, consulte Configurar o registro em log binário do RDS para MySQL para bancos de dados single-AZ.

> **Note**
>
> Você pode configurar a replicação para importar bancos de dados de uma instância MySQL ou MariaDB externa ao Amazon RDS, ou exportar bancos de dados para essas instâncias. Para ter mais informações, consulte Importar dados para um banco de dados do Amazon RDS para MySQL com tempo de inatividade reduzido e Exportar dados de uma instância de banco de dados MySQL usando replicação.

Depois de restaurar a instância de banco de dados com base em um snapshot ou realizar uma recuperação para um ponto no tempo, é possível visualizar a última posição do log binário

recuperada do banco de dados de origem no console do RDS. Em Logs e eventos, insira log binário. A posição do log binário aparece em Notas do sistema.

Tópicos

- Trabalhar com réplicas de leitura MySQL
- Usar a replicação baseada em GTID
- Configurar a replicação da posição do arquivo de log binário com uma instância de origem externa
- Configurar a replicação de várias fontes do Amazon RDS para MySQL

Trabalhar com réplicas de leitura MySQL

A seguir, você pode encontrar informações específicas sobre como trabalhar com réplicas de leitura no RDS para MySQL. Para obter informações gerais sobre as réplicas de leitura e as instruções de como usá-las, consulte Trabalhar com réplicas de leitura de instância de banco de dados.

Para ter mais informações sobre réplicas de leitura do MySQL, consulte os tópicos a seguir.

- Configurar filtros de replicação com MySQL
- Configurar replicação atrasada com o MySQL
- Atualizar réplicas de leitura com o MySQL
- Trabalhar com implantações de réplicas de leitura multi-AZ com o MySQL
- Usar réplicas de leitura em cascata com o RDS para MySQL
- Monitorar o atraso de replicação para réplicas de leitura do MySQL
- Início e interrupção de replicação com réplicas de leitura do MySQL
- Solucionar problemas de uma réplica de leitura do MySQL

Configurar réplicas de leitura com o MySQL

Antes que uma instância de banco de dados do MySQL possa servir como uma fonte de replicação, certifique-se de habilitar os backups automáticos na instância de banco de dados de origem. Para fazer isso, defina o período de retenção de backup como um valor diferente de 0. Esse requisito também se aplica a uma réplica de leitura que seja a instância de banco de dados de origem de outra réplica de leitura. Os backups automáticos são compatíveis somente com réplicas de leitura que executam qualquer versão do MySQL. Você pode configurar a replicação com base nas coordenadas do log binário para uma instância de banco de dados MySQL.

Você pode configurar a replicação usando identificadores de transação global (GTIDs) nas seguintes versões:

- RDS para MySQL versão 5.7.44 e versões 5.7 posteriores
- RDS para MySQL versão 8.0.28 e versões 8.0 posteriores
- RDS para MySQL versão 8.4.3 e versões 8.4 posteriores

Para ter mais informações, consulte Usar a replicação baseada em GTID.

Você pode criar até quinze réplicas de leitura de uma instância de banco de dados na mesma região. Para que a replicação funcione efetivamente, cada réplica de leitura deve ter a mesma quantidade de recursos de computação e de armazenamento que a instância de banco de dados de origem. Se você dimensionar a instância de banco de dados de origem, dimensione as réplicas de leitura também.

O RDS para MySQL é compatível com réplicas de leitura em cascata. Para saber mais sobre como configurar réplicas de leitura em cascata, consulte Usar réplicas de leitura em cascata com o RDS para MySQL.

É possível executar várias ações simultâneas de criação ou exclusão de réplicas de leitura que fazem referência à mesma instância de banco de dados de origem. Ao realizar essas ações, permaneça dentro do limite de quinze réplicas de leitura para cada instância de origem.

Uma réplica de leitura de uma instância de banco de dados MySQL não pode usar uma versão de mecanismo de banco de dados inferior à sua instância de banco de dados de origem.

Preparar instâncias de banco de dados MySQL que usam MyISAM

Se sua instância de banco de dados MySQL usar um mecanismo não transacional, como o MyISAM, será necessário realizar as etapas a seguir para configurar sua réplica de leitura com sucesso. Essas etapas são necessárias para garantir que a réplica de leitura tenha uma cópia consistente de seus dados. Essas etapas não são necessárias caso todas as suas tabelas utilizem um mecanismo transacional como InnoDB.

1. Interrompa todas as operações de linguagem de manipulação de dados (DML) e de linguagem de definição de dados (DDL) em tabelas não transacionais na instância de banco de dados de origem e aguarde pela conclusão delas. Comandos SELECIONAR podem continuar sendo executados.
2. Descarregue e bloqueie as tabelas na instância do banco de dados de origem.
3. Crie a réplica de leitura usando um dos métodos nas seções a seguir.

4. Verifique o andamento da criação da réplica de leitura usando, por exemplo, a operação da API `DescribeDBInstances`. Após a réplica de leitura estar disponível, desbloqueie as tabelas da instância de banco de dados de origem e retome as operações de banco de dados normais.

Configurar filtros de replicação com MySQL

Você pode usar filtros de replicação para especificar quais bancos de dados e tabelas são replicados com uma réplica de leitura. Os filtros de replicação podem incluir bancos de dados e tabelas na replicação ou excluí-los da replicação.

Veja a seguir alguns casos de uso para filtros de replicação:

- Para reduzir o tamanho de uma réplica de leitura. Com a filtragem de replicação, você pode excluir os bancos de dados e tabelas que não são necessários na réplica de leitura.
- Para excluir bancos de dados e tabelas de réplicas de leitura por motivos de segurança.
- Para replicar diferentes bancos de dados e tabelas para casos de uso específicos em diferentes réplicas de leitura. Por exemplo, você pode usar réplicas de leitura específicas para análise ou fragmentação.
- Para uma instância de banco de dados que tenha réplicas de leitura em diferentes Regiões da AWS, para replicar diferentes bancos de dados ou tabelas em diferentes Regiões da AWS.

> **Note**
>
> É possível usar filtros de replicação para especificar quais bancos de dados e tabelas serão replicados com uma instância de banco de dados primária do MySQL configurada como uma réplica em uma topologia de replicação de entrada. Para obter mais informações sobre essa configuração, consulte Configurar a replicação da posição do arquivo de log binário com uma instância de origem externa.

Tópicos

- Configurar parâmetros de filtragem de replicação para o RDS para MySQL
- Limitações de filtragem de replicação do RDS para MySQL
- Exemplos de filtragem de replicação do RDS para MySQL
- Visualizar os filtros de replicação para uma réplica de leitura

Configurar parâmetros de filtragem de replicação para o RDS para MySQL

Para configurar filtros de replicação, defina os seguintes parâmetros de filtragem de replicação na réplica de leitura:

- `replicate-do-db` – Replicar alterações nos bancos de dados especificados. Quando você define esse parâmetro para uma réplica de leitura, somente os bancos de dados especificados no parâmetro são replicados.
- `replicate-ignore-db` – Não replique as alterações nos bancos de dados especificados. Quando o parâmetro `replicate-do-db` é definido para uma réplica de leitura, esse parâmetro não é avaliado.
- `replicate-do-table` – Replicar alterações nas tabelas especificadas. Quando você define esse parâmetro para uma réplica de leitura, somente as tabelas especificadas no parâmetro são replicadas. Além disso, quando o parâmetro `replicate-do-db` ou `replicate-ignore-db` estiver definido, certifique-se de incluir o banco de dados que inclui as tabelas especificadas na replicação com a réplica de leitura.
- `replicate-ignore-table` – Não replique as alterações nas tabelas especificadas. Quando o parâmetro `replicate-do-table` é definido para uma réplica de leitura, esse parâmetro não é avaliado.
- `replicate-wild-do-table` – Replicar tabelas com base nos padrões de nome de banco de dados e tabela especificados. Os caracteres curinga `%` e `_` são compatíveis. Quando o parâmetro `replicate-do-db` ou `replicate-ignore-db` estiver definido, certifique-se de incluir o banco de dados que inclui as tabelas especificadas na replicação com a réplica de leitura.
- `replicate-wild-ignore-table` – Não replique tabelas com base nos padrões de nome de banco de dados e tabela especificados. Os caracteres curinga `%` e `_` são compatíveis. Quando o parâmetro `replicate-do-table` ou `replicate-wild-do-table` é definido para uma réplica de leitura, esse parâmetro não é avaliado.

Os parâmetros são avaliados na ordem em que estão listados. Para obter mais informações sobre como esses parâmetros funcionam, consulte a documentação do MySQL:

- Para obter informações gerais, consulte Opções e variáveis do servidor de réplica.
- Para obter informações sobre como os parâmetros de filtragem de replicação de banco de dados são avaliados, consulte Avaliação de opções de replicação em nível de banco de dados e log binário.

- Para obter informações sobre como os parâmetros de filtragem de replicação de tabela são avaliados, consulte Avaliação de opções de replicação em nível de tabela.

Por padrão, cada um desses parâmetros tem um valor vazio. Em cada réplica de leitura, você pode usar esses parâmetros para definir, alterar e excluir filtros de replicação. Quando você define um desses parâmetros, separe cada filtro dos outros com uma vírgula.

Você pode usar `%` os caracteres curinga `_` e nos parâmetros `replicate-wild-do-table` e `replicate-wild-ignore-table`. O curinga `%` corresponde a qualquer número de caracteres e o caractere curinga `_` corresponde apenas a um caractere.

O formato de log binário da instância de banco de dados de origem é importante para replicação porque determina o registro de alterações de dados. A configuração do parâmetro `binlog_format` determina se a replicação é baseada em linha ou baseada em declaração. Para obter mais informações, consulte Configurar o registro em log binário do RDS para MySQL para bancos de dados single-AZ.

> **Note**
>
> Todas as instruções DDL (Data Definition Language, linguagem de definição de dados) são replicadas como instruções, independentemente da `binlog_format` configuração na instância de banco de dados de origem.

Limitações de filtragem de replicação do RDS para MySQL

As seguintes limitações aplicam-se à filtragem de replicação para o RDS para MySQL:

- Cada parâmetro de filtragem de replicação tem um limite de 2.000 caracteres.
- As vírgulas não são aceitas em filtros de replicação para valores de parâmetros. Em uma lista de parâmetros, as vírgulas só podem ser usadas como separadores de valores. Por exemplo, `ParameterValue='`a,b`'` não é aceito, mas `ParameterValue='a,b'` é.
- O MySQL `--binlog-do-db` e `--binlog-ignore-db` as opções para filtragem de log binário não são compatíveis.
- A filtragem de replicação não suporta transações XA.

  Para obter mais informações, consulte Restrictions on XA Transactions na documentação do MySQL.

Exemplos de filtragem de replicação do RDS para MySQL

Para configurar a filtragem de replicação para uma réplica de leitura, modifique os parâmetros de filtragem de replicação no grupo de parâmetros associado à réplica de leitura.

> **Note**
>
> Não é possível modificar um grupo de parâmetros padrão. Se a réplica de leitura estiver usando um grupo de parâmetros padrão, crie um novo grupo de parâmetros e o associe à instância de banco de dados. Para obter mais informações sobre grupos de parâmetros de banco de dados, consulte Grupos de parâmetros para Amazon RDS.

Você pode definir parâmetros em um grupo de parâmetros usando a AWS Management Console, a AWS CLI ou a API do RDS. Para obter informações sobre como configurar parâmetros, consulte Modificar parâmetros em um grupo de parâmetros de banco de dados no Amazon RDS. Quando você define parâmetros em um grupo de parâmetros, todas as instâncias de banco de dados associadas ao grupo de parâmetros usam as configurações de parâmetro. Se você definir os parâmetros de filtragem de replicação em um grupo de parâmetros, verifique se o grupo de parâmetros está associado apenas a réplicas de leitura. Deixe os parâmetros de filtragem de replicação vazios para instâncias de banco de dados de origem.

Os exemplos a seguir definem os parâmetros usando o AWS CLI. Estes exemplos definem `ApplyMethod` para `immediate` de modo que as mudanças do parâmetro ocorram imediatamente depois que o comando CLI termina. Se você quiser que uma alteração pendente seja aplicada depois que a réplica de leitura for reinicializada, defina como `ApplyMethod` `pending-reboot`.

Os exemplos a seguir definem filtros de replicação:

- Including databases in replication
- Including tables in replication
- Including tables in replication with wildcard characters
- Excluding databases from replication
- Excluding tables from replication
- Excluding tables from replication using wildcard characters

Example Incluir bancos de dados em replicação

O exemplo a seguir inclui os bancos de dados `mydb1` e `mydb2` na replicação.

Para Linux, macOS ou Unix:

```
aws rds modify-db-parameter-group \
  --db-parameter-group-name myparametergroup \
  --parameters "ParameterName=replicate-do-
db,ParameterValue='mydb1,mydb2',ApplyMethod=immediate"
```

Para Windows:

```
aws rds modify-db-parameter-group ^
  --db-parameter-group-name myparametergroup ^
  --parameters "ParameterName=replicate-do-
db,ParameterValue='mydb1,mydb2',ApplyMethod=immediate"
```

Example Incluir tabelas na replicação

O exemplo a seguir inclui as tabelas `table1` e `table2` no banco de dados `mydb1` na replicação.

Para Linux, macOS ou Unix:

```
aws rds modify-db-parameter-group \
  --db-parameter-group-name myparametergroup \
  --parameters "ParameterName=replicate-do-
table,ParameterValue='mydb1.table1,mydb1.table2',ApplyMethod=immediate"
```

Para Windows:

```
aws rds modify-db-parameter-group ^
  --db-parameter-group-name myparametergroup ^
  --parameters "ParameterName=replicate-do-
table,ParameterValue='mydb1.table1,mydb1.table2',ApplyMethod=immediate"
```

Example Incluir tabelas na replicação usando caracteres curinga

O exemplo a seguir inclui tabelas com nomes que começam com `order` e `return` no banco de dados `mydb` na replicação.

Para Linux, macOS ou Unix:

```
aws rds modify-db-parameter-group \
  --db-parameter-group-name myparametergroup \
  --parameters "ParameterName=replicate-wild-do-table,ParameterValue='mydb.order
%,mydb.return%',ApplyMethod=immediate"
```

Para Windows:

```
aws rds modify-db-parameter-group ^
  --db-parameter-group-name myparametergroup ^
  --parameters "ParameterName=replicate-wild-do-table,ParameterValue='mydb.order
%,mydb.return%',ApplyMethod=immediate"
```

Example Excluir bancos de dados da replicação

O exemplo a seguir exclui os bancos de dados `mydb5` e `mydb6` da replicação.

Para Linux, macOS ou Unix:

```
aws rds modify-db-parameter-group \
  --db-parameter-group-name myparametergroup \
  --parameters "ParameterName=replicate-ignore-
db,ParameterValue='mydb5,mydb6',ApplyMethod=immediate"
```

Para Windows:

```
aws rds modify-db-parameter-group ^
  --db-parameter-group-name myparametergroup ^
  --parameters "ParameterName=replicate-ignore-
db,ParameterValue='mydb5,mydb6',ApplyMethod=immediate"
```

Example Excluir tabelas da replicação

O exemplo a seguir exclui a tabela `table1` no banco de dados `mydb5` e a tabela `table2` no banco de dados `mydb6` da replicação.

Para Linux, macOS ou Unix:

```
aws rds modify-db-parameter-group \
  --db-parameter-group-name myparametergroup \
```

```
  --parameters "ParameterName=replicate-ignore-
table,ParameterValue='mydb5.table1,mydb6.table2',ApplyMethod=immediate"
```

Para Windows:

```
aws rds modify-db-parameter-group ^
  --db-parameter-group-name myparametergroup ^
  --parameters "ParameterName=replicate-ignore-
table,ParameterValue='mydb5.table1,mydb6.table2',ApplyMethod=immediate"
```

Example Excluir tabelas da replicação usando caracteres curinga

O exemplo a seguir exclui tabelas com nomes que começam com `order` e `return` no banco de dados `mydb7` da replicação.

Para Linux, macOS ou Unix:

```
aws rds modify-db-parameter-group \
  --db-parameter-group-name myparametergroup \
  --parameters "ParameterName=replicate-wild-ignore-table,ParameterValue='mydb7.order
%,mydb7.return%',ApplyMethod=immediate"
```

Para Windows:

```
aws rds modify-db-parameter-group ^
  --db-parameter-group-name myparametergroup ^
  --parameters "ParameterName=replicate-wild-ignore-table,ParameterValue='mydb7.order
%,mydb7.return%',ApplyMethod=immediate"
```

Visualizar os filtros de replicação para uma réplica de leitura

Você pode visualizar os filtros de replicação de uma réplica de leitura das seguintes maneiras:

- Verifique as configurações dos parâmetros de filtragem de replicação no grupo de parâmetros associado à réplica de leitura.

  Para obter instruções, consulte Visualizar valores de parâmetros para um grupo de parâmetros de banco de dados no Amazon RDS.
- Em um cliente MySQL, conecte-se à réplica de leitura e execute a instrução `SHOW REPLICA STATUS`.

Na saída, os campos a seguir mostram os filtros de replicação para a réplica de leitura:

- `Replicate_Do_DB`
- `Replicate_Ignore_DB`
- `Replicate_Do_Table`
- `Replicate_Ignore_Table`
- `Replicate_Wild_Do_Table`
- `Replicate_Wild_Ignore_Table`

Para obter mais informações sobre esses campos, consulte Verificar o status da replicação na documentação do MySQL.

Configurar replicação atrasada com o MySQL

Você pode usar a replicação atrasada como uma estratégia para a recuperação de desastres. Com a replicação atrasada, você especifica o tempo mínimo, em segundos, para atrasar a replicação da origem para a réplica de leitura. Em caso de um desastre, como uma tabela excluída acidentalmente, você executa as seguintes etapas para recuperar-se rapidamente do desastre:

- Interrompa a replicação da réplica de leitura antes que a alteração que causou o desastre seja enviada para ela.

  Use o procedimento armazenado mysql.rds_stop_replication para interromper a replicação.
- Inicie a replicação e especifique que a replicação deve ser interrompida automaticamente em um local do arquivo de log.

  Especifique um local imediatamente antes do desastre usando o procedimento armazenado mysql.rds_start_replication_until.
- Promova a réplica de leitura para ser a nova instância de banco de dados de origem usando as instruções em Promoção de uma réplica de leitura a uma instância de banco de dados autônoma.

> **Note**
>
> - No RDS para MySQL 8.4, a replicação atrasada tem suporte para o MySQL 8.4.3 e versões posteriores. No RDS para MySQL 8.0, a replicação atrasada tem suporte para o

MySQL 8.0.28 e versões posteriores. No RDS para MySQL 5.7, a replicação atrasada é compatível com o MySQL 5.7.44 e posterior.

- Use procedimentos armazenados para configurar a replicação atrasada. Você não pode configurar a replicação atrasada com o AWS Management Console, a AWS CLI ou a API do Amazon RDS.
- Você pode usar a replicação com base em identificadores de transação global (GTIDs) em uma configuração de replicação atrasada para as seguintes versões:
  - RDS para MySQL versão 5.7.44 e versões 5.7 posteriores
  - RDS para MySQL versão 8.0.28 e versões 8.0 posteriores
  - RDS para MySQL versão 8.4.3 e versões 8.4 posteriores

  Caso você use a replicação baseada em GTID, use o procedimento armazenado mysql.rds_start_replication_until_gtid em vez do procedimento armazenado mysql.rds_start_replication_until. Para obter mais informações sobre a replicação baseada em GTID, consulte Usar a replicação baseada em GTID.

Tópicos

- Configurar replicação atrasada durante a criação da réplica de leitura
- Modificar replicação atrasada de uma réplica de leitura existente
- Configurar um local para interromper a replicação para uma réplica de leitura
- Promover uma réplica de leitura

Configurar replicação atrasada durante a criação da réplica de leitura

Para configurar a replicação atrasada para qualquer réplica de leitura futura criada a partir de uma instância de banco de dados, execute o procedimento armazenado mysql.rds_set_configuration com o parâmetro `target delay`.

Para configurar a replicação atrasada durante a criação da réplica de leitura

1. Usando um cliente MySQL, conecte-se à instância de banco de dados MySQL que será a origem para réplicas de leitura como o usuário mestre.
2. Execute o procedimento armazenado mysql.rds_set_configuration com o parâmetro `target delay`.

Por exemplo, execute o procedimento armazenado a seguir para especificar que a replicação é atrasada em pelo menos uma hora (3,600 segundos) para qualquer réplica de leitura criada a partir da instância de banco de dados atual.

```
call mysql.rds_set_configuration('target delay', 3600);
```

> **Note**
>
> Após executar esse procedimento armazenado, qualquer réplica de leitura que você criar usando a AWS CLI ou a API do Amazon RDS será configurada com a replicação atrasada pelo número de segundos especificado.

Modificar replicação atrasada de uma réplica de leitura existente

Para modificar a replicação atrasada para uma réplica de leitura existente, execute o procedimento armazenado mysql.rds_set_source_delay.

Para modificar a replicação atrasada para uma réplica de leitura existente

1. Usando um cliente do MySQL, conecte-se à réplica de leitura como o usuário mestre.
2. Use o procedimento armazenado mysql.rds_stop_replication para interromper a replicação.
3. Execute o procedimento armazenado mysql.rds_set_source_delay.

   Por exemplo, execute o procedimento armazenado a seguir para especificar que a replicação para a réplica de leitura é atrasada em pelo menos uma hora (3.600 segundos) para qualquer réplica de leitura criada a partir da instância de banco de dados atual.

   ```
   call mysql.rds_set_source_delay(3600);
   ```
4. Use o procedimento armazenado mysql.rds_start_replication para iniciar a replicação.

Configurar um local para interromper a replicação para uma réplica de leitura

Após interromper a replicação para a réplica de leitura, você pode começar a replicação e interrompê-la em um local especificado do arquivo de log binário usando o procedimento armazenado mysql.rds_start_replication_until.

Para iniciar a replicação para uma Réplica de leitura e interrompê-la em um local específico

1. Usando um cliente do MySQL, conecte-se à instância de banco de dados MySQL de origem como o usuário mestre.
2. Execute o procedimento armazenado mysql.rds_start_replication_until.

   O exemplo a seguir inicia a replicação e replica as alterações até que ela atinja o local `120` no arquivo de log binário `mysql-bin-changelog.000777`. Em um cenário de recuperação de desastres, suponha que o local `120` é imediatamente antes do desastre.

```
call mysql.rds_start_replication_until(
  'mysql-bin-changelog.000777',
  120);
```

A replicação é interrompida automaticamente quando o ponto de interrupção é atingido. O seguinte evento do RDS é gerado: `Replication has been stopped since the replica reached the stop point specified by the rds_start_replication_until stored procedure`.

Promover uma réplica de leitura

Após a replicação ser interrompida, em um cenário de recuperação de desastres, você pode promover uma réplica de leitura para ser a nova instância de banco de dados de origem. Para obter informações sobre como promover uma réplica de leitura, consulte Promoção de uma réplica de leitura a uma instância de banco de dados autônoma.

Atualizar réplicas de leitura com o MySQL

As réplicas de leitura foram projetadas para oferecer suporte a consultas de leitura, mas você pode precisar fazer atualizações ocasionais. Por exemplo, talvez seja necessário adicionar um índice para otimizar tipos específicos de consultas que acessam a réplica.

Embora você possa habilitar as atualizações definindo o parâmetro `read_only` como `0` no grupo de parâmetros de banco de dados para a réplica de leitura, é recomendável não fazer isso, pois essa ação pode causar problemas se a réplica de leitura tornar-se incompatível com a instância de banco de dados de origem. Para operações de manutenção, recomendamos que você use implantações azul/verde. Para ter mais informações, consulte Usar implantações azul/verde para atualizações de banco de dados.

Se você desabilitar a opção somente leitura em uma réplica de leitura, altere o valor do parâmetro `read_only` de volta para 1 o mais rápido possível.

Trabalhar com implantações de réplicas de leitura multi-AZ com o MySQL

É possível criar uma réplica de leitura a partir de implantações de instâncias de banco de dados single-AZ ou multi-AZ. Você pode usar implantações multi-AZ para melhorar a durabilidade e a disponibilidade de dados essenciais. No entanto, não é possível usar o multi-AZ secundário para atender a consultas somente leitura. Em vez disso, crie réplicas de leitura de instâncias de banco de dados multi-AZ de alto tráfego para descarregar consultas somente leitura. Se a instância de origem de uma implantação multi-AZ falhar na secundária, todas as réplicas de leitura associadas serão automaticamente alteradas para usar a secundária (não a primária) como a origem de replicação. Para obter mais informações, consulte Configurar e gerenciar uma implantação multi-AZ para o Amazon RDS.

É possível criar uma réplica de leitura como uma instância de banco de dados multi-AZ. O Amazon RDS cria um em modo de espera de sua réplica em outra zona de disponibilidade para suporte a failover da réplica. Você pode criar a réplica de leitura como uma instância de banco de dados multi-AZ independentemente de o banco de dados de origem ser ou não uma instância de banco de dados multi-AZ.

Usar réplicas de leitura em cascata com o RDS para MySQL

O RDS para MySQL é compatível com réplicas de leitura em cascata. Com réplicas de leitura em cascata, é possível escalar leituras sem adicionar sobrecarga à instância de banco de dados do RDS para MySQL de origem.

Com réplicas de leitura em cascata, sua instância de banco de dados do RDS para MySQL envia dados para a primeira réplica de leitura da cadeia. Essa réplica de leitura envia dados para a segunda réplica na cadeia e assim por diante. O resultado final é que todas as réplicas de leitura na cadeia têm as alterações da instância de banco de dados do RDS para MySQL DB, mas sem a sobrecarga apenas na instância de banco de dados de origem.

É possível criar uma série de até três réplicas de leitura em uma cadeia de uma instância de banco de dados de origem do RDS para MySQL. Por exemplo, suponha que você tenha uma instância de banco de dados do RDS para MySQL, `mysql-main`. Você pode fazer o seguinte:

- Começando com `mysql-main`, crie a primeira réplica de leitura na cadeia, `read-replica-1`.
- Na `read-replica-1`, crie a próxima réplica de leitura na cadeia, `read-replica-2`.

- Finalmente, na `read-replica-2`, crie a terceira réplica de leitura na cadeia, `read-replica-3`.

Não é possível criar outra réplica de leitura além dessa terceira réplica de leitura em cascata na série de `mysql-main`. Uma série completa de instâncias de uma instância de banco de dados de origem do RDS para MySQL até o final de uma série de réplicas de leitura em cascata pode consistir em, no máximo, quatro instâncias de banco de dados.

Para que as réplicas de leitura em cascata funcionem, cada instância de banco de dados do RDS para MySQL de origem deve ter os backups automatizados ativados. Para ativar backups automáticos em uma réplica de leitura, primeiro crie a réplica de leitura e a modifique para ativar backups automáticos. Para ter mais informações, consulte Como criar uma réplica de leitura.

Como em qualquer réplica de leitura, é possível promover uma réplica de leitura que faz parte de uma cascata. A promoção de uma réplica de leitura de uma cadeia de réplicas de leitura remove essa réplica da cadeia. Por exemplo, suponha que você queira mover parte da workload da instância de banco de dados `mysql-main` para uma nova instância para uso somente pelo departamento de contabilidade. Pressupondo a cadeia com três réplicas de leitura do exemplo, você decide promover `read-replica-2`. A cadeia é afetada da seguinte forma:

- A promoção de `read-replica-2` a remove da cadeia de replicação.
  - Ela agora é uma instância de banco de dados de leitura/gravação completa.
  - Ela continua replicando para `read-replica-3`, da mesma forma como estava fazendo antes da promoção.
- A `mysql-main` continua a replicar para a `read-replica-1`.

Para obter mais informações sobre como promover réplicas de leitura, consulte Promoção de uma réplica de leitura a uma instância de banco de dados autônoma.

Monitorar o atraso de replicação para réplicas de leitura do MySQL

Para as réplicas de leitura do MySQL, você pode monitorar o atraso da replicação no Amazon CloudWatch visualizando a métrica `ReplicaLag` do Amazon RDS. A métrica `ReplicaLag` relata o valor do campo `Seconds_Behind_Master` do comando `SHOW REPLICA STATUS`.

As causas comuns para o atraso da replicação do MySQL são as seguintes:

- Uma queda de rede.

- Gravação em tabelas com índices diferentes em uma réplica de leitura. Se o parâmetro `read_only` estiver definido como `0` na réplica de leitura, a replicação poderá ser interrompida se a réplica de leitura torna-se incompatível com a instância de banco de dados de origem. Depois de realizar tarefas de manutenção na réplica de leitura, recomendamos que você defina o parâmetro `read_only` de volta para 1.
- Uso de um mecanismo de armazenamento não transacional, como o MyISAM. A replicação só é compatível com o mecanismo de armazenamento InnoDB no MySQL.

Quando a métrica `ReplicaLag` chega a 0, isso mostra que a réplica alcançou a instância do banco de dados de origem. Se a métrica `ReplicaLag` retornar -1, então a replicação não está ativa no momento. `ReplicaLag` = -1 é equivalente a `Seconds_Behind_Master` = `NULL`.

Início e interrupção de replicação com réplicas de leitura do MySQL

Você pode interromper e reiniciar o processo de replicação em uma instância de banco de dados do Amazon RDS ao chamar os procedimentos armazenados do sistema mysql.rds_stop_replication e mysql.rds_start_replication. Você pode fazer isso ao replicar entre duas instâncias do Amazon RDS para operações de longa duração, como a criação de índices grandes. Você também precisa interromper e iniciar a replicação ao importar ou exportar bancos de dados. Para obter mais informações, consulte Importar dados para um banco de dados do Amazon RDS para MySQL com tempo de inatividade reduzido e Exportar dados de uma instância de banco de dados MySQL usando replicação.

Se a replicação for interrompida por mais de 30 dias consecutivos, seja manualmente ou devido a um erro de replicação, o Amazon RDS a encerrará entre a instância de banco de dados de origem e todas as réplicas de leitura. Isso acontece para evitar um aumento nos requisitos de armazenamento da instância de banco de dados de origem e nos tempos de failover prolongado. A instância de banco de dados da réplica de leitura ainda está disponível. No entanto, a replicação não pode ser retomada porque os logs binários exigidos pela réplica de leitura são excluídos da instância de banco de dados de origem após a conclusão da replicação. Você pode criar uma nova réplica de leitura para a instância de banco de dados de origem a fim de restabelecer a replicação.

Solucionar problemas de uma réplica de leitura do MySQL

Para instâncias de bancos de dados MySQL, em alguns casos as réplicas de leitura apresentam erros de replicação ou inconsistências de dados entre a réplica de leitura e sua fonte da instância de banco de dados. Esse problema ocorre quando alguns eventos de log binário (binlog) ou logs redo do InnoDB não são liberados durante uma falha da réplica de leitura ou da instância de banco

de dados de origem. Nesses casos, exclua e recrie manualmente as réplicas de leitura. Você pode reduzir a chance disso acontecer definindo os seguintes valores de parâmetro: `sync_binlog=1` e `innodb_flush_log_at_trx_commit=1`. Essas configurações podem reduzir a performance, portanto, teste o impacto delas antes de implantar as alterações em um ambiente de produção.

> **Warning**
>
> No grupo de parâmetros associado à fonte da instância de banco de dados, recomendamos manter esses valores de parâmetros: `sync_binlog=1` e `innodb_flush_log_at_trx_commit=1`. Esses parâmetros são dinâmicos. Se você não quiser usar essas configurações, recomendamos definir temporariamente esses valores antes de executar qualquer operação na fonte da instância de banco de dados que possa fazer com que ela seja reiniciada. Essas operações incluem, entre outras, a reinicialização, a reinicialização com failover, a atualização da versão do banco de dados e a alteração da classe de instância de banco de dados ou seu armazenamento. A mesma recomendação se aplica à criação de novas réplicas de leitura para a fonte da instância de banco de dados. Não seguir essa orientação aumenta o risco de réplicas de leitura apresentarem erros de replicação ou inconsistências de dados (ou ambas) entre a réplica de leitura e sua fonte da instância de banco de dados.

As tecnologias de replicação do MySQL são assíncronas. Como são assíncronas, são esperados ocasionais aumentos de `BinLogDiskUsage` na instância de banco de dados de origem e `ReplicaLag` na réplica de leitura. Por exemplo, um volume elevado de operações de gravação para a instância de banco de dados de origem pode ocorrer em paralelo. Por outro lado, as operações de gravação na réplica de leitura são serializadas usando um único thread de E/S, o que pode ocasionar um atraso entre a instância de origem e a réplica de leitura. Para obter mais informações sobre réplicas somente leitura na documentação do MySQL, consulte Detalhes de implantação da replicação.

Você pode fazer várias coisas para reduzir o atraso entre as atualizações de uma instância de banco de dados de origem e as atualizações subsequentes da réplica de leitura, como o seguinte:

- Dimensionar uma réplica de leitura para ter um tamanho de armazenamento e uma categoria de instância de banco de dados comparáveis à da instância de banco de dados de origem.
- Assegurar-se de que as configurações de parâmetros nos grupos de parâmetros de banco de dados utilizados pela instância de banco de dados de origem e pela réplica de leitura são

compatíveis. Para mais informações e um exemplo, consulte a discussão sobre o parâmetro `max_allowed_packet` posteriormente nesta seção.

O Amazon RDS monitora o status de replicação de suas réplicas de leitura e atualiza o campo `Replication State` da instância da réplica de leitura para `Error` caso a replicação seja interrompida por qualquer motivo. Um exemplo pode ser se as consultas DML forem executadas no seu conflito de réplica de leitura com as atualizações feitas na instância de banco de dados de origem.

Você pode analisar os detalhes do erro associado gerado pelo mecanismo do MySQL visualizando o campo `Replication Error`. Os eventos que indicam o status da réplica de leitura também são gerados, incluindo RDS-EVENT-0045, RDS-EVENT-0046 e RDS-EVENT-0047. Para mais informações sobre eventos e como se inscrever neles, consulte Trabalhar com a notificação de eventos do Amazon RDS. Se for retornada uma mensagem de erro do MySQL, analise o número de erro na documentação de mensagens de erro do MySQL.

Um problema comum que pode causar erros de replicação é quando o valor do parâmetro `max_allowed_packet` para uma réplica de leitura é menor que o do parâmetro `max_allowed_packet` para a instância de banco de dados de origem. O parâmetro `max_allowed_packet` é um parâmetro personalizado que você pode definir em um grupo de parâmetros de banco de dados. Use o `max_allowed_packet` para especificar o tamanho máximo do código DML que pode ser executado no banco de dados. Em alguns casos, o valor do `max_allowed_packet` no grupo de parâmetros de banco de dados associado a uma réplica de leitura é menor do que o valor de `max_allowed_packet` no grupo de parâmetros de banco de dados associado à instância de banco de dados de origem. Nesses casos, o processo de replicação pode lançar o erro `Packet bigger than 'max_allowed_packet' bytes` e interromper a replicação. Para corrigir o erro, faça com que a instância de banco de dados de origem e a réplica de leitura usem grupos de parâmetros de banco de dados com os mesmos valores de parâmetro `max_allowed_packet`

Outras situações comuns que podem causar erros de replicação incluem o seguinte:

- A gravação em tabelas em uma réplica de leitura. Em alguns casos, é possível criar índices em uma réplica de leitura diferentes dos índices na instância de banco de dados da origem. Se você fizer isso, defina o parâmetro `read_only` como `0` para criar índices. Se você gravar em tabelas na réplica de leitura, isso poderá interromper a replicação se a réplica de leitura tornar-se incompatível com a instância de banco de dados de origem. Depois de realizar tarefas de

manutenção na réplica de leitura, recomendamos que você defina o parâmetro `read_only` de volta para 1.

- Uso de um mecanismo de armazenamento não transacional, como o MyISAM. As réplicas de leitura exigem um mecanismo de armazenamento transacional. A replicação só é compatível com o mecanismo de armazenamento InnoDB no MySQL.
- Usando consultas não deterministas inseguras, como `SYSDATE()`. Para obter mais informações, consulte Determinação de instruções seguras e inseguras no registro de logs binários.

Se você acreditar que pode ignorar um erro com segurança, siga as etapas descritas na seção Ignorar o erro de replicação atual para o RDS para MySQL. Caso contrário, você pode excluir a réplica de leitura primeiro. Crie uma instância usando o mesmo identificador de instância de banco de dados para que o endpoint permaneça o mesmo que o da sua antiga réplica de leitura. Se um erro de replicação for corrigido, o `Replication State` mudará para replicating.

Usar a replicação baseada em GTID

O conteúdo a seguir explica como usar identificadores de transações globais (GTIDs) com a replicação de logs binários (binlog) entre instâncias de banco de dados do Amazon RDS para MySQL.

Se você usa a replicação de logs binários e não conhece a replicação baseada em GTID com o MySQL, consulte Replication with global transaction identifiers na documentação do MySQL.

A replicação baseada em GTID é compatível com as seguintes versões:

- Todas as versões do RDS para MySQL 8.4
- Todas as versões do RDS para MySQL 8.0
- Todas as versões do RDS para MySQL 5.7

Todas as instâncias de banco de dados do MySQL em uma configuração de replicação devem atender a esse requisito de versão.

Tópicos

- Visão geral dos identificadores de transações globais (GTIDs)
- Parâmetros para replicação baseada em GTID
- Ativar a replicação baseada em GTID para novas réplicas de leitura para RDS para MySQL

- Ativar a replicação baseada em GTID para réplicas de leitura existentes para RDS para MySQL
- Desabilitação da replicação baseada em GTID para uma instância de banco de dados do RDS for MySQL com réplicas de leitura

Visão geral dos identificadores de transações globais (GTIDs)

GTIDs são identificadores exclusivos gerados para transações MySQL confirmadas. Você pode usar GTIDs para tornar a replicação de log binário mais simples e fácil de solucionar.

O MySQL usa dois tipos diferentes de transações para a replicação de log binário:

- Transações GTID – Transações identificadas por um GTID.
- Transações anônimas – transações que não têm um GTID atribuído.

Em uma configuração de replicação, GTIDs são exclusivos entre todas as instâncias de banco de dados. Os GTIDs simplificam a configuração da replicação porque ao usá-los você não precisa consultar posições de arquivo de log. Os GTIDs também facilitam o rastreamento de transações replicadas e a determinação da consistência da instância de origem e de réplicas.

É possível usar replicação baseada em GTID para replicar dados com réplicas de leitura do RDS for MySQL. É possível configurar a replicação baseada em GTID ao criar réplicas de leitura ou você pode converter réplicas de leitura existentes para usar a replicação baseada em GTID.

Você também pode utilizar a replicação baseada em GTID em uma configuração de replicação atrasada com o RDS for MySQL. Para ter mais informações, consulte Configurar replicação atrasada com o MySQL.

Parâmetros para replicação baseada em GTID

Use os parâmetros a seguir para configurar a replicação baseada em GTID.

Parâmetro	Valores válidos	Descrição
`gtid_mode`	`OFF`, `OFF_PERMISSIVE` , `ON_PERMISSIVE` , `ON`	`OFF` especifica que novas transações são anônimas (ou seja, não têm GTIDs) e que uma transação deve ser anônima para ser replicada.

Parâmetro	Valores válidos	Descrição
		`OFF_PERMISSIVE` especifica que novas transações são anônimas, mas todas podem ser replicadas. `ON_PERMISSIVE` especifica que novas transações são GTID, mas todas podem ser replicadas. `ON` especifica que novas transações são GTID e que uma transação deve ser GTID para ser replicada.
`enforce_gtid_consistency`	`OFF`, `ON`, `WARN`	`OFF` permite que as transações violem a consistência GTID. `ON` evita que as transações violem a consistência GTID. `WARN` permite que as transações violem a consistência GTID, mas gera um aviso quando ocorre uma violação.

> **Note**
>
> No AWS Management Console, o parâmetro `gtid_mode` aparece como `gtid-mode`.

Para a replicação baseada em GTID, use estas configurações para o grupo de parâmetros da instância de banco de dados ou réplica de leitura:

- `ON` e `ON_PERMISSIVE` somente são aplicáveis à replicação de saída de uma instância de banco de dados do RDS. Esses dois valores fazem com que a sua instância de banco de dados do RDS usem GTIDs para transações que são replicadas. O `ON` requer que o banco de dados de destino também use a replicação baseada em GTID. O `ON_PERMISSIVE` torna a replicação baseada em GTID opcional no banco de dados de destino.

- `OFF_PERMISSIVE`, se definido, significa que as instâncias de banco de dados do RDS podem aceitar a replicação de entrada de um banco de dados de origem. Eles podem fazer isso independentemente de o banco de dados de origem usar a replicação baseada em GTID.
- `OFF`, se definido, significa que sua instância de banco de dados do RDS apenas aceita a replicação de entrada de bancos de dados de origem que não usam a replicação baseada em GTID.

Para obter mais informações sobre parameter groups, consulte Grupos de parâmetros para Amazon RDS.

Ativar a replicação baseada em GTID para novas réplicas de leitura para RDS para MySQL

Quando a replicação baseada em GTID é habilitada para uma instância de banco de dados do RDS for MySQL, a replicação é configurada automaticamente para réplicas de leitura da instância de banco de dados.

Como habilitar a replicação baseada em GTID para novas réplicas de leitura

1. Verifique se o grupo de parâmetros associado a uma instância de banco de dados tem as seguintes configurações de parâmetro:

   - `gtid_mode` – `ON` ou `ON_PERMISSIVE`
   - `enforce_gtid_consistency` – `ON`

   Para obter mais informações sobre a definição de parâmetros de configuração usando grupos de parâmetros, consulte Grupos de parâmetros para Amazon RDS.
2. Se você tiver alterado o grupo de parâmetros da instância de banco de dados, reinicie a instância de banco de dados. Para obter mais informações sobre como fazer isso, consulte Reinicializar uma instância de banco de dados .
3. Crie uma ou mais réplicas de leitura da instância de banco de dados. Para obter mais informações sobre como fazer isso, consulte Como criar uma réplica de leitura.

O Amazon RDS tenta estabelecer a replicação baseada em GTID entre a instância de banco de dados do MySQL e as réplicas de leitura usando o `MASTER_AUTO_POSITION`. Caso a tentativa falhe, o Amazon RDS usa posições de arquivo de log na replicação com as réplicas de leitura. Para

obter mais informações sobre o `MASTER_AUTO_POSITION`, consulte GTID Auto-Positioning na documentação do MySQL.

Ativar a replicação baseada em GTID para réplicas de leitura existentes para RDS para MySQL

Para uma instância de banco de dados do RDS for MySQL existente com réplicas de leitura que não usam a replicação baseada em GTID, é possível configurar a replicação baseada em GTID entre a instância de banco de dados e as réplicas de leitura.

Como habilitar a replicação baseada em GTID para réplicas de leitura existentes

1. Se a instância de banco de dados ou qualquer réplica de leitura estiver utilizando uma versão 8.0 do RDS for MySQL inferior a 8.0.26, faça upgrade da instância de banco de dados ou da réplica de leitura para 8.0.26 ou uma versão posterior do MySQL 8.0. Todas as versões 5.7 do RDS para MySQL versões 8.4 e 5.7 são compatíveis com replicação baseada em GTID.

   Para ter mais informações, consulte Atualizações do mecanismo de banco de dados do RDS para MySQL.
2. (Opcional) Redefina os parâmetros GTID e teste o comportamento da instância de banco de dados e as réplicas de leitura:

   a. Verifique se o grupo de parâmetros associado a uma instância de banco de dados e cada réplica de leitura tem o parâmetro `enforce_gtid_consistency` definido como `WARN`.

      Para obter mais informações sobre a definição de parâmetros de configuração usando grupos de parâmetros, consulte Grupos de parâmetros para Amazon RDS.

   b. Se você tiver alterado o grupo de parâmetros da instância de banco de dados, reinicie a instância de banco de dados. Se você tiver alterado o grupo de parâmetros para uma réplica de leitura, reinicie a réplica de leitura.

      Para obter mais informações, consulte Reinicializar uma instância de banco de dados .

   c. Execute a instância de banco de dados e as réplicas de leitura com a workload normal e monitore os arquivos de log.

      Caso você veja avisos sobre transações incompatíveis com GTID, ajuste o aplicativo de maneira que ele só use recursos compatíveis com GTID. Verifique se a instância de banco de dados não estiver gerando avisos sobre transações incompatíveis com GTID antes de avançar à próxima etapa.

3. Redefina os parâmetros GTID para replicação baseada em GTID que permite transações anônimas até que as réplicas de leitura tenham processado todas elas.

   a. Verifique se o grupo de parâmetros associado a uma instância de banco de dados e cada réplica de leitura tem as seguintes configurações de parâmetro:

      - `gtid_mode` – `ON_PERMISSIVE`
      - `enforce_gtid_consistency` – `ON`

   b. Se você tiver alterado o grupo de parâmetros da instância de banco de dados, reinicie a instância de banco de dados. Se você tiver alterado o grupo de parâmetros para uma réplica de leitura, reinicie a réplica de leitura.

4. Aguarde a replicação de todas as transações anônimas. Para verificar se elas estão replicadas, faça o seguinte:

   a. Execute a instrução a seguir na instância de banco de dados de origem.

      MySQL 8.4

      ```
      SHOW BINARY LOG STATUS;
      ```

      MySQL 5.7 e 8.0

      ```
      SHOW MASTER STATUS;
      ```

      Observe os valores nas colunas `File` e `Position`.

   b. Em cada réplica de leitura, use as informações de arquivo e posição de sua instância de origem na etapa anterior para executar a consulta a seguir.

      ```
      SELECT MASTER_POS_WAIT('file', position);
      ```

      Por exemplo, caso o nome do arquivo seja `mysql-bin-changelog.000031` e a posição seja `107`, execute a declaração a seguir.

      ```
      SELECT MASTER_POS_WAIT('mysql-bin-changelog.000031', 107);
      ```

Se a réplica de leitura ultrapassar a posição especificada, a consulta retornará imediatamente. Do contrário, a função aguarda. Prossiga para a próxima etapa quando a consulta retornar para todas as réplicas de leitura.

5. Redefina os parâmetros GTID apenas para replicação baseada em GTID.

   a. Verifique se o grupo de parâmetros associado a uma instância de banco de dados e cada réplica de leitura tem as seguintes configurações de parâmetro:

      - `gtid_mode` – `ON`
      - `enforce_gtid_consistency` – `ON`

   b. Reinicialize a instância de banco de dados e cada réplica de leitura.

6. Em cada réplica de leitura, execute o procedimento a seguir.

   MySQL 8.4 e versões principais posteriores

   ```
   CALL mysql.rds_set_source_auto_position(1);
   ```

   MySQL 8.0 e versões principais anteriores

   ```
   CALL mysql.rds_set_master_auto_position(1);
   ```

Desabilitação da replicação baseada em GTID para uma instância de banco de dados do RDS for MySQL com réplicas de leitura

Você pode desabilitar a replicação baseada em GTID para uma instância de banco de dados MySQL com réplicas de leitura.

Para desabilitar a replicação baseada em GTID para uma instância de banco de dados do RDS for MySQL com réplicas de leitura

1. Em cada réplica de leitura, execute o seguinte procedimento:

   MySQL 8.4 e versões principais posteriores

   ```
   CALL mysql.rds_set_source_auto_position(0);
   ```

   MySQL 8.0 e versões principais anteriores

```
CALL mysql.rds_set_master_auto_position(0);
```

2. Redefina o `gtid_mode` como `ON_PERMISSIVE`.

   a. Verifique se o grupo de parâmetros associado a uma instância de banco de dados do MySQL e cada réplica de leitura tem possuem `gtid_mode` definido como `ON_PERMISSIVE`.

      Para obter mais informações sobre a definição de parâmetros de configuração usando grupos de parâmetros, consulte Grupos de parâmetros para Amazon RDS.

   b. Reinicialize a instância de banco de dados do MySQL e cada réplica de leitura. Para obter mais informações sobre como reiniciar, consulte Reinicializar uma instância de banco de dados .

3. Redefina o `gtid_mode` como `OFF_PERMISSIVE`.

   a. Verifique se o grupo de parâmetros associado a uma instância de banco de dados do MySQL e cada réplica de leitura tem possuem `gtid_mode` definido como `OFF_PERMISSIVE`.

   b. Reinicialize a instância de banco de dados do MySQL e cada réplica de leitura.

4. Aguarde todas as transações GTID serem aplicadas a todas as réplicas de leitura. Para conferir se elas foram aplicadas, siga estas etapas:

   a. Na instância de banco de dados do MySQL, execute o comando a seguir:

      MySQL 8.4

      ```
      SHOW BINARY LOG STATUS
      ```

      MySQL 5.7 e 8.0

      ```
      SHOW MASTER STATUS
      ```

      Sua saída deve ser semelhante ao resultado a seguir.

      ```
      File                        Position
      ------------------------------------
      ```

```
mysql-bin-changelog.000031      107
-----------------------------------
```

Observe o arquivo e posicione na saída.

b. Em cada réplica de leitura, use as informações de arquivo e posição de sua instância de origem na etapa anterior para executar a seguinte consulta:

MySQL 8.4 e MySQL 8.0.26 e versões posteriores do MySQL 8.0

```
SELECT SOURCE_POS_WAIT('file', position);
```

MySQL 5.7

```
SELECT MASTER_POS_WAIT('file', position);
```

Por exemplo, caso o nome do arquivo seja `mysql-bin-changelog.000031` e a posição seja `107`, execute a seguinte declaração:

MySQL 8.4 e MySQL 8.0.26 e versões posteriores do MySQL 8.0

```
SELECT SOURCE_POS_WAIT('mysql-bin-changelog.000031', 107);
```

MySQL 5.7

```
SELECT MASTER_POS_WAIT('mysql-bin-changelog.000031', 107);
```

5. Redefina os parâmetros de GTID para desabilitar a replicação baseada em GTID.

a. Verifique se o grupo de parâmetros associado à instância de banco de dados MySQL e cada réplica de leitura possuem as seguintes configurações de parâmetro:

- `gtid_mode` – `OFF`
- `enforce_gtid_consistency` – `OFF`

b. Reinicialize a instância de banco de dados do MySQL e cada réplica de leitura.

Configurar a replicação da posição do arquivo de log binário com uma instância de origem externa

É possível configurar a replicação entre uma instância de banco de dados do RDS para MySQL ou do MariaDB e uma instância do MySQL ou do MariaDB externa ao Amazon RDS usando a replicação de arquivos de log binários.

Tópicos

- Antes de começar
- Configurar a replicação da posição do arquivo de log binário com uma instância de origem externa

Antes de começar

Você pode configurar a replicação usando a posição do arquivo de log binário de transações replicadas.

As permissões necessárias para iniciar a replicação em uma instância de banco de dados do Amazon RDS são restritas e não estão disponíveis ao seu usuário mestre do Amazon RDS. Por isso, use os comandos mysql.rds_set_external_master (RDS para MariaDB e RDS para MySQL versões principais 8.0 e anteriores) ou mysql.rds_set_external_source (RDS para MySQL versões principais 8.4 e posteriores) e mysql.rds_start_replication do Amazon RDS para configurar a replicação entre seu banco de dados on-line e seu banco de dados do Amazon RDS.

Para definir o formato de registro em log binário para um banco de dados MySQL e MariaDB, atualize o parâmetro `binlog_format`. Se a instância de banco de dados usar o grupo de parâmetros da instância de banco de dados padrão, crie um novo grupo de parâmetros de banco de dados para modificar o parâmetro `binlog_format`. No MariaDB e no MySQL 8.0 e versões anteriores, `binlog_format` tem como padrão `MIXED`. No entanto, você também pode definir o `binlog_format` como `ROW` ou `STATEMENT` se precisar de um formato específico de log binário (binlog). Reinicie a instância de banco de dados para que a alteração entre em vigor. No MySQL 8.4 e versões superiores, `binlog_format` tem como padrão `ROW`.

Para obter informações sobre como definir o parâmetro `binlog_format`, consulte Configurar o registro em log binário do RDS para MySQL para bancos de dados single-AZ. Para obter informações sobre as implicações de tipos de replicação do MySQL diferentes, consulte Vantagens e desvantagens da replicação baseada em instrução e baseada em linha na documentação do MySQL.

Configurar a replicação da posição do arquivo de log binário com uma instância de origem externa

Siga estas diretrizes ao configurar uma instância de origem externa e uma réplica no Amazon RDS:

- Monitore eventos de failover para a instância do banco de dados do Amazon RDS, que é sua réplica. Se ocorrer um failover, a instância de banco de dados que for a sua réplica poderá ser recriada em um novo host com um endereço de rede diferente. Para obter informações sobre como monitorar eventos de failover, consulte Trabalhar com a notificação de eventos do Amazon RDS.
- Retenha os logs binários na instância de origem até confirmar que eles foram aplicados à réplica. Esta manutenção garante que seja possível restaurar sua instância de origem em caso de falha.
- Ative backups automatizados na sua instância de banco de dados do Amazon RDS. Ativar backups automáticos garante que seja possível restaurar sua réplica para um determinado ponto no tempo, caso seja necessário ressincronizar sua instância de origem e sua réplica. Para obter informações sobre backups e restauração Point-In-Time, consulte Backup, restauração e exportação de dados.

Como configurar a replicação do arquivo de log binário com uma instância de origem externa

1. Torne a instância de origem MySQL ou MariaDB somente leitura.

   ```
   mysql> FLUSH TABLES WITH READ LOCK;
   mysql> SET GLOBAL read_only = ON;
   ```

2. Execute o comando `SHOW MASTER STATUS` na instância do MySQL ou MariaDB de origem para determinar a localização do log binário.

   Você recebe um resultado semelhante ao seguinte exemplo.

   ```
   File                          Position
   ------------------------------------
    mysql-bin-changelog.000031      107
   ------------------------------------
   ```

3. Copie o banco de dados da instância externa para a instância de banco de dados do Amazon RDS usando `mysqldump`. Para bancos de dados muito grandes, convém usar o procedimento em Importar dados para um banco de dados do Amazon RDS para MySQL com tempo de inatividade reduzido.

Para Linux, macOS ou Unix:

```
mysqldump --databases database_name \
    --single-transaction \
    --compress \
    --order-by-primary \
    -u local_user \
    -plocal_password | mysql \
        --host=hostname \
        --port=3306 \
        -u RDS_user_name \
        -pRDS_password
```

Para Windows:

```
mysqldump --databases database_name ^
    --single-transaction ^
    --compress ^
    --order-by-primary ^
    -u local_user ^
    -plocal_password | mysql ^
        --host=hostname ^
        --port=3306 ^
        -u RDS_user_name ^
        -pRDS_password
```

> **Note**
>
> Confirme que não há um espaço entre a opção `-p` e a senha inserida.

Para especificar o nome do host, o nome do usuário, a porta e a senha para se conectar à instância de banco de dados Amazon RDS, use as opções `--host`, `--user (-u)`, `--port` e `-p` no comando `mysql`. O nome do host é o nome do Serviço de Nome de Domínio (DNS) do endpoint da instância de banco de dados do Amazon RDS, por exemplo, `myinstance.123456789012.us-east-1.rds.amazonaws.com`. Você pode encontrar o valor do endpoint nos detalhes da instância no AWS Management Console.

4. Torne a instância MySQL ou MariaDB de origem novamente gravável.

```
mysql> SET GLOBAL read_only = OFF;
mysql> UNLOCK TABLES;
```

Para obter mais informações sobre como fazer backups para usar com a replicação, consulte a documentação do MySQL.

5. No AWS Management Console, adicione o endereço IP do servidor que hospeda o banco de dados externo ao grupo de segurança da nuvem privada virtual (VPC) para a instância de banco de dados Amazon RDS. Para obter mais informações sobre como modificar um grupo de segurança da VPC, consulte Grupos de segurança para a VPC no Guia do usuário da Amazon Virtual Private Cloud.

   O endereço IP poderá ser alterado se as seguintes condições forem atendidas:

   - Se você estiver usando um endereço IP público para a comunicação entre a instância de origem externa e a instância de banco de dados.
   - A instância de origem externa foi interrompida e reiniciada.

   Se essas condições forem atendidas, verifique o endereço IP antes de adicioná-lo.

   Você também pode precisar configurar a rede local para permitir conexões com o endereço IP da instância de banco de dados do Amazon RDS. Isso é realizado para que a rede local consiga se comunicar com a instância do MySQL ou do MariaDB externa. Para localizar o endereço IP da instância de banco de dados do Amazon RDS, use o comando `host`.

   ```
   host db_instance_endpoint
   ```

   O nome do host é o nome de DNS do endpoint da instância de banco de dados do Amazon RDS.

6. Usando o cliente de sua preferência, conecte-se à instância externa e crie um usuário a ser usado para a replicação. Use essa conta unicamente para replicação e restrinja-a ao seu domínio para melhorar a segurança. Veja um exemplo a seguir.

   ```
   CREATE USER 'repl_user'@'mydomain.com' IDENTIFIED BY 'password';
   ```

> **Note**
>
> Especifique uma senha diferente do prompt mostrado aqui como prática recomendada de segurança.

7. Para a instância externa, conceda os privilégios `REPLICATION CLIENT` e `REPLICATION SLAVE` para seu usuário de replicação. Por exemplo, para conceder os privilégios de `REPLICATION CLIENT` e `REPLICATION SLAVE` em todos os bancos de dados para o usuário '`repl_user`' de seu domínio, emita o seguinte comando.

```
GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'mydomain.com';
```

8. Transforma a instância de banco de dados do Amazon RDS na réplica. Para fazer isso, primeiro se conecte à instância de banco de dados do Amazon RDS como usuário mestre. Depois disso, identifique o banco de dados MySQL ou MariaDB externo como a instância de origem usando o comando mysql.rds_set_external_source (RDS para MySQL versões principais 8.4 e posteriores) ou mysql.rds_set_external_master (RDS para MariaDB e RDS para MySQL versões principais 8.0 e anteriores). Use o nome do arquivo de log mestre e a posição do log mestre que você determinou na etapa 2. Os comandos a seguir são exemplos.

   MySQL 8.4

```
CALL mysql.rds_set_external_source ('mysourceserver.mydomain.com', 3306,
 'repl_user', 'password', 'mysql-bin-changelog.000031', 107, 1);
```

   MariaDB e MySQL 8.0 e 5.7

```
CALL mysql.rds_set_external_master ('mymasterserver.mydomain.com', 3306,
 'repl_user', 'password', 'mysql-bin-changelog.000031', 107, 1);
```

> **Note**
>
> No RDS para MySQL, é possível optar por usar a replicação atrasada executando o procedimento armazenado mysql.rds_set_external_source_with_delay (RDS para MySQL versões principais 8.4 e superiores) ou mysql.rds_set_external_master_with_delay (RDS para MariaDB e RDS para MySQL versões principais 8.0 e anteriores). No RDS para MySQL, um motivo

para usar a replicação atrasada é ativar a recuperação de desastres com o procedimento armazenado mysql.rds_start_replication_until. Atualmente, o RDS para MariaDB é compatível com a replicação atrasada, mas não com o procedimento `mysql.rds_start_replication_until`.

9. Na instância de banco de dados do Amazon RDS, emita o comando mysql.rds_start_replication para iniciar a replicação.

```
CALL mysql.rds_start_replication;
```

Configurar a replicação de várias fontes do Amazon RDS para MySQL

Com a replicação de várias fontes, é possível configurar uma instância de banco de dados do Amazon RDS para MySQL como uma réplica que recebe eventos de logs binários de mais de uma instância de banco de dados de origem do RDS para MySQL. A replicação de várias fontes é compatível com instâncias de banco de dados do RDS para MySQL que executam as seguintes versões de mecanismo:

- Todas as versões do MySQL 8.4
- 8.0.35 e versões secundárias posteriores
- 5.7.44 e versões secundárias posteriores

Para ter informações sobre a replicação de várias fontes do MySQL, consulte MySQL Multi-Source Replication na documentação do MySQL. A documentação do MySQL contém informações detalhadas sobre esse recurso, enquanto este tópico descreve como configurar e gerenciar canais de replicação de várias fontes nas instâncias de banco de dados do RDS para MySQL.

Casos de uso da replicação de várias fontes

Os casos a seguir são bons candidatos para usar a replicação de várias fontes no RDS para MySQL:

- Aplicações que precisam mesclar ou combinar vários fragmentos em instâncias de banco de dados separadas em um único fragmento.
- Aplicações que precisam gerar relatórios a partir de dados consolidados de várias fontes.
- Requisitos para criar backups consolidados de longo prazo de dados distribuídos entre várias instâncias de banco de dados do RDS para MySQL.

Pré-requisitos para replicação de várias fontes

Antes de configurar a replicação de várias fontes, cumpra os pré-requisitos a seguir.

- Garanta que cada instância de banco de dados do RDS para MySQL de origem tenha backups automáticos habilitados. A ativação de backups automáticos habilita o registro em log binário. Para saber como habilitar os backups automáticos, consulte the section called “Ativar backups automáticos”.
- Para evitar erros de replicação, recomendamos bloquear as operações de gravação nas instâncias de banco de dados de origem. É possível fazer isso definindo o parâmetro `read-only` como `ON` em um grupo de parâmetros personalizado associado à instância de banco de dados de origem do RDS para MySQL. É possível usar o AWS Management Console ou a AWS CLI para criar um grupo de parâmetros personalizado ou modificar um existente. Para obter mais informações, consulte the section called “Criar um grupo de parâmetros de banco de dados” e the section called “Modificar parâmetros em um grupo de parâmetros de banco de dados”.
- Para cada instância de banco de dados de origem, adicione o endereço IP da instância ao grupo de segurança da Amazon Virtual Private Cloud (VPC) para a instância de banco de dados de várias fontes de dados. Para identificar o endereço IP de uma instância de banco de dados de origem, é possível executar o comando `dig` *`RDS Endpoint`*. Execute o comando a partir de uma instância do Amazon EC2 na mesma VPC da instância de banco de dados de várias fontes de destino.
- Para cada instância de banco de dados de origem, use um cliente para se conectar à instância de banco de dados e criar um usuário de banco de dados com os privilégios necessários para replicação, como no exemplo a seguir.

```
CREATE USER 'repl_user' IDENTIFIED BY 'password';
GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user';
```

> **Note**
>
> A partir do MySQL 8.4, o privilégio `REPLICATION SLAVE` foi descontinuado e substituído por `REPLICATION REPLICA`. Para MySQL 8.4 e versões posteriores, use a seguinte sintaxe:
>
> ```
> CREATE USER 'repl_user' IDENTIFIED BY 'password';
> GRANT REPLICATION CLIENT, REPLICATION REPLICA ON *.* TO 'repl_user';
> ```

Configurar canais de replicação de várias fontes nas instâncias de banco de dados do RDS para MySQL

A configuração de canais de replicação de várias fontes é semelhante à configuração da replicação de uma única fonte. Para replicação de várias fontes, primeiro habilite o registro em log binário na instância de origem. Depois, importe os dados das fontes para a réplica de várias fontes. Então, inicie a replicação de cada fonte usando as coordenadas binárias do log binário ou usando o posicionamento automático do GTID.

Para configurar uma instância de banco de dados do RDS para MySQL como uma réplica de várias fontes de duas ou mais instâncias de banco de dados do RDS para MySQL, execute as etapas a seguir.

Tópicos

- Etapa 1: Importar dados das instâncias de banco de dados de origem para a réplica de várias fontes
- Etapa 2: Iniciar a replicação das instâncias de banco de dados de origem para a réplica de várias fontes

Etapa 1: Importar dados das instâncias de banco de dados de origem para a réplica de várias fontes

Execute as etapas a seguir em cada instância de banco de dados de origem.

Antes de importar os dados de uma fonte para a réplica de várias fontes, determine o arquivo de log binário atual e a posição executando o comando `SHOW MASTER STATUS`. Anote esses detalhes para uso na próxima etapa. Neste exemplo de saída, o arquivo é `mysql-bin-changelog.000031` e a posição é `107`.

> **Note**
>
> A partir do MySQL 8.4, o comando `SHOW MASTER STATUS` foi descontinuado e substituído por `SHOW BINARY LOG STATUS`. Para MySQL 8.4 e versões posteriores, use `SHOW BINARY LOG STATUS`.

```
File                        Position
-----------------------------------
mysql-bin-changelog.000031      107
```

```
----------------------------------
```

Agora, copie o banco de dados da instância de banco de dados de origem para a réplica de várias fontes usando `mysqldump`, como no exemplo a seguir.

```
mysqldump --databases database_name \
 --single-transaction \
 --compress \
 --order-by-primary \
 -u RDS_user_name \
 -p RDS_password \
 --host=RDS Endpoint | mysql \
 --host=RDS Endpoint \
 --port=3306 \
 -u RDS_user_name \
-p RDS_password
```

Depois de copiar o banco de dados, é possível definir o parâmetro somente leitura como `OFF` na instância de banco de dados de origem.

Etapa 2: Iniciar a replicação das instâncias de banco de dados de origem para a réplica de várias fontes

Para cada instância de banco de dados de origem, use as credenciais de usuário administrativas para se conectar à instância e executar os dois procedimentos armazenados a seguir. Esses procedimentos armazenados configuram a replicação em um canal e iniciam a replicação. Este exemplo usa o nome do arquivo de log binário e a posição da saída do exemplo na etapa anterior.

```
CALL mysql.rds_set_external_source_for_channel('mysourcehost.example.com', 3306,
 'repl_user', 'password', 'mysql-bin-changelog.000031', 107, 1, 'channel_1');
CALL mysql.rds_start_replication_for_channel('channel_1');
```

Para ter mais informações sobre como usar esses procedimentos armazenados e outros para configurar e gerenciar os canais de replicação, consulte the section called “Gerenciar a replicação de várias fontes”.

Usar filtros com replicação de várias fontes

É possível usar filtros de replicação para especificar quais bancos de dados e tabelas são replicados em uma réplica de várias fontes. Os filtros de replicação podem incluir bancos de dados e tabelas

na replicação ou excluí-los da replicação. Para ter mais informações sobre filtros de replicação , consulte the section called “Configurar filtros de replicação”.

Com a replicação de várias fontes, é possível configurar filtros de replicação globalmente ou em nível de canal. A filtragem em nível de canal está disponível somente com instâncias de banco de dados compatíveis que executam a versão 8.0 ou 8.4. Os exemplos a seguir mostram como gerenciar filtros globalmente ou em nível de canal.

Observe os seguintes requisitos e comportamento com a filtragem na replicação de várias fontes:

- Aspas invertidas (``) ao redor dos nomes dos canais são obrigatórias.
- Se você alterar os filtros de replicação no grupo de parâmetros, `sql_thread` das réplicas de várias fontes de todos os canais com atualizações será reiniciado para aplicar as alterações dinamicamente. Se uma atualização envolver um filtro global, todos os canais de replicação no estado de execução serão reiniciados.
- Todos os filtros globais são aplicados antes de qualquer filtro específico do canal.
- Se um filtro for aplicado globalmente e em nível de canal, somente o filtro em nível de canal será aplicado. Por exemplo, se os filtros forem `replicate_ignore_db="db1,`channel_22`:db2"`, `replicate_ignore_db` definido como `db1` será aplicado a todos os canais, exceto a `channel_22` e somente `channel_22` ignorará as alterações de `db2`.

Exemplo 1: Definir um filtro global

No exemplo a seguir, o banco de dados `temp_data` é excluído da replicação em todos os canais.

Para Linux, macOS ou Unix:

```
aws rds modify-db-parameter-group \
--db-parameter-group-name myparametergroup \
--parameters "ParameterName=replicate-ignore-
db,ParameterValue='temp_data',ApplyMethod=immediate"
```

Exemplo 2: Definir um filtro em nível de canal

No exemplo a seguir, as alterações do banco de dados `sample22` são incluídas somente no canal `channel_22`. Da mesma forma, as alterações do banco de dados `sample99` são incluídas apenas no canal `channel_99`.

Para Linux, macOS ou Unix:

```
aws rds modify-db-parameter-group \
--db-parameter-group-name myparametergroup \
--parameters "ParameterName=replicate-do-db,ParameterValue='\`channel_22\`:sample22,
\`channel_99\`:sample99',ApplyMethod=immediate"
```

Monitorar canais de replicação de várias fontes

É possível monitorar canais individuais em uma réplica de várias fontes usando os seguintes métodos:

- Para monitorar o status de todos os canais ou de um canal específico, conecte-se à réplica de várias fontes e execute o comando `SHOW REPLICA STATUS` ou `SHOW REPLICA STATUS FOR CHANNEL 'channel_name'`. Para ter mais informações, consulte Checking Replication Status na documentação do MySQL.
- Para receber uma notificação quando um canal de replicação for iniciado, interrompido ou removido, use a notificação de eventos do RDS. Para obter mais informações, consulte the section called “Trabalhar com a notificação de eventos do Amazon RDS”.
- Para monitorar o atraso de um canal específico, confira a métrica `ReplicationChannelLag`. Os pontos de dados dessa métrica têm um período de 60 segundos (1 minuto) e ficam disponíveis por 15 dias. Para localizar o atraso de replicação de um canal, use o identificador da instância e o nome do canal de replicação. Para receber uma notificação quando esse atraso exceder um determinado limite, configure um alarme do CloudWatch. Para obter mais informações, consulte the section called “Monitorando RDS com CloudWatch”.

Considerações e práticas recomendadas para replicação de várias fontes

Antes de usar a replicação de várias fontes no RDS para MySQL, analise as seguintes considerações e práticas recomendadas:

- Garanta que uma instância de banco de dados configurada como uma réplica de várias fontes tenha recursos suficientes, como throughput, memória, CPU e IOPS, para lidar com a workload de instâncias de várias fontes.
- Monitore regularmente a utilização de recursos na réplica de várias fontes e ajuste a configuração do armazenamento ou da instância para lidar com a workload sem sobrecarregar os recursos.

- É possível configurar a replicação multithread em uma réplica de várias fontes definindo a variável do sistema `replica_parallel_workers` com um valor maior que `0`. Nesse caso, o número de threads alocados para cada canal é o valor dessa variável, mais um thread coordenador para gerenciar os threads aplicadores.
- Configure os filtros de replicação corretamente para evitar conflitos. Para replicar um banco de dados inteiro em outro banco de dados em uma réplica, é possível usar a opção `--replicate-rewrite-db`. Por exemplo, é possível replicar todas as tabelas no banco de dados A no banco de dados B em uma instância de réplica. Essa abordagem pode ser útil quando todas as instâncias de origem estão usando a mesma convenção de nomenclatura de esquema. Para ter informações sobre a opção `--replicate-rewrite-db`, consulte Replica Server Options and Variables na documentação do MySQL.
- Para evitar erros de replicação, evite gravar na réplica. Recomendamos habilitar o parâmetro `read_only` em réplicas de várias fontes para bloquear as operações de gravação. Isso ajuda a eliminar problemas de replicação causados por operações de gravação conflitantes.
- Para melhor a performance das operações de leitura, como classificações e junções de alta carga que são executadas na réplica de várias fontes, pense em usar leituras otimizadas pelo RDS. Esse recurso pode ajudar com consultas que dependem de grandes tabelas temporárias ou arquivos de classificação. Para obter mais informações, consulte the section called “Melhorar a performance das consultas com leituras otimizadas do RDS”.
- Para minimizar o atraso na replicação e melhorar a performance de uma réplica de várias fontes, pense em habilitar gravações otimizadas. Para obter mais informações, consulte the section called “Melhorar a performance das gravações com gravações otimizadas pelo RDS para MySQL”.
- Execute operações de gerenciamento (como alterar a configuração) em um canal por vez e evite realizar alterações em vários canais a partir de várias conexões. Essas práticas podem causar conflitos nas operações de replicação. Por exemplo, a execução simultânea de procedimentos `rds_skip_repl_error_for_channel` e `rds_start_replication_for_channel` de várias conexões pode causar a omissão de eventos em um canal diferente do pretendido.
- É possível habilitar backups em uma instância de replicação de várias fontes e exportar dados dessa instância para um bucket do Amazon S3 para armazená-los para fins de longo prazo. No entanto, é importante também configurar backups com retenção adequada nas instâncias de origem individuais. Para ter informações sobre como exportar dados de snapshot para o Amazon S3, consulte the section called “Exportação de dados de snapshot de banco de dados para o Amazon S3”.
- Para distribuir a workload de leitura em uma réplica de várias fontes, é possível criar réplicas de leitura de uma réplica de várias fontes. É possível localizar essas réplicas de leitura em Regiões da

AWS diferentes com base nos requisitos da aplicação. Para ter mais informações sobre réplicas de leitura, consulte the section called “Réplicas de leitura do MySQL”.

Limitações da replicação de várias fontes do RDS para MySQL

As seguintes limitações se aplicam à replicação de várias fontes no RDS para MySQL:

- No momento, o RDS para MySQL comporta a configuração de no máximo 15 canais para uma réplica de várias fontes.
- Uma instância de réplica de leitura não pode ser configurada como uma réplica de várias fontes.
- Para configurar a replicação de várias fontes no RDS para MySQL executando a versão 5.7 do mecanismo, o Performance Schema deve estar habilitado na instância de réplica. A ativação do Performance Schema é opcional no RDS para MySQL executando a versão 8.0 ou 8.4 do mecanismo.
- Para o RDS para MySQL que executa a versão 5.7 do mecanismo, os filtros de replicação se aplicam a todos os canais de replicação. Para o RDS para MySQL que executa a versão 8.0 ou 8.4 do mecanismo, é possível configurar filtros que se aplicam a todos os canais de replicação ou a canais individuais.
- Restaurar um snapshot do RDS ou realizar uma recuperação para um ponto no tempo (PITR) não restaura configurações de canais de réplica de várias fontes.
- Ao criar uma réplica de leitura por meio de uma réplica de várias fontes, ela replica apenas os dados da instância de várias fontes. Ela não restaura a configuração de nenhum canal.
- O MySQL não comporta a configuração de um número diferente de operadores paralelos para cada canal. Cada canal recebe o mesmo número de operadores paralelos com base no valor `replica_parallel_workers`.

As seguintes limitações adicionais se aplicarão se o destino de replicação de várias fontes for um cluster de banco de dados multi-AZ:

- É necessário configurar um canal para uma instância de origem do RDS para MySQL antes de qualquer gravação nessa instância.
- Cada instância do RDS para MySQL de origem deve ter a replicação baseada em GTID habilitada.
- Um evento de failover no cluster de banco de dados remove a configuração de replicação de várias fontes. A restauração dessa configuração requer a repetição das etapas de configuração.

Configurar clusters ativos-ativos para RDS para MySQL

Um cluster ativo-ativo no Amazon RDS é uma configuração de banco de dados em que vários nós lidam ativamente com operações de leitura e gravação, distribuindo a workload entre instâncias para melhorar a disponibilidade e a escalabilidade. Cada nó no cluster é sincronizado para manter a consistência de dados, permitindo alta disponibilidade e um failover mais rápido em caso de falha do nó.

É possível configurar um cluster ativo-ativo para o RDS para MySQL usando o plug-in MySQL Group Replication. O plug-in de replicação de grupo é compatível com instâncias de banco de dados do RDS para MySQL que executam as seguintes versões:

- Todas as versões do MySQL 8.4
- MySQL 8.0.35 e versões secundárias posteriores

Para ter informações sobre a MySQL Group Replication, consulte Group Replication na documentação do MySQL. A documentação do MySQL contém informações detalhadas sobre esse recurso, enquanto este tópico descreve como configurar e gerenciar o plug-in nas instâncias de banco de dados do RDS para MySQL.

> **Note**
>
> Para resumir, todas as menções de cluster “ativo-ativo” neste tópico se referem a clusters ativos-ativos usando o plug-in MySQL Group Replication.

Casos de uso de clusters ativos-ativos

Os casos a seguir são bons candidatos ao uso de clusters ativos-ativos:

- Aplicações que precisam de todas as instâncias de banco de dados do cluster para comportar operações de gravação. O plug-in Group Replication mantém os dados consistentes em cada instância de banco de dados no cluster ativo-ativo. Para ter informações sobre como isso funciona, consulte Group Replication na documentação do MySQL.
- Aplicações que exigem disponibilidade contínua do banco de dados. Com um cluster ativo-ativo, os dados são retidos em todas as instâncias de banco de dados contidas no cluster. Se uma instância de banco de dados falhar, a aplicação poderá redirecionar o tráfego para outra instância de banco de dados no cluster.

- Aplicações que talvez precisem dividir as operações de leitura e gravação entre diferentes instâncias de banco de dados no cluster para fins de balanceamento de carga. Com um cluster ativo-ativo, as aplicações podem enviar tráfego de leitura a instâncias de banco de dados específicas e tráfego de gravação a outras. Também é possível alternar para quais instâncias de banco de dados enviar leituras ou gravações a qualquer momento.

Tópicos

- Limitações e considerações sobre cluster ativo-ativo
- Preparar um cluster ativo-ativo entre VPCs
- Configurações de parâmetros obrigatórias para clusters ativos-ativos
- Converter uma instância de banco de dados existente em um cluster ativo-ativo
- Configurar um cluster ativo-ativo com novas instâncias de banco de dados
- Adicionar uma instância de banco de dados em um cluster ativo-ativo
- Monitorar clusters ativos-ativos
- Interromper a Group Replication em uma instância de banco de dados em um cluster ativo-ativo
- Renomear uma instância de banco de dados em um cluster ativo-ativo
- Remover uma instância de banco de dados de um cluster ativo-ativo

Limitações e considerações sobre cluster ativo-ativo

O clusters ativo-ativo no Amazon RDS oferece maior disponibilidade e escalabilidade ao distribuir workloads em várias instâncias. No entanto, há limitações e considerações importantes a saber ao usar essa arquitetura.

As seções a seguir descrevem os principais fatores, como atrasos na replicação, resolução de conflitos, alocação de recursos e comportamento de failover. Compreender essas considerações pode ajudar a garantir o desempenho e a confiabilidade ideais em implantações de cluster ativo-ativo.

Tópicos

- Limitações dos clusters ativos-ativos do RDS para MySQL
- Considerações e práticas recomendadas para cluster ativo-ativo do RDS para MySQL

Limitações dos clusters ativos-ativos do RDS para MySQL

As seguintes limitações se aplicam a clusters ativos-ativos do RDS para MySQL:

- O nome de usuário principal não pode ser `rdsgrprepladmin` para instâncias de banco de dados em um cluster ativo-ativo. Esse nome de usuário é reservado para conexões da Group Replication.
- Para instâncias de banco de dados com réplicas de leitura em clusters ativos-ativos, um status de replicação prolongado diferente de `Replicating` pode fazer com que os arquivos de log excedam os limites de armazenamento. Para ter informações sobre o status de réplicas de leitura, consulte Monitoramento da replicação de leitura.
- As implantações azul/verde não são compatíveis com instâncias de banco de dados em um cluster ativo-ativo. Para ter mais informações, consulte Usar implantações azul/verde do Amazon RDS para atualizações de banco de dados.
- A autenticação Kerberos não é compatível com instâncias de banco de dados em um cluster ativo-ativo. Para ter mais informações, consulte Uso da autenticação do Kerberos para o Amazon RDS para MySQL.
- As instâncias de banco de dados em um cluster de banco de dados multi-AZ não podem ser adicionadas a um cluster ativo-ativo. No entanto, as instâncias de banco de dados em uma implantação de instância de banco de dados multi-AZ podem ser adicionadas a um cluster ativo-ativo. Para ter mais informações, consulte Configurar e gerenciar uma implantação multi-AZ para o Amazon RDS.
- As tabelas que não têm uma chave primária não são replicadas em um cluster ativo-ativo porque as gravações são rejeitadas pelo plug-in Group Replication.
- As tabelas que não são do InnoDB não são replicadas em um cluster ativo-ativo.
- Clusters ativos-ativos não comportam declarações DML e DDL simultâneas em diferentes instâncias de banco de dados no cluster.
- Não é possível configurar um cluster ativo-ativo para usar o modo primário único para o modo de replicação do grupo. Para essa configuração, recomendamos usar um cluster de banco de dados multi-AZ. Para ter mais informações, consulte Implantações de cluster de banco de dados multi-AZ para o Amazon RDS.
- A replicação de várias fontes não é compatível com instâncias de banco de dados em um cluster ativo-ativo.
- Um cluster ativo-ativo entre regiões não pode impor a verificação da autoridade de certificação (CA) para conexões da Group Replication.

Considerações e práticas recomendadas para cluster ativo-ativo do RDS para MySQL

Antes de usar clusters ativos-ativos do RDS para MySQL, analise as seguintes considerações e práticas recomendadas:

- Os clusters ativos-ativos não podem ter mais de nove instâncias de banco de dados.
- Com o plug-in Group Replication, é possível controlar as garantias de consistência da transação do cluster ativo-ativo. Para ter mais informações, consulte Transaction Consistency Guarantees na documentação do MySQL.
- Conflitos são possíveis quando diferentes instâncias de banco de dados atualizam a mesma linha em um cluster ativo-ativo. Para ter informações sobre conflitos e sua resolução, consulte Group Replication na documentação do MySQL.
- Para tolerância a falhas, inclua pelo menos três instâncias de banco de dados no cluster ativo-ativo. É possível configurar um cluster ativo-ativo com apenas uma ou duas instâncias de banco de dados, mas o cluster não tolerará falhas. Para ter informações sobre tolerância a falhas, consulte Fault -tolerance na documentação do MySQL.
- Quando uma instância de banco de dados ingressa em um cluster ativo-ativo existente e está executando a mesma versão que a mais baixa do mecanismo no cluster, a instância de banco de dados ingressa no modo de leitura-gravação.
- Quando uma instância de banco de dados ingressa em um cluster ativo-ativo existente e está executando uma versão mais alta do que a mais do mecanismo no cluster, a instância de banco de dados deve permanecer no modo de leitura-gravação.
- Se você habilitar a Group Replication para uma instância de banco de dados definindo o parâmetro `rds.group_replication_enabled` como `1` no grupo de parâmetros do banco de dados, mas a replicação não foi iniciada ou falhou ao iniciar, a instância de banco de dados será colocada no modo de superleitura para evitar inconsistências de dados. Para ter informações sobre o modo superleitura, consulte a documentação do MySQL.
- É possível atualizar uma instância de banco de dados em um cluster ativo-ativo, mas a instância de banco de dados é somente leitura até que todas as outras instâncias de banco de dados no cluster ativo-ativo sejam atualizadas para a mesma versão do mecanismo ou uma versão superior. Quando você atualiza uma instância de banco de dados, a instância de banco de dados ingressa automaticamente no mesmo cluster ativo-ativo quando a atualização é concluída. Para evitar uma mudança não intencional para o modo somente leitura de uma instância de banco de dados, desabilite as atualizações automáticas de versões secundárias para ela. Para ter mais informações sobre como atualizar uma instância de banco de dados MySQL, consulte Atualizações do mecanismo de banco de dados do RDS para MySQL.

- É possível adicionar uma instância de banco de dados em uma implantação de instância de banco de dados multi-AZ para um cluster ativo-ativo existente. Também é possível converter uma instância de banco de dados single-AZ em um cluster ativo-ativo em uma implantação de instância de banco de dados multi-AZ. Se uma instância de banco de dados primária em uma implantação multi-AZ falhar, essa instância primária fará o failover para a instância em espera. A nova instância de banco de dados primária ingressa automaticamente no mesmo cluster após a conclusão do failover. Para ter mais informações sobre implantações de instâncias de banco de dados multi-AZ, consulte Implantações de instâncias de banco de dados multi-AZ para o Amazon RDS.
- Recomendamos que as instâncias de banco de dados em um cluster ativo-ativo tenham intervalos de tempo diferentes para as janelas de manutenção. Essa prática evita que várias instâncias de banco de dados no cluster fiquem off-line para manutenção ao mesmo tempo. Para ter mais informações, consulte A janela de manutenção do Amazon RDS.
- Clusters ativos-ativos podem usar SSL para conexões entre instâncias de banco de dados. Para configurar conexões SSL, defina os parâmetros group_replication_recovery_use_ssl e group_replication_ssl_mode. Os valores desses parâmetros devem coincidir em todas as instâncias de banco de dados no cluster ativo-ativo.

  Atualmente, os clusters ativos-ativos não comportam a verificação de autoridade de certificação (CA) para conexões entre Regiões da AWS. Portanto, o parâmetro group_replication_ssl_mode deve ser definido como `DISABLED` (o padrão) ou como `REQUIRED` para clusters entre regiões.
- Um cluster ativo-ativo do RDS para MySQL é executado no modo multiprimário. O valor padrão de group_replication_enforce_update_everywhere_checks é e o parâmetro é `ON` e o parâmetro é estático. Quando esse parâmetro é definido como `ON`, as aplicações não podem ser inseridas em uma tabela que tenha restrições de chave externa em cascata.
- Um cluster ativo do RDS para MySQL usa a pilha de comunicação MySQL para segurança de conexão em vez de XCOM. Para ter mais informações, consulte Communication Stack for Connection Security Management na documentação do MySQL.
- Quando um grupo de parâmetros de banco de dados está associado a uma instância de banco de dados em um cluster ativo-ativo, recomendamos associar esse grupo de parâmetros de banco de dados somente a outras instâncias de banco de dados que estejam no cluster.
- Os clusters ativos-ativos são compatíveis somente com instâncias de banco de dados do RDS para MySQL. Essas instâncias de banco de dados devem estar executando versões compatíveis do mecanismo de banco de dados.
- Quando uma instância de banco de dados em um cluster ativo-ativo sofre uma falha inesperada, o RDS inicia a recuperação dela automaticamente. Se a instância de banco de dados não se

recuperar, recomendamos substituí-la por uma nova instância de banco de dados executando uma recuperação pontual com uma instância de banco de dados íntegra no cluster. Para obter instruções, consulte Adicionar uma instância de banco de dados a um cluster ativo-ativo usando a recuperação para um ponto no tempo.

- É possível excluir uma instância de banco de dados em um cluster ativo-ativo sem afetar as outras instâncias de banco de dados no cluster. Para obter informações sobre como excluir uma instância de banco de dados, consulte Excluir uma instância de banco de dados.
- Quando uma instância de banco de dados deixa acidentalmente um cluster ativo-ativo, por padrão, o parâmetro `group_replication_exit_state_action` é alterado para `OFFLINE_MODE`. Nesse estado, a instância de banco de dados fica inacessível e você deve reinicializá-la para colocá-la novamente online e para que ela volte a fazer parte do cluster. Você pode alterar esse comportamento modificando o parâmetro `group_replication_exit_state_action` em um grupo de parâmetros personalizado. Ao definir o parâmetro como `READ_ONLY`, quando a instância de banco de dados sai acidentalmente de um cluster, ela entra em um super estado de somente leitura em vez de ficar offline.

Preparar um cluster ativo-ativo entre VPCs

É possível configurar um cluster ativo-ativo com instâncias de banco de dados do Amazon RDS para MySQL em mais de uma VPC. As VPCs podem se encontrar na mesma Região da AWS ou em diferentes Regiões da AWS.

> **Note**
>
> O envio de tráfego entre várias Regiões da AWS pode gerar custos adicionais. Para ter mais informações, consulte Overview of Data Transfer Costs for Common Architectures.

Se você estiver configurando um cluster ativo-ativo em uma única VPC, poderá ignorar essas etapas e ir para Configurar um cluster ativo-ativo com novas instâncias de banco de dados.

Como se preparar para um cluster ativo-ativo com instâncias de banco de dados em mais de uma VPC

1. Garanta que os intervalos de endereços IPv4 nos blocos CIDR atendam aos seguintes requisitos:

- Os intervalos de endereços IPv4 nos blocos CIDR das VPCs não podem se sobrepor.
- Todos os intervalos de endereços IPv4 nos blocos CIDR devem ser menores que `128.0.0.0/subnet_mask` ou maiores que 128.0.0.0/`subnet_mask`.

Os seguintes intervalos ilustram estes requisitos:

- `10.1.0.0/16` em uma VPC e `10.2.0.0/16` na outra VPC são aceitos.
- `172.1.0.0/16` em uma VPC e `172.2.0.0/16` na outra VPC são aceitos.
- `10.1.0.0/16` em uma VPC e `10.1.0.0/16` na outra VPC não são aceitos porque os intervalos se sobrepõem.
- `10.1.0.0/16` em uma VPC e `172.1.0.0/16` na outra VPC não são aceitos porque um está abaixo de `128.0.0.0/subnet_mask` e o outro está acima de `128.0.0.0/subnet_mask`.

Para ter informações sobre os blocos CIDR, consulte VPC CIDR blocks no Manual do usuário da Amazon VPC.

2. Em cada VPC, garanta que os nomes de host e a resolução do DNS estejam habilitados.

   Para ter instruções, consulte Exibir e atualizar atributos DNS para sua VPC no Manual do usuário da Amazon VPC.

3. Configure as VPCs para que você possa rotear o tráfego entre elas de uma das seguintes maneiras:

   - Crie uma conexão de emparelhamento da VPC entre as VPCs.

     Para ter instruções, consulte Create a VPC peering connection no Guia de emparelhamento da Amazon VPC. Em cada VPC, verifique se há regras de entrada para os grupos de segurança que façam referência a grupos de segurança na VPC emparelhada. Fazendo isso, você permite que o tráfego flua entre as instâncias associadas com o grupo de segurança referenciado na VPC emparelhada. Para ter instruções, consulte Update your security groups to reference peer security groups no Guia de emparelhamento da Amazon VPC.

   - Crie um gateway de trânsito entre as VPCs.

     Para ter instruções, consulte Getting started with transit gateways em Amazon VPC Transit Gateways. Em cada VPC, verifique se há regras de entrada para os grupos de segurança que permitam o tráfego da outra VPC, como regras de entrada que especifiquem o CIDR da outra

VPC. Ao fazer isso, você permite que o tráfego flua entre as instâncias associadas ao grupo de segurança referenciado no cluster ativo-ativo. Para ter mais informações, consulte Control traffic to your AWS resources using security groups no Manual do usuário da Amazon VPC.

Configurações de parâmetros obrigatórias para clusters ativos-ativos

Configurar parâmetros para clusters ativos-ativos no Amazon RDS para MySQL é essencial para manter o desempenho consistente e a estabilidade operacional. Esta tabela detalha os principais parâmetros que controlam a replicação, a resolução de conflitos e a distribuição da workload. A configuração correta garante uma sincronização eficiente entre os nós, minimiza o atraso na replicação e otimiza a utilização de recursos em ambientes distribuídos ou de alto tráfego.

Parameter	Descrição	Configuração necessária
`binlog_format`	Define o formato de registro em log binário. O valor padrão para o RDS para MySQL versão 8.0 e anterior é `MIXED`. O valor padrão para o RDS para MySQL versão 8.4 é `ROW`. Para ter mais informaçõ es, consulte a documentação do MySQL.	`ROW`
`enforce_gtid_consi stency`	Reforça a consistência do GTID para execução de declarações. O valor padrão do RDS para MySQL é `OFF`. Para ter mais informações, consulte a documentação do MySQL.	`ON`
`group_replication_ group_name`	Define o nome da Group Replication como um UUID. O formato do UUID é `11111111-2222-3333 -4444-555555555555`	Um UUID do MySQL

Parameter	Descrição	Configuração necessária
	. É possível gerar um UUID do MySQL conectando-se a uma instância de banco de dados MySQL e executando o `SELECT UUID()`. O valor deve ser o mesmo para todas as instâncias de banco de dados no cluster ativo-ativo. Para ter mais informações, consulte a documentação do MySQL.	
`gtid-mode`	Controla o registro em log baseado em GTID. O valor padrão do RDS para MySQL é `OFF_PERMISSIVE` . Para ter mais informações, consulte a documentação do MySQL.	`ON`

Parameter	Descrição	Configuração necessária
`rds.custom_dns_resolution`	Especifica se você deve permitir a resolução de DNS do servidor Amazon DNS na VPC. A resolução de DNS deve ser habilitada quando a Group Replication é habilitada com o parâmetro `rds.group_replication_enabled` . A resolução de DNS não deve ser habilitada quando a Group Replication é desabilitada com o parâmetro `rds.group_replication_enabled` . Para ter mais informações, consulte Amazon DNS server no Manual do usuário da Amazon VPC.	1
`rds.group_replication_enabled`	Especifica se a Group Replication está habilitada para uma instância de banco de dados. A Group Replication deve estar habilitada em uma instância de banco de dados em um cluster ativo-ativo.	1
`replica_preserve_commit_order` (RDS para MySQL 8.4 e versões posteriores) ou `slave_preserve_commit_order` (RDS para versões 8.0 do MySQL)	Controla a ordem em que as transações são confirmadas em uma réplica. O valor padrão do RDS para MySQL é `ON`. Para ter mais informações, consulte a documentação do MySQL.	`ON`

Converter uma instância de banco de dados existente em um cluster ativo-ativo

A versão do mecanismo de banco de dados da instância de banco de dados que você deseja migrar para um cluster ativo-ativo deve ser uma destas:

- Todas as versões do MySQL 8.4
- MySQL 8.0.35 e versões secundárias posteriores

Se você precisar atualizar a versão do mecanismo, consulte Atualizações do mecanismo de banco de dados do RDS para MySQL.

Se estiver configurando um cluster ativo-ativo com instâncias de banco de dados em mais de uma VPC, cumpra os pré-requisitos em Preparar um cluster ativo-ativo entre VPCs.

Conclua as etapas a seguir para migrar uma instância de banco de dados existente para um cluster ativo-ativo do RDS para MySQL.

Tópicos

- Etapa 1: Definir os parâmetros de cluster ativo-ativo em um ou mais grupos de parâmetros personalizados
- Etapa 2: Associar a instância de banco de dados a um grupo de parâmetros de banco de dados que tenha os parâmetros obrigatórios da Group Replication definidos
- Etapa 3: Criar o cluster ativo-ativo
- Etapa 4: Criar instâncias de banco de dados do RDS para MySQL adicionais para o cluster ativo-ativo
- Etapa 5: Inicializar o grupo na instância de banco de dados que você está convertendo
- Etapa 6: Iniciar a replicação nas outras instâncias de banco de dados no cluster ativo-ativo
- Etapa 7: (Recomendado) Conferir o status do cluster ativo-ativo

Etapa 1: Definir os parâmetros de cluster ativo-ativo em um ou mais grupos de parâmetros personalizados

As instâncias de banco de dados do RDS para MySQL em um cluster ativo-ativo devem estar associadas a um grupo de parâmetros personalizado que tenha a configuração correta para os

parâmetros obrigatórios. Para ter informações sobre os parâmetros e a configuração necessária para cada um, consulte Configurações de parâmetros obrigatórias para clusters ativos-ativos.

É possível definir esses parâmetros em novos grupos de parâmetros ou em grupos de parâmetros existentes. No entanto, para não afetar acidentalmente as instâncias de banco de dados que não fazem parte do cluster ativo-ativo, é altamente recomendável criar um grupo de parâmetros personalizado. As instâncias de banco de dados em um cluster ativo-ativo podem ser associadas ao mesmo grupo de parâmetros de banco de dados ou a grupos de parâmetros de banco de dados diferentes.

É possível usar o AWS Management Console ou a AWS CLI para criar um grupo de parâmetros personalizado. Para ter mais informações, consulte Criar um grupo de parâmetros de banco de dados no Amazon RDS. O seguinte exemplo executa o comando create-db-parameter-group da AWS CLI para criar um grupo de parâmetros de banco de dados personalizado chamado *myactivepg* para o RDS para MySQL 8.0:

Para Linux, macOS ou Unix:

```
aws rds create-db-parameter-group \
  --db-parameter-group-name myactivepg \
  --db-parameter-group-family mysql8.0 \
  --description "Parameter group for active-active clusters"
```

Para Windows:

```
aws rds create-db-parameter-group ^
  --db-parameter-group-name myactivepg ^
  --db-parameter-group-family mysql8.0 ^
  --description "Parameter group for active-active clusters"
```

Também é possível usar o AWS Management Console ou a AWS CLI para definir os parâmetros no grupo de parâmetros personalizado. Para ter mais informações, consulte Modificar parâmetros em um grupo de parâmetros de banco de dados no Amazon RDS.

O seguinte exemplo executa o comando modify-db-parameter-group da AWS CLI para definir os parâmetros para o RDS para MySQL 8.0. Para usar esse exemplo com o RDS para MySQL 8.4, altere `slave_preserve_commit_order` para `replica_preserve_commit_order`.

Para Linux, macOS ou Unix:

```
aws rds modify-db-parameter-group \
  --db-parameter-group-name myactivepg \
  --parameters
 "ParameterName='rds.group_replication_enabled',ParameterValue='1',ApplyMethod=pending-
reboot" \

 "ParameterName='rds.custom_dns_resolution',ParameterValue='1',ApplyMethod=pending-
reboot" \

 "ParameterName='enforce_gtid_consistency',ParameterValue='ON',ApplyMethod=pending-
reboot" \
              "ParameterName='gtid-mode',ParameterValue='ON',ApplyMethod=pending-
reboot" \

 "ParameterName='binlog_format',ParameterValue='ROW',ApplyMethod=immediate" \

 "ParameterName='slave_preserve_commit_order',ParameterValue='ON',ApplyMethod=immediate"
 \

 "ParameterName='group_replication_group_name',ParameterValue='11111111-2222-3333-4444-55555555
reboot"
```

Para Windows:

```
aws rds modify-db-parameter-group ^
  --db-parameter-group-name myactivepg ^
  --parameters
 "ParameterName='rds.group_replication_enabled',ParameterValue='1',ApplyMethod=pending-
reboot" ^

 "ParameterName='rds.custom_dns_resolution',ParameterValue='1',ApplyMethod=pending-
reboot" ^

 "ParameterName='enforce_gtid_consistency',ParameterValue='ON',ApplyMethod=pending-
reboot" ^
              "ParameterName='gtid-mode',ParameterValue='ON',ApplyMethod=pending-
reboot" ^

 "ParameterName='binlog_format',ParameterValue='ROW',ApplyMethod=immediate" ^

 "ParameterName='slave_preserve_commit_order',ParameterValue='ON',ApplyMethod=immediate"
 ^
```

```
 "ParameterName='group_replication_group_name',ParameterValue='11111111-2222-3333-4444-55555555
reboot"
```

Etapa 2: Associar a instância de banco de dados a um grupo de parâmetros de banco de dados que tenha os parâmetros obrigatórios da Group Replication definidos

Associe a instância de banco de dados a um grupo de parâmetros que você criou ou modificou na etapa anterior. Para obter instruções, consulte Associar um grupo de parâmetros de banco de dados a uma instância de banco de dados no Amazon RDS.

Reinicialize a instância de banco de dados para que as novas configurações dos parâmetros tenham efeito. Para obter instruções, consulte Reinicializar uma instância de banco de dados .

Etapa 3: Criar o cluster ativo-ativo

No grupo de parâmetros de banco de dados associado à instância de banco de dados, defina o parâmetro `group_replication_group_seeds` como o endpoint da instância de banco de dados que você está convertendo.

É possível usar o AWS Management Console ou a AWS CLI para definir o parâmetro. Não é necessário reinicializar a instância de banco de dados depois de definir esse parâmetro. Para obter mais informações sobre como configurar parâmetros, consulte Modificar parâmetros em um grupo de parâmetros de banco de dados no Amazon RDS.

O seguinte exemplo executa o comando modify-db-parameter-group da AWS CLI para definir os parâmetros:

Para Linux, macOS ou Unix:

```
aws rds modify-db-parameter-group \
  --db-parameter-group-name myactivepg \
  --parameters
 "ParameterName='group_replication_group_seeds',ParameterValue='myactivedb1.123456789012.us-
east-1.rds.amazonaws.com:3306',ApplyMethod=immediate"
```

Para Windows:

```
aws rds modify-db-parameter-group ^
  --db-parameter-group-name myactivepg ^
```

```
  --parameters
 "ParameterName='group_replication_group_seeds',ParameterValue='myactivedb1.123456789012.us-
east-1.rds.amazonaws.com:3306',ApplyMethod=immediate"
```

Etapa 4: Criar instâncias de banco de dados do RDS para MySQL adicionais para o cluster ativo-ativo

Para criar instâncias de banco de dados adicionais para o cluster ativo-ativo, execute a recuperação pontual na instância de banco de dados que você está convertendo. Para obter instruções, consulte Adicionar uma instância de banco de dados a um cluster ativo-ativo usando a recuperação para um ponto no tempo.

Um cluster ativo-ativo pode ter até nove instâncias de banco de dados. Execute a recuperação pontual na instância de banco de dados até ter o número de instâncias de banco de dados que você deseja para o cluster. Ao realizar a recuperação pontual, não se esqueça de associar a instância de banco de dados que você está adicionando a um grupo de parâmetros de banco de dados que tenha `rds.group_replication_enabled` definida como 1. Caso contrário, a Group Replication não será iniciada na instância de banco de dados recém-adicionada.

Etapa 5: Inicializar o grupo na instância de banco de dados que você está convertendo

Inicialize o grupo e inicie a replicação:

1. Conecte-se à instância de banco de dados que você está convertendo em um cliente SQL. Para ter mais informações sobre como se conectar a uma instância de banco de dados do RDS para MySQL, consulte Conectar-se à instância de banco de dados do MySQL.
2. No cliente SQL, execute os procedimentos armazenados a seguir e substitua *group_replication_user_password* pela senha do usuário `rdsgrprepladmin`. O usuário `rdsgrprepladmin` está reservado para conexões da Group Replication em um cluster ativo-ativo. A senha desse usuário deve ser a mesma em todas as instâncias de banco de dados em um cluster ativo-ativo.

   ```
   call mysql.rds_set_configuration('binlog retention hours', 168); -- 7 days binlog
   call mysql.rds_group_replication_create_user('group_replication_user_password');
   call
    mysql.rds_group_replication_set_recovery_channel('group_replication_user_password');
   call mysql.rds_group_replication_start(1);
   ```

Este exemplo define o valor `binlog retention hours` como 168, o que significa que os arquivos de log binários são retidos por sete dias na instância de banco de dados. É possível ajustar esse valor para atender aos requisitos.

Este exemplo especifica 1 no procedimento armazenado `mysql.rds_group_replication_start` para inicializar um novo grupo com a instância de banco de dados atual.

Para ter mais informações sobre os procedimentos armazenados chamados no exemplo, consulte Gerenciar clusters ativos-ativos.

Etapa 6: Iniciar a replicação nas outras instâncias de banco de dados no cluster ativo-ativo

Para cada uma das instâncias de banco de dados no cluster ativo-ativo, use um cliente SQL para se conectar à instância e execute os procedimentos armazenados a seguir. Substitua *`group_replication_user_password`* pela senha do usuário `rdsgrprepladmin`.

```
call mysql.rds_set_configuration('binlog retention hours', 168); -- 7 days binlog
call mysql.rds_group_replication_create_user('group_replication_user_password');
call
 mysql.rds_group_replication_set_recovery_channel('group_replication_user_password');
call mysql.rds_group_replication_start(0);
```

Este exemplo define o valor `binlog retention hours` como 168, o que significa que os arquivos de log binários são retidos por sete dias em cada instância de banco de dados. É possível ajustar esse valor para atender aos requisitos.

Este exemplo especifica `0` no procedimento armazenado `mysql.rds_group_replication_start` para inserir a instância de banco de dados atual em um grupo existente.

> **Tip**
>
> Execute esses procedimentos armazenados em todas as outras instâncias de banco de dados no cluster ativo-ativo.

Etapa 7: (Recomendado) Conferir o status do cluster ativo-ativo

Para garantir que cada membro do cluster esteja configurado corretamente, confira o status do cluster conectando-se a uma instância de banco de dados no cluster ativo-ativo e executando o seguinte comando SQL:

```
SELECT * FROM performance_schema.replication_group_members;
```

A saída deve exibir `ONLINE` para `MEMBER_STATE` de cada instância de banco de dados, como neste exemplo de saída:

```
+---------------------------+--------------------------------------
+---------------+-------------+--------------+-------------+---------------
+----------------------------+
| CHANNEL_NAME              | MEMBER_ID                            | MEMBER_HOST    |
 MEMBER_PORT | MEMBER_STATE | MEMBER_ROLE | MEMBER_VERSION | MEMBER_COMMUNICATION_STACK
 |
+---------------------------+--------------------------------------
+---------------+-------------+--------------+-------------+---------------
+----------------------------+
| group_replication_applier | 9854d4a2-5d7f-11ee-b8ec-0ec88c43c251 | ip-10-15-3-137 |
      3306 | ONLINE       | PRIMARY     | 8.0.35         | MySQL                       |
| group_replication_applier | 9e2e9c28-5d7f-11ee-8039-0e5d58f05fef | ip-10-15-3-225 |
      3306 | ONLINE       | PRIMARY     | 8.0.35         | MySQL                       |
| group_replication_applier | a6ba332d-5d7f-11ee-a025-0a5c6971197d | ip-10-15-1-83  |
      3306 | ONLINE       | PRIMARY     | 8.0.35         | MySQL                       |
+---------------------------+--------------------------------------
+---------------+-------------+--------------+-------------+---------------
+----------------------------+
3 rows in set (0.00 sec)
```

Para ter informações sobre os valores `MEMBER_STATE` possíveis, consulte Group Replication Server States na documentação do MySQL.

Configurar um cluster ativo-ativo com novas instâncias de banco de dados

Conclua as etapas a seguir para configurar um cluster ativo-ativo usando novas instâncias de banco de dados do Amazon RDS para MySQL.

Se estiver configurando um cluster ativo-ativo com instâncias de banco de dados em mais de uma VPC, cumpra os pré-requisitos em Preparar um cluster ativo-ativo entre VPCs.

Tópicos

- Etapa 1: Definir os parâmetros de cluster ativo-ativo em um ou mais grupos de parâmetros personalizados
- Etapa 2: Criar instâncias de banco de dados do RDS para MySQL para o cluster ativo-ativo
- Etapa 3: especificar as instâncias de banco de dados no cluster ativo-ativo
- Etapa 4: inicializar o grupo em uma instância de banco de dados e iniciar a replicação
- Etapa 5: iniciar a replicação nas outras instâncias de banco de dados no cluster ativo-ativo
- Etapa 6: (recomendado) conferir o status do cluster ativo-ativo
- Etapa 7: (opcional) importar dados para uma instância de banco de dados no cluster ativo-ativo

Etapa 1: Definir os parâmetros de cluster ativo-ativo em um ou mais grupos de parâmetros personalizados

As instâncias de banco de dados do RDS para MySQL em um cluster ativo-ativo devem estar associadas a um grupo de parâmetros personalizado que tenha a configuração correta para os parâmetros obrigatórios. Para ter informações sobre os parâmetros e a configuração necessária para cada um, consulte Configurações de parâmetros obrigatórias para clusters ativos-ativos.

É possível definir esses parâmetros em novos grupos de parâmetros ou em grupos de parâmetros existentes. No entanto, para não afetar acidentalmente as instâncias de banco de dados que não fazem parte do cluster ativo-ativo, é altamente recomendável criar um grupo de parâmetros personalizado. As instâncias de banco de dados em um cluster ativo-ativo podem ser associadas ao mesmo grupo de parâmetros de banco de dados ou a grupos de parâmetros de banco de dados diferentes.

É possível usar o AWS Management Console ou a AWS CLI para criar um grupo de parâmetros personalizado. Para ter mais informações, consulte Criar um grupo de parâmetros de banco de dados no Amazon RDS. O seguinte exemplo executa o comando create-db-parameter-group da AWS CLI para criar um grupo de parâmetros de banco de dados personalizado chamado *myactivepg* para o RDS para MySQL 8.0:

Para Linux, macOS ou Unix:

```
aws rds create-db-parameter-group \
  --db-parameter-group-name myactivepg \
```

```
  --db-parameter-group-family mysql8.0 \
  --description "Parameter group for active-active clusters"
```

Para Windows:

```
aws rds create-db-parameter-group ^
  --db-parameter-group-name myactivepg ^
  --db-parameter-group-family mysql8.0 ^
  --description "Parameter group for active-active clusters"
```

Também é possível usar o AWS Management Console ou a AWS CLI para definir os parâmetros no grupo de parâmetros personalizado. Para ter mais informações, consulte Modificar parâmetros em um grupo de parâmetros de banco de dados no Amazon RDS.

O seguinte exemplo executa o comando modify-db-parameter-group da AWS CLI para definir os parâmetros para o RDS para MySQL 8.0. Para usar esse exemplo com o RDS para MySQL 8.4, altere `slave_preserve_commit_order` para `replica_preserve_commit_order`.

Para Linux, macOS ou Unix:

```
aws rds modify-db-parameter-group \
  --db-parameter-group-name myactivepg \
  --parameters
 "ParameterName='rds.group_replication_enabled',ParameterValue='1',ApplyMethod=pending-
reboot" \

 "ParameterName='rds.custom_dns_resolution',ParameterValue='1',ApplyMethod=pending-
reboot" \

 "ParameterName='enforce_gtid_consistency',ParameterValue='ON',ApplyMethod=pending-
reboot" \
               "ParameterName='gtid-mode',ParameterValue='ON',ApplyMethod=pending-
reboot" \

 "ParameterName='binlog_format',ParameterValue='ROW',ApplyMethod=immediate" \

 "ParameterName='slave_preserve_commit_order',ParameterValue='ON',ApplyMethod=immediate"
 \

 "ParameterName='group_replication_group_name',ParameterValue='11111111-2222-3333-4444-55555555
reboot"
```

Para Windows:

```
aws rds modify-db-parameter-group ^
  --db-parameter-group-name myactivepg ^
  --parameters
 "ParameterName='rds.group_replication_enabled',ParameterValue='1',ApplyMethod=pending-
reboot" ^

 "ParameterName='rds.custom_dns_resolution',ParameterValue='1',ApplyMethod=pending-
reboot" ^

 "ParameterName='enforce_gtid_consistency',ParameterValue='ON',ApplyMethod=pending-
reboot" ^
               "ParameterName='gtid-mode',ParameterValue='ON',ApplyMethod=pending-
reboot" ^

 "ParameterName='binlog_format',ParameterValue='ROW',ApplyMethod=immediate" ^

 "ParameterName='slave_preserve_commit_order',ParameterValue='ON',ApplyMethod=immediate"
 ^

 "ParameterName='group_replication_group_name',ParameterValue='11111111-2222-3333-4444-55555555
reboot"
```

Etapa 2: Criar instâncias de banco de dados do RDS para MySQL para o cluster ativo-ativo

Clusters ativos-ativos são compatíveis com as seguintes versões de instâncias de banco de dados do RDS para MySQL:

- Todas as versões do MySQL 8.4
- MySQL versão 8.0.35 e versões secundárias posteriores

É possível criar até nove novas instâncias de banco de dados para o cluster.

É possível usar o AWS Management Console ou a AWS CLI para criar instâncias de banco de dados. Para ter mais informações sobre como criar uma instância de banco de dados, consulte Criar uma instância de banco de dados do Amazon RDS. Ao criar a instância de banco de dados, associe-a a um grupo de parâmetros de banco de dados que você criou ou modificou na etapa anterior.

Etapa 3: especificar as instâncias de banco de dados no cluster ativo-ativo

No grupo de parâmetros de banco de dados associado a cada instância de banco de dados, defina o parâmetro `group_replication_group_seeds` como os endpoints das instâncias de banco de dados que você deseja incluir no cluster.

É possível usar o AWS Management Console ou a AWS CLI para definir o parâmetro. Não é necessário reinicializar a instância de banco de dados depois de definir esse parâmetro. Para obter mais informações sobre como configurar parâmetros, consulte Modificar parâmetros em um grupo de parâmetros de banco de dados no Amazon RDS.

O seguinte exemplo executa o comando modify-db-parameter-group da AWS CLI para definir os parâmetros:

Para Linux, macOS ou Unix:

```
aws rds modify-db-parameter-group \
  --db-parameter-group-name myactivepg \
  --parameters
 "ParameterName='group_replication_group_seeds',ParameterValue='myactivedb1.123456789012.us-
east-1.rds.amazonaws.com:3306,myactivedb2.123456789012.us-
east-1.rds.amazonaws.com:3306,myactivedb3.123456789012.us-
east-1.rds.amazonaws.com:3306',ApplyMethod=immediate"
```

Para Windows:

```
aws rds modify-db-parameter-group ^
  --db-parameter-group-name myactivepg ^
  --parameters
 "ParameterName='group_replication_group_seeds',ParameterValue='myactivedb1.123456789012.us-
east-1.rds.amazonaws.com:3306,myactivedb2.123456789012.us-
east-1.rds.amazonaws.com:3306,myactivedb3.123456789012.us-
east-1.rds.amazonaws.com:3306',ApplyMethod=immediate"
```

> **Tip**
>
> Defina o parâmetro `group_replication_group_seeds` em cada grupo de parâmetros de banco de dados associado a uma instância de banco de dados no cluster ativo-ativo.

Etapa 4: inicializar o grupo em uma instância de banco de dados e iniciar a replicação

É possível selecionar qualquer novo banco de dados para inicializar o grupo e iniciar a replicação. Para fazer isso, realize as etapas a seguir:

1. Selecione uma instância de banco de dados no cluster ativo-ativo e conecte-se a essa instância de banco de dados em um cliente SQL. Para ter mais informações sobre como se conectar a uma instância de banco de dados do RDS para MySQL, consulte Conectar-se à instância de banco de dados do MySQL.
2. No cliente SQL, execute os procedimentos armazenados a seguir e substitua *group_replication_user_password* pela senha do usuário `rdsgrprepladmin`. O usuário `rdsgrprepladmin` está reservado para conexões da Group Replication em um cluster ativo-ativo. A senha desse usuário deve ser a mesma em todas as instâncias de banco de dados em um cluster ativo-ativo.

   ```
   call mysql.rds_set_configuration('binlog retention hours', 168); -- 7 days binlog
   call mysql.rds_group_replication_create_user('group_replication_user_password');
   call
    mysql.rds_group_replication_set_recovery_channel('group_replication_user_password');
   call mysql.rds_group_replication_start(1);
   ```

   Este exemplo define o valor `binlog retention hours` como 168, o que significa que os arquivos de log binários são retidos por sete dias na instância de banco de dados. É possível ajustar esse valor para atender aos requisitos.

   Este exemplo especifica 1 no procedimento armazenado `mysql.rds_group_replication_start` para inicializar um novo grupo com a instância de banco de dados atual.

   Para ter mais informações sobre os procedimentos armazenados chamados no exemplo, consulte Gerenciar clusters ativos-ativos.

Etapa 5: iniciar a replicação nas outras instâncias de banco de dados no cluster ativo-ativo

Para cada uma das instâncias de banco de dados no cluster ativo-ativo, use um cliente SQL para se conectar à instância e execute os procedimentos armazenados a seguir. Substitua *group_replication_user_password* pela senha do usuário `rdsgrprepladmin`.

```
call mysql.rds_set_configuration('binlog retention hours', 168); -- 7 days binlog
call mysql.rds_group_replication_create_user('group_replication_user_password');
call
 mysql.rds_group_replication_set_recovery_channel('group_replication_user_password');
call mysql.rds_group_replication_start(0);
```

Este exemplo define o valor `binlog retention hours` como 168, o que significa que os arquivos de log binários são retidos por sete dias em cada instância de banco de dados. É possível ajustar esse valor para atender aos requisitos.

Este exemplo especifica `0` no procedimento armazenado `mysql.rds_group_replication_start` para inserir a instância de banco de dados atual em um grupo existente.

> **Tip**
>
> Execute esses procedimentos armazenados em todas as outras instâncias de banco de dados no cluster ativo-ativo.

Etapa 6: (recomendado) conferir o status do cluster ativo-ativo

Para garantir que cada membro do cluster esteja configurado corretamente, confira o status do cluster conectando-se a uma instância de banco de dados no cluster ativo-ativo e executando o seguinte comando SQL:

```
SELECT * FROM performance_schema.replication_group_members;
```

A saída deve exibir `ONLINE` para `MEMBER_STATE` de cada instância de banco de dados, como neste exemplo de saída:

```
+---------------------------+--------------------------------------
+---------------+-------------+--------------+-------------+----------------
+----------------------------+
| CHANNEL_NAME              | MEMBER_ID                            | MEMBER_HOST    |
 MEMBER_PORT | MEMBER_STATE | MEMBER_ROLE | MEMBER_VERSION | MEMBER_COMMUNICATION_STACK
 |
+---------------------------+--------------------------------------
+---------------+-------------+--------------+-------------+----------------
+----------------------------+
```

```
| group_replication_applier | 9854d4a2-5d7f-11ee-b8ec-0ec88c43c251 | ip-10-15-3-137 |
      3306 | ONLINE       | PRIMARY     | 8.0.35         | MySQL                         |
| group_replication_applier | 9e2e9c28-5d7f-11ee-8039-0e5d58f05fef | ip-10-15-3-225 |
      3306 | ONLINE       | PRIMARY     | 8.0.35         | MySQL                         |
| group_replication_applier | a6ba332d-5d7f-11ee-a025-0a5c6971197d | ip-10-15-1-83  |
      3306 | ONLINE       | PRIMARY     | 8.0.35         | MySQL                         |
+---------------------------+--------------------------------------
+---------------+-------------+-------------+-------------+---------------
+----------------------------+
3 rows in set (0.00 sec)
```

Para ter informações sobre os valores `MEMBER_STATE` possíveis, consulte Group Replication Server States na documentação do MySQL.

Etapa 7: (opcional) importar dados para uma instância de banco de dados no cluster ativo-ativo

É possível importar dados de um banco de dados MySQL para uma instância de banco de dados no cluster ativo-ativo. Depois que os dados são importados, a Group Replication os replica para as outras instâncias de banco de dados no cluster.

Para ter informações sobre como importar dados, consulte Importar dados para um banco de dados do Amazon RDS para MySQL com tempo de inatividade reduzido.

Adicionar uma instância de banco de dados em um cluster ativo-ativo

É possível adicionar uma instância de banco de dados a um cluster ativo-ativo do Amazon RDS para MySQL restaurando um snapshot de banco de dados ou restaurando uma instância de banco de dados para um ponto no tempo. Um cluster ativo-ativo pode incluir até nove instâncias de banco de dados.

Ao recuperar uma instância de banco de dados em um ponto no tempo, geralmente ela inclui transações mais recentes do que uma instância de banco de dados que foi restaurada por meio de um snapshot de banco de dados. Quando a instância de banco de dados tem transações mais recentes, menos transações precisam ser aplicadas ao iniciar a replicação. Portanto, usar a recuperação para um ponto no tempo para adicionar uma instância de banco de dados a um cluster geralmente é mais rápido do que restaurar por meio de um snapshot de banco de dados.

Tópicos

- Adicionar uma instância de banco de dados a um cluster ativo-ativo usando a recuperação para um ponto no tempo
- Adicionar uma instância de banco de dados em um cluster ativo-ativo usando um snapshot de banco de dados

Adicionar uma instância de banco de dados a um cluster ativo-ativo usando a recuperação para um ponto no tempo

É possível adicionar uma instância de banco de dados a um cluster ativo-ativo realizando a recuperação para um ponto no tempo em uma instância de banco de dados no cluster.

Para ter informações sobre como recuperar uma instância de banco de dados para um ponto no tempo em uma Região da AWS diferente, consulte Replicar backups automatizados para outra Região da AWS.

Como adicionar uma instância de banco de dados a um cluster ativo-ativo usando a recuperação para um ponto no tempo

1. Crie uma instância de banco de dados realizando a recuperação para um ponto no tempo em uma instância de banco de dados no cluster ativo-ativo.

   É possível realizar a recuperação para um ponto no tempo em qualquer instância de banco de dados no cluster para criar a instância de banco de dados. Para obter instruções, consulte Restaurar uma instância de banco de dados para um momento especificado no Amazon RDS.

   > **Important**
   >
   > Durante a recuperação para um ponto no tempo, associe a nova instância de banco de dados a um grupo de parâmetros de banco de dados que tenha os parâmetros de cluster ativo-ativo definidos. Caso contrário, a Group Replication não será iniciada na nova instância de banco de dados. Para ter informações sobre os parâmetros e a configuração necessária para cada um, consulte Configurações de parâmetros obrigatórias para clusters ativos-ativos.

> **Tip**
>
> Se você criar um snapshot da instância de banco de dados antes de iniciar a recuperação para um ponto no tempo, poderá reduzir o tempo necessário para aplicar transações na nova instância de banco de dados.

2. Adicione a instância de banco de dados ao parâmetro `group_replication_group_seeds` em cada grupo de parâmetros de banco de dados associado a uma instância de banco de dados no cluster ativo-ativo, incluindo o grupo de parâmetros de banco de dados que você associou à nova instância de banco de dados.

   Para obter mais informações sobre como configurar parâmetros, consulte Modificar parâmetros em um grupo de parâmetros de banco de dados no Amazon RDS.

3. Em um cliente SQL, conecte-se à nova instância de banco de dados e chame o procedimento armazenado mysql.rds_group_replication_set_recovery_channel. Substitua *group_replication_user_password* pela senha do usuário `rdsgrprepladmin`.

```
call
 mysql.rds_group_replication_set_recovery_channel('group_replication_user_password');
```

4. Usando o cliente SQL, chame o procedimento armazenado mysql.rds_group_replication_start para iniciar a replicação:

```
call mysql.rds_group_replication_start(0);
```

Adicionar uma instância de banco de dados em um cluster ativo-ativo usando um snapshot de banco de dados

É possível adicionar uma instância de banco de dados a um cluster ativo-ativo criando um snapshot de banco de dados de uma instância de banco de dados no cluster e, depois, restaurando o snapshot de banco de dados.

Para ter mais informações como copiar um snapshot em uma Região da AWS diferente, consulte the section called “Cópia entre regiões”.

Como adicionar uma instância de banco de dados em um cluster ativo-ativo usando um snapshot de banco de dados

1. Crie um snapshot de banco de dados de uma instância de banco de dados no cluster ativo-ativo.

   É possível criar um snapshot de banco de dados de qualquer instância de banco de dados no cluster. Para obter instruções, consulte Criar um snapshot de banco de dados para uma instância de banco de dados single-AZ para o Amazon RDS.

2. Restaure uma instância de banco de dados por meio do snapshot de banco de dados.

   Durante a operação de recuperação de snapshot, associe a nova instância de banco de dados a um grupo de parâmetros de banco de dados que tenha os parâmetros de cluster ativo-ativo definidos. Para ter informações sobre os parâmetros e a configuração necessária para cada um, consulte Configurações de parâmetros obrigatórias para clusters ativos-ativos.

   Para ter mais informações sobre como restaurar uma instância de banco de dados por meio de um snapshot de banco de dados, consulte Restaurar uma instância de banco de dados.

3. Adicione a instância de banco de dados ao parâmetro `group_replication_group_seeds` em cada grupo de parâmetros de banco de dados associado a uma instância de banco de dados no cluster ativo-ativo, incluindo o grupo de parâmetros de banco de dados que você associou à nova instância de banco de dados.

   Para obter mais informações sobre como configurar parâmetros, consulte Modificar parâmetros em um grupo de parâmetros de banco de dados no Amazon RDS.

4. Em um cliente SQL, conecte-se à nova instância de banco de dados e chame o procedimento armazenado mysql.rds_group_replication_set_recovery_channel. Substitua *`group_replication_user_password`* pela senha do usuário `rdsgrprepladmin`.

   ```
   call
    mysql.rds_group_replication_set_recovery_channel('group_replication_user_password');
   ```

5. Usando o cliente SQL, chame o procedimento armazenado mysql.rds_group_replication_start para iniciar a replicação:

   ```
   call mysql.rds_group_replication_start(0);
   ```

Monitorar clusters ativos-ativos

O monitoramento de clusters ativos-ativos no Amazon RDS para MySQL é crucial para monitorar o desempenho, a integridade da replicação e a sincronização de nós. É possível monitorar o cluster ativo-ativo conectando-se a uma instância de banco de dados no cluster e executando o seguinte comando SQL:

```
SELECT * FROM performance_schema.replication_group_members;
```

A saída deve exibir `ONLINE` para `MEMBER_STATE` de cada instância de banco de dados, como neste exemplo de saída:

```
+---------------------------+--------------------------------------
+---------------+-------------+--------------+-------------+---------------
+----------------------------+
| CHANNEL_NAME              | MEMBER_ID                            | MEMBER_HOST    |
 MEMBER_PORT | MEMBER_STATE | MEMBER_ROLE | MEMBER_VERSION | MEMBER_COMMUNICATION_STACK
 |
+---------------------------+--------------------------------------
+---------------+-------------+--------------+-------------+---------------
+----------------------------+
| group_replication_applier | 9854d4a2-5d7f-11ee-b8ec-0ec88c43c251 | ip-10-15-3-137 |
      3306 | ONLINE       | PRIMARY     | 8.0.35         | MySQL                       |
| group_replication_applier | 9e2e9c28-5d7f-11ee-8039-0e5d58f05fef | ip-10-15-3-225 |
      3306 | ONLINE       | PRIMARY     | 8.0.35         | MySQL                       |
| group_replication_applier | a6ba332d-5d7f-11ee-a025-0a5c6971197d | ip-10-15-1-83  |
      3306 | ONLINE       | PRIMARY     | 8.0.35         | MySQL                       |
+---------------------------+--------------------------------------
+---------------+-------------+--------------+-------------+---------------
+----------------------------+
3 rows in set (0.00 sec)
```

Para ter informações sobre os valores `MEMBER_STATE` possíveis, consulte Group Replication Server States na documentação do MySQL.

Interromper a Group Replication em uma instância de banco de dados em um cluster ativo-ativo

É possível interromper a Group Replication em uma instância de banco de dados em um cluster ativo-ativo. Ao interromper a Group Replication, a instância de banco de dados é colocada no modo

somente superleitura até que a replicação seja reiniciada ou essa instância de banco de dados seja removida do cluster ativo-ativo. Para ter informações sobre o modo superleitura, consulte a documentação do MySQL.

Como interromper temporariamente a Group Replication para um cluster ativo-ativo

1. Conecte-se a uma instância de banco de dados no cluster ativo-ativo usando um cliente SQL.

   Para ter mais informações sobre como se conectar a uma instância de banco de dados do RDS para MySQL, consulte Conectar-se à instância de banco de dados do MySQL.
2. No cliente SQL, chame o procedimento armazenado mysql.rds_group_replication_stop:

```
call mysql.rds_group_replication_stop();
```

Renomear uma instância de banco de dados em um cluster ativo-ativo

É possível alterar o nome de uma instância de banco de dados em um cluster ativo-ativo. Para renomear mais de uma instância de banco de dados em um cluster ativo-ativo, faça isso em uma instância de banco de dados por vez. Então, renomeie uma instância de banco de dados e insira-a novamente no cluster antes de renomear a próxima instância de banco de dados.

Como renomear uma instância de banco de dados em um cluster ativo-ativo

1. Conecte-se à instância de banco de dados em um cliente SQL e chame o procedimento armazenado mysql.rds_group_replication_stop:

```
call mysql.rds_group_replication_stop();
```

2. Renomeie a instância de banco de dados seguindo as instruções em Renomear uma instância de banco de dados.
3. Modifique o parâmetro `group_replication_group_seeds` em cada grupo de parâmetros de banco de dados associado a uma instância de banco de dados no cluster ativo-ativo.

   Na configuração de parâmetros, substitua o endpoint antigo da instância de banco de dados pelo novo endpoint da instância de banco de dados. Para obter mais informações sobre como configurar parâmetros, consulte Modificar parâmetros em um grupo de parâmetros de banco de dados no Amazon RDS.

4. Conecte-se à instância de banco de dados em um cliente SQL e chame o procedimento armazenado mysql.rds_group_replication_start:

```
call mysql.rds_group_replication_start(0);
```

Remover uma instância de banco de dados de um cluster ativo-ativo

Ao remover uma instância de banco de dados de um cluster ativo-ativo, ela é revertida para uma instância de banco de dados autônoma.

Como remover uma instância de banco de dados de um cluster ativo-ativo

1. Conecte-se à instância de banco de dados em um cliente SQL e chame o procedimento armazenado mysql.rds_group_replication_stop:

```
call mysql.rds_group_replication_stop();
```

2. Modifique o parâmetro `group_replication_group_seeds` para as instâncias de banco de dados que permanecerão no cluster ativo-ativo.

   No parâmetro `group_replication_group_seeds`, exclua a instância de banco de dados que você está removendo do cluster ativo-ativo. Para obter mais informações sobre como configurar parâmetros, consulte Modificar parâmetros em um grupo de parâmetros de banco de dados no Amazon RDS.

3. Modifique os parâmetros da instância de banco de dados que você está removendo do cluster ativo-ativo para que ela não faça mais parte do cluster.

   É possível associar a instância de banco de dados a um grupo de parâmetros diferente ou modificar os parâmetros no grupo de parâmetros de banco de dados associado à instância. Os parâmetros a serem modificados incluem `group_replication_group_name`, `rds.group_replication_enabled` e `group_replication_group_seeds`. Para ter mais informações sobre parâmetros de cluster ativo-ativo, consulte Configurações de parâmetros obrigatórias para clusters ativos-ativos.

   Se você modificar os parâmetros em um grupo de parâmetros de banco de dados, garanta que o grupo de parâmetros de banco de dados não esteja associado a outras instâncias de banco de dados no cluster ativo-ativo.

4. Reinicialize a instância de banco de dados que você removeu do cluster ativo-ativo para que as novas configurações de parâmetros tenham efeito.

   Para obter instruções, consulte Reinicializar uma instância de banco de dados .

Exportar dados de uma instância de banco de dados MySQL usando replicação

Para exportar dados de uma instância de banco de dados do RDS para MySQL para uma instância do MySQL executada externamente ao Amazon RDS, é possível usar a replicação. Nesse cenário, a instância de banco de dados MySQL é a instância de banco de dados MySQL de origem e a instância MySQL executada externamente em relação ao Amazon RDS é o banco de dados MySQL externo.

O banco de dados MySQL externo pode ser executado localmente em seu datacenter ou em uma instância do Amazon EC2. O banco de dados MySQL externo deve executar a mesma versão da instância de banco de dados MySQL de origem ou uma versão posterior.

A replicação para um banco de dados MySQL externo só é compatível durante o tempo necessário para exportar um banco de dados da instância de banco de dados MySQL de origem. A replicação deve ser encerrada quando os dados tiverem sido exportados e os aplicativos puderem começar a acessar a instância MySQL externa.

A lista a seguir mostra as etapas a serem realizadas. Cada etapa é discutida em maiores detalhes nas seções mais adiante.

1. Prepare uma instância de banco de dados MySQL externa.
2. Prepare a instância de banco de dados MySQL de origem para replicação.
3. Use o utilitário mysqldump para transferir o banco de dados da instância de banco de dados MySQL de origem para o banco de dados MySQL externo.
4. Inicie a replicação para o banco de dados MySQL externo.
5. Após a conclusão da exportação, pare a replicação.

Preparar um banco de dados MySQL externo

Execute as etapas a seguir para preparar o banco de dados MySQL externo.

Como preparar o banco de dados MySQL externo

1. Instale o banco de dados MySQL externo.

2. Conecte-se ao banco de dados MySQL externo como o usuário mestre. Depois, crie os usuários necessários para dar suporte aos administradores, aplicativos e serviços que acessam o banco de dados.
3. Siga as instruções na documentação do MySQL para preparar o banco de dados MySQL externo como uma réplica. Para ter mais informações, consulte Setting the Replica Configuration, na documentação do MySQL.
4. Configure uma regra de saída para o banco de dados MySQL externo operar como uma réplica de leitura durante a exportação. A regra de saída permite que o banco de dados MySQL externo se conecte à instância de banco de dados MySQL de origem durante a replicação. Especifique uma regra de saída que permita conexões Transmission Control Protocol (TCP) com a porta e o endereço IP da instância de banco de dados MySQL de origem.

   Especifique as regras de saída apropriadas para o ambiente:

   - Se o banco de dados MySQL externo estiver sendo executado em uma instância do Amazon EC2 em uma nuvem privada virtual (VPC) com base no serviço da Amazon VPC, especifique as regras de saída em um grupo de segurança da VPC. Para obter mais informações, consulteControlar acesso com grupos de segurança
   - Se o banco de dados MySQL externo estiver instalado localmente, especifique as regras de saída em um firewall.
5. Se o banco de dados MySQL externo estiver sendo executado em uma VPC, configure regras para a lista de controle de acesso (ACL) da VPC, além da regra de saída do grupo de segurança:

   - Configure uma regra de entrada da ACL que permita o tráfego TCP para as portas 1024–65535 a partir do endereço IP da instância de banco de dados MySQL de origem.
   - Configure uma regra de saída da ACL que permita o tráfego TCP de saída para a porta e o endereço IP da instância de banco de dados MySQL de origem.

   Para obter mais informações sobre network ACLs da Amazon VPC, consulte Network ACLs no Guia do usuário da Amazon VPC.
6. (Opcional) Defina o parâmetro `max_allowed_packet` como o tamanho máximo para evitar erros de replicação. Recomendamos essa configuração.

Preparar a instância de banco de dados MySQL

Execute as etapas a seguir para preparar a instância de banco de dados MySQL de origem como a origem de replicação.

Como preparar a instância de banco de dados MySQL de origem

1. Ao configurar a replicação, verifique se o computador cliente tem espaço suficiente em disco disponível para salvar os logs binários.
2. Conecte-se à instância de banco de dados do MySQL de origem e crie uma conta de replicação seguindo as instruções em Creating a User for Replication na documentação do MySQL.
3. Configure regras de entrada no sistema que executa a instância de banco de dados MySQL de origem para permitir que o banco de dados MySQL externo se conecte durante a replicação. Especifique uma regra de entrada que permita conexões TCP com a porta usada pela instância de banco de dados MySQL de origem do endereço IP do banco de dados MySQL externo.
4. Especifique as regras de saída:

   - Se a instância de banco de dados MySQL de origem estiver em execução em uma VPC, especifique as regras de entrada em um grupo de segurança da VPC. Para ter mais informações, consulte Controlar acesso com grupos de segurança.
5. Se a instância de banco de dados MySQL de origem estiver sendo executada em uma VPC, configure regras de ACL da VPC, além da regra de entrada do grupo de segurança:

   - Configure uma regra de entrada de ACL para permitir conexões TCP com a porta usada pela instância do Amazon RDS do endereço IP do banco de dados MySQL externo.
   - Configure uma regra de saída de ACL para permitir conexões TCP das portas 1024–65535 com o endereço IP do banco de dados MySQL externo.

   Para obter mais informações sobre network ACLs da Amazon VPC, consulte Network ACLs no Guia do usuário da Amazon VPC.
6. Certifique-se de que o período de retenção de backup seja definido o tempo suficiente para que nenhum log binário seja removido durante a exportação. Se algum dos logs for limpo antes da conclusão da exportação, será necessário reiniciar a replicação desde o início. Para obter mais informações sobre como configurar o período de retenção de backup, consulte Introdução aos backups.

7. Use o procedimento armazenado `mysql.rds_set_configuration` para definir o período de retenção de logs binários pelo tempo suficiente para que os logs binários não sejam limpos durante a exportação. Para obter mais informações, consulteAcessar logs binários do MySQL
8. Crie uma réplica de leitura do Amazon RDS na instância de banco de dados MySQL de origem como uma garantia adicional de que os logs binários da instância de banco de dados MySQL de origem não sejam limpos. Para obter mais informações, consulteComo criar uma réplica de leitura
9. Depois que a réplica de leitura do Amazon RDS tiver sido criada, chame o procedimento armazenado `mysql.rds_stop_replication` para interromper o processo de replicação. A instância de banco de dados MySQL de origem não limpa mais seus arquivos de log binários, portanto, eles estão disponíveis para o processo de replicação.
10. (Opcional) Configure os parâmetros `max_allowed_packet` e `slave_max_allowed_packet` com o tamanho máximo, para evitar erros de replicação. O tamanho máximo de ambos os parâmetros é de 1 GB. Recomendamos essa configuração para ambos os parâmetros. Para obter informações sobre como configurar parâmetros, consulte Modificar parâmetros em um grupo de parâmetros de banco de dados no Amazon RDS.

Copiar o banco de dados

Execute as etapas a seguir para copiar o banco de dados.

Como copiar o banco de dados

1. Conecte-se à réplica de leitura do RDS da instância de banco de dados MySQL de origem e execute a instrução `SHOW REPLICA STATUS\G` do MySQL. Anote os seguintes valores:
   - `Master_Host`
   - `Master_Port`
   - `Master_Log_File`
   - `Exec_Master_Log_Pos`

> **Note**
>
> As versões anteriores do MySQL usavam `SHOW SLAVE STATUS` em vez de `SHOW REPLICA STATUS`. Se você estiver usando uma versão do MySQL anterior à 8.0.23, use `SHOW SLAVE STATUS`.

2. Use o utilitário mysqldump para criar um snapshot, que copia os dados do Amazon RDS para seu computador cliente local. Verifique se o computador cliente tem espaço suficiente para armazenar os arquivos `mysqldump` dos bancos de dados a serem replicados. Esse processo pode levar várias horas para bancos de dados muito grandes. Siga as instruções em Creating a Data Snapshot Using mysqldump na documentação do MySQL.

   O exemplo a seguir executa `mysqldump` em um cliente e grava o despejo em um arquivo.

   Para Linux, macOS ou Unix:

   ```
   mysqldump -h source_MySQL_DB_instance_endpoint \
       -u user \
       -ppassword \
       --port=3306 \
       --single-transaction \
       --routines \
       --triggers \
       --databases  database database2 > path/rds-dump.sql
   ```

   Para Windows:

   ```
   mysqldump -h source_MySQL_DB_instance_endpoint ^
       -u user ^
       -ppassword ^
       --port=3306 ^
       --single-transaction ^
       --routines ^
       --triggers ^
       --databases  database database2 > path\rds-dump.sql
   ```

   Você pode carregar o arquivo de backup no banco de dados MySQL externo. Para obter mais informações, consulte Reloading SQL-Format Backups (Recarregar backups do formato SQL)

na documentação do MySQL. É possível executar outro utilitário para carregar os dados no banco de dados MySQL externo.

Concluir a exportação

Execute as etapas a seguir para concluir a exportação.

Como concluir a exportação

1. Use a instrução `CHANGE MASTER` do MySQL para configurar o banco de dados MySQL externo. Especifique o ID e a senha das permissões `REPLICATION SLAVE` concedidas pelo usuário. Especifique os valores `Master_Host`, `Master_Port`, `Relay_Master_Log_File` e `Exec_Master_Log_Pos` obtidos da instrução `SHOW REPLICA STATUS\G` do MySQL executada na réplica de leitura do RDS. Consulte mais informações em CHANGE MASTER TO Statement na documentação do MySQL.

   > **Note**
   >
   > As versões anteriores do MySQL usavam `SHOW SLAVE STATUS` em vez de `SHOW REPLICA STATUS`. Se você estiver usando uma versão do MySQL anterior à 8.0.23, use `SHOW SLAVE STATUS`.

2. Use o comando `START REPLICA` do MySQL para iniciar a replicação da instância de banco de dados MySQL de origem para o banco de dados MySQL externo.

   Isso inicia a replicação da instância de banco de dados MySQL de origem e exporta todas as alterações de origem que ocorreram após a interrupção da replicação da réplica de leitura do Amazon RDS.

   > **Note**
   >
   > As versões anteriores do MySQL usavam `START SLAVE` em vez de `START REPLICA`. Se você estiver usando uma versão do MySQL anterior à 8.0.23, use `START SLAVE`.

3. Execute o comando `SHOW REPLICA STATUS\G` do MySQL no banco de dados MySQL externo para verificar se ele está operando como uma réplica de leitura. Para ter mais informações sobre como interpretar os resultados, consulte SHOW SLAVE | REPLICA STATUS Statement na documentação do MySQL.

4. Após a replicação no banco de dados MySQL externo estar sincronizada com a instância de banco de dados MySQL de origem, use o comando `STOP REPLICA` do MySQL para interromper a replicação da instância de banco de dados MySQL de origem.

   > **Note**
   >
   > As versões anteriores do MySQL usavam `STOP SLAVE` em vez de `STOP REPLICA`. Se você estiver usando uma versão do MySQL anterior à 8.0.23, use `STOP SLAVE`.

5. Na réplica de leitura do Amazon RDS, chame o procedimento armazenado `mysql.rds_start_replication`. Isso permite que o Amazon RDS comece a limpar os arquivos de log binários da instância de banco de dados MySQL de origem.

Opções para instâncias de bancos de dados MySQL

A seguir, você encontrará uma descrição das opções, ou recursos adicionais, que estão disponíveis para instâncias do Amazon RDS que executam o mecanismo de banco de dados do MySQL. Para habilitar essas opções, você pode adicioná-las a um grupo de opções personalizado e, em seguida, associe o grupo de opções à sua instância de banco de dados. Para mais informações sobre como trabalhar com grupos de opções, consulte Trabalhar com grupos de opções.

O Amazon RDS oferece suporte às seguintes opções do MySQL:

Opção	ID da opção	Versões do mecanismo
Suporte ao plug-in de auditoria do MariaDB para MySQL	`MARIADB_AUDIT_PLUGIN`	Todas as versões do MySQL 8.4 MySQL 8.0.28 e versões 8.0 posteriores Todas as versões do MySQL 5.7
Suporte a memcached MySQL	`MEMCACHED`	Todas as versões do MySQL 5.7 e 8.0

Suporte ao plug-in de auditoria do MariaDB para MySQL

O Amazon RDS oferece um plug-in de auditoria para instâncias de banco de dados MySQL que se baseia no plug-in de auditoria de código aberto do MariaDB. Para obter mais informações, consulte o Plugin de auditoria para o repositório do GitHub do MySQL Server.

> **Note**
>
> O plug-in de auditoria para MySQL é baseado no plug-in de auditoria do MariaDB. Ao longo deste artigo, nos referimos a ele como plug-in de auditoria do MariaDB.

O plug-in de auditoria do MariaDB registra a atividade do banco de dados, incluindo usuários que fazem login no banco de dados e as consultas ao banco de dados que são executadas. O registro da atividade do banco de dados é armazenado em um arquivo de log.

Definições de opções do plugin de auditoria

O Amazon RDS oferece suporte para as seguintes configurações para a opção MariaDB Audit Plugin.

Configuração da opção	Valores válidos	Valor padrão	Descrição
`SERVER_AUDIT_FILE_PATH`	`/rdsdbdata/log/audit/`	`/rdsdbdata/log/audit/`	A localização dos arquivos de log. O arquivo de log contém o registro de atividades especificadas em `SERVER_AUDIT_EVENTS` . Para obter mais informações, consulte Como visualizar e listar arquivos de log do banco de dados e Arquivos de log do banco de dados MySQL.
`SERVER_AUDIT_FILE_ROTATE_SIZE`	1–1000000000	1000000	O tamanho em bytes que, quando alcançado , faz com que o arquivo rotacione. Para obter mais informações, consulte Visão geral dos logs de banco de dados do RDS para MySQL.
`SERVER_AUDIT_FILE_`	0–100	9	O número de rotações de log para salvar quando `server_audit_output_type=fi`

Configuração da opção	Valores válidos	Valor padrão	Descrição
ROTATIONS			le . Se definido como 0, o arquivo de log nunca é alternado. Para ter mais informaçõ es, consulte Visão geral dos logs de banco de dados do RDS para MySQL e Como baixar um arquivo de log de banco de dados.
SERVER_AUDIT_EVENTS	CONNECT, QUERY, QUERY_DDL , QUERY_DML , QUERY_DML_NO_SELECT , QUERY_DCL	CONNECT, QUERY	Os tipos de atividades a serem gravados no log. A instalação do MariaDB Audit Plugin é registrada em log. • CONNECT: registrar conexões bem-suced idas e sem êxito com o banco de dados e desconexões do banco de dados. • QUERY: registrar o texto de todas as consultas executadas no banco de dados. • QUERY_DDL : semelhante ao evento de QUERY, mas retorna somente consultas de linguagem de definição de dados (DDL) (CREATE, ALTER, etc.) • QUERY_DML : semelhante ao evento de QUERY, mas retorna somente consultas de linguagem de manipulação de dados (DML) (INSERT, UPDATE, etc., bem como SELECT). • QUERY_DML_NO_SELECT : semelhante ao evento QUERY_DML , mas não registra consultas de log SELECT. • QUERY_DCL : semelhante ao evento de QUERY, mas retorna somente consultas de linguagem de controle de dados (DCL) (GRANT, REVOKE, etc.) Para MySQL, TABLE não é compatível.

Configuração da opção	Valores válidos	Valor padrão	Descrição
SERVER_AUDIT_INCL_USERS	Vários valores separados por vírgulas	Nenhum	Inclua apenas atividades dos usuários especificados. Por padrão, a atividade é registrada para todos os usuários. SERVER_AUDIT_INCL_USERS e SERVER_AUDIT_EXCL_USERS são mutuamente exclusivos. Se você adicionar valores ao SERVER_AUDIT_INCL_USERS , certifique-se de que nenhum valor seja adicionado ao SERVER_AUDIT_EXCL_USERS .

Configuração da opção	Valores válidos	Valor padrão	Descrição
SERVER_AUDIT_EXCL_USERS	Vários valores separados por vírgulas	Nenhum	Exclua a atividade dos usuários especificados. Por padrão, a atividade é registrada para todos os usuários. SERVER_AUDIT_INCL_USERS e SERVER_AUDIT_EXCL_USERS são mutuamente exclusivos. Se você adicionar valores ao SERVER_AUDIT_EXCL_USERS , certifique-se de que nenhum valor seja adicionado ao SERVER_AUDIT_INCL_USERS . O usuário rdsadmin consulta o banco de dados a cada segundo para verificar a integridade do banco de dados. Dependendo das suas outras configurações, essa atividade pode fazer com que o tamanho do seu arquivo de log cresça muito rapidamente. Se você não precisa registrar essa atividade, adicione o usuário rdsadmin à lista SERVER_AUDIT_EXCL_USERS . **Note** CONNECTA atividade é sempre registrada para todos os usuários, mesmo se o usuário é especificado para essa configuração de opção.

Configuração da opção	Valores válidos	Valor padrão	Descrição
SERVER_AUDIT_LOGGING	ON	ON	O registro em log está ativo. O único valor válido é ON. O Amazon RDS não oferece suporte à desativação do registro em log. Se quiser desativar o registro log, remova o MariaDB Audit Plugin. Para obter mais informações, consulte Remover o MariaDB Audit Plugin.
SERVER_AUDIT_QUERY_LOG_LIMIT	0–2147483647	1024	O limite do tamanho da string de consulta em um registro.

Adicionar o MariaDB Audit Plugin

O processo geral para adicionar o MariaDB Audit Plugin a uma instância de banco de dados é o seguinte:

- Crie um novo grupo de opções, ou copie ou modifique um existente
- Adicione opções ao grupo de opções
- Associe o grupo de opções à instância de banco de dados

Depois de adicionar o MariaDB Audit Plugin, você não precisará reiniciar sua instância de banco de dados. Assim que o grupo de opções estiver ativo, a auditoria começará imediatamente.

> **Important**
>
> Adicionar o plugin de auditoria do MariaDB a uma instância de banco de dados ode causar uma interrupção. Recomendamos adicionar o plugin de auditoria do MariaDB durante uma janela de manutenção ou durante um horário de baixa workload de banco de dados.

Para adicionar o MariaDB Audit Plugin

1. Determine o grupo de opções que você deseja usar. Você pode criar um novo grupo de opções ou usar um existente. Se você quiser usar um grupo de opções existente, vá para a próxima etapa. Caso contrário, crie um grupo de opções de banco de dados personalizado. Selecione mysql em Mecanismo e escolha 5.7, 8.0 ou 8.4 em Versão principal do mecanismo. Para ter mais informações, consulte Criar um grupo de opções.
2. Adicione a opção MARIADB_AUDIT_PLUGIN ao grupo de opções e defina as configurações da opção. Para ter mais informações sobre a adição de opções, consulte Adicionar uma opção a um grupo de opções. Para ter mais informações sobre cada configuração, consulte Definições de opções do plugin de auditoria.
3. Aplique o grupo de opções a uma instância de banco de dados nova ou existente.

   - Para uma nova instância de banco de dados, você aplica o grupo de opções ao executar a instância. Para ter mais informações, consulte Criar uma instância de banco de dados do Amazon RDS.
   - Para uma instância de banco de dados existente, você aplica o grupo de opções modificando a instância e anexando o novo grupo de opções. Para obter mais informações, consulteModificar uma instância de banco de dados do Amazon RDS

Formato do log de auditoria

Os arquivos de log são representados como arquivos de variáveis separadas por vírgula (CSV) no formato UTF-8.

> **Tip**
>
> As entradas do log não estão em ordem sequencial. Para ordenar as entradas, use o valor do carimbo de data/hora. Para ver os eventos mais recentes, talvez seja preciso analisar todos os arquivos de log. Para obter mais flexibilidade na classificação e pesquisa dos dados de log, ative a configuração para carregar os logs de auditoria no CloudWatch e visualizá-los usando a interface do CloudWatch.
>
> Para visualizar dados de auditoria com mais tipos de campos e com saída no formato JSON, você também pode usar o recurso Database Activity Streams. Para obter mais informações, consulteMonitorar o Amazon RDS com o recurso Database Activity Streams

Os arquivos de log de auditoria incluem as seguintes informações delimitadas por vírgulas nas linhas, na ordem especificada:

Campo	Descrição
timestamp	O `YYYYMMDD` seguido pelo `HH:MI:SS` (24 horas) para o evento registrado em log.
serverhost	O nome da instância em que o evento foi registrado.
username	O nome de usuário conectado.
host	O host no qual o usuário está conectado.
connectionid	O número de ID da conexão para a operação registrada.
queryid	O número do ID de consulta, que pode ser usado para encontrar os eventos relacionais da tabela e consultas relacionadas. Para eventos `TABLE`, são adicionadas várias linhas.
operação	O tipo de ação gravado. Os valores possíveis são: `CONNECT`, `QUERY`, `READ`, `WRITE`, `CREATE`, `ALTER`, `RENAME` e `DROP`.
banco de dados	O banco de dados ativo, conforme definido pelo comando `USE`.
objeto	Para eventos de `QUERY`, esse valor indica a consulta executada pelo banco de dados. Para eventos de `TABLE`, indica o nome da tabela.
retcode	O código de retorno da operação registrada.
connection_type	O estado de segurança da conexão com o servidor. Os valores possíveis são: • `0` – Não definido • `1` – TCP/IP • `2` – Soquete • `3` – Pipe nomeado • `4` – SSL/TLS • `5` – Memória compartilhada

Visualizar e baixar o log do plugin de auditoria do MariaDB

Depois de habilitar o MariaDB Audit Plugin, você acessará os resultados nos arquivos de log da mesma forma que você acessa outros arquivos de log baseados em texto. Os arquivos de log de auditoria estão localizados em `/rdsdbdata/log/audit/`. Para obter informações sobre como visualizar o arquivo de log no console, consulte Como visualizar e listar arquivos de log do banco de dados. Para obter informações sobre como baixar o arquivo de log, consulte Como baixar um arquivo de log de banco de dados.

Modificar as configurações do plugin de auditoria do MariaDB

Depois de habilitar o MariaDB Audit Plugin, você pode modificar as configurações. Para mais informações sobre como modificar as configurações da opção, consulte Modificar uma configuração de opção. Para ter mais informações sobre cada configuração, consulte Definições de opções do plugin de auditoria.

Remover o MariaDB Audit Plugin

O Amazon RDS não oferece suporte à desativação do registro em log no MariaDB Audit Plugin. No entanto, você pode remover o plugin de uma instância de banco de dados. Quando você remove o MariaDB Audit Plugin, a instância de banco de dados é reiniciada automaticamente para interromper a auditoria.

Para remover o MariaDB Audit Plugin de uma instância de banco de dados, siga um destes procedimentos:

- Remova a opção MariaDB Audit Plugin do grupo de opções ao qual ela pertence. Essa alteração afeta todas as instâncias de bancos de dados que usam o grupo de opções. Para ter mais informações, consulte Remover uma opção de um grupo de opções
- Modifique a instância de banco de dados e especifique um grupo de opções diferente que não inclua o plugin. Essa alteração afeta uma única instância de banco de dados. Você pode especificar um grupo de opções padrão (vazio) ou criar um grupo de opções personalizado diferente. Para ter mais informações, consulte Modificar uma instância de banco de dados do Amazon RDS.

Suporte a memcached MySQL

O Amazon RDS oferece suporte ao uso da interface `memcached` para tabelas do InnoDB introduzido no MySQL 5.6. A API `memcached` permite que os aplicativos usem as tabelas do InnoDB de forma semelhante aos armazenamentos de dados de chave/valor NoSQL.

> **Note**
>
> A interface do memcached não está mais disponível no MySQL 8.4. Ao atualizar suas instâncias de banco de dados para o MySQL 8.4, você deve desativar `memcached` nos grupos de opções existentes.

A interface `memcached` é um cache simples baseado em chave. Os aplicativos usam `memcached` para inserir, manipular e recuperar pares de dados de valor/chave do cache. O MySQL 5.6 introduziu um plugin que implementa um serviço de daemon que expõe os dados das tabelas do InnoDB através do protocolo `memcached`. Para ter mais informações sobre o plug-in `memcached` MySQL, consulte Integração do InnoDB com memcached.

Como habilitar o suporte a memcached para uma instância de banco de dados do RDS para MySQL

1. Determine o grupo de segurança a ser usado para controlar o acesso à interface de `memcached`. Se o conjunto de aplicações que já estão usando a interface SQL for o mesmo conjunto que acessará a interface `memcached`, você poderá usar o grupo de segurança de VPC existente usado pela interface SQL. Se um conjunto diferente de aplicativos acessar a interface `memcached`, defina um novo grupo de segurança de VPC ou grupo de segurança de banco de dados. Para ter mais informações sobre como gerenciar grupo de seguranças, consulte Controlar acesso com grupos de segurança
2. Crie um grupo de opções de banco de dados personalizado, selecionando MySQL como o tipo e a versão do mecanismo. Para ter mais informações sobre como criar um grupo de opções, consulte Criar um grupo de opções.
3. Adicione a opção `MEMCACHED` ao grupo de opções. Especifique a porta que a interface `memcached` usará e o grupo de segurança a ser usado para controlar o acesso à interface. Para mais informações sobre a adição de opções, consulte Adicionar uma opção a um grupo de opções.

4. Modifique as configurações de opções para configurar os parâmetros `memcached`, se necessário. Para mais informações sobre como modificar as configurações da opção, consulte Modificar uma configuração de opção.
5. Aplique o grupo de opções a uma instância. O Amazon RDS habilita o suporte ao `memcached` para essa instância quando o grupo de opções for aplicado:
   - Você habilita o suporte de `memcached` para uma nova instância especificando o grupo de opções personalizado ao executar a instância. Para ter mais informações sobre como executar uma instância MySQL, consulte Criar uma instância de banco de dados do Amazon RDS.
   - Você habilita o suporte de `memcached` para uma instância existente especificando o grupo de opções personalizado ao modificar a instância. Para ter mais informações sobre a modificação de uma instância de banco de dados , consulte Modificar uma instância de banco de dados do Amazon RDS.
6. Especifique quais colunas em suas tabelas MySQL podem ser acessadas por meio da interface de `memcached`. O plug-in `memcached` cria uma tabela de catálogo chamada `containers` em um banco de dados dedicado chamado `innodb_memcache`. Você insere uma linha na tabela `containers` para mapear uma tabela do InnoDB para acesso via `memcached`. Você especifica uma coluna na tabela do InnoDB que é usada para armazenar os valores das chaves do `memcached` e uma ou mais colunas que são usadas para armazenar os valores de dados associados à chave. Você também especifica um nome que um aplicativo `memcached` usa para se referir a esse conjunto de colunas. Para obter detalhes sobre como inserir linhas na tabela `containers`, consulte Elementos internos do plug-in memcached do InnoDB. Para obter um exemplo de mapeamento de uma tabela do InnoDB e acessá-la por meio de `memcached`, consulte Writing applications for the InnoDB memcached plugin.
7. Se as aplicações que acessarem a interface `memcached` estiverem em computadores ou instâncias do EC2 diferentes das aplicações que usam a interface SQL, adicione as informações de conexão desses computadores ao grupo de segurança de VPC associado à instância MySQL. Para ter mais informações sobre como gerenciar grupo de seguranças, consulte Controlar acesso com grupos de segurança.

Você desliga o suporte `memcached` para uma instância modificando essa instância e especificando o grupo de opções padrão para sua versão do MySQL. Para ter mais informações sobre a modificação de uma instância de banco de dados , consulte Modificar uma instância de banco de dados do Amazon RDS.

Considerações sobre segurança do MySQL memcached

O protocolo `memcached` não oferece suporte para a autenticação do usuário. Para ter mais informações sobre as considerações de segurança do MySQL `memcached`, consulte Considerações de segurança para o plug-in memcached do InnoDB na documentação do MySQL.

Você pode realizar as seguintes ações para ajudar a aumentar a segurança da interface `memcached`:

- Especifique uma porta diferente do padrão de 11211 ao adicionar a opção `MEMCACHED` ao grupo de opções.
- Certifique-se de associar a interface `memcached` com um grupo de segurança de VPC que limite o acesso a endereços de clientes conhecidos ou confiáveis e a instâncias do EC2. Para ter mais informações sobre como gerenciar grupo de seguranças, consulte Controlar acesso com grupos de segurança.

Informações sobre conexão com MySQL memcached

Para acessar a interface `memcached`, um aplicativo deve especificar o nome DNS da instância do Amazon RDS e o número da porta `memcached`. Por exemplo, se uma instância tiver um nome DNS `my-cache-instance.cg034hpkmmjt.region.rds.amazonaws.com` e a interface memcached estiver usando a porta 11212, as informações de conexão especificadas no PHP serão:

```
<?php

$cache = new Memcache;
$cache->connect('my-cache-instance.cg034hpkmmjt.region.rds.amazonaws.com',11212);
?>
```

Para encontrar o nome DNS e a porta memcached de uma instância do MySQL

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No canto superior direito do AWS Management Console, selecione a região que contém a instância de banco de dados.
3. No painel de navegação, escolha Databases (Bancos de dados).
4. Escolha o nome da instância de banco de dados MySQL para exibir os detalhes.

5. Na seção Connect (Conectar), anote o valor do campo Endpoint. O nome do DNS é o mesmo que o do endpoint. Além disso, observe que a porta na seção Connect (Conectar) não é usada para acessar a interface memcached.
6. Na seção Details (Detalhes), anote o nome listado no campo Option Group (Grupo de opções).
7. No painel de navegação, escolha Option groups (Grupos de opções).
8. Escolha o nome do grupo de opções usado pela instância de banco de dados MySQL para mostrar os detalhes do grupo de opções. Na seção Options (Opções), anote o valor da configuração de Port (Porta) na opção MEMCACHED.

Configurações de opções de MySQL memcached

O Amazon RDS expõe os parâmetros memcached do MySQL como configurações de opções na opção MEMCACHED do Amazon RDS.

Parâmetros de MySQL memcached

- DAEMON_MEMCACHED_R_BATCH_SIZE – um inteiro que especifica quantas operações de leitura memcached devem ser realizadas antes de fazer um COMMIT para iniciar uma nova transação. Os valores permitidos são de 1 a 4294967295; o padrão é 1. A opção não terá efeito até que a instância seja reiniciada.
- DAEMON_MEMCACHED_W_BATCH_SIZE – um inteiro que especifica quantas operações de gravação memcached, como adicionar, definir ou aumentar, devem ser realizadas antes de fazer um COMMIT para iniciar uma nova transação. Os valores permitidos são de 1 a 4294967295; o padrão é 1. A opção não terá efeito até que a instância seja reiniciada.
- INNODB_API_BK_COMMIT_INTERVAL – um inteiro que especifica com que frequência as conexões inativas de confirmação automática usam a interface memcached do InnoDB. Os valores permitidos são de 1 a 1073741824; o padrão é 5. A opção entra em vigor imediatamente, sem exigir que você reinicie a instância.
- INNODB_API_DISABLE_ROWLOCK – um booliano que desabilita (1 (true)) ou habilita (0 (false)) o uso de bloqueios de linha ao usar a interface memcached do InnoDB. O padrão é 0 (false). A opção não terá efeito até que a instância seja reiniciada.
- INNODB_API_ENABLE_MDL – um booliano que, quando definido como 0 (false), bloqueia a tabela usada pelo plug-in memcached do InnoDB, de modo que ela não possa ser descartada ou alterado pela DDL por meio da interface SQL. O padrão é 0 (false). A opção não terá efeito até que a instância seja reiniciada.

- INNODB_API_TRX_LEVEL – um número inteiro que especifica o nível de isolamento da transação para consultas processadas pela interface memcached. Os valores permitidos são de 0 a 3. O padrão é 0. A opção não terá efeito até que a instância seja reiniciada.

O Amazon RDS configura esses parâmetros memcached MySQL, e eles não podem ser modificados: DAEMON_MEMCACHED_LIB_NAME, DAEMON_MEMCACHED_LIB_PATH e INNODB_API_ENABLE_BINLOG. Os parâmetros que os administradores MySQL definem usando daemon_memcached_options estão disponíveis como configurações da opção MEMCACHED individuais no Amazon RDS.

Parâmetros daemon_memcached_options MySQL

- BINDING_PROTOCOL – uma string que especifica o protocolo de associação a ser usado. Os valores permitidos são auto, ascii ou binary. O padrão é auto, o que significa que o servidor negocia automaticamente o protocolo com o cliente. A opção não terá efeito até que a instância seja reiniciada.
- BACKLOG_QUEUE_LIMIT – um número inteiro que especifica quantas conexões de rede podem estar aguardando para serem processadas por memcached. Aumentar esse limite pode reduzir os erros recebidos por um cliente que não seja capaz de se conectar à instância memcached, mas não melhora a performance do servidor. Os valores permitidos são de 1 a 2048; o padrão é 1024. A opção não terá efeito até que a instância seja reiniciada.
- CAS_DISABLED – um booliano que permite (1 (true)) ou desabilite (0 (false)) o uso do recurso de comparação e permuta (CAS), o que reduz o tamanho por item em 8 bytes. O padrão é 0 (false). A opção não terá efeito até que a instância seja reiniciada.
- CHUNK_SIZE – um número inteiro que especifica o tamanho mínimo do bloco, em bytes, para alocar a chave, o valor e os sinalizadores do menor item. Os valores permitidos são de 1 a 48. O padrão é 48 e você pode melhorar significativamente a eficiência da memória com um valor menor. A opção não terá efeito até que a instância seja reiniciada.
- CHUNK_SIZE_GROWTH_FACTOR – um flutuante que controla o tamanho de novos blocos. O tamanho de um novo bloco é o tamanho do bloco anterior vezes CHUNK_SIZE_GROWTH_FACTOR. Os valores permitidos são de 1 a 2; o padrão é 1.25. A opção não terá efeito até que a instância seja reiniciada.
- ERROR_ON_MEMORY_EXHAUSTED – um booliano, quando definido como 1 (true), especifica que memcached retornará um erro ao invés de despejar itens quando não houver mais memória para armazenar itens. Se definido como 0 (false), memcached expulsará itens se não houver mais memória. O padrão é 0 (false). A opção não terá efeito até que a instância seja reiniciada.

- MAX_SIMULTANEOUS_CONNECTIONS – um inteiro que especifica o número máximo de conexões simultâneas. Definir esse valor como qualquer coisa abaixo de 10 impede que o MySQL seja iniciado. Os valores permitidos são de 10 a 1024; o padrão é 1024. A opção não terá efeito até que a instância seja reiniciada.
- VERBOSITY – uma string que especifica o nível de informações registradas no log de erros do MySQL pelo serviço memcached. O padrão é v. A opção não terá efeito até que a instância seja reiniciada. Os valores permitidos são:
  - v – registra erros e avisos durante a execução do loop de eventos principal.
  - vv – Além das informações registradas por v, também registra cada comando do cliente e a resposta.
  - vvv – Além das informações registradas por vv, também registra transições de estado internas.

O Amazon RDS configura esses parâmetros DAEMON_MEMCACHED_OPTIONS MySQL, eles não podem ser modificados: DAEMON_PROCESS, LARGE_MEMORY_PAGES, MAXIMUM_CORE_FILE_LIMIT, MAX_ITEM_SIZE, LOCK_DOWN_PAGE_MEMORY, MASK, IDFILE, REQUESTS_PER_EVENT, SOCKET e USER.

Parâmetros do MySQL

Por padrão, uma instância de banco de dados MySQL usa um grupo de parâmetros de banco de dados específico para um banco de dados MySQL. Esse grupo de parâmetros contém parâmetros para o mecanismo de banco de dados MySQL. Para obter informações sobre como trabalhar com grupos de parâmetros e definir parâmetros, consulte Grupos de parâmetros para Amazon RDS.

Os parâmetros do RDS for MySQL são definidos com os valores padrão do mecanismo de armazenamento que você selecionou. Para obter mais informações sobre os parâmetros do MySQL, consulte a documentação do MySQL. Para obter mais informações sobre os mecanismos de armazenamento do MySQL, consulte Suporte aos mecanismos de armazenamento no RDS para MySQL.

Você pode exibir os parâmetros disponíveis para uma versão específica do RDS for MySQL usando o console do RDS ou a AWS CLI. Para obter informações sobre como visualizar os parâmetros em um grupo de parâmetros do MySQL no console do RDS, consulte Visualizar valores de parâmetros para um grupo de parâmetros de banco de dados no Amazon RDS.

Usando a AWS CLI, você pode visualizar os parâmetros de uma versão do RDS for MySQL executando o comando `describe-engine-default-parameters`. Especifique um destes valores a seguir para a opção `--db-parameter-group-family`:

- `mysql8.4`
- `mysql8.0`
- `mysql5.7`

Por exemplo, para visualizar parâmetros do RDS for MySQL, versão 8.0, execute o comando a seguir.

```
aws rds describe-engine-default-parameters --db-parameter-group-family mysql8.0
```

O resultado será semelhante ao mostrado a seguir.

```
{
    "EngineDefaults": {
        "Parameters": [
            {
                "ParameterName": "activate_all_roles_on_login",
                "ParameterValue": "0",
```

```
                "Description": "Automatically set all granted roles as active after the
 user has authenticated successfully.",
                "Source": "engine-default",
                "ApplyType": "dynamic",
                "DataType": "boolean",
                "AllowedValues": "0,1",
                "IsModifiable": true
            },
            {
                "ParameterName": "allow-suspicious-udfs",
                "Description": "Controls whether user-defined functions that have only
 an xxx symbol for the main function can be loaded",
                "Source": "engine-default",
                "ApplyType": "static",
                "DataType": "boolean",
                "AllowedValues": "0,1",
                "IsModifiable": false
            },
            {
                "ParameterName": "auto_generate_certs",
                "Description": "Controls whether the server autogenerates SSL key and
 certificate files in the data directory, if they do not already exist.",
                "Source": "engine-default",
                "ApplyType": "static",
                "DataType": "boolean",
                "AllowedValues": "0,1",
                "IsModifiable": false
            },
        ...
```

Para listar somente os parâmetros modificáveis do RDS for MySQL versão 8.0, execute o comando a seguir.

Para Linux, macOS ou Unix:

```
aws rds describe-engine-default-parameters --db-parameter-group-family mysql8.0 \
   --query 'EngineDefaults.Parameters[?IsModifiable==`true`]'
```

Para Windows:

```
aws rds describe-engine-default-parameters --db-parameter-group-family mysql8.0 ^
   --query "EngineDefaults.Parameters[?IsModifiable==`true`]"
```

Tarefas comuns de DBA para instâncias de banco de dados MySQL

No conteúdo a seguir, é possível encontrar descrições das implementações específicas do Amazon RDS de algumas tarefas de DBA comuns para instâncias de banco de dados que executam o mecanismo de banco de dados MySQL. Para oferecer uma experiência de serviço gerenciada, o Amazon RDS não fornece acesso ao shell para as instâncias de banco de dados. Ele também restringe o acesso a determinados procedimentos e tabelas do sistema que exigem privilégios avançados.

Para obter informações sobre como trabalhar com arquivos de log do MySQL no Amazon RDS, consult Arquivos de log do banco de dados MySQL.

Noções básicas sobre usuários predefinidos

O Amazon RDS cria automaticamente vários usuários predefinidos com novas instâncias de banco de dados do RDS para MySQL. Os usuários predefinidos e seus privilégios não podem ser alterados. Não é possível descartar, renomear ou modificar os privilégios desses usuários predefinidos. Qualquer tentativa de fazer isso gerará um erro.

- rdsadmin: um usuário criado para lidar com muitas das tarefas de gerenciamento que o administrador com privilégios de `superuser` executaria em um banco de dados MySQL autônomo. Esse usuário é usado internamente pelo RDS para MySQL para várias tarefas de gerenciamento.
- rdsrepladmin: um usuário usado internamente pelo Amazon RDS para oferecer suporte a atividades de replicação no RDS para instâncias e clusters de banco de dados do MySQL.

Consulte informações sobre outras tarefas de comuns de DBA nos tópicos a seguir.

Tópicos

- Modelo de privilégios baseados em perfis para o RDS para MySQL
- Privilégios dinâmicos para o RDS para MySQL
- Encerrar uma sessão ou consulta do RDS para MySQL
- Ignorar o erro de replicação atual para o RDS para MySQL
- Trabalhar com espaços de tabela do InnoDB para melhorar o tempo de recuperação de falhas do RDS para MySQL

- Gerenciar o histórico de status global do RDS para MySQL
- Configuração do tamanho do pool de buffer e a capacidade de log redo no MySQL 8.4

Modelo de privilégios baseados em perfis para o RDS para MySQL

Iniciando com o RDS para MySQL versão 8.0.36, não é possível modificar as tabelas no banco de dados `mysql` diretamente. Especificamente, não é possível criar usuários de banco de dados executando operações de linguagem de manipulação de dados (DML) nas tabelas `grant`. Em vez disso, use declarações de gerenciamento de contas do MySQL, como `CREATE USER`, `GRANT` e `REVOKE`, para conceder privilégios baseados em perfil aos usuários. Você também não pode criar outros tipos de objeto, como procedimentos armazenados no banco de dados do `mysql`. Você ainda pode consultar as tabelas `mysql`. Se você utilizar a replicação de logs binários, as alterações feitas diretamente nas tabelas `mysql` na instância de banco de dados de origem não serão replicadas no cluster de destino.

Em alguns casos, sua aplicação pode utilizar atalhos para criar usuários ou outros objetos inserindo-os nas tabelas `mysql`. Em caso afirmativo, altere o código da aplicação para utilizar as instruções correspondentes, como `CREATE USER`.

Para exportar metadados para usuários de banco de dados durante a migração de um banco de dados MySQL externo, use um dos seguintes comandos:

- Use o utilitário de despejo de instância do MySQL Shell com um filtro para excluir usuários, perfis e concessões. O exemplo a seguir mostra a sintaxe de comando a ser usada. Verifique se `outputUrl` está vazio.

  ```
  mysqlsh user@host -- util.dumpInstance(outputUrl,{excludeSchemas:['mysql'],users:
   true})
  ```

  Para ter mais informações, consulte Instance Dump Utility, Schema Dump Utility, and Table Dump Utility no MySQL Reference Manual.

- Use o utilitário de cliente `mysqlpump`. Este exemplo inclui todas as tabelas, exceto aquelas no banco de dados do sistema `mysql`. Ela também inclui as instruções `CREATE USER` e `GRANT` para reproduzir todos os usuários do MySQL no banco de dados migrado.

  ```
  mysqlpump --exclude-databases=mysql --users
  ```

O utilitário cliente `mysqlpump` não está mais disponível no MySQL 8.4. Em seu lugar, use `mysqldump`.

Para simplificar o gerenciamento de permissões para vários usuários ou aplicações, é possível utilizar a instrução `CREATE ROLE` para criar uma função que tenha um conjunto de permissões. Em seguida, você pode utilizar as instruções `GRANT` e `SET ROLE` e a função `current_role` para atribuir funções a usuários ou aplicações, alternar a função atual e verificar quais funções estão em vigor. Para obter mais informações sobre o sistema de permissões baseadas em funções no MySQL 8.0, consulte o tópico sobre como Usar funções, no Guia de referência do MySQL.

Important

É altamente recomendável não usar o usuário mestre diretamente nas aplicações. Em vez disso, siga as práticas recomendadas de usar um usuário do banco de dados criado com os privilégios mínimos obrigatórios para a aplicação.

Iniciando com a versão 8.0.36, o RDS para MySQL inclui um perfil especial que tem todos os privilégios a seguir. Essa função se chama `rds_superuser_role`. O usuário administrativo principal de cada instância de banco de dados já tem esse perfil concedido. A função `rds_superuser_role` inclui os seguintes privilégios para todos os objetos de banco de dados:

- `ALTER`
- `APPLICATION_PASSWORD_ADMIN`
- `ALTER ROUTINE`
- `CREATE`
- `CREATE ROLE`
- `CREATE ROUTINE`
- `CREATE TEMPORARY TABLES`
- `CREATE USER`
- `CREATE VIEW`
- `DELETE`
- `DROP`

- DROP ROLE
- EVENT
- EXECUTE
- INDEX
- INSERT
- LOCK TABLES
- PROCESS
- REFERENCES
- RELOAD
- REPLICATION CLIENT
- REPLICATION SLAVE
- ROLE_ADMIN
- SET_USER_ID
- SELECT
- SHOW DATABASES
- SHOW VIEW
- TRIGGER
- UPDATE
- XA_RECOVER_ADMIN

A definição da função também inclui `WITH GRANT OPTION`, permitindo que um usuário administrativo a conceda para outros usuários. Especificamente, o administrador deve conceder quaisquer privilégios necessários para executar a replicação de logs binários com o cluster do MySQL como destino.

Tip

Para ver os detalhes completos das permissões, use a declaração a seguir.

```
SHOW GRANTS FOR rds_superuser_role@'%';
```

Ao conceder acesso utilizando perfis no RDS para MySQL versão 8.0.36 e posterior, você também ativa o perfil utilizando a declaração `SET ROLE role_name` ou `SET ROLE ALL`. O exemplo a seguir mostra como. Substitua o nome da função apropriado para `CUSTOM_ROLE`.

```
# Grant role to user
mysql> GRANT CUSTOM_ROLE TO 'user'@'domain-or-ip-address'

# Check the current roles for your user. In this case, the CUSTOM_ROLE role has not
 been activated.
# Only the rds_superuser_role is currently in effect.
mysql> SELECT CURRENT_ROLE();
+--------------------------+
| CURRENT_ROLE()           |
+--------------------------+
| `rds_superuser_role`@`%` |
+--------------------------+
1 row in set (0.00 sec)

# Activate all roles associated with this user using SET ROLE.
# You can activate specific roles or all roles.
# In this case, the user only has 2 roles, so we specify ALL.
mysql> SET ROLE ALL;
Query OK, 0 rows affected (0.00 sec)

# Verify role is now active
mysql> SELECT CURRENT_ROLE();
+-------------------------------------------------+
| CURRENT_ROLE()                                  |
+-------------------------------------------------+
| `CUSTOM_ROLE`@`%`,`rds_superuser_role`@`%` |
+-------------------------------------------------+
```

Privilégios dinâmicos para o RDS para MySQL

Privilégios dinâmicos são privilégios do MySQL que você pode conceder explicitamente usando a declaração `GRANT`. Dependendo da sua versão do RDS para MySQL, o RDS permite que você conceda somente privilégios dinâmicos específicos. O RDS proíbe alguns desses privilégios porque eles podem interferir nas operações específicas do banco de dados, como replicação e backup.

A tabela a seguir mostra quais desses privilégios é possível conceder para diferentes versões do MySQL. Se você estiver atualizando de uma versão do MySQL inferior à 8.0.36 para a versão 8.0.36

ou posterior, talvez seja necessário atualizar o código da aplicação se a concessão de um privilégio específico não for mais permitida.

Privilege	MySQL 8.0.35 e anterior	MySQL 8.0.36 e versões secundárias posteriores	MySQL 8.4.3 e posterior
ALLOW_NONEXISTENT_DEFINER	Não disponível	Não disponível	Não permitido
APPLICATION_PASSWORD_ADMIN	Permitido	Permitido	Permitido
AUDIT_ABORT_EXEMPT	Permitido	Não permitido	Não permitido
AUDIT_ADMIN	Não permitido	Não permitido	Não permitido
AUTHENTICATION_POLICY_ADMIN	Permitido	Não permitido	Não permitido
BACKUP_ADMIN	Permitido	Não permitido	Não permitido
BINLOG_ADMIN	Permitido	Não permitido	Não permitido
BINLOG_ENCRYPTION_ADMIN	Não permitido	Não permitido	Não permitido
CLONE_ADMIN	Não permitido	Não permitido	Não permitido
CONNECTION_ADMIN	Permitido	Não permitido	Não permitido
ENCRYPTION_KEY_ADMIN	Não permitido	Não permitido	Não permitido
FIREWALL_ADMIN	Não permitido	Não permitido	Não permitido

Privilege	MySQL 8.0.35 e anterior	MySQL 8.0.36 e versões secundárias posteriores	MySQL 8.4.3 e posterior
FIREWALL_EXEMPT	Permitido	Não permitido	Não permitido
FIREWALL_USER	Não permitido	Não permitido	Não permitido
FLUSH_OPTIMIZER_COSTS	Permitido	Permitido	Permitido
FLUSH_PRIVILEGES	Não disponível	Não disponível	Permitido
FLUSH_STATUS	Permitido	Permitido	Permitido
FLUSH_TABLES	Permitido	Permitido	Permitido
FLUSH_USER_RESOURCES	Permitido	Permitido	Permitido
GROUP_REPLICATION_ADMIN	Não permitido	Não permitido	Não permitido
GROUP_REPLICATION_STREAM	Não permitido	Não permitido	Não permitido
INNODB_REDO_LOG_ARCHIVE	Não permitido	Não permitido	Não permitido
INNODB_REDO_LOG_ENABLE	Não permitido	Não permitido	Não permitido
MASKING_DICTIONARIES_ADMIN	Não permitido	Não permitido	Não permitido
NDB_STORED_USER	Não permitido	Não permitido	Não permitido

Privilege	MySQL 8.0.35 e anterior	MySQL 8.0.36 e versões secundárias posteriores	MySQL 8.4.3 e posterior
OPTIMIZE_LOCAL_TABLE	Não disponível	Não disponível	Não permitido
PASSWORDLESS_USER_ADMIN	Não permitido	Não permitido	Não permitido
PERSIST_RO_VARIABLES_ADMIN	Não permitido	Não permitido	Não permitido
REPLICATION_APPLIER	Permitido	Não permitido	Não permitido
REPLICATION_SLAVE_ADMIN	Não permitido	Não permitido	Não permitido
RESOURCE_GROUP_ADMIN	Permitido	Não permitido	Não permitido
RESOURCE_GROUP_USER	Permitido	Não permitido	Não permitido
ROLE_ADMIN	Permitido	Permitido	Permitido
SENSITIVE_VARIABLES_OBSERVER	Permitido	Permitido	Permitido
SERVICE_CONNECTION_ADMIN	Permitido	Não permitido	Não permitido
SESSION_VARIABLES_ADMIN	Permitido	Permitido	Permitido
SET_ANY_DEFINER	Não disponível	Não disponível	Permitido

Privilege	MySQL 8.0.35 e anterior	MySQL 8.0.36 e versões secundárias posteriores	MySQL 8.4.3 e posterior
SET_USER_ID	Permitido	Permitido	Indisponível
SHOW_ROUTINE	Permitido	Permitido	Permitido
SKIP_QUERY_REWRITE	Não permitido	Não permitido	Não permitido
SYSTEM_USER	Não permitido	Não permitido	Não permitido
SYSTEM_VARIABLES_ADMIN	Não permitido	Não permitido	Não permitido
TABLE_ENCRYPTION_ADMIN	Não permitido	Não permitido	Não permitido
TELEMETRY_LOG_ADMIN	Permitido	Não permitido	Não permitido
TP_CONNECTION_ADMIN	Não permitido	Não permitido	Não permitido
TRANSACTION_GTID_TAG	Não disponível	Não disponível	Não permitido
VERSION_TOKEN_ADMIN	Não permitido	Não permitido	Não permitido
XA_RECOVER_ADMIN	Permitido	Permitido	Permitido

Encerrar uma sessão ou consulta do RDS para MySQL

É possível encerrar sessões ou consultas de usuários em instâncias de banco de dados usando os comandos `rds_kill` e `rds_kill_query`. Primeiro, conecte-se à sua instância de banco de dados

MySQL. Depois disso, emita o comando apropriado, conforme mostrado a seguir. Para obter mais informações, consulte Conectar-se à instância de banco de dados do MySQL.

```
CALL mysql.rds_kill(thread-ID)
CALL mysql.rds_kill_query(thread-ID)
```

Por exemplo, para encerrar a sessão que está sendo executada no thread 99, digite o seguinte:

```
CALL mysql.rds_kill(99);
```

Para encerrar a consulta que está sendo executada no thread 99, digite o seguinte:

```
CALL mysql.rds_kill_query(99);
```

Ignorar o erro de replicação atual para o RDS para MySQL

Você pode ignorar um erro na réplica de leitura se ele estiver fazendo com que a réplica de leitura pare de responder e não afetar a integridade dos dados.

> **Note**
>
> Primeiro, você deve verificar se é possível ignorar o erro com segurança. Em um utilitário do MySQL, conecte-se à réplica de leitura e execute o seguinte comando do MySQL.
>
> ```
> SHOW REPLICA STATUS\G
> ```
>
> Para obter informações sobre os valores retornados, consulte a documentação do MySQL. As versões anteriores do MySQL usavam `SHOW SLAVE STATUS` em vez de `SHOW REPLICA STATUS`. Se você estiver usando uma versão do MySQL anterior à 8.0.23, use `SHOW SLAVE STATUS`.

É possível ignorar um erro na réplica de leitura das seguintes maneiras.

Tópicos

- Chamando o procedimento mysql.rds_skip_repl_error
- Configurar o parâmetro slave_skip_errors

Chamando o procedimento mysql.rds_skip_repl_error

O Amazon RDS fornece um procedimento armazenado que é possível chamar para ignorar um erro nas suas réplicas de leitura. Primeiro, conecte-se à sua réplica de leitura. Depois disso, emita os comandos apropriados, conforme mostrado a seguir. Para obter mais informações, consulteConectar-se à instância de banco de dados do MySQL

Para ignorar o erro, emita o seguinte comando.

```
CALL mysql.rds_skip_repl_error;
```

Esse comando não terá efeito se você o executar na instância de banco de dados de origem ou em uma réplica de leitura que não apresentou um erro de replicação.

Para mais informações, como as versões compatíveis com o MySQL `mysql.rds_skip_repl_error`, veja mysql.rds_skip_repl_error.

> **Important**
>
> Se você tentar chamar `mysql.rds_skip_repl_error` e obter o seguinte erro: `ERROR 1305 (42000): PROCEDURE mysql.rds_skip_repl_error does not exist`, faça upgrade da sua instância de banco de dados do MySQL para a versão secundária mais recente ou uma das versões secundárias mínimas listadas em mysql.rds_skip_repl_error.

Configurar o parâmetro slave_skip_errors

Para ignorar um ou mais erros, é possível definir o parâmetro estático `slave_skip_errors` na réplica de leitura. Esse parâmetro pode ser definido para ignorar um ou mais códigos de erro de replicação específicos. Atualmente, é possível definir esse parâmetro somente para instâncias de banco de dados do RDS para MySQL 5.7. Depois de alterar a configuração desse parâmetro, certifique-se de reinicializar sua instância de banco de dados para que a nova configuração entre em vigor. Para obter mais informações sobre definir esse parâmetro, consulte a documentação do MySQL.

Convém definir esse parâmetro em um grupo de parâmetros de banco de dados separado. É possível associar esse grupo de parâmetros de banco de dados somente às réplicas de leitura que precisam ignorar erros. Seguir essa prática recomendada reduz o possível impacto em outras instâncias de banco de dados e réplicas de leitura.

> **Important**
>
> Definir um valor não padrão para esse parâmetro pode resultar em inconsistência de replicação. Apenas defina esse parâmetro como um valor não padrão se você tiver esgotado outras opções para resolver o problema e tiver certeza do impacto potencial sobre os dados da réplica de leitura.

Trabalhar com espaços de tabela do InnoDB para melhorar o tempo de recuperação de falhas do RDS para MySQL

Cada tabela no MySQL consiste de uma definição, de dados e de índices de tabela. O mecanismo de armazenamento InnoDB do MySQL armazena dados e índices de tabela em um espaço de tabela. O InnoDB cria um tablespace compartilhado global que contém um dicionário de dados e outros metadados relevantes, e pode conter dados e índices de tabela. O InnoDB também pode criar tablespaces separados para cada tabela e partição. Esses tablespaces separados são armazenados em arquivos de extensão .ibd e o cabeçalho de cada tablespaces contém um número que o identifica de forma exclusiva.

O Amazon RDS fornece um parâmetro em um grupo de parâmetros do MySQL chamado `innodb_file_per_table`. Esses parâmetros controlam se o InnoDB adiciona novos dados de tabela e índices ao espaço de tabela compartilhado (definindo o valor do parâmetro como 0) ou aos espaços de tabelas individuais (definindo o valor do parâmetro como 1). O Amazon RDS define o valor padrão para `innodb_file_per_table` como 1, o que permite eliminar tabelas individuais do InnoDB e recuperar o armazenamento usado por essas tabelas para a instância de banco de dados. Na maioria dos casos de uso, definir o parâmetro `innodb_file_per_table` como 1 é a configuração recomendada.

Você deve definir o parâmetro `innodb_file_per_table` como 0 quando você tiver um grande número de tabelas, por exemplo mais de 1.000 tabelas, ao usar armazenamento SSD padrão (magnético) ou de propósito geral ou mais de 10.000 tabelas ao usar armazenamento de IOPS provisionadas. Quando você define esse parâmetro como 0, os tablespaces individuais não são criados e isso pode melhorar o tempo necessário para a recuperação da falha no banco de dados.

O MySQL processa cada arquivo de metadados, que inclui tablespaces, durante o ciclo de recuperação de falhas. O tempo que o MySQL leva para processar as informações de metadados no tablespace compartilhado é insignificante em comparação ao tempo necessário para processar milhares de arquivos de tablespace quando há vários tablespaces. Como o número do tablespace é

armazenado no cabeçalho de cada arquivo, o tempo total para ler todos os arquivos do tablespace pode levar até várias horas. Por exemplo, pode levar de cinco a oito horas para processar um milhão de tablespaces do InnoDB no armazenamento padrão durante um ciclo de recuperação de falhas. Em alguns casos, o InnoDB pode determinar que precisa de limpeza adicional após um ciclo de recuperação de falhas para poder começar outro ciclo de recuperação de falhas, o que prolongará o tempo de recuperação. Lembre-se que um ciclo de recuperação de falhas também envolve transações de retorno, correção de páginas com defeito e outras operações além do processamento de informações de tablespace.

Como o parâmetro `innodb_file_per_table` reside em um grupo de parâmetros, você pode alterar o valor do parâmetro editando o grupo de parâmetros usado por sua instância de banco de dados sem ter que reiniciar a instância de banco de dados. Após a configuração ser alterada, por exemplo, de 1 (criar tabelas individuais) para 0 (usar tablespace compartilhado), novas tabelas do InnoDB serão adicionadas ao tablespace compartilhado, enquanto as tabelas existentes continuam a ter tablespaces individuais. Para mover uma tabela do InnoDB para o tablespace compartilhado, você deve usar o comando `ALTER TABLE`.

Migração de vários tablespaces ao tablespace compartilhado

Você pode mover os metadados da tabela do InnoDB de seu próprio tablespace ao tablespace compartilhado, que reconstruirá os metadados da tabela de acordo com a configuração do parâmetro `innodb_file_per_table`. Primeiro, conecte-se à sua instância de banco de dados MySQL. Depois disso, emita os comandos apropriados, conforme mostrado a seguir. Para obter mais informações, consulte Conectar-se à instância de banco de dados do MySQL.

```
ALTER TABLE table_name ENGINE = InnoDB, ALGORITHM=COPY;
```

Por exemplo, a seguinte consulta retorna uma instrução `ALTER TABLE` para cada tabela InnoDB que não esteja no espaço de tabela compartilhado.

Para instâncias de banco de dados do MySQL 5.7:

```
SELECT CONCAT('ALTER TABLE `',
REPLACE(LEFT(NAME , INSTR((NAME), '/') - 1), '`', '``'), '`.`',
REPLACE(SUBSTR(NAME FROM INSTR(NAME, '/') + 1), '`', '``'), `` ENGINE=InnoDB,
 ALGORITHM=COPY;') AS Query
FROM INFORMATION_SCHEMA.INNODB_SYS_TABLES
WHERE SPACE <> 0 AND LEFT(NAME, INSTR((NAME), '/') - 1) NOT IN ('mysql','');
```

Para instâncias de banco de dados MySQL 8.4 e 8.0:

```
SELECT CONCAT('ALTER TABLE `',
REPLACE(LEFT(NAME , INSTR((NAME), '/') - 1), '`', '``'), '`.`',
REPLACE(SUBSTR(NAME FROM INSTR(NAME, '/') + 1), '`', '``'), '` ENGINE=InnoDB,
 ALGORITHM=COPY;') AS Query
FROM INFORMATION_SCHEMA.INNODB_TABLES
WHERE SPACE <> 0 AND LEFT(NAME, INSTR((NAME), '/') - 1) NOT IN ('mysql','');
```

Recriar uma tabela do MySQL para mover os metadados da tabela para o tablespace compartilhado requer temporariamente espaço de armazenamento adicional para recriar a tabela, portanto a instância do banco de dados deve ter espaço de armazenamento disponível. Durante a recriação, a tabela é bloqueada e inacessível para consultas. Para tabelas pequenas ou tabelas que não são acessadas com frequência, isso pode não ser um problema. Para tabelas grandes ou tabelas acessadas com frequência em um ambiente fortemente simultâneo, é possível recriar tabelas em uma réplica de leitura.

É possível criar uma réplica de leitura e migrar metadados da tabela para o tablespace compartilhado na réplica de leitura. Embora a instrução ALTER TABLE bloqueie o acesso na réplica de leitura, a instância de banco de dados de origem não é afetada. A instância de banco de dados de origem continuará a gerar seus logs binários enquanto a réplica de leitura apresentar atraso durante o processo de recriação da tabela. Como a recriação requer espaço de armazenamento adicional e o arquivo de log de reprodução pode ficar grande, você deve criar uma réplica de leitura com o armazenamento alocado maior que a instância de banco de dados de origem.

Para criar uma réplica de leitura e recriar tabelas do InnoDB a fim de usar o tablespace compartilhado, execute as seguintes etapas:

1. Verifique se a retenção de backup está habilitada na instância do banco de dados de origem, de modo que o registro de log binário esteja habilitado
2. Use o AWS Management Console ou a AWS CLI para criar uma réplica de leitura para a instância de banco de dados de origem. Como a criação de uma réplica de leitura envolve muitos dos mesmos processos que os da recuperação de falhas, o processo de criação pode levar algum tempo se houver um grande número de tablespaces do InnoDB. Aloque mais espaço de armazenamento na réplica de leitura do que é usado atualmente na instância de banco de dados de origem.
3. Quando a réplica de leitura tiver sido criada, crie um grupo de parâmetros com as configurações de parâmetro `read_only = 0` e `innodb_file_per_table = 0`. Depois, associe o grupo de parâmetros à réplica de leitura.
4. Emita a seguinte instrução SQL para todas as tabelas que você deseja migrar na réplica:

```
ALTER TABLE name ENGINE = InnoDB
```

5. Quando todas as instruções `ALTER TABLE` forem concluídas na réplica de leitura, verifique se a réplica de leitura está conectada à instância de banco de dados de origem e se as duas instâncias estão sincronizadas.
6. Use o console ou a CLI para promover a réplica de leitura para ser a instância. Verifique se o grupo de parâmetros usado para a nova instância de banco de dados autônoma tem o parâmetro `innodb_file_per_table` definido como 0. Altere o nome da nova instância de banco de dados autônoma e aponte todos os aplicativos para a nova instância de banco de dados autônoma.

Gerenciar o histórico de status global do RDS para MySQL

> **Tip**
>
> Para analisar a performance do banco de dados, você também pode usar o Insights de Performance no Amazon RDS. Para ter mais informações, consulte Monitorar a carga de banco de dados com o Performance Insights no Amazon RDS.

O MySQL mantém muitas variáveis de status que fornecem informações sobre sua operação. Os valores podem lhe ajudar a detectar problemas de bloqueio ou memória em uma instância de banco de dados. Os valores dessas variáveis de status se acumulam desde a última vez que a instância de banco de dados foi iniciada. Você pode redefinir a maioria das variáveis de status para 0 usando o comando `FLUSH STATUS`.

Para permitir o monitoramento desses valores ao longo do tempo, o Amazon RDS fornece um conjunto de procedimentos que fará o snapshot dos valores dessas variáveis de status ao longo do tempo e as registrará em uma tabela, juntamente com quaisquer alterações feitas desde o último snapshot. Esta infraestrutura, denominada GoSH (Histórico de status global), está instalada em todas as instâncias de banco de dados MySQL, desde as versões 5.5.23. O GoSH é desabilitado por padrão.

Para habilitar o GoSH, primeiro habilite o programador de eventos por meio de um grupo de parâmetros de banco de dados, definindo o parâmetro `event_scheduler` como `ON`. Para instâncias de banco de dados do MySQL que executam o MySQL 5.7, defina também o parâmetro `show_compatibility_56` para `1`. Para obter informações sobre como criar e modificar um grupo de parâmetros de banco de dados, consulte Grupos de parâmetros para Amazon

RDS. Para obter informações sobre os efeitos colaterais da habilitação desse parâmetro, consulteshow_compatibility_56 noMySQL 5.7 Reference Manual.

Você poderá, então, usar os procedimentos na tabela a seguir para habilitar e configurar o GoSH. Primeiro, conecte-se à sua instância de banco de dados MySQL. Depois disso, emita os comandos apropriados, conforme mostrado a seguir. Para ter mais informações, consulte Conectar-se à instância de banco de dados do MySQL. Para cada procedimento, execute o comando a seguir e substitua *procedure-name*:

```
CALL procedure-name;
```

A tabela a seguir lista todos os procedimentos que você pode usar para *procedure-name* no comando anterior.

Procedimento	Descrição
mysql.rds_enable_gsh_collector	Permite que o GoSH faça snapshots padrão em intervalos especificados por rds_set_gsh_collector .
mysql.rds_set_gsh_collector	Especifica o intervalo em minutos entre snapshots. O valor padrão é 5.
mysql.rds_disable_gsh_collector	Desabilita os snapshots.
mysql.rds_collect_global_status_history	Faz um snapshot sob demanda.
mysql.rds_enable_gsh_rotation	Permite o revezamento do conteúdo da tabela mysql.rds_global_status_history com o da mysql.rds_global_status_history_old em intervalos especificados pelo rds_set_gsh_rotation .
mysql.rds_set_gsh_rotation	Especifica o intervalo em dias entre os revezamentos de tabelas. O valor padrão é 7.
mysql.rds_disable_gsh_rotation	Desabilita o revezamento de tabelas.

Procedimento	Descrição
`mysql.rds_rotate_global_status_history`	Reveza o conteúdo da tabela `mysql.rds_global_status_history` para o da `mysql.rds_global_status_history_old` sob demanda.

Quando o GoSH estiver sendo executado, você pode consultar as tabelas nas quais ele grava. Por exemplo, para consultar a taxa de acerto do grupo de buffer do Innodb, você deve emitir a seguinte consulta:

```
select a.collection_end, a.collection_start, (( a.variable_Delta-b.variable_delta)/
a.variable_delta)*100 as "HitRatio"
    from mysql.rds_global_status_history as a join mysql.rds_global_status_history as b
 on a.collection_end = b.collection_end
    where a. variable_name = 'Innodb_buffer_pool_read_requests' and b.variable_name =
 'Innodb_buffer_pool_reads'
```

Configuração do tamanho do pool de buffer e a capacidade de log redo no MySQL 8.4

No MySQL 8.4, o Amazon RDS ativa o parâmetro `innodb_dedicated_server` por padrão. Com o parâmetro `innodb_dedicated_server`, o mecanismo do banco de dados calcula os parâmetros `innodb_buffer_pool_size` e `innodb_redo_log_capacity`. Para obter informações sobre como esses parâmetros são calculados, consulte Configuração do tamanho do pool de buffer do InnoDB e Log Redo na documentação do MySQL.

Com `innodb_dedicated_server` habilitado, o parâmetro `innodb_buffer_pool_size` é calculado com base na memória da classe da instância de banco de dados. A tabela a seguir exibe a memória do servidor detectada e o tamanho do pool de buffer correspondente.

Memória do servidor detectada	Tamanho do pool de buffer
< 1 GB	Valor padrão de 128 MB
1 GB a 4 GB	*Memória de servidor detectada* * 0,5
> 4 GB	*Memória de servidor detectada* * 0,75

O parâmetro `innodb_redo_log_capacity` é escalado automaticamente com a classe da instância para (número de vCPUs/2) GB até um máximo de 16 GB. Classes de instâncias maiores têm uma maior capacidade de log redo, o que pode melhorar o desempenho e a resiliência de workloads com uso intenso de gravação.

Antes de atualizar do MySQL 8.0 para o MySQL 8.4, aumente o espaço de armazenamento para acomodar um possível aumento no tamanho dos logs redo que pode ocorrer após a conclusão da atualização. Para ter mais informações, consulte Aumento da capacidade de armazenamento da instância de banco de dados.

Se você não quiser que o parâmetro `innodb_dedicated_server` calcule os valores para os parâmetros `innodb_buffer_pool_size` e `innodb_redo_log_capacity`, substitua esses valores definindo valores específicos para eles em um grupo de parâmetros personalizado. Como alternativa, você pode desativar o parâmetro `innodb_dedicated_server` e definir valores para os parâmetros `innodb_buffer_pool_size` e `innodb_redo_log_capacity` em um grupo de parâmetros personalizado. Para ter mais informações, consulte Grupos de parâmetros padrão e personalizados.

Se você desabilitar o parâmetro `innodb_dedicated_server` definindo-o como `0` e não definir valores para os parâmetros `innodb_buffer_pool_size` e `innodb_redo_log_capacity`, o Amazon RDS definirá os dois últimos parâmetros como 128 MB e 100 MB, respectivamente. Esses padrões resultam em desempenho ruim em classes de instâncias maiores.

Fuso horário local para instâncias de banco de dados MySQL

Por padrão, o fuso horário de uma instância de banco de dados MySQL é o Tempo Universal Coordenado (UTC). Você pode definir o fuso horário de sua instância de banco de dados como o fuso horário local de seu aplicativo.

Para definir o fuso horário local de uma instância de banco de dados, defina o parâmetro `time_zone` no grupo de parâmetros para sua instância de banco de dados como um dos valores compatíveis listados posteriormente nesta seção. Quando você configura o parâmetro `time_zone` para um grupo de parâmetros, todas as instâncias de banco de dados e as réplicas de leitura que estiverem usando esse grupo de parâmetros mudarão para usar o novo fuso horário local. Para obter informações sobre como configurar parâmetros em um grupo de parâmetros, consulte Grupos de parâmetros para Amazon RDS.

Após definir o fuso horário local, todas as novas conexões ao banco de dados refletirão a alteração. Se você tiver conexões abertas em seu banco de dados quando mudar o fuso horário local, você não verá a atualização do fuso horário local até que encerre a conexão e abra uma nova conexão.

É possível definir um fuso horário local diferente para uma instância de banco de dados e uma ou mais de suas réplicas de leitura. Para fazer isso, use um grupo de parâmetros diferente para a instância de banco de dados e a(s) replica(s), e defina o parâmetro `time_zone` em cada grupo de parâmetros para um fuso horário local diferente.

Se você estiver replicando entre as Regiões da AWS, a instância de banco de dados de origem e a réplica de leitura usarão grupos de parâmetros diferentes (os grupos de parâmetros são exclusivos de cada Região da AWS). Para usar o mesmo fuso horário local a cada instância, é necessário configurar o parâmetro `time_zone` nos grupos de parâmetros da instância e da réplica de leitura.

Quando você restaura uma instância de banco de dados de um snapshot de banco de dados, o fuso horário local é definido como UTC. Você pode atualizar o fuso horário para o fuso horário local após a conclusão da restauração. Se você restaurar uma instância de banco de dados para um período específico, o fuso horário local para a instância de banco de dados restaurada representará a configuração de fuso horário do grupo de parâmetros da instância de banco de dados restaurada.

A Internet Assigned Numbers Authority (IANA) publica novos fusos horários em https://www.iana.org/time-zones várias vezes por ano. Toda vez que o RDS lança uma nova versão de manutenção secundária do MySQL, ela vem com os dados de fuso horário mais recentes no momento do lançamento. Ao usar as versões mais recentes do RDS para MySQL, você tem dados de fuso horário

recentes do RDS. Para garantir que sua instância de banco de dados tenha dados de fuso horário recentes, recomendamos atualizar para uma versão superior do mecanismo de banco de dados. Como alternativa, você pode modificar as tabelas de fuso horário nas instâncias de banco de dados do MariaDB manualmente. Para fazer isso, você pode usar comandos SQL ou executar a ferramenta mysql_tzinfo_to_sql em um cliente SQL. Depois de atualizar os dados do fuso horário manualmente, reinicialize sua instância de banco de dados para que as alterações tenham efeito. O RDS não modifica nem redefine os dados de fuso horário das instâncias de banco de dados em execução. Os novos dados de fuso horário são instalados somente quando você executa uma atualização da versão do mecanismo de banco de dados.

Você pode definir seu fuso horário local para um dos seguintes valores.

Zona	Time zone (Fuso horário)
África	África/Cairo, África/Casablanca, África/Harare, África/Monrovia, África/Nairobi, África/Tripoli, África/Windhoek
América	América/Araguaina, América/Assunção, América/Bogotá, América/Buenos_Aires, América/Caracas, América/Chihuahua, América/Cuiabá, América/Denver, América/Fortaleza, América/Guatemala, América/Halifax, América/Manaus, América/Matamoros, América/Monterrey, América/Montevidéu, América/Phoenix, América/Santiago, América/Tijuana
Ásia	Ásia/Amman, Ásia/Ashgabat, Ásia/Baghdad, Ásia/Baku, Ásia/Bangkok, Ásia/Beirut, Ásia/Calcutta, Ásia/Damascus, Ásia/Dhaka, Ásia/Irkutsk, Ásia/Jerusalem, Ásia/Kabul, Ásia/Karachi, Ásia/Kathmandu, Ásia/Krasnoyarsk, Ásia/Magadan, Ásia/Muscat, Ásia/Novosibirsk, Ásia/Riyadh, Ásia/Seoul, Ásia/Shanghai, Ásia/Singapore, Ásia/Taipei, Ásia/Tehran, Ásia/Tokyo, Ásia/Ulaanbaatar, Ásia/Vladivostok, Ásia/Yakutsk, Ásia/Yerevan
Atlântico	Atlântico/Açores
Austrália	Austrália/Adelaide, Austrália/Brisbane, Austrália/Darwin, Austrália/Hobart, Austrália/Perth, Austrália/Sydney
Brasil	Brasil/DeNoronha, Brasil/Leste
Canadá	Canadá/Newfoundland, Canadá/Saskatchewan, Canadá/Yukon

Zona	Time zone (Fuso horário)
Europa	Europa/Amsterdam, Europa/Athens, Europa/Dublin, Europa/Helsinki, Europa/Istanbul, Europa/Kaliningrad, Europa/Moscow, Europa/Paris, Europa/Prague, Europa/Sarajevo
Pacífico	Pacífico/Auckland, Pacífico/Fiji, Pacífico/Guam, Pacífico/Honolulu, Pacífico/Samoa
EUA	EUA/Alasca, EUA/Central, EUA/East-Indiana, EUA/Eastern, EUA/Pacífico
UTC	UTC

Problemas conhecidos e limitações do Amazon RDS para MySQL

Veja a seguir os problemas e as limitações conhecidos no trabalho com o Amazon RDS para MySQL.

Tópicos

- Palavra reservada InnoDB
- Comportamento de armazenamento cheio para Amazon RDS para MySQL
- Tamanho do grupo de buffers do InnoDB inconsistente
- A otimização de mesclagem de índice retorna resultados errados
- Exceções de parâmetros do MySQL para instâncias de bancos de dados do Amazon RDS
- Limites de tamanho de arquivo do MySQL no Amazon RDS
- Não há suporte ao plugin Keyring do MySQL
- Portas personalizadas
- Limitações de procedimentos armazenados do MySQL
- Replicação baseada em GTID com uma instância de origem externa
- Plug-in de autenticação padrão do MySQL
- Substituir innodb_buffer_pool_size
- Atualização do MySQL 5.7 para o MySQL 8.4
- Compressão de páginas do InnoDB

Palavra reservada InnoDB

`InnoDB` é uma palavra reservada para o RDS para MySQL. Você não pode usar esse nome para um banco de dados MySQL.

Comportamento de armazenamento cheio para Amazon RDS para MySQL

Quando o armazenamento fica cheio em uma instância de banco de dados MySQL, pode haver inconsistências de metadados, incompatibilidade de dicionários e tabelas órfãs. Para evitar esses problemas, o Amazon RDS interrompe automaticamente uma instância de banco de dados que atinja o estado `storage-full`.

Uma instância de banco de dados MySQL atinge o estado `storage-full` nos seguintes casos:

- A instância de banco de dados tem menos de 20.000 MiB de armazenamento e o armazenamento disponível atinge 200 MiB ou menos.
- A instância de banco de dados tem mais de 102.400 MiB de armazenamento e o armazenamento disponível só chega a 1024 MiB ou menos.
- A instância de banco de dados tem entre 20.000 MiB e 102.400 MiB de armazenamento e está com menos de 1% de armazenamento disponível.

Depois que o Amazon RDS interromper uma instância de banco de dados automaticamente porque ela atingiu o estado `storage-full`, você ainda poderá modificá-la. Para reiniciar a instância de banco de dados, conclua pelo menos um dos seguintes procedimentos:

- Modifique a instância de banco de dados para habilitar o autoscaling do armazenamento.

  Para ter mais informações sobre o autoscaling do armazenamento, consulte Gerenciar a capacidade automaticamente com a escalabilidade automática de armazenamento do Amazon RDS.
- Modifique a instância de banco de dados para aumentar sua capacidade de armazenamento.

  Para ter mais informações sobre o aumento da capacidade de armazenamento, consulte Aumento da capacidade de armazenamento da instância de banco de dados.

Depois de fazer uma dessas alterações, a instância de banco de dados será reiniciada automaticamente. Para mais informações sobre a modificação de uma instância de banco de dados , consulte Modificar uma instância de banco de dados do Amazon RDS.

Tamanho do grupo de buffers do InnoDB inconsistente

Para o MySQL 5.7, existe atualmente um bug na forma como o tamanho do grupo de buffers do InnoDB é gerenciado. O MySQL 5.7 pode ajustar o parâmetro `innodb_buffer_pool_size` para um valor muito grande, que pode fazer com que o grupo de buffers do InnoDB cresça demais e consuma muita memória. Esse efeito pode fazer com que o mecanismo de banco de dados MySQL pare de funcionar ou pode impedir que ele seja iniciado. Esse problema é mais comum para classes de instâncias de banco de dados que têm menos memória disponível.

Para resolver esse problema, defina o valor do parâmetro `innodb_buffer_pool_size` como um múltiplo do produto do valor do parâmetro `innodb_buffer_pool_instances` e do valor do parâmetro `innodb_buffer_pool_chunk_size`. Por exemplo, você pode configurar o valor do

parâmetro `innodb_buffer_pool_size` como um múltiplo de oito vezes o produto dos valores dos parâmetros `innodb_buffer_pool_instances` e `innodb_buffer_pool_chunk_size`, conforme mostrado no exemplo a seguir.

```
innodb_buffer_pool_chunk_size = 536870912
innodb_buffer_pool_instances = 4
innodb_buffer_pool_size = (536870912 * 4) * 8 = 17179869184
```

Para obter detalhes sobre esse bug do MySQL 5.7, acesse https://bugs.mysql.com/bug.php?id=79379 na documentação do MySQL.

A otimização de mesclagem de índice retorna resultados errados

As consultas que utilizam a otimização de mesclagem de índice podem retornar resultados incorretos devido a um bug no otimizador de consulta MySQL que foi introduzido no MySQL 5.5.37. Ao emitir uma consulta em uma tabela com vários índices, o otimizador verifica intervalos de linhas com base em vários índices, mas não mescla os resultados corretamente. Para ter mais informações sobre o bug do otimizador de consulta, acesse http://bugs.mysql.com/bug.php?id=72745 e http://bugs.mysql.com/bug.php?id=68194 no banco de dados de bugs do MySQL.

Por exemplo, considere uma consulta em uma tabela com dois índices em que os argumentos de pesquisa fazem referência às colunas indexadas.

```
SELECT * FROM table1
WHERE indexed_col1 = 'value1' AND indexed_col2 = 'value2';
```

Nesse caso, o mecanismo de pesquisa pesquisará ambos os índices. No entanto, devido ao erro, os resultados mesclados estão incorretos.

Para resolver esse problema, você pode realizar um dos seguintes procedimentos:

- Defina o parâmetro `optimizer_switch` como `index_merge=off` no grupo de parâmetros de banco de dados para a sua instância de banco de dados MySQL. Para obter informações sobre como definir os parâmetros do grupo de parâmetros de banco de dados, consulte Grupos de parâmetros para Amazon RDS.
- Atualize a instância de banco de dados do MySQL para o MySQL versão 5.7 ou 8.0. Para ter mais informações, consulte Atualizações do mecanismo de banco de dados do RDS para MySQL.
- Se você não puder atualizar sua instância ou alterar o parâmetro `optimizer_switch`, poderá contornar o bug identificando explicitamente um índice para a consulta, por exemplo:

```
SELECT * FROM table1
USE INDEX covering_index
WHERE indexed_col1 = 'value1' AND indexed_col2 = 'value2';
```

Para ter mais informações, consulte Index merge optimization (Otimizar a mesclagem de índice) na documentação do MySQL.

Exceções de parâmetros do MySQL para instâncias de bancos de dados do Amazon RDS

Alguns parâmetros do MySQL requerem considerações especiais quando usados com uma instância de banco de dados do Amazon RDS.

lower_case_table_names

Como o Amazon RDS usa um sistema de arquivos que diferencia maiúsculas de minúsculas, não há suporte para definir o valor do parâmetro de servidor `lower_case_table_names` como 2 (nomes armazenados conforme especificados, mas comparados em minúsculas). Veja a seguir os valores compatíveis com as instâncias de banco de dados do Amazon RDS para MySQL:

- 0 (nomes armazenados como dados e as comparações diferenciam maiúsculas de minúsculas) é compatível com todas as versões do RDS para MySQL.
- 1 (nomes armazenados em letras minúsculas e comparações não diferenciam maiúsculas de minúsculas) é compatível com o RDS para MySQL versão 5.7, versão 8.0.28 e versões 8.0 posteriores, e versão 8.4.

Defina o parâmetro `lower_case_table_names` em um grupo de parâmetros de banco de dados personalizado antes de criar uma instância de banco de dados. Em seguida, especifique o grupo de parâmetros de banco de dados personalizado ao criar a instância de banco de dados.

Quando um grupo de parâmetros é associado a uma instância de banco de dados MySQL com uma versão inferior a 8.0, recomendamos que você evite alterar o parâmetro `lower_case_table_names` no grupo de parâmetros. Alterá-lo pode causar inconsistências com backups de recuperação para um ponto no tempo e instâncias de bancos de dados de réplica de leitura.

Quando um grupo de parâmetros é associado a uma instância de banco de dados MySQL versão 8.0 ou 8.4, não é possível modificar o parâmetro `lower_case_table_names` no grupo de parâmetros.

Réplicas de leitura sempre devem usar o mesmo valor de parâmetro `lower_case_table_names` que a instância de banco de dados de origem.

long_query_time

Você pode definir o parâmetro `long_query_time` como um valor de ponto flutuante para conseguir registrar consultas lentas no log de consultas lentas do MySQL com resolução de microssegundos. Você pode definir um valor como 0,1 segundos, que seria de 100 milissegundos, para ajudar ao depurar transações lentas que demoram menos de um segundo.

Limites de tamanho de arquivo do MySQL no Amazon RDS

Para instâncias de banco de dados do MySQL versões 8.0 e posteriores, o tamanho máximo do arquivo é de 16 TiB. Ao usar espaços de tabela de arquivo por tabela, o tamanho máximo do arquivo limita o tamanho de uma tabela do InnoDB a 16 TiB. Os espaços de tabelas de arquivo por tabela do InnoDB (com cada tabela em seu próprio espaço de tabela) são definidos por padrão para instâncias de bancos de dados MySQL. Consulte mais informações em InnoDB Limits na documentação do MySQL.

> **Note**
>
> Algumas instâncias de Banco de Dados existentes têm um limite menor. Por exemplo, as instâncias de banco de dados MySQL criadas antes de abril de 2014 têm um limite de tamanho de arquivo e de tabela de 2 TB. Esse limite de tamanho de arquivo de 2 TB também se aplica a instâncias de banco de dados ou a réplicas de leitura criadas de snapshots de banco de dados tirados antes de abril de 2014, independentemente de quando a instância de banco de dados foi criada.

Existem vantagens e desvantagens na utilização de espaços de tabela de arquivo por tabela do InnoDB, dependendo do seu aplicativo. Para determinar a melhor abordagem para a aplicação, consulte File-per-table tablespaces na documentação do MySQL.

Não recomendamos permitir que as tabelas cresçam até o tamanho máximo do arquivo. Em geral, uma prática recomendada é particionar dados em tabelas menores, o que pode melhorar a performance e os tempos de recuperação.

Uma opção que você pode usar para dividir uma tabela grande em tabelas menores é o particionamento. O particionamento distribui partes da sua tabela grande em arquivos separados com base em regras que você especifica. Por exemplo, se você armazenar transações por data, poderá criar regras de particionamento que distribuem transações antigas em arquivos separados usando o particionamento. Em seguida, periodicamente, você pode arquivar os dados históricos de transações que não precisam estar prontamente disponíveis para o seu aplicativo. Para ter mais informações, consulte Partitioning na documentação do MySQL.

Como não há uma única tabela ou visualização do sistema que indique o tamanho de todas as tabelas e do espaço de tabela do sistema InnoDB, consulte várias tabelas para definir o tamanho dos espaços de tabela.

Como determinar o tamanho do espaço de tabela do sistema InnoDB e do espaço de tabela do dicionário de dados

- Use o seguinte comando SQL para determinar se alguns dos seus espaços de tabela são muito grandes e são candidatos para particionamento.

> **Note**
>
> O espaço de tabela do dicionário de dados é específico para o MySQL 8.0 e versões posteriores.

```
select FILE_NAME,TABLESPACE_NAME, ROUND(((TOTAL_EXTENTS*EXTENT_SIZE)
/1024/1024/1024), 2)  as "File Size (GB)" from information_schema.FILES
where tablespace_name in ('mysql','innodb_system');
```

Como determinar o tamanho das tabelas de usuários do InnoDB fora do espaço de tabela do sistema InnoDB (para versões do MySQL 5.7)

- Use o seguinte comando SQL para determinar se algumas das suas tabelas são muito grandes e são candidatas para particionamento.

```
SELECT SPACE,NAME,ROUND((ALLOCATED_SIZE/1024/1024/1024), 2)
as "Tablespace Size (GB)"
FROM information_schema.INNODB_SYS_TABLESPACES ORDER BY 3 DESC;
```

Como determinar o tamanho das tabelas de usuários do InnoDB fora do espaço de tabela do sistema InnoDB (para versões do MySQL 8.0 e posteriores)

- Use o seguinte comando SQL para determinar se algumas das suas tabelas são muito grandes e são candidatas para particionamento.

```
SELECT SPACE,NAME,ROUND((ALLOCATED_SIZE/1024/1024/1024), 2)
as "Tablespace Size (GB)"
FROM information_schema.INNODB_TABLESPACES ORDER BY 3 DESC;
```

Como determinar o tamanho das tabelas de usuários que não são do InnoDB

- Use o seguinte comando SQL para determinar se algumas das suas tabelas de usuários que não são do InnoDB são muito grandes.

```
SELECT TABLE_SCHEMA, TABLE_NAME, round(((DATA_LENGTH + INDEX_LENGTH+DATA_FREE)
/ 1024 / 1024/ 1024), 2) As "Approximate size (GB)" FROM information_schema.TABLES
WHERE TABLE_SCHEMA NOT IN ('mysql', 'information_schema', 'performance_schema')
and ENGINE<>'InnoDB';
```

Para habilitar espaços de tabela de arquivo por tabela do InnoDB

- Defina o parâmetro innodb_file_per_table como 1 no grupo de parâmetros da instância de banco de dados.

Para desabilitar espaços de tabela de arquivo por tabela do InnoDB

- Defina o parâmetro innodb_file_per_table como `0` no grupo de parâmetros da instância de banco de dados.

Para obter informações sobre como atualizar um grupo de parâmetros, consulte Grupos de parâmetros para Amazon RDS.

Quando tiver habilitado ou desabilitado espaços de tabelas de arquivo por tabela do InnoDB, você poderá emitir um comando `ALTER TABLE` para mover uma tabela do espaço de tabela global para seu próprio espaço de tabela, ou do seu próprio espaço de tabela para o espaço de tabela global, conforme mostrado no exemplo a seguir:

```
ALTER TABLE table_name TABLESPACE=innodb_file_per_table;
```

Não há suporte ao plugin Keyring do MySQL

No momento, o Amazon RDS para MySQL não oferece suporte ao plug-in Keyring `keyring_aws` da Amazon Web Services do MySQL.

Portas personalizadas

O Amazon RDS bloqueia conexões com a porta personalizada 33060 para o mecanismo MySQL. Escolha uma porta diferente para o seu mecanismo MySQL.

Limitações de procedimentos armazenados do MySQL

Os procedimentos armazenados mysql.rds_kill e mysql.rds_kill_query não podem encerrar sessões ou consultas pertencentes a usuários do MySQL que tenham nomes de usuário com mais de 16 caracteres nas seguintes versões do RDS para MySQL:

- Versão 8.0.32 e versões 8 anteriores
- Versão 5.7.41 e versões 5.7 anteriores

Replicação baseada em GTID com uma instância de origem externa

O Amazon RDS comporta a replicação baseada em identificadores de transação global (GTIDs) de uma instância externa do MySQL externa em uma instância de banco de dados do Amazon RDS para MySQL que exija a definição de GTID_PURGED durante a configuração. No entanto, somente o RDS para MySQL 8.0.37 e versões posteriores comportam essa funcionalidade.

Plug-in de autenticação padrão do MySQL

O RDS para MySQL versão 8.0.34 e 8.0 posteriores e posterior usam o plug-in `mysql_native_password`. Você não pode alterar a configuração `default_authentication_plugin`.

O RDS para MySQL versão 8.4 e versões posteriores usam o plug-in `caching_sha2_password` como o plug-in de autenticação padrão. Você pode alterar o plug-in de autenticação padrão para o MySQL 8.4. O plug-in `mysql_native_password` ainda funciona com o MySQL 8.4, mas o suporte a esse plug-in termina com o MySQL 8.4. Para alterar o plug-in de autenticação padrão, crie um grupo de parâmetros personalizado e modifique o valor do parâmetro `authentication_policy`. Para ter mais informações, consulte the section called “Grupos de parâmetros padrão e personalizados”.

Substituir innodb_buffer_pool_size

Com classes de instância de banco de dados micro ou small, o valor padrão do parâmetro `innodb_buffer_pool_size` pode ser diferente do valor exibido ao executar o seguinte comando:

```
mysql> SELECT @@innodb_buffer_pool_size;
```

Essa diferença pode ocorrer quando o Amazon RDS precisa substituir o valor padrão como parte do gerenciamento das classes de instância de banco de dados. Se necessário, você poderá substituir o valor padrão e defini-lo como um valor compatível com a classe de instância de banco de dados. Para determinar um valor válido, adicione o uso da memória e a memória total disponível na instância de banco de dados. Para ter mais informações, consulte Tipos de instâncias do Amazon RDS.

Se a instância de banco de dados tiver apenas 4 GB de memória, você não poderá definir `innodb_buffer_pool_size` como 8 GB, mas poderá defini-lo como 3 GB, dependendo da quantidade de memória alocada a outros parâmetros.

Se o valor inserido for muito grande, o Amazon RDS o reduzirá para os seguintes limites:

- Classes de instância de banco de dados micro: 256 MB
- Classes de instância de banco de dados db.t4g.micro: 128 MB

Atualização do MySQL 5.7 para o MySQL 8.4

Você não pode atualizar diretamente do MySQL 5.7 para o MySQL 8.4. Primeiro, você deve atualizar do MySQL 5.7 para o MySQL 8.0 e, em seguida, do MySQL 8.0 para o MySQL 8.4. Para ter mais informações, consulte Atualizações da versão principal do RDS para MySQL.

Compressão de páginas do InnoDB

A compressão de páginas do InnoDB não funciona com instâncias de banco de dados do Amazon RDS que têm um tamanho de bloco do sistema de arquivos de 16k porque o tamanho do bloco do sistema de arquivos deve ser menor que o tamanho da página do InnoDB. A partir de fevereiro de 2024, todas as instâncias de banco de dados recém-criadas têm um tamanho de bloco do sistema de arquivos de 16k, o que aumenta o throughput e diminui o consumo de IOPS durante a liberação de páginas.

Referência de procedimentos armazenados do RDS para MySQL

Esses tópicos descrevem procedimentos armazenados no sistema que estão disponíveis para instâncias do Amazon RDS que executam o mecanismo de banco de dados MySQL. O usuário primário deve executar esses procedimentos.

Tópicos

- Coletar e manter o histórico de status global
- Configurar, iniciar e interromper a replicação de logs binários (binlogs)
- Encerrar uma sessão ou consulta
- Gerenciar clusters ativos-ativos
- Gerenciar a replicação de várias fontes
- Replicar transações com GTIDs
- Alternar os logs de consulta
- Definir e mostrar a configuração de logs binários
- Aquecer o armazenamento em cache do InnoDB

Coletar e manter o histórico de status global

O Amazon RDS fornece um conjunto de procedimentos que gera snapshots dos valores dessas variáveis de status ao longo do tempo e as grava em uma tabela, juntamente com quaisquer alterações feitas desde o último snapshot. Essa infraestrutura é chamada de histórico de status global. Para obter mais informações, consulte Gerenciar o histórico de status global do RDS para MySQL.

Os procedimentos armazenados a seguir gerenciam a forma como o histórico de status global é coletado e mantido.

Tópicos

- mysql.rds_collect_global_status_history
- mysql.rds_disable_gsh_collector
- mysql.rds_disable_gsh_rotation
- mysql.rds_enable_gsh_collector
- mysql.rds_enable_gsh_rotation
- mysql.rds_rotate_global_status_history
- mysql.rds_set_gsh_collector
- mysql.rds_set_gsh_rotation

mysql.rds_collect_global_status_history

Gera um snapshot sob demanda para o histórico de status global.

Sintaxe

```
CALL mysql.rds_collect_global_status_history;
```

mysql.rds_disable_gsh_collector

Desativa os snapshots gerados pelo histórico de status global.

Sintaxe

```
CALL mysql.rds_disable_gsh_collector;
```

mysql.rds_disable_gsh_rotation

Desativa a rotação da tabela `mysql.global_status_history`.

Sintaxe

```
CALL mysql.rds_disable_gsh_rotation;
```

mysql.rds_enable_gsh_collector

Ativa o histórico de status global para gerar snapshots padrão em intervalos especificados por `rds_set_gsh_collector`.

Sintaxe

```
CALL mysql.rds_enable_gsh_collector;
```

mysql.rds_enable_gsh_rotation

Ativa a rotação do conteúdo da tabela `mysql.global_status_history` com o da `mysql.global_status_history_old` em intervalos especificados por `rds_set_gsh_rotation`.

Sintaxe

```
CALL mysql.rds_enable_gsh_rotation;
```

mysql.rds_rotate_global_status_history

Reveza o conteúdo da tabela `mysql.global_status_history` para o da `mysql.global_status_history_old` sob demanda.

Sintaxe

```
CALL mysql.rds_rotate_global_status_history;
```

mysql.rds_set_gsh_collector

Especifica o intervalo em minutos entre os snapshots gerados pelo histórico de status global.

Sintaxe

```
CALL mysql.rds_set_gsh_collector(intervalPeriod);
```

Parâmetros

intervalPeriod

O intervalo em minutos entre snapshots. O valor padrão é 5.

mysql.rds_set_gsh_rotation

Especifica o intervalo em dias entre os revezamentos da tabela `mysql.global_status_history`.

Sintaxe

```
CALL mysql.rds_set_gsh_rotation(intervalPeriod);
```

Parâmetros

intervalPeriod

O intervalo em dias entre os revezamentos da tabela. O valor padrão é 7.

Configurar, iniciar e interromper a replicação de logs binários (binlogs)

Os procedimentos armazenados a seguir controlam como as transações são replicadas de um banco de dados externo para o RDS para MySQL, ou do RDS para MySQL para um banco de dados externo.

Ao usar esses procedimentos armazenados para gerenciar a replicação com um usuário de replicação configurado com `caching_sha2_password`, você deve configurar o TLS especificando `SOURCE_SSL=1`. `caching_sha2_password` é o plugin de autenticação padrão do RDS para MySQL 8.4. Para obter mais informações, consulte Criptografia com SSL/TLS.

Para obter informações sobre como configurar, usar e gerenciar réplicas de leitura, consulte the section called “Réplicas de leitura do MySQL”.

Tópicos

- mysql.rds_next_master_log (RDS para MariaDB e RDS para MySQL versões principais 8.0 e anteriores)
- mysql.rds_next_source_log (RDS para MySQL versões principais 8.4 e posteriores)
- mysql.rds_reset_external_master (RDS para MariaDB e RDS para MySQL versões principais 8.0 e anteriores)
- mysql.rds_reset_external_source (RDS para MySQL versões principais 8.4 e posteriores)
- mysql.rds_set_external_master (RDS para MariaDB e RDS para MySQL versões principais 8.0 e anteriores)
- mysql.rds_set_external_source (RDS para MySQL versões principais 8.4 e posteriores)
- mysql.rds_set_external_master_with_auto_position (RDS para MySQL versões principais 8.0 e anteriores)
- mysql.rds_set_external_source_with_auto_position (RDS para MySQL versões principais 8.4 e posteriores)
- mysql.rds_set_external_master_with_delay (RDS para MariaDB e RDS para MySQL versões principais 8.0 e anteriores)
- mysql.rds_set_external_source_with_delay (RDS para MySQL versões principais 8.4 e superiores)
- mysql.rds_set_external_source_gtid_purged
- mysql.rds_set_master_auto_position (RDS para MySQL versões principais 8.0 e anteriores)
- mysql.rds_set_source_auto_position (RDS para MySQL versões principais 8.4 e posteriores)
- mysql.rds_set_source_delay

- mysql.rds_skip_repl_error
- mysql.rds_start_replication
- mysql.rds_start_replication_until
- mysql.rds_stop_replication

mysql.rds_next_master_log (RDS para MariaDB e RDS para MySQL versões principais 8.0 e anteriores)

Altera a posição do log da instância de banco de dados de origem para o início do próximo log binário na instância de banco de dados de origem. Use este procedimento somente se estiver recebendo o erro 1236 de E/S de replicação em uma réplica de leitura.

Sintaxe

```
CALL mysql.rds_next_master_log(
curr_master_log
);
```

Parâmetros

curr_master_log

O índice do arquivo de log mestre atual. Por exemplo, se o arquivo atual dor denominado `mysql-bin-changelog.012345`, o índice será 12345. Para determinar o nome do arquivo de log do mestre, execute o comando `SHOW REPLICA STATUS` e veja o campo `Master_Log_File`.

Observações de uso

O usuário mestre deve executar o procedimento `mysql.rds_next_master_log`.

> **Warning**
>
> Chame `mysql.rds_next_master_log` somente se a replicação falhar após um failover de uma instância de banco de dados multi-AZ que for a origem da replicação, e o campo `Last_IO_Errno` do `SHOW REPLICA STATUS` reportar o erro 1236 de E/S. Chamar `mysql.rds_next_master_log` pode resultar em perda de dados na réplica de leitura caso as transações na instância de origem não tenham sido gravadas no log binário no disco antes do evento de failover. Você pode reduzir a chance de que isso aconteça

configurando os parâmetros `sync_binlog` e `innodb_support_xa` da instância de origem como 1, embora isso possa reduzir o desempenho. Para obter mais informações, consulte Solucionar problemas de uma réplica de leitura do MySQL.

Exemplos

Suponha que a replicação falhe em uma réplica de leitura do RDS para MySQL. A execução de `SHOW REPLICA STATUS\G` na réplica de leitura retorna o seguinte resultado:

```
*************************** 1. row ***************************
             Replica_IO_State:
                  Source_Host: myhost.XXXXXXXXXXXXXX.rr-rrrr-1.rds.amazonaws.com
                  Source_User: MasterUser
                  Source_Port: 3306
                Connect_Retry: 10
              Source_Log_File: mysql-bin-changelog.012345
          Read_Source_Log_Pos: 1219393
               Relay_Log_File: relaylog.012340
                Relay_Log_Pos: 30223388
        Relay_Source_Log_File: mysql-bin-changelog.012345
           Replica_IO_Running: No
          Replica_SQL_Running: Yes
              Replicate_Do_DB:
          Replicate_Ignore_DB:
           Replicate_Do_Table:
       Replicate_Ignore_Table:
      Replicate_Wild_Do_Table:
  Replicate_Wild_Ignore_Table:
                   Last_Errno: 0
                   Last_Error:
                 Skip_Counter: 0
          Exec_Source_Log_Pos: 30223232
              Relay_Log_Space: 5248928866
              Until_Condition: None
               Until_Log_File:
                Until_Log_Pos: 0
           Source_SSL_Allowed: No
           Source_SSL_CA_File:
           Source_SSL_CA_Path:
              Source_SSL_Cert:
            Source_SSL_Cipher:
               Source_SSL_Key:
```

```
        Seconds_Behind_Master: NULL
Source_SSL_Verify_Server_Cert: No
                Last_IO_Errno: 1236
                Last_IO_Error: Got fatal error 1236 from master when reading data from
 binary log: 'Client requested master to start replication from impossible position;
 the first event 'mysql-bin-changelog.013406' at 1219393, the last event read from
 '/rdsdbdata/log/binlog/mysql-bin-changelog.012345' at 4, the last byte read from '/
rdsdbdata/log/binlog/mysql-bin-changelog.012345' at 4.'
               Last_SQL_Errno: 0
               Last_SQL_Error:
  Replicate_Ignore_Server_Ids:
             Source_Server_Id: 67285976
```

O campo `Last_IO_Errno` mostra que a instância está recebendo o erro 1236 de E/S. O campo `Master_Log_File` mostra que o nome do arquivo é `mysql-bin-changelog.012345`, o que significa que o índice de arquivos de log é `12345`. Para resolver o erro, chame `mysql.rds_next_master_log` com o seguinte parâmetro:

```
CALL mysql.rds_next_master_log(12345);
```

mysql.rds_next_source_log (RDS para MySQL versões principais 8.4 e posteriores)

Altera a posição do log da instância de banco de dados de origem para o início do próximo log binário na instância de banco de dados de origem. Use este procedimento somente se estiver recebendo o erro 1236 de E/S de replicação em uma réplica de leitura.

Sintaxe

```
CALL mysql.rds_next_source_log(
curr_source_log
);
```

Parâmetros

`curr_source_log`

O índice do arquivo de log de origem atual. Por exemplo, se o arquivo atual dor denominado `mysql-bin-changelog.012345`, o índice será 12345. Para determinar o nome do arquivo de log de origem atual, execute o comando `SHOW REPLICA STATUS` e veja o campo `Source_Log_File`.

Observações de uso

O usuário administrativo precisa executar o procedimento `mysql.rds_next_source_log`.

> **Warning**
>
> Chame `mysql.rds_next_source_log` somente se a replicação falhar após um failover de uma instância de banco de dados multi-AZ que for a origem da replicação, e o campo `Last_IO_Errno` do `SHOW REPLICA STATUS` reportar o erro 1236 de E/S. Chamar `mysql.rds_next_source_log` pode resultar em perda de dados na réplica de leitura caso as transações na instância de origem não tenham sido gravadas no log binário no disco antes do evento de failover. Você pode reduzir a chance de que isso aconteça configurando os parâmetros `sync_binlog` e `innodb_support_xa` da instância de origem como 1, embora isso possa reduzir o desempenho. Para obter mais informações, consulte Solucionar problemas de uma réplica de leitura do MySQL.

Exemplos

Suponha que a replicação falhe em uma réplica de leitura do RDS para MySQL. A execução de `SHOW REPLICA STATUS\G` na réplica de leitura retorna o seguinte resultado:

```
*************************** 1. row ***************************
             Replica_IO_State:
                  Source_Host: myhost.XXXXXXXXXXXXXX.rr-rrrr-1.rds.amazonaws.com
                  Source_User: MasterUser
                  Source_Port: 3306
                Connect_Retry: 10
              Source_Log_File: mysql-bin-changelog.012345
          Read_Source_Log_Pos: 1219393
               Relay_Log_File: relaylog.012340
                Relay_Log_Pos: 30223388
        Relay_Source_Log_File: mysql-bin-changelog.012345
           Replica_IO_Running: No
          Replica_SQL_Running: Yes
              Replicate_Do_DB:
          Replicate_Ignore_DB:
           Replicate_Do_Table:
       Replicate_Ignore_Table:
      Replicate_Wild_Do_Table:
  Replicate_Wild_Ignore_Table:
```

```
                   Last_Errno: 0
                   Last_Error:
                 Skip_Counter: 0
          Exec_Source_Log_Pos: 30223232
              Relay_Log_Space: 5248928866
              Until_Condition: None
               Until_Log_File:
                Until_Log_Pos: 0
           Source_SSL_Allowed: No
           Source_SSL_CA_File:
           Source_SSL_CA_Path:
              Source_SSL_Cert:
            Source_SSL_Cipher:
               Source_SSL_Key:
        Seconds_Behind_Source: NULL
Source_SSL_Verify_Server_Cert: No
                Last_IO_Errno: 1236
                Last_IO_Error: Got fatal error 1236 from source when reading data from
 binary log: 'Client requested source to start replication from impossible position;
 the first event 'mysql-bin-changelog.013406' at 1219393, the last event read from
 '/rdsdbdata/log/binlog/mysql-bin-changelog.012345' at 4, the last byte read from '/
rdsdbdata/log/binlog/mysql-bin-changelog.012345' at 4.'
               Last_SQL_Errno: 0
               Last_SQL_Error:
  Replicate_Ignore_Server_Ids:
             Source_Server_Id: 67285976
```

O campo `Last_IO_Errno` mostra que a instância está recebendo o erro 1236 de E/S. O campo `Source_Log_File` mostra que o nome do arquivo é `mysql-bin-changelog.012345`, o que significa que o índice de arquivos de log é `12345`. Para resolver o erro, chame `mysql.rds_next_source_log` com o seguinte parâmetro:

```
CALL mysql.rds_next_source_log(12345);
```

mysql.rds_reset_external_master (RDS para MariaDB e RDS para MySQL versões principais 8.0 e anteriores)

Reconfigura uma instância de banco de dados do RDS para MySQL para que não seja mais uma réplica de leitura de uma instância do MySQL executada fora do Amazon RDS.

> **Important**
>
> Para executar esse procedimento, `autocommit` deve estar habilitado. Para habilitá-lo, defina o parâmetro `autocommit` como 1. Para obter informações sobre como modificar parâmetros, consulte Modificar parâmetros em um grupo de parâmetros de banco de dados no Amazon RDS.

Sintaxe

```
CALL mysql.rds_reset_external_master;
```

Observações de uso

O usuário mestre deve executar o procedimento `mysql.rds_reset_external_master`. Esse procedimento deve ser executado na instância de banco de dados MySQL a ser removida como a réplica de leitura de uma instância do MySQL sendo executada externamente ao Amazon RDS.

> **Note**
>
> Recomendamos que você use réplicas de leitura para gerenciar a replicação entre duas instâncias de banco de dados do Amazon RDS, sempre que possível. Quando você o fizer, recomendamos o uso somente desse e de outros procedimentos armazenados relacionados a replicação. Essas práticas permitem o uso de topologias de replicação mais complexas entre as instâncias de banco de dados do Amazon RDS. Oferecemos esses procedimentos armazenados principalmente para permitir a replicação com instâncias do MySQL externas ao Amazon RDS. Para obter informações sobre como gerenciar a replicação entre instâncias de banco de dados do Amazon RDS, consulte Trabalhar com réplicas de leitura de instância de banco de dados.

Para obter mais informações sobre como usar a replicação para importar dados de uma instância do MySQL sendo executada externamente ao Amazon RDS, consulte Configurar a replicação da posição do arquivo de log binário com uma instância de origem externa.

mysql.rds_reset_external_source (RDS para MySQL versões principais 8.4 e posteriores)

Reconfigura uma instância de banco de dados do RDS para MySQL para que não seja mais uma réplica de leitura de uma instância do MySQL executada fora do Amazon RDS.

> **Important**
>
> Para executar esse procedimento, `autocommit` deve estar habilitado. Para habilitá-lo, defina o parâmetro `autocommit` como 1. Para obter informações sobre como modificar parâmetros, consulte Modificar parâmetros em um grupo de parâmetros de banco de dados no Amazon RDS.

Sintaxe

```
CALL mysql.rds_reset_external_source;
```

Observações de uso

O usuário administrativo precisa executar o procedimento `mysql.rds_reset_external_source`. Esse procedimento deve ser executado na instância de banco de dados MySQL a ser removida como a réplica de leitura de uma instância do MySQL sendo executada externamente ao Amazon RDS.

> **Note**
>
> Recomendamos que você use réplicas de leitura para gerenciar a replicação entre duas instâncias de banco de dados do Amazon RDS, sempre que possível. Quando você o fizer, recomendamos o uso somente desse e de outros procedimentos armazenados relacionados a replicação. Essas práticas permitem o uso de topologias de replicação mais complexas entre as instâncias de banco de dados do Amazon RDS. Oferecemos esses procedimentos armazenados principalmente para permitir a replicação com instâncias do MySQL externas ao Amazon RDS.
>
> Para obter informações sobre como gerenciar a replicação entre instâncias de banco de dados do Amazon RDS, consulte Trabalhar com réplicas de leitura de instância de banco de dados. Para obter mais informações sobre como usar a replicação para importar dados

de uma instância do MySQL sendo executada externamente ao Amazon RDS, consulte Configurar a replicação da posição do arquivo de log binário com uma instância de origem externa.

mysql.rds_set_external_master (RDS para MariaDB e RDS para MySQL versões principais 8.0 e anteriores)

Configura uma instância de banco de dados do RDS para MySQL para que seja uma réplica de leitura de uma instância do MySQL executada fora do Amazon RDS.

Important

Para executar esse procedimento, `autocommit` deve estar habilitado. Para habilitá-lo, defina o parâmetro `autocommit` como 1. Para obter informações sobre como modificar parâmetros, consulte Modificar parâmetros em um grupo de parâmetros de banco de dados no Amazon RDS.

Note

É possível usar o procedimento armazenado mysql.rds_set_external_master_with_delay (RDS para MariaDB e RDS para MySQL versões principais 8.0 e anteriores) para configurar uma instância de banco de dados de origem externa e uma replicação atrasada.

Sintaxe

```
CALL mysql.rds_set_external_master (
  host_name
  , host_port
  , replication_user_name
  , replication_user_password
  , mysql_binary_log_file_name
  , mysql_binary_log_file_location
  , ssl_encryption
);
```

Parâmetros

host_name

O nome de host ou o endereço IP da instância do MySQL executada externamente ao Amazon RDS que se tornará a instância de banco de dados de origem.

host_port

A porta usada para executar a instância do MySQL executada externamente ao Amazon RDS a ser configurada como a instância de banco de dados de origem. Se sua configuração de rede inclui replicação de porta Secure Shell (SSH) que converte o número da porta, especifique o número da porta exposto pelo SSH.

replication_user_name

O ID de um usuário com as permissões `REPLICATION CLIENT` e `REPLICATION SLAVE` na instância do MySQL executada externamente ao Amazon RDS. Recomendamos que você forneça uma conta que seja usada unicamente para a replicação com a instância externa.

replication_user_password

A senha do ID de usuário especificada em `replication_user_name`.

mysql_binary_log_file_name

O nome do log binário na instância de banco de dados de origem que contém as informações de replicação.

mysql_binary_log_file_location

O local no log binário `mysql_binary_log_file_name` no qual a replicação começa a ler as informações de replicação.

É possível determinar o nome e a localização do arquivo de binlog executando `SHOW MASTER STATUS` na instância do banco de dados de origem.

ssl_encryption

Um valor que especifica se a criptografia do Secure Sockets Layer (SSL) será usada na conexão de replicação. 1 especifica para usar criptografia de SSL, 0 especifica para não usar criptografia. O padrão é 0.

> **Note**
>
> A opção `MASTER_SSL_VERIFY_SERVER_CERT` não tem suporte. Essa opção é definida como 0, o que significa que a conexão é criptografada, mas os certificados não são verificados.

Observações de uso

O usuário mestre deve executar o procedimento `mysql.rds_set_external_master`. Esse procedimento deve ser executado na instância de banco de dados MySQL que será configurada como a réplica de leitura de uma instância do MySQL externa ao Amazon RDS.

Antes de executar `mysql.rds_set_external_master`, é necessário configurar a instância do MySQL executada externamente ao Amazon RDS para ser uma instância de banco de dados de origem. Para conectar-se à instância do MySQL sendo executada externamente ao Amazon RDS, você deve especificar valores de `replication_user_name` e `replication_user_password` que indicam um usuário de replicação com permissões de `REPLICATION CLIENT` e `REPLICATION SLAVE` na instância externa do MySQL.

Como configurar uma instância externa do MySQL como uma instância de banco de dados de origem

1. Usando o cliente do MySQL de sua escolha, conecte-se à instância externa do MySQL e crie uma conta de usuário a ser usada para a replicação. Veja um exemplo a seguir.

   MySQL 5.7

   ```
   CREATE USER 'repl_user'@'mydomain.com' IDENTIFIED BY 'password';
   ```

   MySQL 8.0

   ```
   CREATE USER 'repl_user'@'mydomain.com' IDENTIFIED WITH mysql_native_password BY
    'password';
   ```

> **Note**
>
> Especifique uma senha diferente do prompt mostrado aqui como prática recomendada de segurança.

2. Na instância externa do MySQL, conceda privilégios de `REPLICATION CLIENT` e `REPLICATION SLAVE` para seu usuário de replicação. O exemplo a seguir concede privilégios de `REPLICATION CLIENT` e `REPLICATION SLAVE` ao usuário 'repl_user' em todos os bancos de dados de seu domínio.

   MySQL 5.7

   ```
   GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'mydomain.com'
    IDENTIFIED BY 'password';
   ```

   MySQL 8.0

   ```
   GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'mydomain.com';
   ```

Para usar a replicação criptografada, configure a instância de banco de dados de origem para usar conexões SSL.

> **Note**
>
> Recomendamos que você use réplicas de leitura para gerenciar a replicação entre duas instâncias de banco de dados do Amazon RDS, sempre que possível. Quando você o fizer, recomendamos o uso somente desse e de outros procedimentos armazenados relacionados a replicação. Essas práticas permitem o uso de topologias de replicação mais complexas entre as instâncias de banco de dados do Amazon RDS. Oferecemos esses procedimentos armazenados principalmente para permitir a replicação com instâncias do MySQL externas ao Amazon RDS. Para obter informações sobre como gerenciar a replicação entre instâncias de banco de dados do Amazon RDS, consulte Trabalhar com réplicas de leitura de instância de banco de dados.

Depois de chamar `mysql.rds_set_external_master` para configurar uma instância de banco de dados do Amazon RDS, é possível chamar mysql.rds_start_replication na réplica de leitura para iniciar o processo de replicação. É possível chamar mysql.rds_reset_external_master (RDS para MariaDB e RDS para MySQL versões principais 8.0 e anteriores) para remover a configuração da réplica de leitura.

Quando `mysql.rds_set_external_master` é chamado, o Amazon RDS grava a hora, o usuário e uma ação do `set master` nas tabelas `mysql.rds_history` e `mysql.rds_replication_status`.

Exemplos

Ao executar em uma instância de banco de dados MySQL, o exemplo a seguir configura a instância do banco de dados para ser uma réplica de leitura de uma instância do MySQL que é executada externamente ao Amazon RDS.

```
call mysql.rds_set_external_master(
  'Externaldb.some.com',
  3306,
  'repl_user',
  'password',
  'mysql-bin-changelog.0777',
  120,
  1);
```

mysql.rds_set_external_source (RDS para MySQL versões principais 8.4 e posteriores)

Configura uma instância de banco de dados do RDS para MySQL para que seja uma réplica de leitura de uma instância do MySQL executada fora do Amazon RDS.

> **Important**
>
> Para executar esse procedimento, `autocommit` deve estar habilitado. Para habilitá-lo, defina o parâmetro `autocommit` como 1. Para obter informações sobre como modificar parâmetros, consulte Modificar parâmetros em um grupo de parâmetros de banco de dados no Amazon RDS.

Sintaxe

```
CALL mysql.rds_set_external_source (
  host_name
  , host_port
  , replication_user_name
  , replication_user_password
  , mysql_binary_log_file_name
  , mysql_binary_log_file_location
  , ssl_encryption
);
```

Parâmetros

host_name

O nome de host ou o endereço IP da instância do MySQL executada externamente ao Amazon RDS que se tornará a instância de banco de dados de origem.

host_port

A porta usada para executar a instância do MySQL executada externamente ao Amazon RDS a ser configurada como a instância de banco de dados de origem. Se sua configuração de rede inclui replicação de porta Secure Shell (SSH) que converte o número da porta, especifique o número da porta exposto pelo SSH.

replication_user_name

O ID de um usuário com as permissões `REPLICATION CLIENT` e `REPLICATION SLAVE` na instância do MySQL executada externamente ao Amazon RDS. Recomendamos que você forneça uma conta que seja usada unicamente para a replicação com a instância externa.

replication_user_password

A senha do ID de usuário especificada em `replication_user_name`.

mysql_binary_log_file_name

O nome do log binário na instância de banco de dados de origem que contém as informações de replicação.

mysql_binary_log_file_location

O local no log binário `mysql_binary_log_file_name` no qual a replicação começa a ler as informações de replicação.

É possível determinar o nome e a localização do arquivo de binlog executando `SHOW MASTER STATUS` na instância do banco de dados de origem.

`ssl_encryption`

Um valor que especifica se a criptografia do Secure Sockets Layer (SSL) será usada na conexão de replicação. 1 especifica para usar criptografia de SSL, 0 especifica para não usar criptografia. O padrão é 0.

> **Note**
>
> A opção `SOURCE_SSL_VERIFY_SERVER_CERT` não tem suporte. Essa opção é definida como 0, o que significa que a conexão é criptografada, mas os certificados não são verificados.

Observações de uso

O usuário administrativo precisa executar o procedimento `mysql.rds_set_external_source`. Esse procedimento deve ser executado na instância de banco de dados do RDS para MySQL que será configurada como a réplica de leitura de uma instância do MySQL externa ao Amazon RDS.

Antes de executar `mysql.rds_set_external_source`, é necessário configurar a instância do MySQL executada externamente ao Amazon RDS para ser uma instância de banco de dados de origem. Para conectar-se à instância do MySQL sendo executada externamente ao Amazon RDS, você deve especificar valores de `replication_user_name` e `replication_user_password` que indicam um usuário de replicação com permissões de `REPLICATION CLIENT` e `REPLICATION SLAVE` na instância externa do MySQL.

Como configurar uma instância externa do MySQL como uma instância de banco de dados de origem

1. Usando o cliente do MySQL de sua escolha, conecte-se à instância externa do MySQL e crie uma conta de usuário a ser usada para a replicação. Veja um exemplo a seguir.

```
CREATE USER 'repl_user'@'mydomain.com' IDENTIFIED BY 'password';
```

> **Note**
>
> Especifique uma senha diferente do prompt mostrado aqui como prática recomendada de segurança.

2. Na instância externa do MySQL, conceda privilégios de `REPLICATION CLIENT` e `REPLICATION SLAVE` para seu usuário de replicação. O exemplo a seguir concede privilégios de `REPLICATION CLIENT` e `REPLICATION SLAVE` ao usuário 'repl_user' em todos os bancos de dados de seu domínio.

```
GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'mydomain.com';
```

Para usar a replicação criptografada, configure a instância de banco de dados de origem para usar conexões SSL. Além disso, importe o certificado de autoridade de certificação, o certificado e a chave de clientes na instância de banco de dados ou no cluster de banco de dados usando o procedimento mysql.rds_import_binlog_ssl_material.

> **Note**
>
> Recomendamos que você use réplicas de leitura para gerenciar a replicação entre duas instâncias de banco de dados do Amazon RDS, sempre que possível. Quando você o fizer, recomendamos o uso somente desse e de outros procedimentos armazenados relacionados a replicação. Essas práticas permitem o uso de topologias de replicação mais complexas entre as instâncias de banco de dados do Amazon RDS. Oferecemos esses procedimentos armazenados principalmente para permitir a replicação com instâncias do MySQL externas ao Amazon RDS. Para obter informações sobre como gerenciar a replicação entre instâncias de banco de dados do Amazon RDS, consulte Trabalhar com réplicas de leitura de instância de banco de dados.

Depois de chamar `mysql.rds_set_external_source` para configurar uma instância de banco de dados do RDS para MySQL, é possível chamar mysql.rds_start_replication na réplica de leitura para iniciar o processo de replicação. É possível chamar mysql.rds_reset_external_source (RDS para MySQL versões principais 8.4 e posteriores) para remover a configuração da réplica de leitura.

Quando `mysql.rds_set_external_source` é chamado, o Amazon RDS grava a hora, o usuário e uma ação do `set master` nas tabelas `mysql.rds_history` e `mysql.rds_replication_status`.

Exemplos

Quando executado em uma instância de banco de dados do RDS para MySQL, o exemplo a seguir configura a instância de banco de dados para ser uma réplica de leitura de uma instância do MySQL que é executada externamente ao Amazon RDS.

```
call mysql.rds_set_external_source(
  'Externaldb.some.com',
  3306,
  'repl_user',
  'password',
  'mysql-bin-changelog.0777',
  120,
  1);
```

mysql.rds_set_external_master_with_auto_position (RDS para MySQL versões principais 8.0 e anteriores)

Configura uma instância de banco de dados do RDS para MySQL para ser uma réplica de leitura de uma instância do MySQL executada externamente em relação ao Amazon RDS. Esse procedimento também configura a replicação atrasada e a replicação baseada em identificadores de transações globais (GTIDs).

> **Important**
>
> Para executar esse procedimento, `autocommit` deve estar habilitado. Para habilitá-lo, defina o parâmetro `autocommit` como 1. Para obter informações sobre como modificar parâmetros, consulte Modificar parâmetros em um grupo de parâmetros de banco de dados no Amazon RDS.

Sintaxe

```
CALL mysql.rds_set_external_master_with_auto_position (
  host_name
```

```
  , host_port
  , replication_user_name
  , replication_user_password
  , ssl_encryption
  , delay
);
```

Parâmetros

host_name

O nome de host ou o endereço IP da instância do MySQL executada externamente ao Amazon RDS que se tornará a instância de banco de dados de origem.

host_port

A porta usada para executar a instância do MySQL executada externamente ao Amazon RDS a ser configurada como a instância de banco de dados de origem. Se sua configuração de rede inclui replicação de porta Secure Shell (SSH) que converte o número da porta, especifique o número da porta exposto pelo SSH.

replication_user_name

O ID de um usuário com as permissões `REPLICATION CLIENT` e `REPLICATION SLAVE` na instância do MySQL executada externamente ao Amazon RDS. Recomendamos que você forneça uma conta que seja usada unicamente para a replicação com a instância externa.

replication_user_password

A senha do ID de usuário especificada em `replication_user_name`.

ssl_encryption

Um valor que especifica se a criptografia do Secure Sockets Layer (SSL) será usada na conexão de replicação. 1 especifica para usar criptografia de SSL, 0 especifica para não usar criptografia. O padrão é 0.

> **Note**
>
> A opção `MASTER_SSL_VERIFY_SERVER_CERT` não tem suporte. Essa opção é definida como 0, o que significa que a conexão é criptografada, mas os certificados não são verificados.

delay

O número mínimo de segundos para atrasar a replicação da instância de banco de dados de origem.

O limite para esse parâmetro é de um dia (86.400 segundos).

Observações de uso

O usuário mestre deve executar o procedimento `mysql.rds_set_external_master_with_auto_position`. Esse procedimento deve ser executado na instância de banco de dados MySQL que será configurada como a réplica de leitura de uma instância do MySQL externa ao Amazon RDS.

Esse procedimento é compatível com todas as versões 5.7 do RDS para MySQL e com o RDS para MySQL 8.0.26 e versões 8.0 posteriores.

Antes de executar `mysql.rds_set_external_master_with_auto_position`, é necessário configurar a instância do MySQL executada externamente ao Amazon RDS para ser uma instância de banco de dados de origem. Para conectar-se à instância do MySQL que é externa ao Amazon RDS, você deve especificar os valores de `replication_user_name` e `replication_user_password`. Esses valores devem indicar um usuário de replicação que tenha as permissões `REPLICATION CLIENT` e `REPLICATION SLAVE` na instância externa do MySQL.

Como configurar uma instância externa do MySQL como uma instância de banco de dados de origem

1. Usando o cliente do MySQL de sua escolha, conecte-se à instância externa do MySQL e crie uma conta de usuário a ser usada para a replicação. Veja um exemplo a seguir.

   ```
   CREATE USER 'repl_user'@'mydomain.com' IDENTIFIED BY 'SomePassW0rd'
   ```

2. Na instância externa do MySQL, conceda privilégios de `REPLICATION CLIENT` e `REPLICATION SLAVE` para seu usuário de replicação. O exemplo a seguir concede privilégios de `REPLICATION CLIENT` e `REPLICATION SLAVE` ao usuário `'repl_user'` em todos os bancos de dados de seu domínio.

   ```
   GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'mydomain.com'
   IDENTIFIED BY 'SomePassW0rd'
   ```

Para obter mais informações, consulte Configurar a replicação da posição do arquivo de log binário com uma instância de origem externa.

> **Note**
>
> Recomendamos que você use réplicas de leitura para gerenciar a replicação entre duas instâncias de banco de dados do Amazon RDS, sempre que possível. Quando você o fizer, recomendamos o uso somente desse e de outros procedimentos armazenados relacionados a replicação. Essas práticas permitem o uso de topologias de replicação mais complexas entre as instâncias de banco de dados do Amazon RDS. Oferecemos esses procedimentos armazenados principalmente para permitir a replicação com instâncias do MySQL externas ao Amazon RDS. Para obter informações sobre como gerenciar a replicação entre instâncias de banco de dados do Amazon RDS, consulte Trabalhar com réplicas de leitura de instância de banco de dados.

Antes de chamar `mysql.rds_set_external_master_with_auto_position`, chame the section called “mysql.rds_set_external_source_gtid_purged” para definir a variável de sistema `gtid_purged` com um intervalo GTID especificado de uma fonte externa.

Depois de chamar `mysql.rds_set_external_master_with_auto_position` para configurar uma instância de banco de dados do Amazon RDS, é possível chamar mysql.rds_start_replication na réplica de leitura para iniciar o processo de replicação. É possível chamar mysql.rds_reset_external_master (RDS para MariaDB e RDS para MySQL versões principais 8.0 e anteriores) para remover a configuração da réplica de leitura.

Ao chamar `mysql.rds_set_external_master_with_auto_position`, o Amazon RDS registra a hora, o usuário e uma ação do `set master` nas tabelas `mysql.rds_history` e `mysql.rds_replication_status`.

Para a recuperação de desastres, você pode usar esse procedimento com o procedimento armazenado mysql.rds_start_replication_until ou o mysql.rds_start_replication_until_gtid. Para efetuar roll forward de alterações para uma réplica de leitura atrasada no momento imediatamente antes de um desastre, é possível executar o procedimento `mysql.rds_set_external_master_with_auto_position`. Depois que o procedimento `mysql.rds_start_replication_until_gtid` interromper a replicação, é possível promover a réplica de leitura para ser a nova instância de banco de dados primária usando as instruções em Promoção de uma réplica de leitura a uma instância de banco de dados autônoma.

Para usar o procedimento `mysql.rds_rds_start_replication_until_gtid`, a replicação baseada em GTID deve estar habilitada. Para ignorar uma transação específica baseada em GTID que seja conhecida por causar desastres, use o procedimento armazenado mysql.rds_skip_transaction_with_gtid. Para obter mais informações sobre como trabalhar com a replicação baseada em GTID, consulte Usar a replicação baseada em GTID.

Exemplos

Ao executar em uma instância de banco de dados MySQL, o exemplo a seguir configura a instância do banco de dados para ser uma réplica de leitura de uma instância do MySQL que é executada externamente ao Amazon RDS. Ele define o atraso mínimo de replicação como uma hora (3.600 segundos) na instância de banco de dados MySQL. Uma alteração da instância do banco de dados de origem MySQL executada externamente ao Amazon RDS não é aplicada à réplica de leitura da instância de banco de dados MySQL por pelo menos uma hora.

```
call mysql.rds_set_external_master_with_auto_position(
  'Externaldb.some.com',
  3306,
  'repl_user',
  'SomePassW0rd',
  1,
  3600);
```

mysql.rds_set_external_source_with_auto_position (RDS para MySQL versões principais 8.4 e posteriores)

Configura uma instância de banco de dados do RDS para MySQL para ser uma réplica de leitura de uma instância do MySQL executada externamente em relação ao Amazon RDS. Esse procedimento também configura a replicação atrasada e a replicação baseada em identificadores de transações globais (GTIDs).

> **Important**
>
> Para executar esse procedimento, `autocommit` deve estar habilitado. Para habilitá-lo, defina o parâmetro `autocommit` como 1. Para obter informações sobre como modificar parâmetros, consulte Modificar parâmetros em um grupo de parâmetros de banco de dados no Amazon RDS.

Sintaxe

```
CALL mysql.rds_set_external_source_with_auto_position (
  host_name
  , host_port
  , replication_user_name
  , replication_user_password
  , ssl_encryption
  , delay
);
```

Parâmetros

host_name

O nome de host ou o endereço IP da instância do MySQL executada externamente ao Amazon RDS que se tornará a instância de banco de dados de origem.

host_port

A porta usada para executar a instância do MySQL executada externamente ao Amazon RDS a ser configurada como a instância de banco de dados de origem. Se sua configuração de rede inclui replicação de porta Secure Shell (SSH) que converte o número da porta, especifique o número da porta exposto pelo SSH.

replication_user_name

O ID de um usuário com as permissões `REPLICATION CLIENT` e `REPLICATION SLAVE` na instância do MySQL executada externamente ao Amazon RDS. Recomendamos que você forneça uma conta que seja usada unicamente para a replicação com a instância externa.

replication_user_password

A senha do ID de usuário especificada em `replication_user_name`.

ssl_encryption

Um valor que especifica se a criptografia do Secure Sockets Layer (SSL) será usada na conexão de replicação. 1 especifica para usar criptografia de SSL, 0 especifica para não usar criptografia. O padrão é 0.

> **Note**
>
> A opção `SOURCE_SSL_VERIFY_SERVER_CERT` não tem suporte. Essa opção é definida como 0, o que significa que a conexão é criptografada, mas os certificados não são verificados.

`delay`

O número mínimo de segundos para atrasar a replicação da instância de banco de dados de origem.

O limite para esse parâmetro é de um dia (86.400 segundos).

Observações de uso

O usuário administrativo precisa executar o procedimento `mysql.rds_set_external_source_with_auto_position`. Esse procedimento deve ser executado na instância de banco de dados MySQL que será configurada como a réplica de leitura de uma instância do MySQL externa ao Amazon RDS.

Antes de executar `mysql.rds_set_external_source_with_auto_position`, é necessário configurar a instância do MySQL executada externamente ao Amazon RDS para ser uma instância de banco de dados de origem. Para conectar-se à instância do MySQL que é externa ao Amazon RDS, você deve especificar os valores de `replication_user_name` e `replication_user_password`. Esses valores devem indicar um usuário de replicação que tenha as permissões `REPLICATION CLIENT` e `REPLICATION SLAVE` na instância externa do MySQL.

Como configurar uma instância externa do MySQL como uma instância de banco de dados de origem

1. Usando o cliente do MySQL de sua escolha, conecte-se à instância externa do MySQL e crie uma conta de usuário a ser usada para a replicação. Veja um exemplo a seguir.

   ```
   CREATE USER 'repl_user'@'mydomain.com' IDENTIFIED BY 'SomePassW0rd'
   ```

2. Na instância externa do MySQL, conceda privilégios de `REPLICATION CLIENT` e `REPLICATION SLAVE` para seu usuário de replicação. O exemplo a seguir concede privilégios

de `REPLICATION CLIENT` e `REPLICATION SLAVE` ao usuário `'repl_user'` em todos os bancos de dados de seu domínio.

```
GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'mydomain.com'
IDENTIFIED BY 'SomePassW0rd'
```

Para obter mais informações, consulte Configurar a replicação da posição do arquivo de log binário com uma instância de origem externa.

> **Note**
>
> Recomendamos que você use réplicas de leitura para gerenciar a replicação entre duas instâncias de banco de dados do Amazon RDS, sempre que possível. Quando você o fizer, recomendamos o uso somente desse e de outros procedimentos armazenados relacionados a replicação. Essas práticas permitem o uso de topologias de replicação mais complexas entre as instâncias de banco de dados do Amazon RDS. Oferecemos esses procedimentos armazenados principalmente para permitir a replicação com instâncias do MySQL externas ao Amazon RDS. Para obter informações sobre como gerenciar a replicação entre instâncias de banco de dados do Amazon RDS, consulte Trabalhar com réplicas de leitura de instância de banco de dados.

Antes de chamar `mysql.rds_set_external_source_with_auto_position`, chame the section called “mysql.rds_set_external_source_gtid_purged” para definir a variável de sistema `gtid_purged` com um intervalo GTID especificado de uma fonte externa.

Depois de chamar `mysql.rds_set_external_source_with_auto_position` para configurar uma instância de banco de dados do Amazon RDS, é possível chamar mysql.rds_start_replication na réplica de leitura para iniciar o processo de replicação. É possível chamar mysql.rds_reset_external_source (RDS para MySQL versões principais 8.4 e posteriores) para remover a configuração da réplica de leitura.

Ao chamar `mysql.rds_set_external_source_with_auto_position`, o Amazon RDS registra a hora, o usuário e uma ação do `set master` nas tabelas `mysql.rds_history` e `mysql.rds_replication_status`.

Para a recuperação de desastres, você pode usar esse procedimento com o procedimento armazenado mysql.rds_start_replication_until ou o mysql.rds_start_replication_until_gtid.

Para efetuar roll forward de alterações para uma réplica de leitura atrasada no momento imediatamente antes de um desastre, é possível executar o procedimento `mysql.rds_set_external_source_with_auto_position`. Depois que o procedimento `mysql.rds_start_replication_until_gtid` interromper a replicação, é possível promover a réplica de leitura para ser a nova instância de banco de dados primária usando as instruções em Promoção de uma réplica de leitura a uma instância de banco de dados autônoma.

Para usar o procedimento `mysql.rds_rds_start_replication_until_gtid`, a replicação baseada em GTID deve estar habilitada. Para ignorar uma transação específica baseada em GTID que seja conhecida por causar desastres, use o procedimento armazenado mysql.rds_skip_transaction_with_gtid. Para obter mais informações sobre como trabalhar com a replicação baseada em GTID, consulte Usar a replicação baseada em GTID.

Exemplos

Ao executar em uma instância de banco de dados MySQL, o exemplo a seguir configura a instância do banco de dados para ser uma réplica de leitura de uma instância do MySQL que é executada externamente ao Amazon RDS. Ele define o atraso mínimo de replicação como uma hora (3.600 segundos) na instância de banco de dados MySQL. Uma alteração da instância do banco de dados de origem MySQL executada externamente ao Amazon RDS não é aplicada à réplica de leitura da instância de banco de dados MySQL por pelo menos uma hora.

```
call mysql.rds_set_external_source_with_auto_position(
  'Externaldb.some.com',
  3306,
  'repl_user',
  'SomePassW0rd',
  1,
  3600);
```

mysql.rds_set_external_master_with_delay (RDS para MariaDB e RDS para MySQL versões principais 8.0 e anteriores)

Configura uma instância de banco de dados do RDS para MySQL para ser uma réplica de leitura de uma instância do MySQL executada externamente em relação ao Amazon RDS e configura a replicação atrasada.

> **Important**
>
> Para executar esse procedimento, `autocommit` deve estar habilitado. Para habilitá-lo, defina o parâmetro `autocommit` como 1. Para obter informações sobre como modificar parâmetros, consulte Modificar parâmetros em um grupo de parâmetros de banco de dados no Amazon RDS.

Sintaxe

```
CALL mysql.rds_set_external_master_with_delay(
  host_name
  , host_port
  , replication_user_name
  , replication_user_password
  , mysql_binary_log_file_name
  , mysql_binary_log_file_location
  , ssl_encryption
  , delay
);
```

Parâmetros

host_name

O nome de host ou o endereço IP da instância do MySQL executada externamente ao Amazon RDS que se tornará a instância de banco de dados de origem.

host_port

A porta usada para executar a instância do MySQL executada externamente ao Amazon RDS a ser configurada como a instância de banco de dados de origem. Se sua configuração de rede inclui replicação de porta SSH que converte o número da porta, especifique o número da porta exposto pelo SSH.

replication_user_name

O ID de um usuário com as permissões `REPLICATION CLIENT` e `REPLICATION SLAVE` na instância do MySQL executada externamente ao Amazon RDS. Recomendamos que você forneça uma conta que seja usada unicamente para a replicação com a instância externa.

replication_user_password

A senha do ID de usuário especificada em `replication_user_name`.

mysql_binary_log_file_name

O nome do log binário na instância de banco de dados de origem contém as informações de replicação.

mysql_binary_log_file_location

O local no log binário `mysql_binary_log_file_name` no qual a replicação começará a ler as informações de replicação.

É possível determinar o nome e a localização do arquivo de binlog executando `SHOW MASTER STATUS` na instância do banco de dados de origem.

ssl_encryption

Um valor que especifica se a criptografia do Secure Sockets Layer (SSL) será usada na conexão de replicação. 1 especifica para usar criptografia de SSL, 0 especifica para não usar criptografia. O padrão é 0.

> **Note**
>
> A opção `MASTER_SSL_VERIFY_SERVER_CERT` não tem suporte. Essa opção é definida como 0, o que significa que a conexão é criptografada, mas os certificados não são verificados.

delay

O número mínimo de segundos para atrasar a replicação da instância de banco de dados de origem.

O limite para esse parâmetro é de um dia (86.400 segundos).

Observações de uso

O usuário mestre deve executar o procedimento `mysql.rds_set_external_master_with_delay`. Esse procedimento deve ser executado na

instância de banco de dados MySQL que será configurada como a réplica de leitura de uma instância do MySQL externa ao Amazon RDS.

Antes de executar `mysql.rds_set_external_master_with_delay`, é necessário configurar a instância do MySQL executada externamente ao Amazon RDS para ser uma instância de banco de dados de origem. Para conectar-se à instância do MySQL que é externa ao Amazon RDS, você deve especificar os valores de `replication_user_name` e `replication_user_password`. Esses valores devem indicar um usuário de replicação que tenha as permissões `REPLICATION CLIENT` e `REPLICATION SLAVE` na instância externa do MySQL.

Como configurar uma instância externa do MySQL como uma instância de banco de dados de origem

1. Usando o cliente do MySQL de sua escolha, conecte-se à instância externa do MySQL e crie uma conta de usuário a ser usada para a replicação. Veja um exemplo a seguir.

   ```
   CREATE USER 'repl_user'@'mydomain.com' IDENTIFIED BY 'SomePassW0rd'
   ```

2. Na instância externa do MySQL, conceda privilégios de `REPLICATION CLIENT` e `REPLICATION SLAVE` para seu usuário de replicação. O exemplo a seguir concede privilégios de `REPLICATION CLIENT` e `REPLICATION SLAVE` ao usuário `'repl_user'` em todos os bancos de dados de seu domínio.

   ```
   GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'mydomain.com'
   IDENTIFIED BY 'SomePassW0rd'
   ```

Para obter mais informações, consulte Configurar a replicação da posição do arquivo de log binário com uma instância de origem externa.

> **Note**
>
> Recomendamos que você use réplicas de leitura para gerenciar a replicação entre duas instâncias de banco de dados do Amazon RDS, sempre que possível. Quando você o fizer, recomendamos o uso somente desse e de outros procedimentos armazenados relacionados a replicação. Essas práticas permitem o uso de topologias de replicação mais complexas entre as instâncias de banco de dados do Amazon RDS. Oferecemos esses procedimentos armazenados principalmente para permitir a replicação com instâncias do MySQL externas ao Amazon RDS. Para obter informações sobre como gerenciar a replicação entre instâncias

de banco de dados do Amazon RDS, consulte Trabalhar com réplicas de leitura de instância de banco de dados.

Depois de chamar `mysql.rds_set_external_master_with_delay` para configurar uma instância de banco de dados do Amazon RDS, é possível chamar mysql.rds_start_replication na réplica de leitura para iniciar o processo de replicação. É possível chamar mysql.rds_reset_external_master (RDS para MariaDB e RDS para MySQL versões principais 8.0 e anteriores) para remover a configuração da réplica de leitura.

Ao chamar `mysql.rds_set_external_master_with_delay`, o Amazon RDS registra a hora, o usuário e uma ação do `set master` nas tabelas `mysql.rds_history` e `mysql.rds_replication_status`.

Para a recuperação de desastres, você pode usar esse procedimento com o procedimento armazenado mysql.rds_start_replication_until ou o mysql.rds_start_replication_until_gtid. Para efetuar roll forward de alterações para uma réplica de leitura atrasada no momento imediatamente antes de um desastre, é possível executar o procedimento `mysql.rds_set_external_master_with_delay`. Depois que o procedimento `mysql.rds_start_replication_until` interromper a replicação, é possível promover a réplica de leitura para ser a nova instância de banco de dados primária usando as instruções em Promoção de uma réplica de leitura a uma instância de banco de dados autônoma.

Para usar o procedimento `mysql.rds_rds_start_replication_until_gtid`, a replicação baseada em GTID deve estar habilitada. Para ignorar uma transação específica baseada em GTID que seja conhecida por causar desastres, use o procedimento armazenado mysql.rds_skip_transaction_with_gtid. Para obter mais informações sobre como trabalhar com a replicação baseada em GTID, consulte Usar a replicação baseada em GTID.

O procedimento `mysql.rds_set_external_master_with_delay` está disponível nestas versões do RDS para MySQL:

- MySQL 8.0.26 e versões 8.0 posteriores
- Todas as versões 5.7

Exemplos

Ao executar em uma instância de banco de dados MySQL, o exemplo a seguir configura a instância do banco de dados para ser uma réplica de leitura de uma instância do MySQL que é executada

externamente ao Amazon RDS. Ele define o atraso mínimo de replicação como uma hora (3.600 segundos) na instância de banco de dados MySQL. Uma alteração da instância do banco de dados de origem MySQL executada externamente ao Amazon RDS não é aplicada à réplica de leitura da instância de banco de dados MySQL por pelo menos uma hora.

```
call mysql.rds_set_external_master_with_delay(
  'Externaldb.some.com',
  3306,
  'repl_user',
  'SomePassW0rd',
  'mysql-bin-changelog.000777',
  120,
  1,
  3600);
```

mysql.rds_set_external_source_with_delay (RDS para MySQL versões principais 8.4 e superiores)

Configura uma instância de banco de dados do RDS para MySQL para ser uma réplica de leitura de uma instância do MySQL executada externamente em relação ao Amazon RDS e configura a replicação atrasada.

> **Important**
>
> Para executar esse procedimento, `autocommit` deve estar habilitado. Para habilitá-lo, defina o parâmetro `autocommit` como 1. Para obter informações sobre como modificar parâmetros, consulte Modificar parâmetros em um grupo de parâmetros de banco de dados no Amazon RDS.

Sintaxe

```
CALL mysql.rds_set_external_source_with_delay (
  host_name
  , host_port
  , replication_user_name
  , replication_user_password
  , mysql_binary_log_file_name
  , mysql_binary_log_file_location
  , ssl_encryption
```

```
  , delay
);
```

Parâmetros

host_name

O nome de host ou o endereço IP da instância do MySQL executada externamente ao Amazon RDS que se tornará a instância de banco de dados de origem.

host_port

A porta usada para executar a instância do MySQL executada externamente ao Amazon RDS a ser configurada como a instância de banco de dados de origem. Se sua configuração de rede inclui replicação de porta SSH que converte o número da porta, especifique o número da porta exposto pelo SSH.

replication_user_name

O ID de um usuário com as permissões `REPLICATION CLIENT` e `REPLICATION SLAVE` na instância do MySQL executada externamente ao Amazon RDS. Recomendamos que você forneça uma conta que seja usada unicamente para a replicação com a instância externa.

replication_user_password

A senha do ID de usuário especificada em `replication_user_name`.

mysql_binary_log_file_name

O nome do log binário na instância de banco de dados de origem contém as informações de replicação.

mysql_binary_log_file_location

O local no log binário `mysql_binary_log_file_name` no qual a replicação começará a ler as informações de replicação.

É possível determinar o nome e a localização do arquivo de binlog executando `SHOW MASTER STATUS` na instância do banco de dados de origem.

ssl_encryption

Um valor que especifica se a criptografia do Secure Sockets Layer (SSL) será usada na conexão de replicação. 1 especifica para usar criptografia de SSL, 0 especifica para não usar criptografia. O padrão é 0.

> **Note**
>
> A opção SOURCE_SSL_VERIFY_SERVER_CERT não tem suporte. Essa opção é definida como 0, o que significa que a conexão é criptografada, mas os certificados não são verificados.

delay

O número mínimo de segundos para atrasar a replicação da instância de banco de dados de origem.

O limite para esse parâmetro é de um dia (86.400 segundos).

Observações de uso

O usuário administrativo precisa executar o procedimento `mysql.rds_set_external_source_with_delay`. Esse procedimento deve ser executado na instância de banco de dados MySQL que será configurada como a réplica de leitura de uma instância do MySQL externa ao Amazon RDS.

Antes de executar `mysql.rds_set_external_source_with_delay`, é necessário configurar a instância do MySQL executada externamente ao Amazon RDS para ser uma instância de banco de dados de origem. Para conectar-se à instância do MySQL que é externa ao Amazon RDS, você deve especificar os valores de `replication_user_name` e `replication_user_password`. Esses valores devem indicar um usuário de replicação que tenha as permissões `REPLICATION CLIENT` e `REPLICATION SLAVE` na instância externa do MySQL.

Como configurar uma instância externa do MySQL como uma instância de banco de dados de origem

1. Usando o cliente do MySQL de sua escolha, conecte-se à instância externa do MySQL e crie uma conta de usuário a ser usada para a replicação. Veja um exemplo a seguir.

   ```
   CREATE USER 'repl_user'@'mydomain.com' IDENTIFIED BY 'SomePassW0rd'
   ```

2. Na instância externa do MySQL, conceda privilégios de `REPLICATION CLIENT` e `REPLICATION SLAVE` para seu usuário de replicação. O exemplo a seguir concede privilégios

de `REPLICATION CLIENT` e `REPLICATION SLAVE` ao usuário `'repl_user'` em todos os bancos de dados de seu domínio.

```
GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'mydomain.com'
IDENTIFIED BY 'SomePassW0rd'
```

Para obter mais informações, consulte Configurar a replicação da posição do arquivo de log binário com uma instância de origem externa.

> **Note**
>
> Recomendamos que você use réplicas de leitura para gerenciar a replicação entre duas instâncias de banco de dados do Amazon RDS, sempre que possível. Quando você o fizer, recomendamos o uso somente desse e de outros procedimentos armazenados relacionados a replicação. Essas práticas permitem o uso de topologias de replicação mais complexas entre as instâncias de banco de dados do Amazon RDS. Oferecemos esses procedimentos armazenados principalmente para permitir a replicação com instâncias do MySQL externas ao Amazon RDS. Para obter informações sobre como gerenciar a replicação entre instâncias de banco de dados do Amazon RDS, consulte Trabalhar com réplicas de leitura de instância de banco de dados.

Depois de chamar `mysql.rds_set_external_source_with_delay` para configurar uma instância de banco de dados do Amazon RDS, é possível chamar mysql.rds_start_replication na réplica de leitura para iniciar o processo de replicação. É possível chamar mysql.rds_reset_external_source (RDS para MySQL versões principais 8.4 e posteriores) para remover a configuração da réplica de leitura.

Ao chamar `mysql.rds_set_external_source_with_delay`, o Amazon RDS registra a hora, o usuário e uma ação do `set master` nas tabelas `mysql.rds_history` e `mysql.rds_replication_status`.

Para a recuperação de desastres, você pode usar esse procedimento com o procedimento armazenado mysql.rds_start_replication_until ou o mysql.rds_start_replication_until_gtid. Para efetuar roll forward de alterações para uma réplica de leitura atrasada no momento imediatamente antes de um desastre, é possível executar o procedimento `mysql.rds_set_external_source_with_delay`. Depois que o procedimento

`mysql.rds_start_replication_until` interromper a replicação, é possível promover a réplica de leitura para ser a nova instância de banco de dados primária usando as instruções em Promoção de uma réplica de leitura a uma instância de banco de dados autônoma.

Para usar o procedimento `mysql.rds_rds_start_replication_until_gtid`, a replicação baseada em GTID deve estar habilitada. Para ignorar uma transação específica baseada em GTID que seja conhecida por causar desastres, use o procedimento armazenado mysql.rds_skip_transaction_with_gtid. Para obter mais informações sobre como trabalhar com a replicação baseada em GTID, consulte Usar a replicação baseada em GTID.

Exemplos

Ao executar em uma instância de banco de dados MySQL, o exemplo a seguir configura a instância do banco de dados para ser uma réplica de leitura de uma instância do MySQL que é executada externamente ao Amazon RDS. Ele define o atraso mínimo de replicação como uma hora (3.600 segundos) na instância de banco de dados MySQL. Uma alteração da instância do banco de dados de origem MySQL executada externamente ao Amazon RDS não é aplicada à réplica de leitura da instância de banco de dados MySQL por pelo menos uma hora.

```
call mysql.rds_set_external_source_with_delay(
  'Externaldb.some.com',
  3306,
  'repl_user',
  'SomePassW0rd',
  'mysql-bin-changelog.000777',
  120,
  1,
  3600);
```

mysql.rds_set_external_source_gtid_purged

Define a variável de sistema gtid_purged com um intervalo GTID especificado de uma fonte externa. O valor `gtid_purged` é necessário para configurar a replicação baseada em GTID para retomar a replicação usando posicionamento automático.

> **Important**
>
> Para executar esse procedimento, `autocommit` deve estar habilitado. Para habilitá-lo, defina o parâmetro `autocommit` como 1. Para obter informações sobre como modificar

parâmetros, consulte Modificar parâmetros em um grupo de parâmetros de banco de dados no Amazon RDS.

Sintaxe

```
CALL mysql.rds_set_external_source_gtid_purged(
  server_uuid
  , start_pos
  , end_pos
);
```

Parâmetros

server_uuid

O identificador universalmente exclusivo (UID) do servidor externo do qual o intervalo GTID está sendo importado.

start_pos

A posição inicial do intervalo GTID a ser definida.

end_pos

A posição final do intervalo GTID a ser definida.

Observações de uso

O procedimento `mysql.rds_set_external_source_gtid_purged` só está disponível com o MySQL 8.0.37 e versões 8.0 posteriores.

Chame `mysql.rds_set_external_source_gtid_purged` antes de chamar mysql.rds_set_external_master_with_auto_position (RDS para MySQL versões principais 8.0 e anteriores), mysql.rds_set_external_source_with_auto_position (RDS para MySQL versões principais 8.4 e posteriores) ou mysql.rds_set_external_source_with_auto_position_for_channel.

Antes de chamar `mysql.rds_set_external_source_gtid_purged`, interrompa todos os canais de replicação ativos do banco de dados. Para conferir o status de um canal, use a declaração `SHOW`

REPLICA STATUS MySQL. Para interromper a replicação em um canal, chame the section called "mysql.rds_stop_replication_for_channel".

O intervalo GTID especificado deverá ser um superconjunto do valor GTID_PURGED existente. Esse procedimento armazenado confere os seguintes valores antes de definir o valor GTID_PURGED:

- O server_uuid é válido.
- O valor de start_pos é maior que 0 e menor que o valor de end_pos.
- O valor de end_pos é maior ou igual ao valor de start_pos.

Se o conjunto de GTID no servidor externo contiver vários intervalos de valores, pense em chamar o procedimento várias vezes com valores de conjunto de GTID diferentes.

Ao chamar mysql.rds_set_external_source_gtid_purged, o Amazon RDS registra a hora, o usuário e uma ação do set gtid_purged na tabela mysql.rds_history.

Se você não definir o valor gtid_purged adequadamente para o backup utilizado para replicação, isso poderá gerar transações perdidas ou duplicadas durante o processo de replicação. Execute as etapas a seguir para corrigir o valor gtid_purged.

Como definir o valor gtid_purged na réplica

1. Determine o ponto no tempo ou o arquivo de backup específico a ser usado como ponto de partida para a replicação. Pode ser um backup lógico (um arquivo mysqldump) ou físico (um snapshot do Amazon RDS).
2. Determine o valor gtid_executed. Esse valor representa o conjunto de todos os GTIDs que foram confirmados no servidor. Para recuperar esse valor, na instância de origem, faça o seguinte:
   - Execute a declaração SELECT @@GLOBAL.GTID_EXECUTED; SQL no momento em que o backup foi feito.
   - Se alguma opção relacionada estiver incluída no respectivo utilitário de backup, extraia o valor do arquivo de backup. Para ter mais informações, consulte a opção set-gtid-purged na documentação do MySQL.
3. Determine o valor gtid_purged a ser usado para a chamada para mysql.rds_set_external_source_gtid_purged. O valor gtid_purged deve incluir todos os GTIDs que foram executados na instância de origem e que não são mais necessários

para a replicação. Portanto, o valor `gtid_purged` deve ser um subconjunto do valor `gtid_executed` recuperado na etapa anterior.

Para determinar o valor `gtid_purged`, identifique os GTIDs que não estão incluídos no backup e não são mais necessários para a replicação. É possível fazer isso analisando os logs binários ou usando uma ferramenta, como mysqlbinlog, para encontrar os GTIDs que foram removidos dos logs binários.

Se você tiver um backup consistente que inclua todos os logs binários até o ponto de backup, também será possível definir o valor `gtid_purged` como igual ao valor `gtid_executed` no ponto de backup.

4. Depois de determinar o valor `gtid_purged` apropriado que seja consistente com o backup, chame o procedimento armazenado `mysql.rds_set_external_source_gtid_purged` na sua instância de banco de dados do RDS para MySQL para definir o valor.

Exemplos

Quando executado em uma instância de banco de dados do MySQL, o exemplo a seguir define o intervalo GTID de um servidor MySQL externo com o UUID `12345678-abcd-1234-efgh-123456789abc`, uma posição inicial de 1 e uma posição final de `100`. O valor do GTID resultante é definido como `+12345678-abcd-1234-efgh-123456789abc:1-100`.

```
CALL mysql.rds_set_external_source_gtid_purged('12345678-abcd-1234-efgh-123456789abc',
 1, 100);
```

mysql.rds_set_master_auto_position (RDS para MySQL versões principais 8.0 e anteriores)

Define o modo de replicação de base nas posições do arquivo de log binário ou nos identificadores de transações globais (GTIDs).

Sintaxe

```
CALL mysql.rds_set_master_auto_position (
auto_position_mode
);
```

Parâmetros

auto_position_mode

Um valor que indica se será usada a replicação de posição do arquivo de log ou a replicação com base no GTID:

- 0: usar o método de replicação com base na posição do arquivo de log binário. O padrão é 0.
- 1: usar o método de replicação com base no GTID.

Observações de uso

O usuário principal deve executar o procedimento mysql.rds_set_master_auto_position.

Esse procedimento é compatível com todas as versões 5.7 do RDS para MySQL e com o RDS para MySQL 8.0.26 e versões 8.0 posteriores.

mysql.rds_set_source_auto_position (RDS para MySQL versões principais 8.4 e posteriores)

Define o modo de replicação de base nas posições do arquivo de log binário ou nos identificadores de transações globais (GTIDs).

Sintaxe

```
CALL mysql.rds_set_source_auto_position (auto_position_mode);
```

Parâmetros

auto_position_mode

Um valor que indica se será usada a replicação de posição do arquivo de log ou a replicação com base no GTID:

- 0: usar o método de replicação com base na posição do arquivo de log binário. O padrão é 0.
- 1: usar o método de replicação com base no GTID.

Observações de uso

O usuário administrativo precisa executar o procedimento mysql.rds_set_source_auto_position.

mysql.rds_set_source_delay

Define o número mínimo de segundos para atrasar a replicação da instância de banco de dados de origem para a réplica de leitura atual. Use esse procedimento quando estiver conectado a uma réplica de leitura para atrasar a replicação da instância de banco de dados de origem.

Sintaxe

```
CALL mysql.rds_set_source_delay(
delay
);
```

Parâmetros

delay

O número mínimo de segundos para atrasar a replicação da instância de banco de dados de origem.

O limite para esse parâmetro é de um dia (86.400 segundos).

Observações de uso

O usuário mestre deve executar o procedimento `mysql.rds_set_source_delay`.

Para a recuperação de desastres, você pode usar esse procedimento com o procedimento armazenado mysql.rds_start_replication_until ou o mysql.rds_start_replication_until_gtid. Para efetuar roll forward de alterações para uma réplica de leitura atrasada no momento imediatamente antes de um desastre, é possível executar o procedimento `mysql.rds_set_source_delay`. Depois que o procedimento `mysql.rds_start_replication_until` ou `mysql.rds_start_replication_until_gtid` interromper a replicação, é possível promover a réplica de leitura para ser a nova instância de banco de dados primária usando as instruções em Promoção de uma réplica de leitura a uma instância de banco de dados autônoma.

Para usar o procedimento `mysql.rds_rds_start_replication_until_gtid`, a replicação baseada em GTID deve estar habilitada. Para ignorar uma transação específica baseada em GTID que seja conhecida por causar desastres, use o procedimento armazenado mysql.rds_skip_transaction_with_gtid. Para obter mais informações sobre a replicação baseada em GTID, consulte Usar a replicação baseada em GTID.

O procedimento `mysql.rds_set_source_delay` está disponível nestas versões do RDS para MySQL:

- Todas as versões do RDS para MySQL 8.4
- MySQL 8.0.26 e versões 8.0 posteriores
- Todas as versões 5.7

Exemplos

Para atrasar a replicação da instância de banco de dados de origem para a réplica de leitura atual por pelo menos uma hora (3.600 segundos), é possível chamar `mysql.rds_set_source_delay` com o seguinte parâmetro:

```
CALL mysql.rds_set_source_delay(3600);
```

mysql.rds_skip_repl_error

Ignora e exclui um erro de replicação em uma réplica de leitura de banco de dados MySQL.

Sintaxe

```
CALL mysql.rds_skip_repl_error;
```

Observações de uso

O usuário principal deve executar o procedimento `mysql.rds_skip_repl_error` em uma réplica de leitura. Para obter mais informações sobre esse procedimento, consulte Chamando o procedimento mysql.rds_skip_repl_error.

Para determinar se há erros, execute o comando `SHOW REPLICA STATUS\G` do MySQL. Se um erro de replicação não for crítico, execute `mysql.rds_skip_repl_error` para ignorá-lo. Se houver vários, `mysql.rds_skip_repl_error` exclui o primeiro erro, depois avisa que há outros ainda. Assim, você pode usar `SHOW REPLICA STATUS\G` para determinar o plano de ação correto para o próximo erro. Para obter informações sobre os valores retornados, consulte Instrução SHOW REPLICA STATUS na documentação do MySQL.

Para obter mais informações sobre como lidar com erros de replicação no Amazon RDS, consulte Solucionar problemas de uma réplica de leitura do MySQL.

Erro de replicação interrompida

Ao chamar o procedimento `mysql.rds_skip_repl_error`, você pode receber uma mensagem de erro informando que a réplica está inativa ou desativada.

Essa mensagem de erro aparece quando você executa o procedimento na instância primária em vez da réplica de leitura. Você deve executar esse procedimento na réplica de leitura para que o procedimento funcione.

Essa mensagem de erro também poderá aparecer se você executar o procedimento na réplica de leitura, mas a replicação não poderá ser reiniciada com êxito.

Se você precisar ignorar um grande número de erros, o atraso de replicação poderá aumentar além do período de retenção padrão para arquivos de log binário (binlog). Nesse caso, você poderá encontrar um erro fatal, com os arquivos binlog sendo limpos antes de sua reprodução na réplica de leitura. Essa remoção faz com que a replicação pare, e você não consegue chamar o comando `mysql.rds_skip_repl_error` para ignorar erros de replicação.

É possível mitigar esse problema aumentando o número de horas em que os arquivos binlog são retidos na instância de banco de dados de origem. Após aumentar o período de retenção de log binário, você pode reiniciar a replicação e chamar o comando `mysql.rds_skip_repl_error` conforme necessário.

Para definir o período de retenção do binlog, use o procedimento mysql.rds_set_configuration e especifique um parâmetro de configuração `'binlog retention hours'`, juntamente com o número de horas de retenção dos arquivos binlog no cluster do banco de dados. O exemplo a seguir define o período de retenção para arquivos de log binário em 48 horas.

```
CALL mysql.rds_set_configuration('binlog retention hours', 48);
```

mysql.rds_start_replication

Inicia a replicação de uma instância de banco de dados do RDS para MySQL.

> **Note**
>
> Você pode usar o procedimento armazenado mysql.rds_start_replication_until ou mysql.rds_start_replication_until_gtid para iniciar a replicação de uma instância de banco de dados do RDS para MySQL e interromper a replicação no local do arquivo de log binário especificado.

Sintaxe

```
CALL mysql.rds_start_replication;
```

Observações de uso

O usuário principal deve executar o procedimento `mysql.rds_start_replication`.

Para importar dados de uma instância do MySQL fora do Amazon RDS, chame `mysql.rds_start_replication` na réplica de leitura para iniciar o processo de replicação depois de ter chamado mysql.rds_set_external_master (RDS para MariaDB e RDS para MySQL versões principais 8.0 e anteriores) ou mysql.rds_set_external_source (RDS para MySQL versões principais 8.4 e posteriores) para criar a configuração de replicação. Para ter mais informações, consulte Restauração de um backup em uma instância de banco de dados MySQL.

Para exportar dados para uma instância do MySQL fora do Amazon RDS, chame `mysql.rds_start_replication` e `mysql.rds_stop_replication` na réplica de leitura para controlar algumas ações de replicação, como a remoção de logs binários. Para ter mais informações, consulte Exportar dados de uma instância de banco de dados MySQL usando replicação.

Também é possível chamar `mysql.rds_start_replication` na réplica de leitura para reiniciar qualquer processo de replicação que tenha sido interrompido anteriormente chamando `mysql.rds_stop_replication`. Para ter mais informações, consulte Trabalhar com réplicas de leitura de instância de banco de dados.

mysql.rds_start_replication_until

Inicia a replicação de uma instância de banco de dados do RDS para MySQL e interrompe a replicação no local do arquivo de log binário especificado.

Sintaxe

```
CALL mysql.rds_start_replication_until (
replication_log_file
  , replication_stop_point
);
```

Parâmetros

replication_log_file

O nome do log binário na instância de banco de dados de origem que contém as informações de replicação.

replication_stop_point

O local no log binário `replication_log_file` no qual a replicação será interrompida.

Observações de uso

O usuário mestre deve executar o procedimento `mysql.rds_start_replication_until`.

O procedimento `mysql.rds_start_replication_until` está disponível nestas versões do RDS for MySQL:

- Todas as versões do RDS para MySQL 8.4
- MySQL 8.0.26 e versões 8.0 posteriores
- Todas as versões 5.7

Você pode usar esse procedimento com a replicação atrasada para recuperação de desastres. Se você tiver a replicação atrasada configurada, use esse procedimento para efetuar roll forward de alterações em uma réplica de leitura atrasada para o momento imediatamente antes de um desastre. Depois que esse procedimento interromper a replicação, é possível promover a réplica de leitura para ser a nova instância de banco de dados primária usando as instruções em Promoção de uma réplica de leitura a uma instância de banco de dados autônoma.

Você pode configurar a replicação atrasada usando os seguintes procedimentos armazenados:

- mysql.rds_set_configuration
- mysql.rds_set_external_master_with_delay (RDS para MariaDB e RDS para MySQL versões principais 8.0 e anteriores)
- mysql.rds_set_external_source_with_delay (RDS para MySQL versões principais 8.4 e superiores)
- mysql.rds_set_source_delay

O nome do arquivo especificado para o parâmetro `replication_log_file` deve corresponder ao nome do arquivo do log binário da instância de banco de dados de origem.

Quando o parâmetro `replication_stop_point` especifica um local de parada no passado, a replicação é interrompida imediatamente.

Exemplos

O exemplo a seguir inicia a replicação e replica as alterações até que ela atinja o local `120` no arquivo de log binário `mysql-bin-changelog.000777`.

```
call mysql.rds_start_replication_until(
  'mysql-bin-changelog.000777',
  120);
```

mysql.rds_stop_replication

Interrompe a replicação de uma instância de banco de dados MySQL.

Sintaxe

```
CALL mysql.rds_stop_replication;
```

Observações de uso

O usuário mestre deve executar o procedimento `mysql.rds_stop_replication`.

Se você estiver configurando a replicação para importar dados de uma instância do MySQL em execução externamente ao Amazon RDS, chame `mysql.rds_stop_replication` na réplica de leitura para encerrar o processo de replicação após a importação ter sido concluída. Para ter mais informações, consulte Restauração de um backup em uma instância de banco de dados MySQL.

Se estiver configurando a replicação para exportar dados para uma instância do MySQL externa ao Amazon RDS, chame `mysql.rds_start_replication` e `mysql.rds_stop_replication` na réplica de leitura para controlar algumas ações de replicação, como a remoção de logs binários. Para ter mais informações, consulte Exportar dados de uma instância de banco de dados MySQL usando replicação.

Você também pode usar `mysql.rds_stop_replication` para parar uma replicação entre duas instâncias de banco de dados do Amazon RDS. Normalmente, você interrompe a replicação para executar uma operação de execução longa na réplica de leitura, como ao criar um índice grande na réplica de leitura. Também é possível reiniciar qualquer processo de replicação que tenha sido

interrompido chamando mysql.rds_start_replication na réplica de leitura. Para ter mais informações, consulte Trabalhar com réplicas de leitura de instância de banco de dados.

Encerrar uma sessão ou consulta

Os procedimentos armazenados a seguir encerram uma sessão ou consulta.

Tópicos

- mysql.rds_kill
- mysql.rds_kill_query

mysql.rds_kill

Encerra uma conexão ao servidor MySQL.

Sintaxe

```
CALL mysql.rds_kill(processID);
```

Parâmetros

processID

A identidade do thread de conexão a ser encerrada.

Observações de uso

Cada conexão ao servidor do MySQL é executada em um thread separado. Para encerrar uma conexão, use o procedimento `mysql.rds_kill` e passe o ID de thread dessa conexão. Para obter o ID de thread, use o comando SHOW PROCESSLIST do MySQL.

Para receber informações sobre limitações, consulte Limitações de procedimentos armazenados do MySQL.

Exemplos

O exemplo a seguir encerra uma conexão com um ID de thread de 4243:

```
CALL mysql.rds_kill(4243);
```

mysql.rds_kill_query

Encerra uma consulta em execução no servidor MySQL.

Sintaxe

```
CALL mysql.rds_kill_query(processID);
```

Parâmetros

processID

A identidade do processo ou thread que está executando a consulta a ser encerrada.

Observações de uso

Para encerrar uma consulta em execução no servidor MySQL, use o procedimento `mysql_rds_kill_query` e passe o ID do thread que está executando a consulta. O procedimento então encerra a conexão.

Para obter o ID, consulte a tabela INFORMATION_SCHEMA PROCESSLIST do MySQL ou use o comando SHOW PROCESSLIST do MySQL. O valor na coluna ID de `SHOW PROCESSLIST` ou `SELECT * FROM INFORMATION_SCHEMA.PROCESSLIST` é *processID*.

Para receber informações sobre limitações, consulte Limitações de procedimentos armazenados do MySQL.

Exemplos

O seguinte exemplo encerra uma consulta com um ID de thread de consulta 230040:

```
CALL mysql.rds_kill_query(230040);
```

Gerenciar clusters ativos-ativos

Os procedimentos armazenados a seguir configuram e gerenciam clusters ativos-ativos do RDS para MySQL. Para ter mais informações, consulte the section called “Configurar clusters ativos-ativos”.

Esses procedimentos armazenados só estão disponíveis com as instâncias de banco de dados do RDS para MySQL que executam as seguintes versões:

- Todas as versões do MySQL 8.4
- MySQL 8.0.35 e versões secundárias posteriores

Tópicos

- mysql.rds_group_replication_advance_gtid
- mysql.rds_group_replication_create_user
- mysql.rds_group_replication_set_recovery_channel
- mysql.rds_group_replication_start
- mysql.rds_group_replication_stop

mysql.rds_group_replication_advance_gtid

Cria GTIDs de espaço reservado na instância de banco de dados atual.

Sintaxe

```
CALL mysql.rds_group_replication_advance_gtid(
begin_id
, end_id
, server_uuid
);
```

Parâmetros

begin_id

O ID da transação inicial a ser criado.

end_id

O ID da transação final a ser criado.

begin_id

O `group_replication_group_name` da transação final a ser criado. O `group_replication_group_name` é especificada como um UUID no grupo de parâmetros de banco de dados associado à instância de banco de dados.

Observações de uso

Em um cluster ativo-ativo, para que uma instância de banco de dados ingresse em um grupo, todas as transações GTID executadas na nova instância de banco de dados devem existir nos outros membros do cluster. Em casos incomuns, uma nova instância de banco de dados pode ter mais transações a serem agrupadas quando as transações são executadas antes de ingressar na instância. Nesse caso, não é possível remover nenhuma transação existente, mas é possível usar esse procedimento para criar os GTIDs de espaço reservado correspondentes nas outras instâncias de banco de dados do grupo. Antes de fazer isso, verifique se as transações não afetam os dados replicados.

Ao chamar esse procedimento, as transações GTID do `server_uuid:begin_id-end_id` são criadas com conteúdo vazio. Para evitar problemas de replicação, não use esse procedimento em nenhuma outra condição.

> **Important**
>
> Evite chamar esse procedimento quando o cluster ativo-ativo estiver funcionando normalmente. Não chame esse procedimento a menos que você entenda as possíveis consequências das transações que está criando. Chamar esse procedimento pode gerar dados inconsistentes.

Exemplo

O seguinte exemplo cria GTIDs de espaço reservado na instância de banco de dados atual:

```
CALL mysql.rds_group_replication_advance_gtid(5, 6,
 '11111111-2222-3333-4444-555555555555');
```

mysql.rds_group_replication_create_user

Cria o usuário de replicação `rdsgrprepladmin` para replicação do grupo na instância de banco de dados.

Sintaxe

```
CALL mysql.rds_group_replication_create_user(
replication_user_password
);
```

Parâmetros

`replication_user_password`

A senha do usuário de replicação `rdsgrprepladmin`.

Observações de uso

- A senha do usuário de replicação `rdsgrprepladmin` deve ser a mesma em todas as instâncias de banco de dados em um cluster ativo-ativo.
- O nome do usuário `rdsgrprepladmin` é reservado para conexões da replicação do grupo. Nenhum outro usuário, incluindo o usuário principal, pode ter esse nome de usuário.

Exemplo

O seguinte exemplo cria o usuário de replicação `rdsgrprepladmin` para replicação do grupo na instância de banco de dados:

```
CALL mysql.rds_group_replication_create_user('password');
```

mysql.rds_group_replication_set_recovery_channel

Define o canal `group_replication_recovery` para um cluster ativo-ativo. O procedimento utiliza o usuário reservado `rdsgrprepladmin` para configurar o canal.

Sintaxe

```
CALL mysql.rds_group_replication_set_recovery_channel(
```

```
replication_user_password);
```

Parâmetros

replication_user_password

A senha do usuário de replicação `rdsgrprepladmin`.

Observações de uso

A senha do usuário de replicação `rdsgrprepladmin` deve ser a mesma em todas as instâncias de banco de dados em um cluster ativo-ativo. Uma chamada para o `mysql.rds_group_replication_create_user` especifica a senha.

Exemplo

O seguinte exemplo define o canal `group_replication_recovery` para um cluster ativo-ativo:

```
CALL mysql.rds_group_replication_set_recovery_channel('password');
```

mysql.rds_group_replication_start

Inicia a replicação do grupo na instância de banco de dados atual.

Sintaxe

```
CALL mysql.rds_group_replication_start(
bootstrap
);
```

Parâmetros

bootstrap

Um valor que especifica se deve inicializar um novo grupo ou ingressar em um grupo existente. `1` inicializa um novo grupo com a instância de banco de dados atual. `0` insere a instância de banco de dados atual em um grupo existente conectando-se aos endpoints definidos no parâmetro `group_replication_group_seeds` no grupo de parâmetros do banco de dados associado à instância de banco de dados.

Exemplo

O seguinte exemplo inicializa um novo grupo com a instância de banco de dados atual:

```
CALL mysql.rds_group_replication_start(1);
```

mysql.rds_group_replication_stop

Interrompe a replicação do grupo na instância de banco de dados atual.

Sintaxe

```
CALL mysql.rds_group_replication_stop();
```

Observações de uso

A interrupção da replicação em uma instância de banco de dados não afeta nenhuma outra instância de banco de dados no cluster ativo-ativo.

Gerenciar a replicação de várias fontes

Os procedimentos armazenados a seguir configuram e gerenciam canais de replicação em uma réplica de várias fontes do RDS para MySQL. Para ter mais informações, consulte the section called "Configurar a replicação de várias fontes".

Esses procedimentos armazenados só estão disponíveis com as instâncias de banco de dados do RDS para MySQL que executam as seguintes versões de mecanismo:

- Todas as versões 8.4
- 8.0.35 e versões secundárias posteriores
- 5.7.44 e versões secundárias posteriores

Ao usar procedimentos armazenados para gerenciar a replicação com um usuário de replicação configurado com `caching_sha2_passwword`, você deve configurar o TLS especificando `SOURCE_SSL=1`. `caching_sha2_password` é o plugin de autenticação padrão do RDS para MySQL 8.4.

> **Note**
>
> Embora essa documentação se refira às instâncias de banco de dados de origem como instâncias de banco de dados do RDS para MySQL, esses procedimentos também funcionam para instâncias do MySQL executadas fora do Amazon RDS.

Tópicos

- mysql.rds_next_source_log_for_channel
- mysql.rds_reset_external_source_for_channel
- mysql.rds_set_external_source_for_channel
- mysql.rds_set_external_source_with_auto_position_for_channel
- mysql.rds_set_external_source_with_delay_for_channel
- mysql.rds_set_source_auto_position_for_channel
- mysql.rds_set_source_delay_for_channel
- mysql.rds_skip_repl_error_for_channel
- mysql.rds_start_replication_for_channel

- mysql.rds_start_replication_until_for_channel
- mysql.rds_start_replication_until_gtid_for_channel
- mysql.rds_stop_replication_for_channel

mysql.rds_next_source_log_for_channel

Altera a posição do log da instância de banco de dados de origem para o início do próximo log binário na instância de banco de dados de origem. Use esse procedimento somente se estiver recebendo o erro 1236 de E/S de replicação em uma réplica de várias fontes.

Sintaxe

```
CALL mysql.rds_next_source_log_for_channel(
curr_master_log,
channel_name
);
```

Parâmetros

curr_master_log

O índice do arquivo de log de origem atual. Por exemplo, se o arquivo atual dor denominado `mysql-bin-changelog.012345`, o índice será 12345. Para determinar o nome do arquivo de log de origem atual, execute o comando `SHOW REPLICA STATUS FOR CHANNEL '`*`channel_name`*`'` e veja o campo `Source_Log_File`.

channel_name

O nome do canal de replicação na réplica de várias fontes. Cada canal de replicação recebe os eventos de logs binários de uma única instância de banco de dados do RDS para MySQL de origem executada em uma porta e um host específicos.

Observações de uso

O usuário principal deve executar o procedimento `mysql.rds_next_source_log_for_channel`. Se houver um erro IO_thread, por exemplo, será possível usar esse procedimento para ignorar todos os eventos no arquivo de log binário atual e retomar a replicação do próximo arquivo de log binário para o canal especificado em `channel_name`.

Exemplo

Suponha que a replicação falhe em um canal em uma réplica de várias fontes. A execução de `SHOW REPLICA STATUS FOR CHANNEL 'channel_1'\G` na réplica de leitura exibe o seguinte resultado:

```
mysql> SHOW REPLICA STATUS FOR CHANNEL 'channel_1'\G
*************************** 1. row ***************************
             Replica_IO_State: Waiting for source to send event
                  Source_Host: myhost.XXXXXXXXXXXXXX.rr-rrrr-1.rds.amazonaws.com
                  Source_User: ReplicationUser
                  Source_Port: 3306
                Connect_Retry: 60
              Source_Log_File: mysql-bin-changelog.012345
          Read_Source_Log_Pos: 1219393
               Relay_Log_File: replica-relay-bin.000003
                Relay_Log_Pos: 30223388
        Relay_Source_Log_File: mysql-bin-changelog.012345
           Replica_IO_Running: No
          Replica_SQL_Running: Yes
              Replicate_Do_DB:.
              .
              .
                Last_IO_Errno: 1236
                Last_IO_Error: Got fatal error 1236 from master when reading data from
 binary log: 'Client requested master to start replication from impossible position;
 the first event 'mysql-bin-changelog.013406' at 1219393, the last event read from
 '/rdsdbdata/log/binlog/mysql-bin-changelog.012345' at 4, the last byte read from '/
rdsdbdata/log/binlog/mysql-bin-changelog.012345' at 4.'
               Last_SQL_Errno: 0
               Last_SQL_Error:
               .
               .
                 Channel_name: channel_1
              .
              .
 -- Some fields are omitted in this example output
```

O campo `Last_IO_Errno` mostra que a instância está recebendo o erro 1236 de E/S. O campo `Source_Log_File` mostra que o nome do arquivo é `mysql-bin-changelog.012345`,

o que significa que o índice de arquivos de log é 12345. Para resolver o erro, chame `mysql.rds_next_source_log_for_channel` com os seguintes parâmetros:

```
CALL mysql.rds_next_source_log_for_channel(12345,'channel_1');
```

mysql.rds_reset_external_source_for_channel

Interrompe o processo de replicação no canal especificado e remove o canal e as configurações associadas da réplica de várias fontes.

> **Important**
>
> Para executar esse procedimento, `autocommit` deve estar habilitado. Para habilitá-lo, defina o parâmetro `autocommit` como 1. Para obter informações sobre como modificar parâmetros, consulte Modificar parâmetros em um grupo de parâmetros de banco de dados no Amazon RDS.

Sintaxe

```
CALL mysql.rds_reset_external_source_for_channel (channel_name);
```

Parâmetros

channel_name

O nome do canal de replicação na réplica de várias fontes. Cada canal de replicação recebe os eventos de logs binários de uma única instância de banco de dados do RDS para MySQL de origem executada em uma porta e um host específicos.

Observações de uso

O usuário principal deve executar o procedimento `mysql.rds_reset_external_source_for_channel`. Esse procedimento exclui todos os registros de retransmissão pertencentes ao canal que está sendo removido.

mysql.rds_set_external_source_for_channel

Configura um canal de replicação em uma instância de banco de dados do RDS para MySQL para replicar os dados de outra instância de banco de dados do RDS para MySQL.

> **Important**
>
> Para executar esse procedimento, `autocommit` deve estar habilitado. Para habilitá-lo, defina o parâmetro `autocommit` como 1. Para obter informações sobre como modificar parâmetros, consulte Modificar parâmetros em um grupo de parâmetros de banco de dados no Amazon RDS.

> **Note**
>
> Em vez disso, é possível usar o procedimento armazenado the section called "mysql.rds_set_external_source_with_delay_for_channel" para configurar esse canal com replicação atrasada.

Sintaxe

```
CALL mysql.rds_set_external_source_for_channel (
  host_name
  , host_port
  , replication_user_name
  , replication_user_password
  , mysql_binary_log_file_name
  , mysql_binary_log_file_location
  , ssl_encryption
  , channel_name
);
```

Parâmetros

host_name

O nome de host ou o endereço IP da instância de banco de dados do RDS para MySQL.

host_port

A porta usada pela instância de banco de dados do RDS para MySQL. Se sua configuração de rede inclui replicação de porta Secure Shell (SSH) que converte o número da porta, especifique o número da porta exposto pelo SSH.

replication_user_name

O ID de um usuário com as permissões `REPLICATION CLIENT` e `REPLICATION SLAVE` na instância de banco de dados de origem do RDS para MySQL. Recomendamos fornecer uma conta que seja usada unicamente para a replicação com a instância de banco de dados de origem.

replication_user_password

A senha do ID de usuário especificada em `replication_user_name`.

mysql_binary_log_file_name

O nome do log binário na instância de banco de dados de origem que contém as informações de replicação.

mysql_binary_log_file_location

O local no log binário `mysql_binary_log_file_name` no qual a replicação começa a ler as informações de replicação.

É possível determinar o nome e a localização do arquivo de binlog executando `SHOW BINARY LOG STATUS` na instância do banco de dados de origem.

> **Note**
>
> As versões anteriores do MySQL usavam `SHOW MASTER STATUS` em vez de `SHOW BINARY LOG STATUS`. Se você estiver usando uma versão do MySQL anterior à 8.4, use `SHOW MASTER STATUS`.

ssl_encryption

Um valor que especifica se a criptografia do Secure Sockets Layer (SSL) será usada na conexão de replicação. 1 especifica para usar criptografia de SSL, 0 especifica para não usar criptografia. O padrão é 0.

> **Note**
>
> A opção `SOURCE_SSL_VERIFY_SERVER_CERT` não tem suporte. Essa opção é definida como 0, o que significa que a conexão é criptografada, mas os certificados não são verificados.

`channel_name`

O nome da instância de replicação. Cada canal de replicação recebe os eventos de logs binários de uma única instância de banco de dados do RDS para MySQL de origem executada em uma porta e um host específicos.

Observações de uso

O usuário principal deve executar o procedimento `mysql.rds_set_external_source_for_channel`. Esse procedimento deve ser executado na instância de banco de dados do RDS para MySQL de destino na qual você está criando o canal de replicação.

Antes de executar `mysql.rds_set_external_source_for_channel`, configure um usuário de replicação na instância de banco de dados de origem com os privilégios necessários para a réplica de várias fontes. Para conectar a réplica de várias fontes à instância de banco de dados de origem, é necessário especificar os valores `replication_user_name` e `replication_user_password` de um usuário de replicação que tenha as permissões `REPLICATION CLIENT` e `REPLICATION SLAVE` na instância de banco de dados de origem.

Como configurar um usuário de replicação na instância de banco de dados de origem

1. Usando o cliente MySQL de sua preferência, conecte-se à instância de banco de dados de origem e crie uma conta de usuário a ser usada para a replicação. Veja um exemplo a seguir.

   > **Important**
   >
   > Como prática recomendada de segurança, especifique uma senha diferente do valor do espaço reservado mostrado nos exemplos a seguir.

```
CREATE USER 'repl_user'@'example.com' IDENTIFIED BY 'password';
```

2. Na instância de banco de dados de origem, conceda os privilégios `REPLICATION CLIENT` e `REPLICATION SLAVE` para o usuário de replicação. O exemplo a seguir concede privilégios de `REPLICATION CLIENT` e `REPLICATION SLAVE` ao usuário 'repl_user' em todos os bancos de dados de seu domínio.

```
GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'example.com';
```

Para usar a replicação criptografada, configure a instância de banco de dados de origem para usar conexões SSL.

Depois de chamar `mysql.rds_set_external_source_for_channel` para configurar esse canal de replicação, será possível chamar mysql.rds_start_replication_for_channel na réplica para iniciar o processo de replicação no canal. É possível chamar the section called "mysql.rds_reset_external_source_for_channel" para interromper a replicação no canal e remover a configuração do canal da réplica.

Ao chamar `mysql.rds_set_external_source_for_channel`, o Amazon RDS grava a hora, o usuário e uma ação de `set channel source` na tabela `mysql.rds_history` sem detalhes específicos do canal e na tabela `mysql.rds_replication_status` com o nome do canal. Essas informações são registradas apenas para fins internos de uso e monitoramento. Para registrar a chamada de procedimento completa para fins de auditoria, pense em habilitar logs gerais ou de auditoria, com base nos requisitos específicos da aplicação.

Exemplos

Quando executado em uma instância de banco de dados do RDS para MySQL, o exemplo a seguir configura um canal de replicação denominado `channel_1` nessa instância de banco de dados para replicar dados da fonte especificada pelo host `sourcedb.example.com` e pela porta `3306`.

```
call mysql.rds_set_external_source_for_channel(
  'sourcedb.example.com',
  3306,
  'repl_user',
  'password',
  'mysql-bin-changelog.0777',
```

```
  120,
  0,
  'channel_1');
```

mysql.rds_set_external_source_with_auto_position_for_channel

Configura um canal de replicação em uma instância de banco de dados do RDS para MySQL com um atraso de replicação opcional. A replicação se baseia em identificadores de transação global (GTIDs).

> **Important**
>
> Para executar esse procedimento, `autocommit` deve estar habilitado. Para habilitá-lo, defina o parâmetro `autocommit` como 1. Para obter informações sobre como modificar parâmetros, consulte Modificar parâmetros em um grupo de parâmetros de banco de dados no Amazon RDS.

Sintaxe

```
CALL mysql.rds_set_external_source_with_auto_position_for_channel (
  host_name
  , host_port
  , replication_user_name
  , replication_user_password
  , ssl_encryption
  , delay
  , channel_name
);
```

Parâmetros

host_name

O nome de host ou o endereço IP da instância de banco de dados do RDS para MySQL.

host_port

A porta usada pela instância de banco de dados do RDS para MySQL. Se sua configuração de rede inclui replicação de porta Secure Shell (SSH) que converte o número da porta, especifique o número da porta exposto pelo SSH.

replication_user_name

O ID de um usuário com as permissões `REPLICATION CLIENT` e `REPLICATION SLAVE` na instância de banco de dados de origem do RDS para MySQL. Recomendamos fornecer uma conta que seja usada unicamente para a replicação com a instância de banco de dados de origem.

replication_user_password

A senha do ID de usuário especificada em `replication_user_name`.

ssl_encryption

Um valor que especifica se a criptografia do Secure Sockets Layer (SSL) será usada na conexão de replicação. 1 especifica para usar criptografia de SSL, 0 especifica para não usar criptografia. O padrão é 0.

> **Note**
>
> A opção `SOURCE_SSL_VERIFY_SERVER_CERT` não tem suporte. Essa opção é definida como 0, o que significa que a conexão é criptografada, mas os certificados não são verificados.

delay

O número mínimo de segundos para atrasar a replicação da instância de banco de dados de origem.

O limite para esse parâmetro é de um dia (86.400 segundos).

channel_name

O nome da instância de replicação. Cada canal de replicação recebe os eventos de logs binários de uma única instância de banco de dados do RDS para MySQL de origem executada em uma porta e um host específicos.

Observações de uso

O usuário principal deve executar o procedimento `mysql.rds_set_external_source_with_auto_position_for_channel`. Esse

procedimento deve ser executado na instância de banco de dados do RDS para MySQL de destino na qual você está criando o canal de replicação.

Antes de executar `rds_set_external_source_with_auto_position_for_channel`, configure um usuário de replicação na instância de banco de dados de origem com os privilégios necessários para a réplica de várias fontes. Para conectar a réplica de várias fontes à instância de banco de dados de origem, é necessário especificar os valores `replication_user_name` e `replication_user_password` de um usuário de replicação que tenha as permissões `REPLICATION CLIENT` e `REPLICATION SLAVE` na instância de banco de dados de origem.

Como configurar um usuário de replicação na instância de banco de dados de origem

1. Usando o cliente MySQL de sua preferência, conecte-se à instância de banco de dados de origem e crie uma conta de usuário a ser usada para a replicação. Veja um exemplo a seguir.

   > **Important**
   >
   > Como prática recomendada de segurança, especifique uma senha diferente do valor do espaço reservado mostrado nos exemplos a seguir.

   ```
   CREATE USER 'repl_user'@'example.com' IDENTIFIED BY 'password';
   ```

2. Na instância de banco de dados de origem, conceda os privilégios `REPLICATION CLIENT` e `REPLICATION SLAVE` para o usuário de replicação. O exemplo a seguir concede privilégios de `REPLICATION CLIENT` e `REPLICATION SLAVE` ao usuário 'repl_user' em todos os bancos de dados de seu domínio.

   ```
   GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'example.com';
   ```

Para usar a replicação criptografada, configure a instância de banco de dados de origem para usar conexões SSL.

Antes de chamar `mysql.rds_set_external_source_with_auto_position_for_channel`, chame the section called “mysql.rds_set_external_source_gtid_purged” para definir a variável de sistema `gtid_purged` com um intervalo GTID especificado de uma fonte externa.

Depois de chamar `mysql.rds_set_external_source_with_auto_position_for_channel` para configurar uma instância de banco de dados do Amazon RDS como réplica de leitura em um canal específico, é possível chamar the section called “mysql.rds_start_replication_for_channel” na réplica de leitura para iniciar o processo de replicação nesse canal.

Depois de chamar `mysql.rds_set_external_source_with_auto_position_for_channel` para configurar esse canal de replicação, será possível chamar mysql.rds_start_replication_for_channel na réplica para iniciar o processo de replicação no canal. É possível chamar the section called “mysql.rds_reset_external_source_for_channel” para interromper a replicação no canal e remover a configuração do canal da réplica.

Exemplos

Quando executado em uma instância de banco de dados do RDS para MySQL, o exemplo a seguir configura um canal de replicação denominado `channel_1` nessa instância de banco de dados para replicar dados da fonte especificada pelo host `sourcedb.example.com` e pela porta `3306`. Ele define o atraso de replicação mínimo como uma hora (3.600 segundos). Isso significa que uma alteração da instância de banco de dados do RDS para MySQL não é aplicada à réplica de várias fontes por pelo menos uma hora.

```
call mysql.rds_set_external_source_with_auto_position_for_channel(
  'sourcedb.example.com',
  3306,
  'repl_user',
  'password',
  1,
  3600,
  'channel_1');
```

mysql.rds_set_external_source_with_delay_for_channel

Configura um canal de replicação em uma instância de banco de dados do RDS para MySQL com um atraso de replicação especificado.

> **Important**
>
> Para executar esse procedimento, `autocommit` deve estar habilitado. Para habilitá-lo, defina o parâmetro `autocommit` como 1. Para obter informações sobre como modificar

parâmetros, consulte Modificar parâmetros em um grupo de parâmetros de banco de dados no Amazon RDS.

Sintaxe

```
CALL mysql.rds_set_external_source_with_delay_for_channel (
  host_name
  , host_port
  , replication_user_name
  , replication_user_password
  , mysql_binary_log_file_name
  , mysql_binary_log_file_location
  , ssl_encryption
  , delay
  , channel_name
);
```

Parâmetros

host_name

O nome de host ou o endereço IP da instância de banco de dados do RDS para MySQL.

host_port

A porta usada pela instância de banco de dados do RDS para MySQL. Se sua configuração de rede inclui replicação de porta Secure Shell (SSH) que converte o número da porta, especifique o número da porta exposto pelo SSH.

replication_user_name

O ID de um usuário com as permissões `REPLICATION CLIENT` e `REPLICATION SLAVE` na instância de banco de dados de origem do RDS para MySQL. Recomendamos fornecer uma conta que seja usada unicamente para a replicação com a instância de banco de dados de origem.

replication_user_password

A senha do ID de usuário especificada em `replication_user_name`.

mysql_binary_log_file_name

O nome do log binário na instância de banco de dados de origem contém as informações de replicação.

mysql_binary_log_file_location

O local no log binário `mysql_binary_log_file_name` no qual a replicação começará a ler as informações de replicação.

É possível determinar o nome e a localização do arquivo de binlog executando `SHOW BINARY LOG STATUS` na instância do banco de dados de origem.

> **Note**
>
> As versões anteriores do MySQL usavam `SHOW MASTER STATUS` em vez de `SHOW BINARY LOG STATUS`. Se você estiver usando uma versão do MySQL anterior à 8.4, use `SHOW MASTER STATUS`.

ssl_encryption

Um valor que especifica se a criptografia do Secure Sockets Layer (SSL) será usada na conexão de replicação. 1 especifica para usar criptografia de SSL, 0 especifica para não usar criptografia. O padrão é 0.

> **Note**
>
> A opção `SOURCE_SSL_VERIFY_SERVER_CERT` não tem suporte. Essa opção é definida como 0, o que significa que a conexão é criptografada, mas os certificados não são verificados.

delay

O número mínimo de segundos para atrasar a replicação da instância de banco de dados de origem.

O limite para esse parâmetro é de um dia (86.400 segundos).

channel_name

O nome da instância de replicação. Cada canal de replicação recebe os eventos de logs binários de uma única instância de banco de dados do RDS para MySQL de origem executada em uma porta e um host específicos.

Observações de uso

O usuário principal deve executar o procedimento `mysql.rds_set_external_source_with_delay_for_channel`. Esse procedimento deve ser executado na instância de banco de dados do RDS para MySQL de destino na qual você está criando o canal de replicação.

Antes de executar `mysql.rds_set_external_source_with_delay_for_channel`, configure um usuário de replicação na instância de banco de dados de origem com os privilégios necessários para a réplica de várias fontes. Para conectar a réplica de várias fontes à instância de banco de dados de origem, é necessário especificar os valores `replication_user_name` e `replication_user_password` de um usuário de replicação que tenha as permissões `REPLICATION CLIENT` e `REPLICATION SLAVE` na instância de banco de dados de origem.

Como configurar um usuário de replicação na instância de banco de dados de origem

1. Usando o cliente MySQL de sua preferência, conecte-se à instância de banco de dados de origem e crie uma conta de usuário a ser usada para a replicação. Veja um exemplo a seguir.

   > **Important**
   >
   > Como prática recomendada de segurança, especifique uma senha diferente do valor do espaço reservado mostrado nos exemplos a seguir.

   ```
   CREATE USER 'repl_user'@'example.com' IDENTIFIED BY 'password';
   ```

2. Na instância de banco de dados de origem, conceda os privilégios `REPLICATION CLIENT` e `REPLICATION SLAVE` para o usuário de replicação. O exemplo a seguir concede privilégios de `REPLICATION CLIENT` e `REPLICATION SLAVE` ao usuário 'repl_user' em todos os bancos de dados de seu domínio.

```
GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'example.com';
```

Para usar a replicação criptografada, configure a instância de banco de dados de origem para usar conexões SSL.

Depois de chamar `mysql.rds_set_external_source_with_delay_for_channel` para configurar esse canal de replicação, será possível chamar mysql.rds_start_replication_for_channel na réplica para iniciar o processo de replicação no canal. É possível chamar the section called "mysql.rds_reset_external_source_for_channel" para interromper a replicação no canal e remover a configuração do canal da réplica.

Ao chamar `mysql.rds_set_external_source_with_delay_for_channel`, o Amazon RDS grava a hora, o usuário e uma ação de `set channel source` na tabela `mysql.rds_history` sem detalhes específicos do canal e na tabela `mysql.rds_replication_status` com o nome do canal. Essas informações são registradas apenas para fins internos de uso e monitoramento. Para registrar a chamada de procedimento completa para fins de auditoria, pense em habilitar logs gerais ou de auditoria, com base nos requisitos específicos da aplicação.

Exemplos

Quando executado em uma instância de banco de dados do RDS para MySQL, o exemplo a seguir configura um canal de replicação denominado `channel_1` nessa instância de banco de dados para replicar dados da fonte especificada pelo host `sourcedb.example.com` e pela porta `3306`. Ele define o atraso de replicação mínimo como uma hora (3.600 segundos). Isso significa que uma alteração da instância de banco de dados do RDS para MySQL não é aplicada à réplica de várias fontes por pelo menos uma hora.

```
call mysql.rds_set_external_source_with_delay_for_channel(
  'sourcedb.example.com',
  3306,
  'repl_user',
  'password',
  'mysql-bin-changelog.000777',
  120,
  1,
  3600,
  'channel_1');
```

mysql.rds_set_source_auto_position_for_channel

Define o modo de replicação para o canal especificado como baseado nas posições do arquivo de log binário ou nos identificadores de transações globais (GTIDs).

Sintaxe

```
CALL mysql.rds_set_source_auto_position_for_channel (
auto_position_mode
 , channel_name
);
```

Parâmetros

auto_position_mode

Um valor que indica se será usada a replicação de posição do arquivo de log ou a replicação com base no GTID:

- `0`: usar o método de replicação com base na posição do arquivo de log binário. O padrão é `0`.
- 1: usar o método de replicação com base no GTID.

channel_name

O nome do canal de replicação na réplica de várias fontes. Cada canal de replicação recebe os eventos de logs binários de uma única instância de banco de dados do RDS para MySQL de origem executada em uma porta e um host específicos.

Observações de uso

O usuário principal deve executar o procedimento `mysql.rds_set_source_auto_position_for_channel`. Esse procedimento reinicia a replicação no canal especificado para aplicar o modo de posição automática especificado.

Exemplos

O exemplo a seguir define o modo de posição automática para channel_1 usar o método de replicação baseado em GTID.

```
call mysql.rds_set_source_auto_position_for_channel(1,'channel_1');
```

mysql.rds_set_source_delay_for_channel

Define o número mínimo de segundos para atrasar a replicação da instância de banco de dados de origem para a réplica de várias fontes para o canal especificado.

Sintaxe

```
CALL mysql.rds_set_source_delay_for_channel(delay, channel_name);
```

Parâmetros

delay

O número mínimo de segundos para atrasar a replicação da instância de banco de dados de origem.

O limite para esse parâmetro é de um dia (86.400 segundos).

channel_name

O nome do canal de replicação na réplica de várias fontes. Cada canal de replicação recebe os eventos de logs binários de uma única instância de banco de dados do RDS para MySQL de origem executada em uma porta e um host específicos.

Observações de uso

O usuário principal deve executar o procedimento `mysql.rds_set_source_delay_for_channel`. Para usar o procedimento, primeiro chame `mysql.rds_stop_replication_for_channel` para interromper a replicação. Depois, chame esse procedimento para definir o valor do atraso de replicação. Quando o atraso estiver definido, chame `mysql.rds_start_replication_for_channel` para reiniciar a replicação.

Exemplos

O exemplo a seguir define o atraso da replicação da instância de banco de dados de origem no `channel_1` da réplica de várias fontes por pelo menos uma hora (3.600 segundos).

```
CALL mysql.rds_set_source_delay_for_channel(3600,'channel_1');
```

mysql.rds_skip_repl_error_for_channel

Ignora um evento de logs binários e exclui um erro de replicação em uma réplica de várias fontes de banco de dados do MySQL para o canal especificado.

Sintaxe

```
CALL mysql.rds_skip_repl_error_for_channel(channel_name);
```

Parâmetros

channel_name

O nome do canal de replicação na réplica de várias fontes. Cada canal de replicação recebe os eventos de logs binários de uma única instância de banco de dados do RDS para MySQL de origem executada em uma porta e um host específicos.

Observações de uso

O usuário principal deve executar o procedimento `mysql.rds_skip_repl_error_for_channel` em uma réplica de leitura. É possível usar esse procedimento praticamente da mesma forma que o `mysql.rds_skip_repl_error` é usado para ignorar um erro em uma réplica de leitura. Para ter mais informações, consulte Chamando o procedimento mysql.rds_skip_repl_error.

> **Note**
>
> Para ignorar erros na replicação baseada em GTID, recomendamos usar o procedimento the section called “mysql.rds_skip_transaction_with_gtid”.

Para determinar se há erros, execute o comando `SHOW REPLICA STATUS FOR CHANNEL 'channel_name'\G` do MySQL. Se um erro de replicação não for crítico, execute `mysql.rds_skip_repl_error_for_channel` para ignorá-lo. Se houver vários erros, `mysql.rds_skip_repl_error_for_channel` excluirá o primeiro erro no canal de replicação especificado e, depois, avisará que há outros. Assim, você pode usar `SHOW REPLICA STATUS FOR CHANNEL 'channel_name'\G` para determinar o plano de ação correto para o próximo erro. Para obter informações sobre os valores retornados, consulte Instrução SHOW REPLICA STATUS na documentação do MySQL.

mysql.rds_start_replication_for_channel

Inicia a replicação de uma instância de banco de dados do RDS para MySQL para uma réplica de várias fontes no canal especificado.

> **Note**
>
> Você pode usar o procedimento armazenado mysql.rds_start_replication_until_for_channel ou mysql.rds_start_replication_until_gtid_for_channel para iniciar a replicação de uma instância de banco de dados do RDS para MySQL interromper a replicação no local do arquivo de log binário especificado.

Sintaxe

```
CALL mysql.rds_start_replication_for_channel(channel_name);
```

Parâmetros

channel_name

O nome do canal de replicação na réplica de várias fontes. Cada canal de replicação recebe os eventos de logs binários de uma única instância de banco de dados do RDS para MySQL de origem executada em uma porta e um host específicos.

Observações de uso

O usuário principal deve executar o procedimento `mysql.rds_start_replication_for_channel`. Depois de importar os dados da instância de banco de dados do RDS para MySQL de origem, execute esse comando na réplica de várias fontes para iniciar a replicação no canal especificado.

Exemplos

O exemplo a seguir inicia a replicação em `channel_1` de várias fontes.

```
CALL mysql.rds_start_replication_for_channel('channel_1');
```

mysql.rds_start_replication_until_for_channel

Inicia a replicação de uma instância de banco de dados do RDS para MySQL no canal especificado e interrompe a replicação no local do arquivo de log binário especificado.

Sintaxe

```
CALL mysql.rds_start_replication_until_for_channel (
replication_log_file
  , replication_stop_point
  , channel_name
);
```

Parâmetros

replication_log_file

O nome do log binário na instância de banco de dados de origem contém as informações de replicação.

replication_stop_point

O local no log binário `replication_log_file` no qual a replicação será interrompida.

channel_name

O nome do canal de replicação na réplica de várias fontes. Cada canal de replicação recebe os eventos de logs binários de uma única instância de banco de dados do RDS para MySQL de origem executada em uma porta e um host específicos.

Observações de uso

O usuário principal deve executar o procedimento `mysql.rds_start_replication_until_for_channel`. Com esse procedimento, a replicação começa e depois é interrompida quando a posição especificada do arquivo de log binário é atingida. Esse procedimento interrompe tanto o `SQL_THREAD` quanto o `IO_THREAD`.

O nome do arquivo especificado para o parâmetro `replication_log_file` deve corresponder ao nome do arquivo do log binário da instância de banco de dados de origem.

Quando o parâmetro `replication_stop_point` especifica um local de parada no passado, a replicação é interrompida imediatamente.

Exemplos

O exemplo a seguir inicia a replicação em `channel_1` e replica as alterações até atingir o local `120` no arquivo de log binário `mysql-bin-changelog.000777`.

```
call mysql.rds_start_replication_until_for_channel(
  'mysql-bin-changelog.000777',
  120,
  'channel_1'
  );
```

mysql.rds_start_replication_until_gtid_for_channel

Inicia a replicação no canal especificado de uma instância de banco de dados do RDS para MySQL e interrompe a replicação no identificador de transação global (GTID) especificado.

Sintaxe

```
CALL mysql.rds_start_replication_until_gtid_for_channel(gtid,channel_name);
```

Parâmetros

`gtid`

O GTID após o qual a replicação será interrompida.

`channel_name`

O nome do canal de replicação na réplica de várias fontes. Cada canal de replicação recebe os eventos de logs binários de uma única instância de banco de dados do RDS para MySQL de origem executada em uma porta e um host específicos.

Observações de uso

O usuário principal deve executar o procedimento `mysql.rds_start_replication_until_gtid_for_channel`. O procedimento inicia a replicação no canal especificado e aplica todas as alterações até o valor de GTID especificado. Depois, ele interrompe a replicação no canal.

Quando o parâmetro `gtid` especifica uma transação que já tenha sido executada pela réplica, a replicação é interrompida imediatamente.

Antes de executar esse procedimento, você deve desabilitar a replicação multiencadeada definindo o valor de `replica_parallel_workers` ou `slave_parallel_workers` como `0`.

Exemplos

O exemplo a seguir inicia a replicação em `channel_1` e replica as alterações até atingir o GTID `3E11FA47-71CA-11E1-9E33-C80AA9429562:23`.

```
call mysql.rds_start_replication_until_gtid_for_channel('3E11FA47-71CA-11E1-9E33-
C80AA9429562:23','channel_1');
```

mysql.rds_stop_replication_for_channel

Interrompe a replicação de uma instância de banco de dados do MySQL no canal especificado.

Sintaxe

```
CALL mysql.rds_stop_replication_for_channel(channel_name);
```

Parâmetros

`channel_name`

O nome do canal de replicação na réplica de várias fontes. Cada canal de replicação recebe os eventos de logs binários de uma única instância de banco de dados do RDS para MySQL de origem executada em uma porta e um host específicos.

Observações de uso

O usuário principal deve executar o procedimento `mysql.rds_stop_replication_for_channel`.

Exemplos

O exemplo a seguir interrompe a replicação em `channel_1` da réplica de várias fontes.

```
CALL mysql.rds_stop_replication_for_channel('channel_1');
```

Replicar transações com GTIDs

Os procedimentos armazenados a seguir controlam como as transações são replicadas usando identificadores de transações globais (GTIDs) com o RDS para MySQL. Para ter mais informações sobre replicação com base em GTIDs com o RDS para MySQL, consulte Usar a replicação baseada em GTID.

Ao usar procedimentos armazenados para gerenciar a replicação com um usuário de replicação configurado com `caching_sha2_password`, você deve configurar o TLS especificando `SOURCE_SSL=1`. `caching_sha2_password` é o plugin de autenticação padrão do RDS para MySQL 8.4.

Tópicos

- mysql.rds_skip_transaction_with_gtid
- mysql.rds_start_replication_until_gtid

mysql.rds_skip_transaction_with_gtid

Ignora a replicação de uma transação com o identificador de transação global (GTID) especificado em uma instância de banco de dados MySQL.

Você pode usar esse procedimento para a recuperação de desastres, quando uma transação baseada em GTID específica for conhecida por causar desastres. Use esse procedimento armazenado para ignorar a transação problemática. Exemplos de transações problemáticas incluem transações que desabilitam a replicação, excluem dados importantes ou fazem com que a instância de banco de dados se torne indisponível.

Sintaxe

```
CALL mysql.rds_skip_transaction_with_gtid (
gtid_to_skip
);
```

Parâmetros

gtid_to_skip

O GTID da transação de replicação a ser ignorada.

Observações de uso

O usuário principal deve executar o procedimento `mysql.rds_skip_transaction_with_gtid`.

Esse procedimento é compatível com todas as versões 5.7 do RDS para MySQL, todas as versões 8.0 do RDS para MySQL e todas as versões do RDS para MySQL 8.4.

Exemplos

O exemplo a seguir ignora a replicação da transação com o GTID `3E11FA47-71CA-11E1-9E33-C80AA9429562:23`.

```
CALL mysql.rds_skip_transaction_with_gtid('3E11FA47-71CA-11E1-9E33-C80AA9429562:23');
```

mysql.rds_start_replication_until_gtid

Inicia a replicação de uma instância de banco de dados do RDS para MySQL e interrompe a replicação logo depois do identificador de transação global (GTID) especificado.

Sintaxe

```
CALL mysql.rds_start_replication_until_gtid(gtid);
```

Parâmetros

`gtid`

O GTID após o qual a replicação será interrompida.

Observações de uso

O usuário principal deve executar o procedimento `mysql.rds_start_replication_until_gtid`.

Esse procedimento é compatível com todas as versões 5.7 do RDS para MySQL, todas as versões 8.0 do RDS para MySQL e todas as versões do RDS para MySQL 8.4.

Você pode usar esse procedimento com a replicação atrasada para recuperação de desastres. Se você tiver a replicação atrasada configurada, use esse procedimento para efetuar roll forward de

alterações em uma réplica de leitura atrasada para o momento imediatamente antes de um desastre. Depois que esse procedimento interromper a replicação, é possível promover a réplica de leitura para ser a nova instância de banco de dados primária usando as instruções em Promoção de uma réplica de leitura a uma instância de banco de dados autônoma.

Você pode configurar a replicação atrasada usando os seguintes procedimentos armazenados:

- mysql.rds_set_configuration
- mysql.rds_set_external_master_with_delay (RDS para MariaDB e RDS para MySQL versões principais 8.0 e anteriores)
- mysql.rds_set_external_source_with_delay (RDS para MySQL versões principais 8.4 e superiores)
- mysql.rds_set_source_delay

Quando o parâmetro `gtid` especifica uma transação que já tenha sido executada pela réplica, a replicação é interrompida imediatamente.

Exemplos

O exemplo a seguir inicia a replicação e replica as alterações até atingir o GTID `3E11FA47-71CA-11E1-9E33-C80AA9429562:23`.

```
call mysql.rds_start_replication_until_gtid('3E11FA47-71CA-11E1-9E33-C80AA9429562:23');
```

Alternar os logs de consulta

Os procedimentos armazenados a seguir fazem a rotação dos logs do MySQL para tabelas de backup. Para ter mais informações, consulte Arquivos de log do banco de dados MySQL.

Tópicos

- mysql.rds_rotate_general_log
- mysql.rds_rotate_slow_log

mysql.rds_rotate_general_log

Reveza a tabela `mysql.general_log` com uma tabela de backup.

Sintaxe

```
CALL mysql.rds_rotate_general_log;
```

Observações de uso

Você pode revezar a tabela `mysql.general_log` com uma tabela de backup, chamando o procedimento `mysql.rds_rotate_general_log`. Quando as tabelas de log são revezadas, a tabela de log atual é copiada para uma tabela de log de backup e as entradas na tabela de log atual são removidas. Se uma tabela de log de backup já existir, então ela será excluída antes que a tabela de log atual seja copiada ao backup. Você pode consultar a tabela de log de backup, se necessário. A tabela de log de backup para a tabela `mysql.general_log` é denominada `mysql.general_log_backup`.

É possível executar esse procedimento somente quando o parâmetro `log_output` está definido como `TABLE`.

mysql.rds_rotate_slow_log

Reveza a tabela `mysql.slow_log` com uma tabela de backup.

Sintaxe

```
CALL mysql.rds_rotate_slow_log;
```

Observações de uso

Você pode revezar a tabela `mysql.slow_log` com uma tabela de backup, chamando o procedimento `mysql.rds_rotate_slow_log`. Quando as tabelas de log são revezadas, a tabela de log atual é copiada para uma tabela de log de backup e as entradas na tabela de log atual são removidas. Se uma tabela de log de backup já existir, então ela será excluída antes que a tabela de log atual seja copiada ao backup.

Você pode consultar a tabela de log de backup, se necessário. A tabela de log de backup para a tabela `mysql.slow_log` é denominada `mysql.slow_log_backup`.

Definir e mostrar a configuração de logs binários

Os procedimentos armazenados a seguir definem e mostram parâmetros de configuração, como para retenção de arquivos de log binários.

Tópicos

- mysql.rds_set_configuration
- mysql.rds_show_configuration

mysql.rds_set_configuration

Especifica o número de horas para retenção de logs binários ou o número de segundos para atrasar a replicação.

Sintaxe

```
CALL mysql.rds_set_configuration(name,value);
```

Parâmetros

name

(Opcional) O nome do parâmetro de configuração a definir.

value

O valor do parâmetro de configuração.

Observações de uso

O procedimento `mysql.rds_set_configuration` oferece suporte aos seguintes parâmetros de configuração:

- horas de retenção do log binário
- atraso na origem
- atraso de destino

Os parâmetros de configuração são armazenados permanentemente e sobrevivem a qualquer reinicialização ou failover da instância de banco de dados.

horas de retenção do log binário

O parâmetro `binlog retention hours` é usado para especificar o número de horas para reter arquivos de log binários. O Amazon RDS normalmente elimina um log binário o mais rápido possível, mas o log binário ainda pode ser necessário para a replicação com um banco de dados MySQL externo ao RDS.

O valor padrão de `binlog retention hours` é `NULL`. Para o RDS para MySQL, `NULL` significa que os logs binários não são retidos (0 horas).

Para especificar o número de horas para reter os logs binários em uma instância de banco de dados, use o procedimento armazenado `mysql.rds_set_configuration` e especifique um período com tempo suficiente para que a replicação ocorra, conforme exibido no exemplo a seguir.

```
call mysql.rds_set_configuration('binlog retention hours', 24);
```

> **Note**
>
> Não é possível usar o valor `0` para `binlog retention hours`.

Para instâncias de banco de dados MySQL, o valor máximo de `binlog retention hours` é de 168 (7 dias).

Após configurar o período de retenção, monitore o uso de armazenamento da instância de banco de dados para garantir que os logs binários retidos não consumam muito armazenamento.

Para implantações de cluster de banco de dados multi-AZ, você só pode configurar a retenção de log binário da instância de banco de dados de gravador, e a configuração é propagada para todas as instâncias de banco de dados de leitor de forma assíncrona. Se os logs binários no cluster de banco de dados excederem a metade do espaço total de armazenamento local, o Amazon RDS moverá automaticamente os logs obsoletos para o volume do EBS. No entanto, os logs mais recentes permanecem no armazenamento local e, portanto, poderão ser perdidos se houver uma falha que exija a substituição do host ou se você aumentar ou reduzir a escala do banco de dados verticalmente.

atraso na origem

Use o parâmetro `source delay` em uma réplica de leitura para especificar o número de segundos para atrasar a replicação da réplica de leitura para a instância de banco de dados de origem. O Amazon RDS replica as alterações o mais rápido possível, mas talvez você queira que alguns ambientes atrasem a replicação. Por exemplo, quando a replicação é atrasada, você pode efetuar roll forward de uma réplica de leitura atrasada para o momento imediatamente antes de um desastre. Se uma tabela for descartada acidentalmente, use a replicação atrasada para recuperá-la rapidamente. O valor padrão de `target delay` é `0` (não atrasar a replicação).

Quando você usa esse parâmetro, ele executa mysql.rds_set_source_delay e aplica CHANGE primary TO MASTER_DELAY = valor de entrada. Se for bem-sucedido, o procedimento salvará o parâmetro `source delay` na tabela `mysql.rds_configuration`.

Para especificar o número de segundos em que o Amazon RDS atrasará a replicação de uma instância de banco de dados de origem, use o procedimento armazenado `mysql.rds_set_configuration` e especifique o número de segundos de atraso da replicação. No exemplo a seguir, a replicação é atrasada por pelo menos uma hora (3.600 segundos).

```
call mysql.rds_set_configuration('source delay', 3600);
```

Depois, o procedimento executa `mysql.rds_set_source_delay(3600)`.

O limite do parâmetro `source delay` é de um dia (86.400 segundos).

atraso de destino

Use o parâmetro `target delay` para especificar o número de segundos para atrasar a replicação entre uma instância de banco de dados e quaisquer réplicas de leitura gerenciadas pelo RDS criadas futuramente com base nessa instância. Esse parâmetro é ignorado para réplicas de leitura não gerenciadas pelo RDS. O Amazon RDS replica as alterações o mais rápido possível, mas talvez você queira que alguns ambientes atrasem a replicação. Por exemplo, quando a replicação é atrasada, você pode efetuar roll forward de uma réplica de leitura atrasada para o momento imediatamente antes de um desastre. Se uma tabela for descartada acidentalmente, use a replicação atrasada para recuperá-la rapidamente. O valor padrão de `target delay` é `0` (não atrasar a replicação).

Para a recuperação de desastres, você pode usar esse parâmetro de configuração com o procedimento armazenado mysql.rds_start_replication_until ou o mysql.rds_start_replication_until_gtid. Para efetuar roll forward de alterações para uma

réplica de leitura atrasada no momento imediatamente antes de um desastre, é possível executar o procedimento `mysql.rds_set_configuration` com esse conjunto de parâmetros. Depois que o procedimento `mysql.rds_start_replication_until` ou `mysql.rds_start_replication_until_gtid` interromper a replicação, é possível promover a réplica de leitura para ser a nova instância de banco de dados primária usando as instruções em Promoção de uma réplica de leitura a uma instância de banco de dados autônoma.

Para usar o procedimento `mysql.rds_rds_start_replication_until_gtid`, a replicação baseada em GTID deve estar habilitada. Para ignorar uma transação específica baseada em GTID que seja conhecida por causar desastres, use o procedimento armazenado mysql.rds_skip_transaction_with_gtid. Para ter mais informações sobre como trabalhar com a replicação baseada em GTID, consulte Usar a replicação baseada em GTID.

Para especificar o número de segundos em que o Amazon RDS atrasará a replicação de uma réplica de leitura, use o procedimento armazenado `mysql.rds_set_configuration` e especifique o número de segundos de atraso da replicação. O exemplo a seguir especifica que a replicação é atrasada por pelo menos uma hora (3.600 segundos).

```
call mysql.rds_set_configuration('target delay', 3600);
```

O limite do parâmetro `target delay` é de um dia (86.400 segundos).

mysql.rds_show_configuration

A quantidade de horas em que os logs binários são mantidos.

Sintaxe

```
CALL mysql.rds_show_configuration;
```

Observações de uso

Para verificar o número de horas durante as quais o Amazon RDS vai reter os logs binários, use o procedimento armazenado `mysql.rds_show_configuration`.

Exemplos

O exemplo a seguir mostra exibe o período de retenção:

```
call mysql.rds_show_configuration;
```

```
                name                          value     description
                binlog retention hours        24        binlog retention hours specifies
the duration in hours before binary logs are automatically deleted.
```

Aquecer o armazenamento em cache do InnoDB

Os procedimentos armazenados a seguir salvam, carregam ou cancelam o carregamento do pool de buffers do InnoDB no RDS para instâncias de banco de dados MySQL. Para obter mais informações, consulte Aquecimento de cache do InnoDB para o MySQL no Amazon RDS.

Tópicos

- mysql.rds_innodb_buffer_pool_dump_now
- mysql.rds_innodb_buffer_pool_load_abort
- mysql.rds_innodb_buffer_pool_load_now

mysql.rds_innodb_buffer_pool_dump_now

Descarrega o estado atual do grupo de buffer no disco.

Sintaxe

```
CALL mysql.rds_innodb_buffer_pool_dump_now();
```

Observações de uso

O usuário mestre deve executar o procedimento `mysql.rds_innodb_buffer_pool_dump_now`.

mysql.rds_innodb_buffer_pool_load_abort

Cancela um carregamento do estado salvo do grupo de buffer em andamento.

Sintaxe

```
CALL mysql.rds_innodb_buffer_pool_load_abort();
```

Observações de uso

O usuário mestre deve executar o procedimento `mysql.rds_innodb_buffer_pool_load_abort`.

mysql.rds_innodb_buffer_pool_load_now

Carrega o estado salvo do grupo de buffer a partir do disco.

Sintaxe

```
CALL mysql.rds_innodb_buffer_pool_load_now();
```

Observações de uso

O usuário mestre deve executar o procedimento `mysql.rds_innodb_buffer_pool_load_now`.

Amazon RDS para Oracle

O Amazon RDS é compatível com instâncias de banco de dados que executam as seguintes versões e edições do Oracle Database:

- Oracle Database 21c (21.0.0.0)
- Oracle Database 19c (19.0.0.0)

> **Note**
>
> O Oracle Database 11g, o Oracle Database 12c e o Oracle Database 18c são versões herdadas que não são mais compatíveis com o Amazon RDS.

Antes de criar uma instância de banco de dados, é necessário concluir as etapas na seção Configuração do ambiente do Amazon RDS deste guia. Ao criar uma instância de banco de dados usando sua conta mestra, a conta obtém privilégios de DBA, com algumas limitações. Use essa conta para tarefas administrativas, como a criação de contas de banco de dados adicionais. Não é possível usar SYS, SYSTEM ou outras contas administrativas fornecidas pelo Oracle.

Você pode criar o seguinte:

- Instâncias de banco de dados
- DB snapshots
- Restaurações point-in-time
- Backups automatizados
- Backups manuais

É possível usar instâncias de banco de dados que executam o Oracle Database em uma VPC. Também é possível adicionar recursos à instância de banco de dados habilitando várias opções, como Oracle Spatial ou Oracle Statspack. O Amazon RDS oferece suporte para implantações Multi-AZ para Oracle como uma solução de failover de alta disponibilidade.

> **Important**
>
> Para oferecer uma experiência de serviço gerenciada, o Amazon RDS não fornece acesso ao shell para as instâncias de banco de dados. Também restringe o acesso a determinados procedimentos e tabelas do sistema que necessitam de privilégios avançados. É possível acessar seu banco de dados usando qualquer aplicação cliente SQL padrão, como o Oracle SQL*Plus. No entanto, não é possível acessar o host diretamente usando Telnet ou Secure Shell (SSH).

Tópicos

- Visão geral do Oracle no Amazon RDS
- Conectar-se a uma instância de banco de dados Oracle
- Proteção das conexões de instância de banco de dados Oracle
- Trabalhar com CDBs no RDS para Oracle
- Administração da sua instância de banco de dados Oracle
- Configurar recursos avançados do RDS para Oracle
- Importar dados para o Oracle no Amazon RDS
- Trabalhar com réplicas de leitura do Amazon RDS para Oracle
- Adição de opções a instâncias de banco de dados Oracle
- Fazer upgrade do mecanismo de banco de dados RDS para Oracle
- Usar software de terceiros com a instância de banco de dados do Oracle
- Notas de lançamento do mecanismo Oracle Database

Visão geral do Oracle no Amazon RDS

Você pode ler as seções a seguir para obter uma visão geral do RDS para Oracle.

Tópicos

- Recursos do RDS for Oracle
- Versões do RDS para Oracle
- Opções de licenciamento do RDS para Oracle
- Usuários e privilégios do RDS para Oracle

- Classes de instância de banco de dados do RDS para Oracle
- Arquitetura de banco de dados RDS para Oracle
- Parâmetros de inicialização do RDS para Oracle
- Conjuntos de caracteres do RDS for Oracle
- Limitações do RDS para Oracle

Recursos do RDS for Oracle

O Amazon RDS for Oracle oferece suporte à maioria dos recursos e funcionalidades do banco de dados Oracle. Alguns recursos podem ter suporte limitado ou privilégios restritos. Alguns recursos só estão disponíveis no Enterprise Edition, enquanto outros exigem licenças adicionais. Para ter mais informações sobre os recursos do Oracle Database para versões específicas do Oracle Database, consulte o documento Oracle Database Licensing Information User Manual da versão que você está usando.

Tópicos

- Novos recursos no RDS for Oracle
- Recursos com suporte no RDS for Oracle
- Recursos sem suporte no RDS for Oracle

Novos recursos no RDS for Oracle

Para ver os novos recursos no RDS para Oracle, pesquise a palavra-chave **`Oracle`** em Histórico do documento.

Recursos com suporte no RDS for Oracle

O Amazon RDS para Oracle oferece suporte aos seguintes recursos do Oracle Database:

> **Note**
>
> A lista a seguir não é completa.

- Compressão avançada

- Oracle Application Express (APEX)

  Para obter mais informações, consulte Oracle Application Express (APEX).
- Gerenciamento automatizado da memória
- Gerenciamento automático de operações Desfazer
- AWR (Automatic Workload Repository)

  Para ter mais informações, consulte Gerar relatórios de performance com o AWR (Automatic Workload Repository).
- Ativar o Data Guard com performance máxima na mesma região da AWS ou entre regiões da AWS

  Para ter mais informações, consulte Trabalhar com réplicas de leitura do Amazon RDS para Oracle.
- Tabelas blockchain (Oracle Database 21c e posterior)

  Para ter mais informações, consulte Managing Blockchain Tables (Gerenciar tabelas blockchain) na documentação do Oracle Database.
- Notificação de consulta contínua

  Para ter mais informações, consulte Usar notificação de consulta contínua (CQN) na documentação da Oracle.
- Redação de Dados
- Notificação de consulta contínua

  Para ter mais informações, consulte Notificação de alteração do banco de dados na documentação da Oracle.
- Banco de dados na memória
- Consultas e transações distribuídas
- Redefinição baseada em edição

  Para ter mais informações, consulte Definir a edição padrão para uma instância de banco de dados.
- EM Express (12c e posterior)

  Para ter mais informações, consulte Oracle Enterprise Manager.

- Auditoria granular
- Tabela Flashback, consulta Flashback, consulta de transação Flashback
- Sobreposição gradual de senhas para aplicações (Oracle Database 21c e posterior)

  Para ter mais informações, consulte Managing Gradual Database Password Rollover for Applications (Gerenciar a sobreposição gradual de senhas de banco de dados para aplicações) na documentação do Oracle Database.
- HugePages

  Para ter mais informações, consulte Ativar o HugePages para uma instância do RDS para Oracle.
- Importação/exportação (legadas e Data Pump) e SQL*Loader

  Para ter mais informações, consulte Importar dados para o Oracle no Amazon RDS.
- Java Virtual Machine (JVM)

  Para ter mais informações, consulte Oracle Java Virtual Machine.
- JavaScript (Oracle Database 21c e posterior)

  Para ter mais informações, consulte DBMS_MLE na documentação do Oracle Database.
- Label Security

  Para obter mais informações, consulte Oracle Label Security.
- Localizador

  Para ter mais informações, consulte Oracle Locator.
- Visualizações materializadas
- Multilocatário

  A arquitetura multilocatário da Oracle é compatível com o Oracle Database 19c e todas as versões posteriores. Para obter mais informações, consulte Trabalhar com CDBs no RDS para Oracle.
- Criptografia de rede

  Para ter mais informações, consulte Oracle Native Network Encryption e Oracle Secure Sockets Layer.
- Particionamento
- Testes de aplicativos reais

Para usar os recursos completos de captura e reprodução, você deve usar o Amazon Elastic File System (Amazon EFS) para acessar arquivos gerados pelo Oracle Real Application Testing. Consulte mais informações em Integração do Amazon EFS e na publicação do blog Use Oracle Real Application Testing features with Amazon RDS for Oracle.

- Fragmentação no nível da aplicação (mas não o recurso de fragmentação do Oracle)
- Espacial e gráfico

  Para ter mais informações, consulte Oracle Spatial.
- Iniciar a otimização de consultas
- Streams e enfileiramento avançado
- Gerenciamento de resumos – Regravação de consulta de visualização materializada
- Texto (não há suporte para tipos de datastore de arquivo e URL)
- Recuperação total
- Transparent Data Encryption (TDE)

  Para ter mais informações, consulte Oracle Transparent Data Encryption.
- Auditoria unificada, modo misto

  Para ter mais informações, consulte Auditoria de modo misto na documentação da Oracle.
- XML DB (sem o XML DB Protocol Server)

  Para ter mais informações, consulte Oracle XML DB.
- Banco de dados privado virtual

Recursos sem suporte no RDS for Oracle

O Amazon RDS para Oracle não oferece suporte aos seguintes recursos do Oracle Database:

> **Note**
>
> A lista a seguir não é completa.

- Automatic Storage Management (ASM)

- Database Vault
- Banco de dados Flashback

> **Note**
>
> Para obter soluções alternativas, consulte a entrada do blog de banco de dados Alternativas ao atributo de banco de dados flashback Oracle no Amazon RDS para Oracle da AWS.

- FTP e SFTP
- Tabelas particionadas híbridas
- Gateway de mensagens
- Repositório de gerenciamento do Oracle Enterprise Manager Cloud Control
- Real Application Clusters (Oracle RAC)
- Real Application Security (RAS)
- Auditoria unificada, Modo puro
- Esquema do Workspace Manager (WMSYS)

> **Warning**
>
> Em geral, o Amazon RDS não impede que você crie esquemas para recursos que não têm suporte. No entanto, se você criar esquemas para recursos e componentes Oracle que exigem privilégios SYSDBA, poderá danificar o dicionário de dados e afetar a disponibilidade da instância de banco de dados. Use apenas recursos e esquemas com suporte que estão disponíveis em Adição de opções a instâncias de banco de dados Oracle.

Versões do RDS para Oracle

O RDS para Oracle é compatível com várias versões do Oracle Database.

> **Note**
>
> Para obter informações sobre como atualizar suas versões, consulte Fazer upgrade do mecanismo de banco de dados RDS para Oracle.

Tópicos

- Oracle Database 21c com o Amazon RDS
- Oracle Database 19c com Amazon RDS

Oracle Database 21c com o Amazon RDS

O Amazon RDS é compatível com o Oracle Database 21c, que inclui o Oracle Enterprise Edition e o Oracle Standard Edition 2. O Oracle Database 21c (21.0.0.0) inclui muitos recursos novos e atualizações da versão anterior. Uma alteração importante é que o Oracle Database 21c é compatível somente com a arquitetura multilocatário: você não pode mais criar um banco de dados como um não CDB tradicional. Para saber mais sobre as diferenças entre CDBs e não CDBs, consulte Limitações do RDS for CDBs Oracle.

Nesta seção, você pode encontrar os recursos e as alterações que são importantes para usar o Oracle Database 21c (21.0.0.0) no Amazon RDS. Para obter uma lista completa das alterações, consulte a documentação do Oracle Database 21c. Para obter uma lista completa de recursos compatíveis com cada edição do Oracle Database 21c, consulte Permitted Features, Options, and Management Packs by Oracle Database Offering (Recursos, opções e pacotes de gerenciamento permitidos pela oferta Oracle Database) na documentação do Oracle.

Alterações em parâmetros do Amazon RDS for Oracle Database 21c (21.0.0.0)

O Oracle Database 21c (21.0.0.0) inclui vários parâmetros novos, além de parâmetros com novos intervalos e novos valores padrão.

Tópicos

- Novos parâmetros
- Alterações no parâmetro compatível
- Parâmetros removidos

Novos parâmetros

A tabela a seguir mostra os novos parâmetros do Amazon RDS for Oracle Database 21c (21.0.0.0).

Nome	Intervalo de valores	Valor padrão	Modificável	Descrição
blockchain_table_max_no_drop	NONE | 0	NONE	Y	Permite controlar a quantidade máxima de tempo ocioso que pode ser especificado ao criar uma tabela blockchain.
dbnest_enable	NONE | CDB_RESOURCE_PDB_ALL	NONE	N	Permite habilitar ou desabilitar o dbNest. O dbNest fornece isolamento e gerenciamento de recursos do sistema operacional, isolamento do sistema de arquivos e computação segura para PDBs.
dbnest_pdb_fs_conf	NONE | *pathname*	NONE	N	Especifica o arquivo de configuração do sistema de arquivos do dbNest para um PDB.
diagnostics_control	ERROR | WARNING | IGNORE	IGNORE	Y	Permite controlar e monitorar os usuários que executam operações de diagnóstico de banco de dados possivelmente inseguras.
drcp_dedicated_opt	YES | NO	YES	Y	Habilita ou desabilita o uso de otimização dedicada com o Database Resident Connection Pooling (DRCP).
enable_per_pdb_drcp	true | false	true	N	Controla se o Database Resident Connection Pooling (DRCP) configura um grupo de conexões para todo o CDB ou um grupo de

Nome	Intervalo de valores	Valor padrão	Modificável	Descrição
				conexões isolado para cada PDB.
inmemory_deep_vectorization	`true | false`	`true`	Y	Habilita ou desabilita o framework de vetorização profunda.
mandatory_user_profile	*`profile_name`*	N/D	N	Especifica o perfil de usuário obrigatório para um CDB ou PDB.
optimizer_capture_sql_quarantine	`true | false`	`false`	Y	Habilita ou desabilita o framework de vetorização profunda.
optimizer_use_sql_quarantine	`true | false`	`false`	Y	Habilita ou desabilita a criação automática de configurações do SQL Quarantine.
result_cache_execution_threshold	`0` para `68719476736`	`2`	Y	Especifica o número máximo de vezes que uma função do PL/SQL pode ser executada antes que o respectivo resultado seja armazenado no cache de resultados.
result_cache_max_temp_result	`0` para `100`	`5`	Y	Especifica a porcentagem de `RESULT_CACHE_MAX_TEMP_SIZE` que qualquer resultado de consulta armazenado em cache único pode consumir.

Nome	Intervalo de valores	Valor padrão	Modificável	Descrição
result_cache_max_temp_size	0 para 2199023255552	RESULT_CACHE_SIZE * 10	Y	Especifica a quantidade máxima de espaço de tabela temporário (em bytes) que pode ser consumida pelo cache de resultados.
sga_min_size	De 0 a 2199023255552 (o valor máximo é 50% de sga_target)	0	Y	Indica um possível valor mínimo para o uso da SGA de um banco de dados conectável (PDB).
tablespace_encryption_default_algorithm	GOST256 | SEED128 | ARIA256 | ARIA192 | ARIA128 | 3DES168 | AES256 | AES192 | AES128	AES128	Y	Especifica o algoritmo padrão que o banco de dados usa ao criptografar um espaço de tabela.

Alterações no parâmetro compatível

O parâmetro `compatible` tem um novo valor máximo para o Oracle Database 21c (21.0.0.0) no Amazon RDS. A tabela a seguir mostra o novo valor padrão.

Nome do parâmetro	Valor máximo do Oracle Database 21c (21.0.0.0)
compatible	21.0.0

Parâmetros removidos

Os seguintes parâmetros foram removidos do Oracle Database 21c (21.0.0.0):

- `remote_os_authent`
- `sec_case_sensitive_logon`
- `unified_audit_sga_queue_size`

Oracle Database 19c com Amazon RDS

O Amazon RDS oferece suporte para o Oracle Database 19c, que inclui o Oracle Enterprise Edition e o Oracle Standard Edition Two.

O Oracle Database 19c (19.0.0.0) inclui muitos recursos novos e atualizações da versão anterior. Nesta seção, encontre os recursos e as alterações importantes feitas no uso do Oracle Database 19c (19.0.0.0) no Amazon RDS. Para obter uma lista completa das alterações, consulte a documentação do Oracle Database 19c. Para obter uma lista completa de recursos compatíveis com cada edição do Oracle Database 19c, consulte Permitted Features, Options, and Management Packs by Oracle Database Offering (Recursos, opções e pacotes de gerenciamento permitidos pela oferta Oracle Database) na documentação do Oracle.

Alterações em parâmetros do Amazon RDS for Oracle Database 19c (19.0.0.0)

O Oracle Database 19c (19.0.0.0) inclui vários parâmetros novos, além de parâmetros com novos intervalos e novos valores padrão.

Tópicos

- Novos parâmetros
- Alterações no parâmetro compatível
- Parâmetros removidos

Novos parâmetros

A tabela a seguir mostra os novos parâmetros do Amazon RDS for Oracle Database 19c (19.0.0.0).

Nome	Valores	Permit modifi ão	Descrição
lob_signature_enable	TRUE, FALSE (padrão)	Y	Habilita ou desabilita o recurso de assinatura do localizador LOB.
max_datapump_parallel_per_job	1 a 1024, ou AUTO	Y	Especifica o número máximo de processos paralelos permitidos para cada trabalho do Oracle Data Pump.

Alterações no parâmetro compatível

O parâmetro `compatible` tem um novo valor máximo para o Oracle Database 19c (19.0.0.0) no Amazon RDS. A tabela a seguir mostra o novo valor padrão.

Nome do parâmetro	Valor máximo do Oracle Database 19c (19.0.0.0)
compatible	19.0.0

Parâmetros removidos

Os seguintes parâmetros foram removidos no Oracle Database 19c (19.0.0.0):

- `exafusion_enabled`
- `max_connections`
- `o7_dictionary_access`

Opções de licenciamento do RDS para Oracle

O Amazon RDS for Oracle tem duas opções de licenciamento: Licença inclusa (LI) e Traga sua própria licença (BYOL). Depois de criar uma instância de banco de dados Oracle no Amazon RDS, é possível alterar o modelo de licenciamento ao modificar a instância de banco de dados. Para obter mais informações, consulte Modificar uma instância de banco de dados do Amazon RDS.

> **Important**
>
> Verifique se você tem a licença apropriada do Oracle Database, com Software Update License and Support, para a classe da instância de banco de dados e a edição do Oracle Database. Além disso, verifique se você tem licenças para todos os recursos do Oracle Database licenciados separadamente.

Tópicos

- Modelo de licença incluída para SE2
- Traga a sua própria licença (BYOL) para EE e SE2
- Licenciar implantações Multi-AZ Oracle

Modelo de licença incluída para SE2

No modelo de licença inclusa, você não precisa comprar licenças do Oracle Database separadamente. O AWS tem licença para o software de banco de dados Oracle. O modelo Licença inclusa só é aceito no Amazon RDS para Oracle Database Standard Edition 2 (SE2).

Nesse modelo, se você tiver uma conta do AWS Support com suporte para casos, entre em contato com o Suporte para solicitações de serviço do Amazon RDS e do Oracle Database. O uso da opção LI do RDS para Oracle está sujeito à Seção 10.3.1 dos Termos de serviço da AWS.

Traga a sua própria licença (BYOL) para EE e SE2

No modelo BYOL, é possível usar suas licenças existentes do Oracle Database para implantar bancos de dados no Amazon RDS. O Amazon RDS comporta o modelo BYOL somente para Oracle Database Enterprise Edition (EE) e o Oracle Database Standard Edition 2 (SE2).

Verifique se você tem a licença apropriada do Oracle Database (com Software Update License and Support) para a classe da instância de banco de dados e a edição do Oracle Database que deseja executar. Você também deve seguir as políticas da Oracle para o licenciamento do software Oracle Database no ambiente de computação em nuvem. Para obter mais informações sobre a política de licenciamento da Oracle para o Amazon EC2, consulte Licenciamento de softwares Oracle no ambiente de computação em nuvem.

Nesse modelo, você continua a usar sua conta de suporte ativo do Oracle e entra em contato com a Oracle diretamente para solicitações de serviço do Oracle Database. Caso tenha uma conta do

AWS Support com suporte a caso, você pode entrar em contato com o Suporte para solucionar problemas do Amazon RDS. A Amazon Web Services e o Oracle têm um processo de suporte de vários fornecedores para casos que necessitem de assistência de ambas as organizações.

Integração com o AWS License Manager

Para facilitar o monitoramento do uso de licenças Oracle no modelo BYOL, o AWS License Manager integra-se ao Amazon RDS for Oracle. O License Manager oferece suporte ao rastreamento de edições de mecanismos e pacotes de licenças do RDS for Oracle com base em núcleos virtuais (vCPUs). Você também pode usar o License Manager com o AWS Organizations para gerenciar centralmente todas as suas contas organizacionais.

A tabela a seguir mostra os filtros de informações de produtos do RDS para Oracle.

Filtro	Nome	Descrição
Edição do mecanismo	`oracle-ee`	Oracle Database Enterprise Edition (EE)
	`oracle-se2`	Oracle Database Standard Edition 2 (SE2)
Pacote de licenças	`data guard`	Consulte Trabalhar com réplicas de leitura do Amazon RDS para Oracle (Oracle Active Data Guard)
	`olap`	Consulte Oracle OLAP
	`ols`	Consulte Oracle Label Security
	`diagnostic pack sqlt`	Consulte Oracle SQLT
	`tuning pack sqlt`	Consulte Oracle SQLT

Para rastrear o uso da licença das instâncias de banco de dados do Oracle, é possível criar uma licença autogerenciada utilizando o AWS License Manager. Nesse caso, os recursos do RDS para Oracle que correspondem ao filtro de informações do produto são automaticamente associados à licença autogerenciada. A detecção de instâncias de banco de dados Oracle pode levar até 24 horas. Também é possível rastrear uma licença entre contas usando o AWS Resource Access Manager.

Console

Como criar uma licença autogerenciada no AWS License Manager para rastrear o uso de licença das instâncias de banco de dados do Oracle

1. Acesse https://console.aws.amazon.com/license-manager/.
2. Escolha Criar licença autogerenciada.

   Consulte instruções em Create a self-managed license no Guia do usuário do AWS License Manager.

   Adicione uma regra para um RDS Product Information Filter (Filtro de informações de produto do RDS) no painel Product Information (Informações do produto).

   Para obter mais informações, consulte ProductInformation na Referência da API do AWS License Manager.
3. (Somente rastreamento entre contas) Use o AWS Resource Access Manager para compartilhar suas licenças autogerenciadas com qualquer conta da AWS ou por meio do AWS Organizations. Para mais informações, consulte Compartilhar seus recursos da AWS.

AWS CLI

Para criar uma licença autogerenciada usando a AWS CLI, chame o comando create-license-configuration. Use os parâmetros `--cli-input-json` ou `--cli-input-yaml` para passar os parâmetros para o comando.

Example

O exemplo a seguir cria uma licença autogerenciada para o Oracle Enterprise Edition.

```
aws license-manager create-license-configuration --cli-input-json file://rds-oracle-
ee.json
```

Veja a seguir o arquivo `rds-oracle-ee.json` de exemplo usado no exemplo.

```
{
    "Name": "rds-oracle-ee",
    "Description": "RDS Oracle Enterprise Edition",
    "LicenseCountingType": "vCPU",
    "LicenseCountHardLimit": false,
```

```
    "ProductInformationList": [
        {
            "ResourceType": "RDS",
            "ProductInformationFilterList": [
                {
                    "ProductInformationFilterName": "Engine Edition",
                    "ProductInformationFilterValue": ["oracle-ee"],
                    "ProductInformationFilterComparator": "EQUALS"
                }
            ]
        }
    ]
}
```

Para obter mais informações sobre o produto, consulte Detecção automática do inventário de recursos no Guia do usuário do AWS License Manager.

Para obter mais informações sobre o parâmetro `--cli-input`, consulte Gerar um esqueleto AWS CLI e parâmetros de entrada usando um arquivo de entrada JSON ou YAML no AWS CLI Guia do usuário .

Migrar entre edições do Oracle Database

Se você tiver uma licença BYOL do Oracle Database não utilizada e apropriada para a edição e a classe da instância de banco de dados que planeja executar, poderá migrar da Standard Edition 2 (SE2) para a Enterprise Edition (EE). Não é possível migrar da EE para outras edições.

Como alterar a edição do Oracle Database e reter seus dados

1. Crie um snapshot da instância de banco de dados.

   Para obter mais informações, consulte Criar um snapshot de banco de dados para uma instância de banco de dados single-AZ para o Amazon RDS.
2. Restaure o snapshot para uma nova instância de banco de dados e selecione a edição de banco de dados Oracle que deseja usar.

   Para obter mais informações, consulte Restaurar uma instância de banco de dados.
3. (Opcional) Exclua a antiga instância de banco de dados, a não ser que deseje mantê-la em execução, e tenha as licenças apropriadas do banco de dados Oracle para ela.

   Para obter mais informações, consulte Excluir uma instância de banco de dados.

Licenciar implantações Multi-AZ Oracle

O Amazon RDS oferece suporte para implantações Multi-AZ para Oracle como uma solução de failover de alta disponibilidade. Recomendamos o Multi-AZ para workloads de produção. Para obter mais informações, consulte Configurar e gerenciar uma implantação multi-AZ para o Amazon RDS.

Se você usar o modelo Traga sua própria licença, deverá ter uma licença para a instância de banco de dados primária e a instância de banco de dados em espera em uma implantação Multi-AZ.

Usuários e privilégios do RDS para Oracle

Quando você cria uma instância de banco de dados do Amazon RDS para Oracle, o usuário principal padrão tem a maioria das permissões máximas de usuário na instância de banco de dados. Use a conta de usuário principal para qualquer tarefa administrativa, como criar contas do usuário adicionais no banco de dados. Como o RDS é um serviço gerenciado, você não tem permissão para fazer login como `SYS` e `SYSTEM` e, portanto, não tem privilégios `SYSDBA`.

Tópicos

- Limitações para privilégios do Oracle DBA
- Como gerenciar privilégios em objetos SYS

Limitações para privilégios do Oracle DBA

Uma função é uma coleção de privilégios que podem ser concedidos ou revogados de um usuário. Um banco de dados Oracle usa funções para fornecer segurança. Para receber mais informações, consulte Configuring Privilege and Role Authorization na documentação do Oracle Database.

Uma função predefinida, chamada de `DBA`, normalmente fornece todos os privilégios administrativos em um banco de dados Oracle. Ao criar uma instância de banco de dados, sua conta de usuário mestra obtém privilégios de DBA (com algumas limitações). Para fornecer uma experiência gerenciada, um banco de dados do RDS for Oracle não fornece os seguintes privilégios para a função `DBA`:

- `ALTER DATABASE`
- `ALTER SYSTEM`
- `CREATE ANY DIRECTORY`
- `DROP ANY DIRECTORY`
- `GRANT ANY PRIVILEGE`

- `GRANT ANY ROLE`

Para receber mais informações sobre privilégios e perfis do sistema RDS para Oracle, consulte Privilégios da conta de usuário mestre.

Como gerenciar privilégios em objetos SYS

Você pode gerenciar privilégios em objetos `SYS` usando o pacote `rdsadmin.rdsadmin_util`. Por exemplo, se você criar o usuário do banco de dados `myuser`, poderá usar o procedimento `rdsadmin.rdsadmin_util.grant_sys_object` para conceder privilégios `SELECT` em `V_$SQLAREA` ao `myuser`. Para obter mais informações, consulte os tópicos a seguir:

- Concessão de privilégios SELECT ou EXECUTE a objetos SYS
- Revogar privilégios SELECT ou EXECUTE em objetos SYS
- Concessão de privilégios a usuários não mestres

Classes de instância de banco de dados do RDS para Oracle

A capacidade de computação e memória de uma instância de banco de dados do RDS para Oracle é determinada de acordo com a classe de instância. A classe de instância de banco de dados da qual você precisa depende dos requisitos de energia e memória de processamento.

Classes de instância de banco de dados compatíveis do RDS para Oracle

As classes de instância compatíveis com o RDS para Oracle são um subconjunto das classes de instância de banco de dados do RDS. Para obter a lista completa de classes de instância do RDS, consulte Classes de instâncias de banco de dados .

Classes de instâncias de banco de dados pré-configuradas do RDS para Oracle

O RDS para Oracle também oferece classes de instância pré-configuradas para workloads que necessitem de mais memória, armazenamento e E/S por vCPU. Essas classes de instância usam a seguinte convenção de nomenclatura:

```
db.r5b.instance_size.tpcthreads_per_core.memratio
db.r5.instance_size.tpcthreads_per_core.memratio
```

Veja a seguir um exemplo de classe de instância pré-configurada para memória adicional:

```
db.r5b.4xlarge.tpc2.mem2x
```

Os componentes do nome da classe de instância anterior são os seguintes:

- `db.r5b.4xlarge`: o nome da classe de instância.
- `tpc2`: os threads por núcleo. O valor 2 significa que o multithreading está ativado. O valor 1 significa que o multithreading está ativado.
- `mem2x`: a proporção de memória adicional à memória padrão para a classe de instância. Nesse exemplo, a otimização fornece o dobro de memória que uma instância de banco de dados db.r5.4xlarge padrão.

Combinações de edição, classe de instância e licenciamento compatíveis no RDS para Oracle

Se você estiver usando o console do RDS, poderá descobrir se uma combinação específica de edição, classe de instância e licença é compatível escolhendo Criar banco de dados e especificando uma opção diferente. Na AWS CLI, é possível executar o seguinte comando:

```
aws rds describe-orderable-db-instance-options --engine engine-type --license-
model license-type
```

A tabela a seguir lista todas as edições, classes de instância e tipos de licença compatíveis com o RDS para Oracle. Para obter mais informações sobre os atributos de memória de cada tipo, consulte Tipos de instância do RDS para Oracle. Consulte mais informações sobre preços em Amazon RDS for Oracle pricing models.

Edição do Oracle	Oracle Database 19c e posterior
Enterprise Edition (EE)	Classes de instância padrão
Traga a sua própria licença (BYOL)	db.m7i.large–db.m7i.48xlarge db.m6id.large–db.m6id.32xlarge db.m6i.large–db.m6i.32xlarge db.m5d.large a db.m5d.24xlarge

Edição do Oracle	Oracle Database 19c e posterior
	db.m5.large – db.m5.24xlarge
	Classes de instância otimizada para memória
	db.r7i.large–db.r7i.48xlarge db.r6id.large–db.r6id.32xlarge db.r6i.large–db.r6i.32xlarge (somente 19c) db.r5d.large a db.r5d.24xlarge db.r5b.large–db.r5b.24xlarge db.r5.large–db.r5.24xlarge db.x2iedn.xlarge–db.x2iedn.32xlarge db.x2iezn.2xlarge–db.x2iezn.12xlarge db.x2idn.16xlarge–db.x2idn.32xlarge db.x1e.xlarge–db.x1e.32xlarge db.x1.16xlarge–db.x1.32xlarge[1] db.z1d.large–db.z1d.12xlarge
	Classes de instância pré-configuradas otimizadas para memória

Edição do Oracle	Oracle Database 19c e posterior
	db.r6i.8xlarge.tpc2.mem4x
	db.r6i.8xlarge.tpc2.mem3x
	db.r6i.6xlarge.tpc2.mem4x
	db.r6i.4xlarge.tpc2.mem4x
	db.r6i.4xlarge.tpc2.mem3x
	db.r6i.4xlarge.tpc2.mem2x
	db.r6i.2xlarge.tpc2.mem8x
	db.r6i.2xlarge.tpc2.mem4x
	db.r6i.2xlarge.tpc1.mem2x
	db.r6i.xlarge.tpc2.mem4x
	db.r6i.xlarge.tpc2.mem2x
	db.r6i.large.tpc1.mem2x
	db.r5b.8xlarge.tpc2.mem3x
	db.r5b.6xlarge.tpc2.mem4x
	db.r5b.4xlarge.tpc2.mem4x
	db.r5b.4xlarge.tpc2.mem3x
	db.r5b.4xlarge.tpc2.mem2x
	db.r5b.2xlarge.tpc2.mem8x
	db.r5b.2xlarge.tpc2.mem4x
	db.r5b.2xlarge.tpc1.mem2x
	db.r5b.xlarge.tpc2.mem4x

Edição do Oracle	Oracle Database 19c e posterior
	db.r5b.xlarge.tpc2.mem2x
	db.r5b.large.tpc1.mem2x
	db.r5.12xlarge.tpc2.mem2x
	db.r5.8xlarge.tpc2.mem3x
	db.r5.6xlarge.tpc2.mem4x
	db.r5.4xlarge.tpc2.mem4x
	db.r5.4xlarge.tpc2.mem3x
	db.r5.4xlarge.tpc2.mem2x
	db.r5.2xlarge.tpc2.mem8x
	db.r5.2xlarge.tpc2.mem4x
	db.r5.2xlarge.tpc1.mem2x
	db.r5.xlarge.tpc2.mem4x
	db.r5.xlarge.tpc2.mem2x
	db.r5.large.tpc1.mem2x
	Classes de instância expansível
	db.t3.small–db.t3.2xlarge

Edição do Oracle	Oracle Database 19c e posterior
Standard Edition 2 (SE2)	Classes de instância padrão
Traga a sua própria licença (BYOL)	db.m7i.large–db.m7i.4xlarge db.m6id.large–db.m6id.4xlarge db.m6i.large–db.m6i.4xlarge db.m5d.large a db.m5d.4xlarge db.m5.large – db.m5.4xlarge
	Classes de instância otimizada para memória
	db.r7i.large–db.r7i.4xlarge db.r6id.large–db.r6id.4xlarge db.r6i.large–db.r6i.4xlarge db.r5d.large a db.r5d.4xlarge db.r5b.large–db.r5b.4xlarge db.r5.large–db.r5.4xlarge db.x2iedn.xlarge–db.x2iedn.4xlarge db.x2iezn.2xlarge–db.x2iezn.4xlarge db.z1d.large–db.z1d.3xlarge
	Classes de instância pré-configuradas otimizadas para memória

Edição do Oracle	Oracle Database 19c e posterior
	db.r6i.4xlarge.tpc2.mem4x
	db.r6i.4xlarge.tpc2.mem3x
	db.r6i.4xlarge.tpc2.mem2x
	db.r6i.2xlarge.tpc2.mem8x
	db.r6i.2xlarge.tpc2.mem4x
	db.r6i.2xlarge.tpc1.mem2x
	db.r6i.xlarge.tpc2.mem4x
	db.r6i.xlarge.tpc2.mem2x
	db.r6i.large.tpc1.mem2x
	db.r5b.4xlarge.tpc2.mem4x
	db.r5b.4xlarge.tpc2.mem3x
	db.r5b.4xlarge.tpc2.mem2x
	db.r5b.2xlarge.tpc2.mem8x
	db.r5b.2xlarge.tpc2.mem4x
	db.r5b.2xlarge.tpc1.mem2x
	db.r5b.xlarge.tpc2.mem4x
	db.r5b.xlarge.tpc2.mem2x
	db.r5b.large.tpc1.mem2x
	db.r5.4xlarge.tpc2.mem4x
	db.r5.4xlarge.tpc2.mem3x
	db.r5.4xlarge.tpc2.mem2x

Edição do Oracle	Oracle Database 19c e posterior
	db.r5.2xlarge.tpc2.mem8x
	db.r5.2xlarge.tpc2.mem4x
	db.r5.2xlarge.tpc1.mem2x
	db.r5.xlarge.tpc2.mem4x
	db.r5.xlarge.tpc2.mem2x
	db.r5.large.tpc1.mem2x
	Classes de instância expansível
	db.t3.small–db.t3.2xlarge
Standard Edition 2 (SE2)	Classes de instância padrão
Licença incluída	db.m5.large – db.m5.4xlarge
	Classes de instância otimizada para memória
	db.r6i.large–db.r6i.4xlarge
	db.r5.large–db.r5.4xlarge
	Classes de instância expansível
	db.t3.small–db.t3.2xlarge

[1] Não é mais possível criar instâncias de banco de dados do RDS para Oracle usando a família de classes de instância X1.* (as classes de instância X1e.* ainda são aceitas). Se você usa classes X1.* atualmente, mude para uma classe de instância de nova geração assim que possível. A partir de 22 de janeiro de 2025, o RDS inicia as atualizações automáticas em sua janela de manutenção definida. Durante a atualização, o RDS escolhe o tipo de instância X2iedn equivalente e o atualiza. Para obter mais informações, consulte o artigo do re:Post O Amazon RDS para Oracle encerrará o suporte para instâncias de banco de dados X1 em 22 de janeiro de 2025.

> **Note**
>
> Recomendamos que clientes BYOL consultem seu contrato de licenciamento para avaliar o impacto de reprovações do Amazon RDS for Oracle. Para obter mais informações sobre a capacidade computacional das classes de instância de banco de dados compatíveis com o RDS para Oracle, consulte Classes de instâncias de banco de dados e Configurar o processador para uma classe de instância de banco de dados no RDS para Oracle.

> **Note**
>
> Se você tiver snapshots de banco de dados de instâncias de banco de dados que usavam as classes de instância de banco de dados obsoletas, selecione uma classe de instância de banco de dados que não seja obsoleta ao restaurar os snapshots de banco de dados. Para ter mais informações, consulte Restaurar uma instância de banco de dados.

Classes de instâncias de banco de dados obsoletas do RDS para Oracle

Estas classes de instância de banco de dados estão descontinuadas para o RDS para Oracle:

- db.m1, db.m2, db.m3, db.m4
- db.t1, db.t2
- db.r1, db.r2, db.r3, db.r4

As classes de instâncias de banco de dados anteriores foram substituídas por classes de instâncias de banco de dados com melhor performance que geralmente estão disponíveis a um custo menor. Se você tiver instâncias de banco de dados que usam as classes de instância de banco de dados defasadas, terá as seguintes opções:

- Permitir que o Amazon RDS modifique cada instância de banco de dados automaticamente para usar uma classe de instância de banco de dados não defasada comparável. Para conferir os cronogramas de defasagem, consulte Tipos de classe de instância de banco de dados.
- Alterar a classe da instância de banco de dados por conta própria, modificando a instância de banco de dados. Para ter mais informações, consulte Modificar uma instância de banco de dados do Amazon RDS.

Se você tiver snapshots de banco de dados de instâncias de banco de dados que usavam as classes de instância de banco de dados obsoletas, selecione uma classe de instância de banco de dados que não seja obsoleta ao restaurar os snapshots de banco de dados. Para ter mais informações, consulte Restaurar uma instância de banco de dados.

Arquitetura de banco de dados RDS para Oracle

A arquitetura multilocatário permite que um banco de dados Oracle funcione como um banco de dados de contêiner (CDB) multilocatário. Um CDB pode incluir bancos de dados conectáveis (PDBs) criados pelo cliente. Um não-CDB é um banco de dados Oracle que usa a arquitetura tradicional, que não pode conter PDBs. Para obter mais informações sobre a arquitetura multilocatário, consulte o Oracle Multitenant Administrator's Guide (Guia do administrador do Oracle Multitenant).

Para o Oracle Database 19c e posteriores, você pode criar uma instância de banco de dados do RDS para Oracle que usa a arquitetura CDB. No RDS para Oracle, os PDBs são chamados de bancos de dados de locatários. Suas aplicações cliente se conectam no nível do banco de dados de locatários (PDB), e não no nível de CDB. O RDS for Oracle é compatível com as seguintes configurações da arquitetura CDB:

Configuração de locação múltipla

Essa plataforma do RDS permite que uma instância de CDB do RDS para Oracle contenha de um a trinta bancos de dados de locatários, dependendo da edição do banco de dados e de quaisquer licenças de opção necessárias. Você pode usar APIs do RDS para adicionar, modificar e remover bancos de dados de locatário. A configuração de multilocatário no RDS para Oracle não oferece compatibilidade para PDBs de aplicação ou PDBs de proxy, que são tipos especiais de PDB. Para ter mais informações sobre PDBs de aplicação e PDBs de proxy, consulte Types of PDBs na documentação do Oracle Database.

> **Note**
>
> A configuração do Amazon RDS é chamada de “multi-tenant”, em vez de “multitenant”, porque é um recurso da plataforma RDS, não apenas do mecanismo de banco de dados Oracle. Da mesma forma, o termo “locatário” do RDS refere-se a qualquer locatário em uma configuração do RDS, não apenas aos PDBs da Oracle. Na documentação do RDS, o termo “multilocatário da Oracle” (Oracle multitenant), sem hífen, refere-se exclusivamente à arquitetura CDB de banco de dados Oracle, que é compatível com implantações on-premises e do RDS.

Configuração de locatário único

Esse recurso da plataforma RDS limita uma instância de CDB do RDS para Oracle a um banco de dados de um locatário (PDB). Você não pode adicionar mais PDBs usando as APIs do RDS. A configuração de locatário único da arquitetura multilocatário usa as mesmas APIs do RDS que a arquitetura não CDB. Assim, a experiência de trabalhar com um CDB na configuração de inquilino único é basicamente a mesma de trabalhar com um não-CDB.

É possível converter um CDB que use a configuração de locatário único na configuração multilocatária permitindo, dessa forma, a inclusão de PDBs no CDB. Essa mudança de arquitetura é permanente e irreversível. Para ter mais informações, consulte Converter a configuração de locatário único em multilocatário.

> **Note**
>
> Você não pode acessar o CDB em si.

No Oracle Database 21c e posteriores, todos os bancos de dados são CDBs. Por outro lado, você pode criar uma instância de banco de dados do Oracle Database 19c como CDB ou não CDB. Não é possível fazer upgrade de um não CDB para CDB, mas você pode converter um não CDB Oracle Database 19c em CDB, depois fazer upgrade. Não é possível converter um CDB em um não CDB.

Para obter mais informações, consulte os seguintes recursos:

- Trabalhar com CDBs no RDS para Oracle
- Limitações do RDS for CDBs Oracle
- Criar uma instância de banco de dados do Amazon RDS

Parâmetros de inicialização do RDS para Oracle

No Amazon RDS, você gerencia parâmetros usando um grupo de parâmetros de banco de dados. Usando esse grupo, você pode personalizar os parâmetros de inicialização. Por exemplo, você pode configurar o tamanho da área de classificação com `sort_area_size`. Todas as instâncias de banco de dados do RDS para Oracle associadas a um grupo de parâmetros de banco de dados específico usam as mesmas configurações de parâmetro. Para obter mais informações, consulte Grupos de parâmetros para Amazon RDS.

Parâmetros de inicialização permitidos do RDS para Oracle

Os parâmetros permitidos para a instância de banco de dados dependem da edição e versão do Oracle Database. Para visualizar os parâmetros de inicialização compatíveis de uma edição e uma versão específicas do Oracle Database, execute o comando `describe-engine-default-parameters` da AWS CLI. Por exemplo, para visualizar os parâmetros de inicialização permitidos para a Enterprise Edition do Oracle Database 19c, execute o comando a seguir.

```
aws rds describe-engine-default-parameters \
    --db-parameter-group-family oracle-ee-19 \
    --output json | jq -r '.EngineDefaults.Parameters[].ParameterName'

_add_col_optim_enabled
_adg_parselock_timeout
_allow_insert_with_update_check
_allow_level_without_connect_by
_always_semi_join
_autotask_max_window
_awr_disabled_flush_tables
_awr_mmon_cpuusage
_awr_mmon_deep_purge_all_expired
_b_tree_bitmap_plans
_bct_bitmaps_per_file
_bloom_filter_enabled
_buffered_publisher_flow_control_threshold
_bug29394014_allow_triggers_on_vpd_table
_cleanup_rollback_entries
_client_enable_auto_unregister
_clusterwide_global_transactions
_complex_view_merging
_connect_by_use_union_all
_cost_equality_semi_join
_cursor_features_enabled
_cursor_obsolete_threshold
_datafile_write_errors_crash_instance
_db_block_buffers
...
```

Para descrever um único parâmetro de inicialização, use o comando a seguir, substituindo `sga_max_size` pelo nome do seu parâmetro (exemplo de saída incluído).

```
aws rds describe-engine-default-parameters \
```

```
    --db-parameter-group-family oracle-ee-19 \
    --query 'EngineDefaults.Parameters[?ParameterName==`sga_max_size`]' \
    --output json

[
    {
        "ParameterName": "sga_max_size",
        "Description": "max total SGA size",
        "Source": "engine-default",
        "ApplyType": "static",
        "DataType": "integer",
        "AllowedValues": "0-2199023255552",
        "IsModifiable": true
    }
]
```

Para encontrar a documentação sobre os parâmetros de inicialização do Oracle Database, consulte Initialization Parameters na documentação do Oracle Database. Observe que o parâmetro `ARCHIVE_LAG_TARGET` tem considerações especiais no RDS para Oracle. Esse parâmetro força uma troca de log de redo on-line após o término do tempo especificado. No RDS para Oracle, `ARCHIVE_LAG_TARGET` é definido como `300` porque o objetivo de ponto de recuperação (RPO) é de cinco minutos. Para honrar esse objetivo, o RDS para Oracle troca o log redo on-line a cada cinco minutos e o armazena em um bucket do Amazon S3.

Se a frequência da troca de log de redo on-line reduzir o desempenho em seu banco de dados do RDS para Oracle, você poderá escalar a instância de banco de dados e o armazenamento para um valor maior de IOPS e throughput. Como alternativa, se você usar o RDS Custom para Oracle ou implantar um banco de dados do Oracle no Amazon EC2, poderá ajustar a configuração do parâmetro de inicialização `ARCHIVE_LAG_TARGET`.

Valores de parâmetros válidos no RDS para Oracle

No RDS para Oracle, somente os seguintes caracteres são válidos para valores de parâmetro:

- Letras (`A-Z` e `a-z`)
- Números (`0-9`)
- Espaços em branco (espaços, tabulações e quebras de linha)
- Os seguintes caracteres especiais: `_ / . : + = ( ) ' * , % $ -` (hífen)

Conjuntos de caracteres do RDS for Oracle

O RDS para Oracle é compatível com dois tipos de conjuntos de caracteres: o conjunto de caracteres de banco de dados e o conjunto de caracteres nacionais.

Conjunto de caracteres de banco de dados

O conjunto de caracteres de banco de dados Oracle é usado nos tipos de dados `CHAR`, `VARCHAR2` e `CLOB`. O banco de dados também usa esse conjunto de caracteres para metadados, como nomes de tabelas, nomes de colunas e instruções SQL. O conjunto de caracteres de banco de dados Oracle é normalmente chamado de conjunto de caracteres do banco de dados.

Você pode definir o conjunto de caracteres ao criar uma instância de banco de dados. Você não pode alterar o conjunto de caracteres de banco de dados depois de criar o banco de dados.

Conjuntos de caracteres de banco de dados compatíveis

A tabela a seguir lista os conjuntos de caracteres de banco de dados Oracle com suporte no Amazon RDS. É possível usar um valor dessa tabela com o parâmetro `--character-set-name` do comando da AWS CLI create-db-instance ou com o parâmetro `CharacterSetName` da operação CreateDBInstance da API do Amazon RDS.

> **Note**
>
> O conjunto de caracteres para um CDB é sempre AL32UTF8. Você pode definir um conjunto de caracteres diferente somente para o PDB.

Valor	Descrição
AL32UTF8	Unicode 5.0 UTF-8 Conjunto de caracteres universal (padrão)
AR8ISO8859P6	ISO 8859-6 Latin/Arabic
AR8MSWIN1256	Microsoft Windows Code Page 1256 8 bits Latin/Arabic
BLT8ISO8859P13	ISO 8859-13 Baltic

Valor	Descrição
BLT8MSWIN1257	Microsoft Windows Code Page 1257 8 bits Baltic
CL8ISO8859P5	ISO 88559-5 Latin/Cyrillic
CL8MSWIN1251	Microsoft Windows Code Page 1251 8 bits Latin/Cyrillic
EE8ISO8859P2	ISO 8859-2 East European
EL8ISO8859P7	ISO 8859-7 Latin/Greek
EE8MSWIN1250	Microsoft Windows Code Page 1250 8 bits East European
EL8MSWIN1253	Microsoft Windows Code Page 1253 8 bits Latin/Greek
IW8ISO8859P8	ISO 8859-8 Latin/Hebrew
IW8MSWIN1255	Microsoft Windows Code Page 1255 8 bits Latin/Hebrew
JA16EUC	EUC 24 bits Japanese
JA16EUCTILDE	Igual a JA16EUC, exporto pelo mapeamento do til para e partir do Unicode
JA16SJIS	Shift-JIS 16 bits Japanese
JA16SJISTILDE	Igual a JA16SJIS, exporto pelo mapeamento do til para e partir do Unicode
KO16MSWIN949	Microsoft Windows Code Page 949 Korean
NE8ISO8859P10	ISO 8859-10 North European
NEE8ISO8859P4	ISO 8859-4 North e Northeast European

Valor	Descrição
TH8TISASCII	Thai Industrial Standard 620-2533-ASCII 8 bits
TR8MSWIN1254	Microsoft Windows Code Page 1254 8 bits Turkish
US7ASCII	ASCII 7 bits American
UTF8	Unicode 3.0 UTF-8 conjunto de caracteres universal, compatível com CESU-8
VN8MSWIN1258	Microsoft Windows Code Page 1258 8 bits Vietnamese
WE8ISO8859P1	Western European 8 bits ISO 8859 Part 1
WE8ISO8859P15	ISO 8859-15 West European
WE8ISO8859P9	ISO 8859-9 West European e Turkish
WE8MSWIN1252	Microsoft Windows Code Page 1252 8 bits West European
ZHS16GBK	GBK 16 bits Simplified Chinese
ZHT16HKSCS	Microsoft Windows Code Page 950 com conjunto de caracteres Hong Kong Supplementary HKSCS-2001. A conversão do conjunto de caracteres é baseada em Unicode 3.0.
ZHT16MSWIN950	Microsoft Windows Code Page 950 Traditional Chinese
ZHT32EUC	EUC 32 bits Traditional Chinese

Variável de ambiente NLS_LANG

Um local é um conjunto de informações que abordam requisitos linguísticos e culturais que correspondem a um determinado idioma e país. Definir a variável de ambiente NLS_LANG no

ambiente do seu cliente é a maneira mais simples de especificar o comportamento da localidade para o Oracle. Essa variável define o idioma e o território usados pela aplicação cliente e pelo servidor de banco de dados. Ele também indica o conjunto de caracteres do cliente, que corresponde ao conjunto de caracteres para os dados inseridos ou exibidos por um aplicativo cliente. Para obter mais informações sobre NLS_LANG e conjuntos de caracteres, consulte O que é um conjunto de caracteres ou uma página de código? na documentação da Oracle.

Parâmetros de inicialização do NLS

Você também pode definir os seguintes parâmetros de inicialização do National Language Support (NLS) no nível da instância para uma instância de banco de dados do Oracle no Amazon RDS:

- NLS_DATE_FORMAT
- NLS_LENGTH_SEMANTICS
- NLS_NCHAR_CONV_EXCP
- NLS_TIME_FORMAT
- NLS_TIME_TZ_FORMAT
- NLS_TIMESTAMP_FORMAT
- NLS_TIMESTAMP_TZ_FORMAT

Para obter informações sobre como modificar parâmetros de instâncias, consulte Grupos de parâmetros para Amazon RDS.

Você pode definir outros parâmetros de inicialização do NLS no seu cliente SQL. Por exemplo, a seguinte instrução define o parâmetro de inicialização NLS_LANGUAGE para GERMAN em um cliente SQL que está conectado a uma instância de banco de dados Oracle:

```
ALTER SESSION SET NLS_LANGUAGE=GERMAN;
```

Para obter informações sobre como se conectar a uma instância de banco de dados Oracle com um cliente SQL, consulte Conectar-se a uma instância de banco de dados Oracle.

Conjunto de caracteres nacionais

O conjunto de caracteres nacionais é usado nos tipos de dados `NCHAR`, `NVARCHAR2` e `NCLOB`. O conjunto de caracteres nacional é normalmente chamado de conjunto de caracteres NCHAR. Ao contrário do conjunto de caracteres do banco de dados, o conjunto de caracteres NCHAR não afeta os metadados do banco de dados.

O conjunto de caracteres NCHAR é compatível com os seguintes conjuntos de caracteres:

- AL16UTF16 (padrão)
- UTF8

Você pode especificar qualquer valor com o `--nchar-character-set-name` parâmetro do comando create-db-instance (somente AWS CLI versão 2). Se você usar a API do Amazon RDS, especifique o parâmetro `NcharCharacterSetName`da operação CreateDBInstance. Não é possível alterar o conjunto de caracteres nacionais depois de criar o banco de dados.

Para obter mais informações sobre Unicode em bancos de dados Oracle, consulte Suporte a bancos de dados multilíngues com unicode na documentação da Oracle.

Limitações do RDS para Oracle

Nas seções a seguir, você pode encontrar limitações importantes ao uso do RDS para Oracle. Para limitações específicas dos CDBs, consulte Limitações do RDS for CDBs Oracle.

> **Note**
>
> Essa lista não é exaustiva.

Tópicos

- Limites de tamanho de arquivo do Oracle no Amazon RDS
- Limites de tamanho de bloco no RDS para Oracle
- Sinônimos públicos para schemas fornecidos pela Oracle
- Recursos não compatíveis com o RDS para Oracle
- Limitações a privilégios de DBA no RDS para Oracle.
- Descontinuação dos protocolos Transport Layer Security (TLS) 1.0 e 1.1 no RDS para Oracle

Limites de tamanho de arquivo do Oracle no Amazon RDS

O tamanho máximo de um único arquivo em instâncias de banco de dados do RDS para Oracle é de 16 TiB (tebibytes). Esse limite é imposto pelo sistema de arquivos ext4 usado pela instância. Assim, os arquivos de dados grandes do Oracle são limitados a 16 TiB. Se tentar redimensionar um arquivo

de dados em um espaço de tabela de arquivos grandes para um valor acima do limite, você receberá um erro como o seguinte.

```
ORA-01237: cannot extend datafile 6
ORA-01110: data file 6: '/rdsdbdata/db/mydir/datafile/myfile.dbf'
ORA-27059: could not reduce file size
Linux-x86_64 Error: 27: File too large
Additional information: 2
```

Limites de tamanho de bloco no RDS para Oracle

O tamanho máximo de um único arquivo em instâncias de banco de dados do RDS para Oracle é 8 KB. O tamanho de bloco de 16 KB não é permitido.

Sinônimos públicos para schemas fornecidos pela Oracle

Não crie ou modifique sinônimos públicos para schemas mantidos pela Oracle, incluindo `SYS`, `SYSTEM` e `RDSADMIN`. Isso pode resultar na invalidação de componentes centrais do banco de dados e afetar a disponibilidade da sua instância de banco de dados.

Você pode criar sinônimos públicos fazendo referência a objetos em seus próprios esquemas.

Recursos não compatíveis com o RDS para Oracle

Em geral, o Amazon RDS não impede que você crie esquemas para recursos que não têm suporte. No entanto, se você criar schemas para recursos e componentes Oracle que exigem privilégios SYS, poderá danificar o dicionário de dados e afetar a disponibilidade da instância. Use apenas recursos e esquemas com suporte que estão disponíveis em Adição de opções a instâncias de banco de dados Oracle.

Limitações a privilégios de DBA no RDS para Oracle.

Uma função é uma coleção de privilégios que podem ser concedidos ou revogados de um usuário. Um banco de dados Oracle usa funções para fornecer segurança.

Uma função predefinida, chamada de `DBA`, normalmente fornece todos os privilégios administrativos em um banco de dados Oracle. Ao criar uma instância de banco de dados, sua conta de usuário mestra obtém privilégios de DBA (com algumas limitações). Para fornecer uma experiência gerenciada, um banco de dados do RDS for Oracle não fornece os seguintes privilégios para a função `DBA`:

- ALTER DATABASE
- ALTER SYSTEM
- CREATE ANY DIRECTORY
- DROP ANY DIRECTORY
- GRANT ANY PRIVILEGE
- GRANT ANY ROLE

Use a conta de usuário mestre para tarefas administrativas, como criar contas de usuário adicionais no banco de dados. Não é possível utilizar as contas SYS, SYSTEM nem outras contas administrativas fornecidas pela Oracle.

Descontinuação dos protocolos Transport Layer Security (TLS) 1.0 e 1.1 no RDS para Oracle

Os protocolos Transport Layer Security versões 1.0 e 1.1 (TLS 1.0 e TLS 1.1) foram descontinuados. De acordo com as práticas recomendadas de segurança, a Oracle descontinuou o uso do TLS 1.0 e do TLS 1.1. Para atender aos requisitos de segurança, recomendamos enfaticamente que você use o TLS 1.2 em vez disso.

Conectar-se a uma instância de banco de dados Oracle

Depois que o Amazon RDS provisiona a instância de banco de dados do Oracle, você pode usar qualquer aplicação cliente padrão SQL para fazer login na instância de banco de dados. Como o RDS é um serviço gerenciado, você não pode fazer login como SYS nem SYSTEM. Para ter mais informações, consulte Usuários e privilégios do RDS para Oracle.

Neste tópico, você aprenderá a usar o Oracle SQL Developer ou o SQL*Plus para se conectar a uma instância de banco de dados do RDS para Oracle. Consulte e veja um exemplo do processo de criação e conexão com uma instância de banco de dados de amostr Criar uma instância de banco de dados Oracle e conectar-se a ela.

Tópicos

- Encontrar o endpoint da instância de banco de dados do RDS para Oracle
- Conectar-se à sua instância de banco de dados usando o Oracle SQL Developer
- Conectar-se à sua instância de banco de dados do Linux usando o SQL*Plus
- Considerações para grupos de segurança
- Considerações para arquitetura de processo
- Solução de problema de conexão com a instância de banco de dados Oracle
- Modificação das propriedades de conexão usando parâmetros sqlnet.ora

Encontrar o endpoint da instância de banco de dados do RDS para Oracle

Cada instância de banco de dados do Amazon RDS tem um endpoint, e cada ponto final possui o nome DNS e o número da porta para a instância de banco de dados. Para se conectar à sua instância de banco de dados usando um aplicativo cliente SQL, você precisa do nome DNS e do número da porta para sua instância de banco de dados.

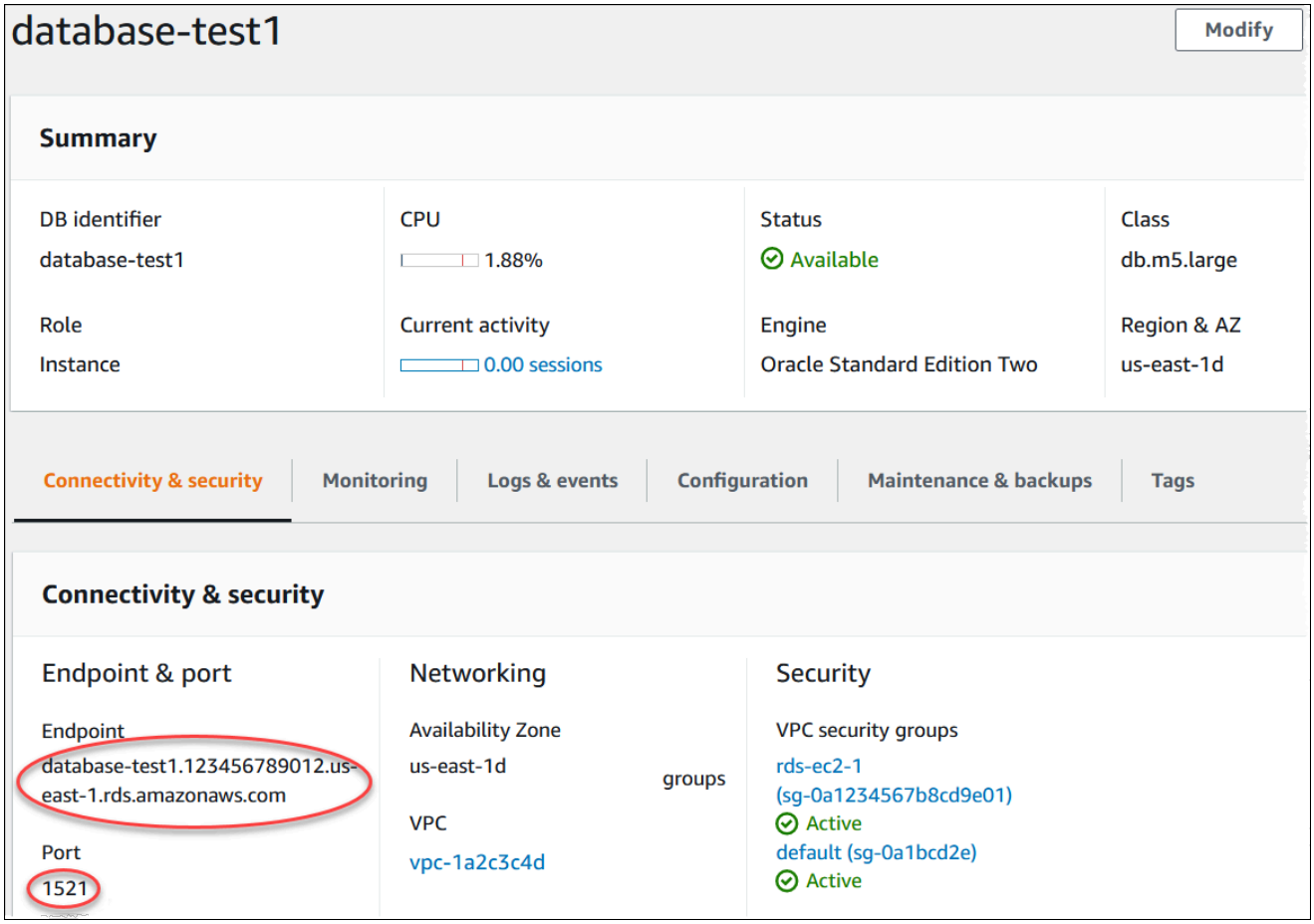
Você pode encontrar o endpoint de uma instância de banco de dados usando o console do Amazon RDS ou a AWS CLI.

> **Note**
>
> Se você estiver usando a autenticação Kerberos, consulte Conectar-se ao Oracle com a autenticação Kerberos.

Console

Para localizar o endpoint usando o console

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No canto superior direito do console do, escolha a região AWS para a instância de banco de dados.
3. Encontre o nome DNS e o número da porta da instância de banco de dados.
   a. Escolha Databases (Bancos de dados) para exibir a lista das instâncias de banco de dados.
   b. Escolha o nome da instância de banco de dados Oracle para exibir os detalhes da instância.
   c. Na guia Connectivity & security (Conectividade e segurança), copie o endpoint. Além disso, anote o número da porta. Você precisará do endpoint e do número da porta para conectar-se à instância de banco de dados.



AWS CLI

Para encontrar o endpoint de uma instância de banco de dados Oracle usando a AWS CLI, chame o comando describe-db-instances.

Example Para localizar o endpoint usando a AWS CLI

```
aws rds describe-db-instances
```

Procure o `Endpoint` na saída para encontrar o nome DNS e o número da porta da sua instância de banco de dados. A linha `Address` na saída contém o nome DNS. Este é um exemplo da saída de endpoint JSON.

```
"Endpoint": {
    "HostedZoneId": "Z1PVIF0B656C1W",
    "Port": 3306,
    "Address": "myinstance.123456789012.us-west-2.rds.amazonaws.com"
},
```

> **Note**
>
> A saída pode conter informações de várias instâncias de banco de dados.

Conectar-se à sua instância de banco de dados usando o Oracle SQL Developer

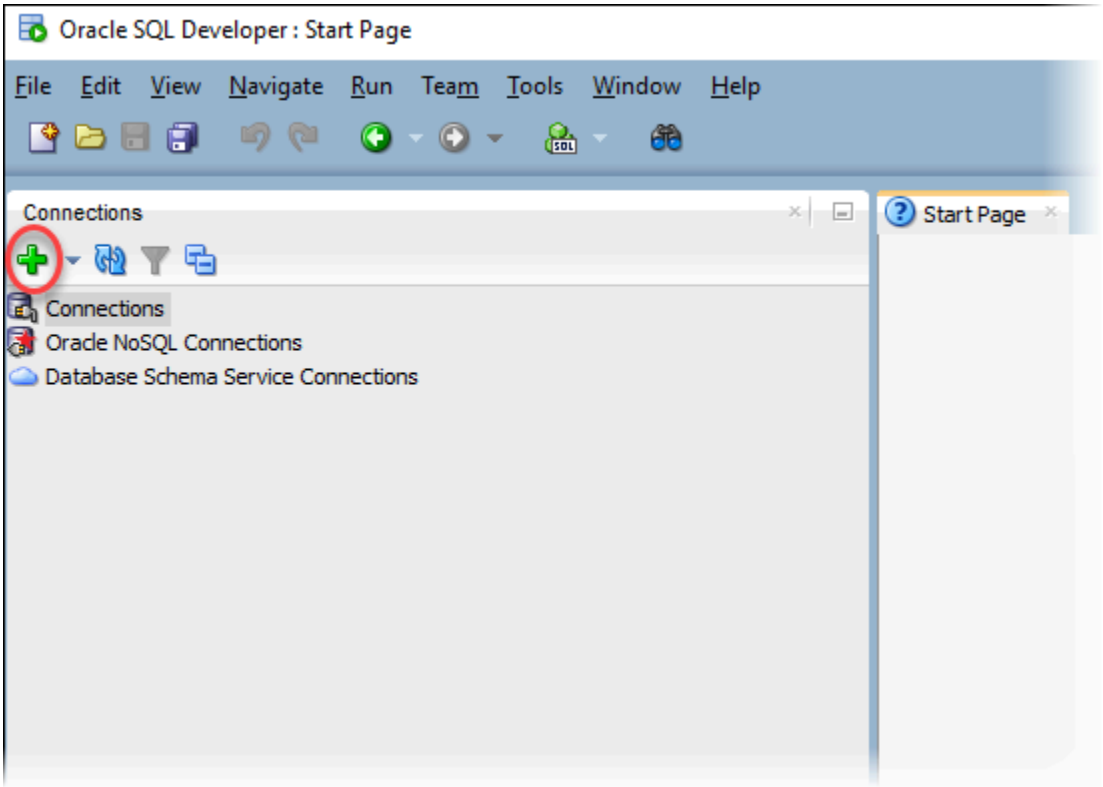
Neste procedimento, você se conectará à sua instância de banco de dados usando o Oracle SQL Developer. Para fazer download de uma versão independente desse utilitário, consulte a página de downloads do Oracle SQL Developer.

Para se conectar à sua instância de banco de dados, você precisa do nome DNS e do número da porta. Para obter informações sobre como encontrar o nome DNS e o número da porta de uma instância de banco de dados, consulte Encontrar o endpoint da instância de banco de dados do RDS para Oracle.

Como conectar-se a uma instância de banco de dados usando o SQL Developer

1. Inicie o Oracle SQL Developer.

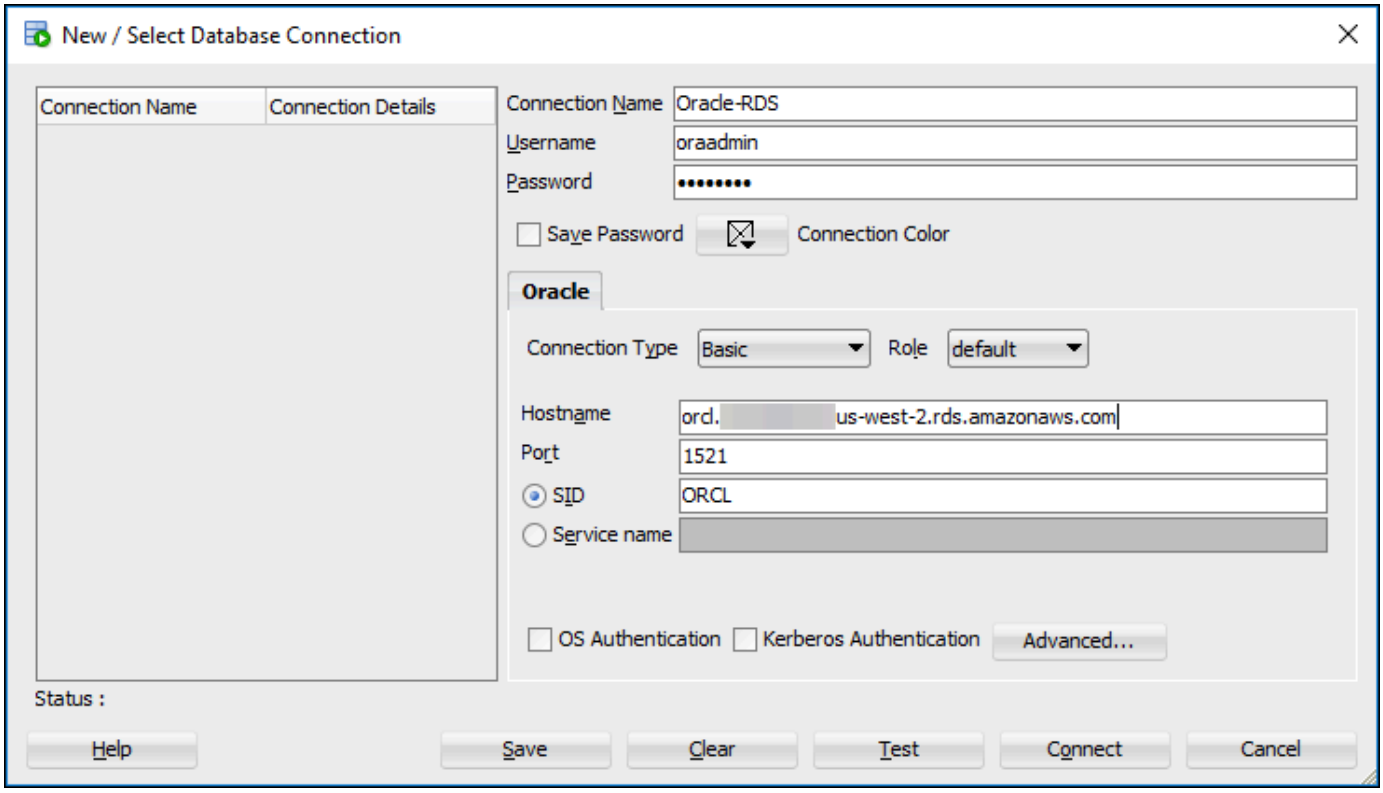
2. Na guia Connections (Conexões), escolha o ícone add (+) (adicionar (+)).



3. Na caixa de diálogo New/Select Database Connection (Nova/selecionar conexão de banco de dados), forneça as informações da sua instância de banco de dados:

   - Em Connection Name (Nome da conexão), digite um nome que descreva a conexão, como `Oracle-RDS`.
   - Em Username (Nome do usuário), digite o nome do administrador do banco de dados da instância de banco de dados.
   - Em Password (Senha), digite a senha do administrador do banco de dados.
   - Em Hostname (Nome do host), digite o nome DNS da instância de banco de dados.
   - Em Port (Porta), digite o número da porta.
   - Em SID, insira o nome do banco de dados. Você pode encontrar o nome do banco de dados na guia Configuration (Configuração) da página de detalhes do banco de dados.

   A caixa de diálogo preenchida deve ser semelhante à seguinte.

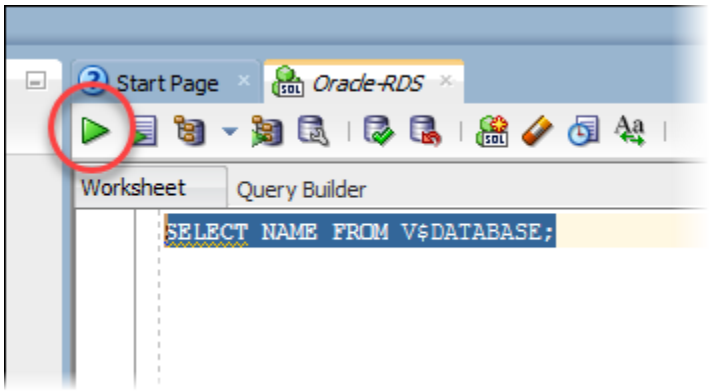


4. Selecione Conectar.
5. Agora, você pode começar a criar seus próprios bancos de dados e executar consultas em instâncias de banco de dados e bancos de dados como de costume. Para executar uma consulta de teste na sua instância de banco de dados, faça o seguinte:

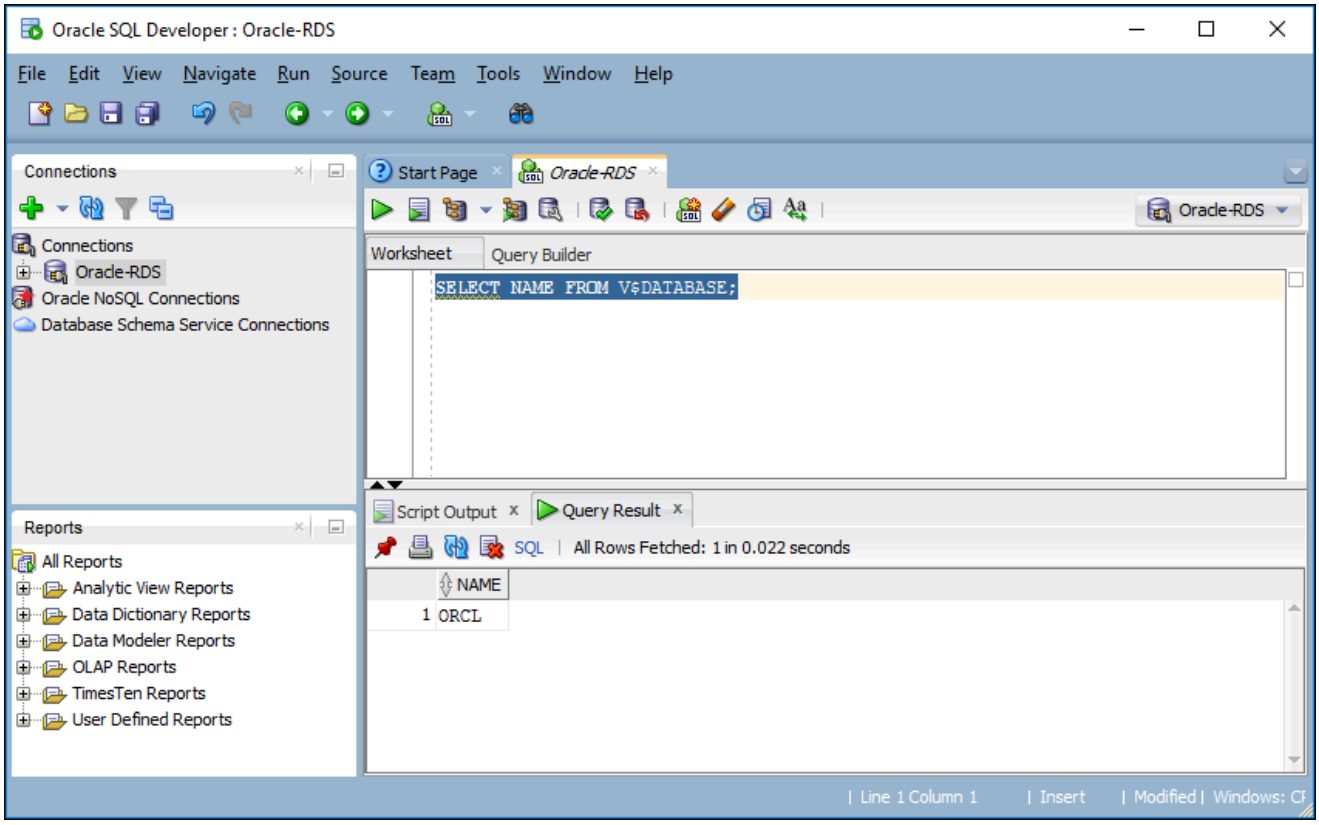
   a. Na guia Worksheet (Planilha) da conexão, digite a consulta SQL a seguir.

   ```
   SELECT NAME FROM V$DATABASE;
   ```

   b. Escolha o ícone execute (execução) para executar a consulta.

   O SQL Developer retorna o nome do banco de dados.



Conectar-se à sua instância de banco de dados do Linux usando o SQL*Plus

Você pode usar um utilitário como SQL*Plus para se conectar a uma instância de banco de dados do Amazon RDS executando o Oracle. Para fazer o download do Oracle Instant Client, que inclui uma versão independente do SQL*Plus, consulte Downloads do Oracle Instant Client.

Para se conectar à sua instância de banco de dados, você precisa do nome DNS e do número da porta. Para obter informações sobre como encontrar o nome DNS e o número da porta de uma instância de banco de dados, consulte Encontrar o endpoint da instância de banco de dados do RDS para Oracle.

Example Para conectar-se a uma instância de banco de dados Oracle usando o SQL*Plus

Nos exemplos a seguir, substitua o nome de usuário do administrador de sua instância de banco de dados. Além disso, substitua o nome do DNS da instância de banco de dados e inclua o número da porta e o Oracle SID. O valor de SID é o nome do banco de dados da instância de banco de dados que você especificou quando a criou, e não o nome da instância de banco de dados.

Para Linux, macOS ou Unix:

```
sqlplus 'user_name@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=dns_name)(PORT=port))
(CONNECT_DATA=(SID=database_name)))'
```

Para Windows:

```
sqlplus user_name@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=dns_name)(PORT=port))
(CONNECT_DATA=(SID=database_name)))
```

Você deve ver saída semelhante ao seguinte:

```
SQL*Plus: Release 12.1.0.2.0 Production on Mon Aug 21 09:42:20 2017
```

Depois que você inserir a senha de usuário, o prompt do SQL será exibido.

```
SQL>
```

> **Note**
>
> Visto que a string de conexão de formato mais curto (EZ Connect), como `sqlplus USER/PASSWORD@longer-than-63-chars-rds-endpoint-here:1521/database-identifier`, pode encontrar um limite máximo de caracteres, recomendamos que não a use para conexão.

Considerações para grupos de segurança

Para se conectar à sua instância de banco de dados, ela deve estar associada a um grupo de segurança que contenha os endereços IP e a configuração de rede que você usa para acessá-la. Sua instância de banco de dados pode usar o grupo de segurança padrão. Se você atribuiu um grupo de segurança padrão não configurado quando criou a instância de banco de dados, o firewall dessa instância impedirá conexões. Para obter informações sobre como criar um novo security group, consulte Controlar acesso com grupos de segurança.

Depois de criar o novo security group, você modifica sua instância de banco de dados para associá-la a esse security group. Para obter mais informações, consulte Modificar uma instância de banco de dados do Amazon RDS.

Você pode aprimorar a segurança usando o SSL para criptografar conexões com sua a instância de banco de dados. Para mais informações, consulte Oracle Secure Sockets Layer.

Considerações para arquitetura de processo

Os processos do servidor lidam com as conexões do usuário em uma instância de banco de dados Oracle. Por padrão, a instância de banco de dados Oracle usa processos de servidores dedicados. Com os processos de servidores dedicados, cada processo de servidor lida somente com um processo de usuário. Se preferir, você pode configurar processos de servidores compartilhados. Com os processos de servidores compartilhados, cada processo de servidor lida com vários processos de usuário.

Você pode considerar processos de servidores compartilhados quando uma quantidade elevada de sessões de usuários estiver consumindo muita memória no servidor. Você também pode considerar os processos do servidores compartilhados quando as sessões se conectarem e desconectarem com muita frequência, resultando em problemas de performance. Há também desvantagens na utilização de processos de servidores compartilhados. Por exemplo, eles podem prejudicar os recursos da CPU e são mais complexos para configurar e administrar.

Para obter mais informações sobre processos de servidores dedicados e compartilhados, consulte Sobre os processos de servidores dedicados e compartilhados na documentação do Oracle. Para obter mais informações sobre como configurar processos de servidores compartilhados em uma instância de banco de dados Oracle, consulte How do I configure Amazon RDS for Oracle database to work with shared servers? na Central de Conhecimento.

Solução de problema de conexão com a instância de banco de dados Oracle

Veja a seguir os problemas que você pode encontrar ao tentar se conectar à sua instância de banco de dados Oracle.

Problema	Sugestões de solução de problemas
Não é possível conectar-se à sua instância de banco de dados.	Uma instância de banco de dados recém-criada fica com o status creating (criando) até que ela esteja pronta para uso. Quando o status muda para available (disponível), você pode se conectar à instância de banco de dados. Dependendo da classe da instância de banco de dados e da quantidade de armazenamento, pode levar

Problema	Sugestões de solução de problemas
	até 20 minutos para que a nova instância de banco de dados esteja disponível.
Não é possível conectar-se à sua instância de banco de dados.	Se você não pode enviar ou receber comunicações através da porta que especificou quando criou a instância de banco de dados, não poderá se conectar à instância de banco de dados. Verifique com o administrador da rede para verificar se a porta que você especificou para a sua instância de banco de dados permite comunicação de entrada e saída.
Não é possível conectar-se à sua instância de banco de dados.	As regras de acesso aplicadas pelo seu firewall local e os endereços IP autorizados para acessar sua instância de banco de dados no security group para a instância de banco de dados podem não corresponder. O problema está provavelmente com as regras de saída ou de entrada no seu firewall. É possível adicionar ou editar uma regra de entrada no grupo de segurança. Em Source (Origem), selecione My IP (Meu IP). Isso concede acesso à instância de banco de dados do endereço IP detectado no navegador. Para obter mais informações, consulte Amazon VPC e Amazon RDS. Para obter mais informações sobre security groups, consulte Controlar acesso com grupos de segurança. Para orientação pelo processo de configuração de regras para o seu security group, consulte Tutorial: Criar uma VPC para usar com uma instância de banco de dados (somente IPv4).

Problema	Sugestões de solução de problemas
Connect failed because target host or object does not exist – Oracle, Error: ORA-12545 (A conexão falhou pois o host de destino ou o objeto não existe – Oracle, erro: ORA-12545)	Verifique se você especificou o nome do servidor e o número da porta corretamente. Em Server name (Nome do servidor), digite o nome DNS no console. Para obter informações sobre como encontrar o nome DNS e o número da porta de uma instância de banco de dados, consulte Encontrar o endpoint da instância de banco de dados do RDS para Oracle.
Invalid username/ password; logon denied – Oracle, Error: ORA-01017 (Nome de usuário/senha inválidos; login negado – Oracle, erro: ORA-01017)	Você conseguiu acessar a instância de banco de dados, mas a conexão foi recusada. Isso geralmente é causado pelo fornecime nto de nome de usuário ou senha incorretos. Verifique o nome do usuário e a senha e tente novamente.
No momento, TNS:liste ner não conhece o SID fornecido no descritor de conexão - Oracle, ERRO: ORA-12505	Certifique-se de que o SID correto esteja inserido. O SID equivale ao nome do seu banco de dados. Encontre o nome do banco de dados na guia Configuration (Configuração) da página Databases (Bancos de dados) para sua instância. Você também pode encontrar o nome do banco de dados usando a AWS CLI: `aws rds describe-db-instances --query 'DBInstances[*].[DBInstanceIdentifier,DBName]' --output text`

Para obter mais informações sobre problemas de conexão, consulte Não é possível conectar-se à instância de banco de dados do Amazon RDS.

Modificação das propriedades de conexão usando parâmetros sqlnet.ora

O arquivo sqlnet.ora inclui os parâmetros que configuram os recursos do Oracle Net em servidores e clientes de banco de dados Oracle. Usando os parâmetros no arquivo sqlnet.ora, você pode modificar propriedades para conexões dentro e fora do banco de dados.

Para obter mais informações sobre por que convém definir parâmetros do sqlnet.ora, consulte Configurar parâmetros do perfil na documentação da Oracle.

Configurar parâmetros do sqlnet.ora

Os grupos de parâmetros do Amazon RDS for Oracle incluem um subconjunto de parâmetros sqlnet.ora. Você os configura da mesma forma que configura outros parâmetros do Oracle. O prefixo `sqlnetora.` identifica quais parâmetros são parâmetros do sqlnet.ora. Por exemplo, em um parameter group Oracle no Amazon RDS, o parâmetro `default_sdu_size` do sqlnet.ora é `sqlnetora.default_sdu_size`.

Para obter informações sobre como gerenciar parameter groups e definir valores de parâmetros, consulte Grupos de parâmetros para Amazon RDS.

Parâmetros do sqlnet.ora compatíveis

O Amazon RDS oferece suporte aos seguintes parâmetros do sqlnet.ora. As alterações nos parâmetros dinâmicos do sqlnet.ora entram em vigor imediatamente.

Parâmetro	Valores válidos	Estático/ dinâmico	Descrição
`sqlnetora.default_sdu_size`	`512` para `20971`	Dinâmico	O tamanho da unidade de dados da sessão (SDU), em bytes. A SDU é a quantidade de dados que são armazenados em um buffer e enviados pela rede ao mesmo tempo.
`sqlnetora.diag_adr_enabled`	`ON`, `OFF`	Dinâmico	Um valor que ativa e desativa o rastreamento do Repositório de Diagnóstico Automático (ADR). `ON` especifica que é usado o rastreamento de arquivos de ADR.

Parâmetro	Valores válidos	Estático/ dinâmico	Descrição
			OFF especifica que é usado o rastreamento de arquivos que não são de ADR.
sqlnetora.recv_buf_size	8192 para 268435	Dinâmico	O limite de espaço de buffer para receber operações de sessões, compatíveis com os protocolos TCP/IP, TCP/IP com SSL e SDP.
sqlnetora.send_buf_size	8192 para 268435	Dinâmico	O limite de espaço de buffer para enviar operações de sessões, compatíveis com os protocolos TCP/IP, TCP/IP com SSL e SDP.
sqlnetora.sqlnet.allowed_logon_version_client	8, 10, 11, 12	Dinâmico	A versão mínima do protocolo de autentica ção permite que clientes e servidores atuando como clientes estabeleçam uma conexão com instâncias de bancos de dados Oracle.
sqlnetora.sqlnet.allowed_logon_version_server	8, 9, 10, 11, 12, 12a	Dinâmico	A versão mínima do protocolo de autenticação permite estabelecer uma conexão com instâncias de bancos de dados Oracle.
sqlnetora.sqlnet.expire_time	0 para 1440	Dinâmico	Intervalo de tempo, em minutos, para enviar uma verificação para conferir se as conexões cliente/servidor estão ativas.

Parâmetro	Valores válidos	Estático/ dinâmico	Descrição
sqlnetora.sqlnet.inbound_connect_timeout	0 ou 10 a 7200	Dinâmico	Tempo, em segundos, para um cliente se conectar ao servidor de banco de dados e fornecer as informações de autenticação necessárias.
sqlnetora.sqlnet.outbound_connect_timeout	0 ou 10 a 7200	Dinâmico	Tempo, em segundos, para um cliente estabelecer uma conexão do Oracle Net com a instância de banco de dados.
sqlnetora.sqlnet.recv_timeout	0 ou 10 a 7200	Dinâmico	Tempo, em segundos, para um servidor de banco de dados aguardar os dados do cliente depois de estabelecer uma conexão.
sqlnetora.sqlnet.send_timeout	0 ou 10 a 7200	Dinâmico	Tempo, em segundos, para um servidor de banco de dados concluir uma operação de envio para os clientes depois de estabelecer uma conexão.
sqlnetora.tcp.connect_timeout	0 ou 10 a 7200	Dinâmico	Tempo, em segundos, para um cliente estabelecer uma conexão TCP com o servidor de banco de dados.

Parâmetro	Valores válidos	Estático/ dinâmico	Descrição
`sqlnetora.trace_level_server`	`0`, `4`, `10`, `16`, `OFF`, `USER`, `ADMIN`, `SUPPOF`	Dinâmico	Para rastreamento que não abrange arquivos de ADR, ativa o rastreamento do servidor em um nível especificado ou o desativa.

O valor padrão para cada parâmetro do sqlnet.ora compatível é o padrão do Oracle Database para a versão.

Exibir parâmetros do sqlnet.ora

Você pode exibir os parâmetros do sqlnet.ora e suas configurações usando o AWS Management Console, a AWS CLI ou um cliente SQL.

Exibir parâmetros do sqlnet.ora usando o console

Para obter informações sobre como exibir parâmetros em um parameter group, consulte Grupos de parâmetros para Amazon RDS.

Nos parameter groups do Oracle, o prefixo `sqlnetora.` identifica quais parâmetros são parâmetros do sqlnet.ora.

Exibir parâmetros do sqlnet.ora usando a AWS CLI

Para exibir os parâmetros do sqlnet.ora que foram configurados em um parameter group do Oracle, use o comando describe-db-parameters da AWS CLI.

Para exibir todos os parâmetros do sqlnet.ora para uma instância de banco de dados Oracle, chame o comando download-db-log-file-portion da AWS CLI. Especifique o DB instance identifier, o nome do arquivo de log e o tipo de saída.

Example

O código a seguir lista todos os parâmetros do sqlnet.ora par `mydbinstance`.

Para Linux, macOS ou Unix:

```
aws rds download-db-log-file-portion \
    --db-instance-identifier mydbinstance \
    --log-file-name trace/sqlnet-parameters \
    --output text
```

Para Windows:

```
aws rds download-db-log-file-portion ^
    --db-instance-identifier mydbinstance ^
    --log-file-name trace/sqlnet-parameters ^
    --output text
```

Exibir parâmetros do sqlnet.ora usando um cliente SQL

Depois de se conectar à instância de banco de dados Oracle em um cliente SQL, a consulta a seguir lista os parâmetros do sqlnet.ora.

```
SELECT * FROM TABLE
   (rdsadmin.rds_file_util.read_text_file(
        p_directory => 'BDUMP',
        p_filename  => 'sqlnet-parameters'));
```

Para obter informações sobre como se conectar a uma instância de banco de dados Oracle em um cliente SQL, consulte Conectar-se a uma instância de banco de dados Oracle.

Proteção das conexões de instância de banco de dados Oracle

O Amazon RDS for Oracle oferece suporte a conexões criptografadas em SSL/TLS, além da opção Oracle Native Network Encryption (NNE), para criptografar conexões entre a aplicação e a instância de banco de dados Oracle. Para obter mais informações sobre a opção Oracle Native Network Encryption, consulte Oracle Native Network Encryption.

Tópicos

- Usar SSL com uma instância de banco de dados Oracle
- Atualizar aplicações para conexão com instâncias de banco de dados Oracle usando novos certificados SSL/TLS
- Usar a criptografia de rede com uma instância de banco de dados do RDS para Oracle
- Configurar a autenticação Kerberos do Amazon RDS para Oracle
- Configurar o acesso UTL_HTTP usando certificados e uma carteira Oracle

Usar SSL com uma instância de banco de dados Oracle

O Secure Sockets Layer (SSL) é um protocolo padrão do setor para proteger as conexões de rede entre o cliente e o servidor. Após o SSL versão 3.0, o nome mudou para Transport Layer Security (TLS), mas ainda costuma ser chamado de SSL. O Amazon RDS é compatível com criptografia SSL para instâncias de banco de dados Oracle. Usando o SSL, você pode criptografar uma conexão entre seu cliente de aplicativo e sua instância de banco de dados Oracle. O suporte a SSL está disponível em todas as regiões da AWS para Oracle.

Para habilitar a criptografia SSL para uma instância de banco de dados Oracle, adicione a opção Oracle SSL ao grupo de opções associado a instância de banco de dados. O Amazon RDS usa uma segunda porta, conforme exigido pela Oracle, para conexões SSL. Isso permite que comunicações de texto sem formatação e de texto com criptografia SSL ocorram ao mesmo tempo entre uma instância de banco de dados e um cliente Oracle. Por exemplo, você pode usar a porta com comunicação de texto simples para se comunicar com outros recursos dentro de uma VPC enquanto usa a porta com comunicação criptografada em SSL para se comunicar com recursos fora da VPC.

Para obter mais informações, consulte Oracle Secure Sockets Layer.

> **Note**
>
> Não é possível usar a Oracle Native Network Encryption (NNE) na mesma instância de banco de dados. Antes de poder usar a criptografia SSL, você deve desabilitar qualquer outra criptografia de conexão.

Atualizar aplicações para conexão com instâncias de banco de dados Oracle usando novos certificados SSL/TLS

Em 13 de janeiro de 2023, o Amazon RDS publicou novos certificados de autoridade de certificação (CA) para se conectar às suas instâncias de banco de dados do RDS via Secure Socket Layer ou Transport Layer Security (SSL/TLS). A seguir, você pode encontrar informações sobre como atualizar seus aplicativos para usar os novos certificados.

Este tópico pode ajudá-lo a determinar se algum aplicativo cliente usa SSL/TLS para conectar-se às suas instâncias de banco de dados.

> **Important**
>
> Quando você altera o certificado de uma instância de banco de dados do Amazon RDS for Oracle, somente o listener do banco de dados é reiniciado. A instância de banco de dados não é reiniciada. As conexões de banco de dados existentes não são afetadas, mas novas conexões encontrarão erros por um breve período enquanto o listener é reiniciado. Recomendamos reiniciar o banco de dados Oracle para evitar erros de conexão.

> **Note**
>
> Para aplicativos cliente que usam SSL/TLS para conectar-se às instâncias de banco de dados, atualize os armazenamentos confiáveis do aplicativo cliente para incluir os novos certificados de CA.

Depois de atualizar seus certificados de CA nos armazenamentos confiáveis do aplicativo cliente, você pode fazer o rodízio dos certificados nas suas instâncias de banco de dados. É altamente recomendável testar esses procedimentos em um ambiente de desenvolvimento ou teste antes de implementá-los em seus ambientes de produção.

Para obter mais informações sobre a mudança de certificados, consulte Alternar o certificado SSL/TLS. Para obter mais informações sobre como baixar certificados, consulte Usar SSL/TLS para criptografar uma conexão com uma instância ou um cluster de banco de dados. Para obter informações sobre o uso de SSL/TLS com instâncias de bancos de dados Oracle, consulte Oracle Secure Sockets Layer.

Tópicos

- Como descobrir se as aplicações se conectam usando SSL
- Atualizar o armazenamento confiável de aplicações
- Exemplo de código Java para estabelecer conexões SSL

Como descobrir se as aplicações se conectam usando SSL

Se a instância banco de dados Oracle usa um grupo de opção com a opção SSL adicionada, você pode estar usando SSL. Confira isso seguindo as instruções em Listar as opções e as configurações de opções para um grupo de opções. Para obter informações sobre a opção SSL, consulte Oracle Secure Sockets Layer.

Verifique o log do ouvinte para determinar se há conexões SSL. Esta é uma saída de exemplo em um log de ouvinte.

```
date time * (CONNECT_DATA=(CID=(PROGRAM=program)
(HOST=host)(USER=user))(SID=sid)) *
(ADDRESS=(PROTOCOL=tcps)(HOST=host)(PORT=port)) * establish * ORCL * 0
```

Quando PROTOCOL tem o valor `tcps` para uma entrada, ele mostra uma conexão SSL. No entanto, quando HOST é `127.0.0.1` , você pode ignorar a entrada. As conexões de `127.0.0.1` são um agente de gerenciamento local na instância de banco de dados. Essas conexões não são conexões SSL externas. Portanto, você terá aplicativos conectados usando SSL se você vir entradas de log do ouvinte em que PROTOCOL é `tcps` e HOST não é `127.0.0.1`.

Para verificar o log do ouvinte, você pode publicá-lo no Amazon CloudWatch Logs. Para obter mais informações, consulte Publicação de logs da Oracle no Amazon CloudWatch Logs.

Atualizar o armazenamento confiável de aplicações

Você pode atualizar o armazenamento confiável para aplicativos que usam conexões SQL*Plus ou JDBC para SSL/TLS.

Atualizar o armazenamento confiável de aplicações para SQL*Plus

Você pode atualizar o armazenamento confiável para aplicativos que usam conexões SQL*Plus para SSL/TLS.

> **Note**
>
> Ao atualizar o armazenamento confiável, é possível reter certificados mais antigos, além de adicionar os novos certificados.

Como atualizar o armazenamento confiável para aplicativos SQL*Plus

1. Baixe o novo certificado raiz que funciona para todas as regiões da AWS e coloque o arquivo no diretório `ssl_wallet`.

   Para obter informações sobre como baixar o certificado raiz, consulte Usar SSL/TLS para criptografar uma conexão com uma instância ou um cluster de banco de dados.
2. Execute o comando a seguir para atualizar o Oracle Wallet.

   ```
   prompt>orapki wallet add -wallet $ORACLE_HOME/ssl_wallet -trusted_cert -cert
         $ORACLE_HOME/ssl_wallet/ssl-cert.pem -auto_login_only
   ```

   Substitua o nome do arquivo pelo arquivo baixado.
3. Execute o seguinte comando para confirmar que a carteira foi atualizada com sucesso.

   ```
   prompt>orapki wallet display -wallet $ORACLE_HOME/ssl_wallet
   ```

   Sua saída deve conter o seguinte.

   ```
   Trusted Certificates:
   Subject: CN=Amazon RDS Root 2019 CA,OU=Amazon RDS,O=Amazon Web Services\,
    Inc.,L=Seattle,ST=Washington,C=US
   ```

Atualizar o armazenamento confiável de aplicações para JDBC

Você pode atualizar o armazenamento confiável para aplicativos que usam conexões JDBC para SSL/TLS.

Para obter informações sobre como baixar o certificado raiz, consulte Usar SSL/TLS para criptografar uma conexão com uma instância ou um cluster de banco de dados.

Para obter exemplos de scripts que importam certificados, consulte Script de exemplo para importar certificados para o seu armazenamento confiável.

Exemplo de código Java para estabelecer conexões SSL

O exemplo de código a seguir mostra como configurar a conexão SSL usando o JDBC.

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.util.Properties;

public class OracleSslConnectionTest {
    private static final String DB_SERVER_NAME = "<dns-name-provided-by-amazon-rds>";
    private static final Integer SSL_PORT = "<ssl-option-port-configured-in-option-
group>";
    private static final String DB_SID = "<oracle-sid>";
    private static final String DB_USER = "<user name>";
    private static final String DB_PASSWORD = "<password>";
    // This key store has only the prod root ca.
    private static final String KEY_STORE_FILE_PATH = "<file-path-to-keystore>";
    private static final String KEY_STORE_PASS = "<keystore-password>";

    public static void main(String[] args) throws SQLException {
        final Properties properties = new Properties();
        final String connectionString = String.format(
                "jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCPS)(HOST=%s)(PORT=
%d))(CONNECT_DATA=(SID=%s)))",
                DB_SERVER_NAME, SSL_PORT, DB_SID);
        properties.put("user", DB_USER);
        properties.put("password", DB_PASSWORD);
        properties.put("oracle.jdbc.J2EE13Compliant", "true");
        properties.put("javax.net.ssl.trustStore", KEY_STORE_FILE_PATH);
        properties.put("javax.net.ssl.trustStoreType", "JKS");
        properties.put("javax.net.ssl.trustStorePassword", KEY_STORE_PASS);
        final Connection connection = DriverManager.getConnection(connectionString,
 properties);
        // If no exception, that means handshake has passed, and an SSL connection can
 be opened
    }
```

```
}
```

> **Important**
>
> Depois de determinar que suas conexões com o banco de dados usam SSL/TLS e ter atualizado o armazenamento confiável de aplicações, você poderá atualizar o banco de dados para usar os certificados rds-ca-rsa2048-g1. Para obter instruções, consulte a etapa 3 em Atualizar o certificado CA modificando a instância ou o cluster de banco de dados.

Usar a criptografia de rede com uma instância de banco de dados do RDS para Oracle

O Oracle Database oferece duas maneiras de criptografar dados pela rede: criptografia de rede nativa (NNE) e Transport Layer Security (TLS). A NNE é um recurso de segurança proprietário da Oracle, enquanto o TLS é um padrão do setor. O RDS para Oracle é compatível com a NNE para todas as edições do Oracle Database.

A NNE tem as seguintes vantagens em relação ao TLS:

- Você pode controlar a NNE no cliente e no servidor usando as configurações na opção NNE:
  - `SQLNET.ALLOW_WEAK_CRYPTO_CLIENTS` e `SQLNET.ALLOW_WEAK_CRYPTO`
  - `SQLNET.CRYPTO_CHECKSUM_CLIENT` e `SQLNET.CRYPTO_CHECKSUM_SERVER`
  - `SQLNET.CRYPTO_CHECKSUM_TYPES_CLIENT` e `SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER`
  - `SQLNET.ENCRYPTION_CLIENT` e `SQLNET.ENCRYPTION_SERVER`
  - `SQLNET.ENCRYPTION_TYPES_CLIENT` e `SQLNET.ENCRYPTION_TYPES_SERVER`
- Na maioria dos casos, não é preciso configurar seu cliente nem o servidor. Por outro lado, o TLS exige que você configure o cliente e o servidor.
- Nenhum certificado é necessário. No TLS, o servidor exige um certificado (que expira) e o cliente exige um certificado raiz confiável para a autoridade de certificação que emitiu o certificado do servidor.

Para ativar a criptografia NNE para uma instância de banco de dados Oracle, adicione a opção Oracle NNE ao grupo de opções associado à instância de banco de dados. Para obter mais informações, consulte Oracle Native Network Encryption.

> **Note**
>
> Não é possível usar a NNE e o TLS na mesma instância de banco de dados.

Configurar a autenticação Kerberos do Amazon RDS para Oracle

É possível usar a autenticação Kerberos para autenticar usuários quando se conectam à instância de banco de dados do Amazon RDS para Oracle. Nesta configuração, sua instância de banco de dados funciona com AWS Directory Service for Microsoft Active Directory, também chamada AWS Managed Microsoft AD. Quando os usuários são autenticados por uma instância de banco de dados do RDS para Oracle unida ao domínio confiável, as solicitações de autenticação são encaminhadas para o diretório criado com o AWS Directory Service.

Manter todas as suas credenciais no mesmo diretório pode economizar tempo e esforço. Você tem um lugar centralizado para armazenar e gerenciar credenciais para várias instâncias de banco de dados. Um diretório também pode melhorar o perfil de segurança geral.

Disponibilidade de região e versão

A disponibilidade e a compatibilidade de recursos varia entre versões específicas de cada mecanismo de banco de dados e entre Regiões da AWS. Para obter mais informações sobre a disponibilidade de versões e regiões do RDS para Oracle com autenticação de Kerberos, consulte Regiões e mecanismos de banco de dados compatíveis com a autenticação Kerberos no Amazon RDS.

> **Note**
>
> A autenticação Kerberos não é compatível com classes de instância de banco de dados que estão obsoletas para instâncias de banco de dados do RDS para Oracle. Para obter mais informações, consulte Classes de instância de banco de dados do RDS para Oracle.

Tópicos

- Configurar o Kerberos para instâncias de banco de dados do Oracle
- Gerenciamento de uma instância de banco de dados em um domínio
- Conectar-se ao Oracle com a autenticação Kerberos

Configurar o Kerberos para instâncias de banco de dados do Oracle

Use o AWS Directory Service for Microsoft Active Directory, também chamado de AWS Managed Microsoft AD, para configurar a autenticação Kerberos para uma instância de banco de dados do Oracle. Para configurar a autenticação Kerberos, conclua as seguintes etapas:

- Etapa 1: Criar um diretório usando o AWS Managed Microsoft AD
- Etapa 2: Criar uma confiança
- Etapa 3: Configurar as permissões do IAM para o Amazon RDS
- Etapa 4: Criar e configurar usuários
- Etapa 5: Ativar o tráfego entre VPCs entre o diretório e a instância de banco de dados
- Etapa 6: Criar ou modificar uma instância de banco de dados Oracle
- Etapa 7: Criar logins do Oracle da autenticação Kerberos
- Etapa 8: Configurar um cliente Oracle

> **Note**
>
> Durante a configuração, o RDS cria um usuário de banco de dados Oracle chamado *managed_service_user*@*example.com* com o privilégio `CREATE SESSION`, onde *example.com* é seu nome de domínio. Este usuário corresponde ao usuário que o Directory Service cria dentro do Active Directory gerenciado. Periodicamente, o RDS usa as credenciais fornecidas pelo Directory Service para efetuar logon no banco de dados Oracle. Depois, o RDS destrói imediatamente o cache do tíquete.

Etapa 1: Criar um diretório usando o AWS Managed Microsoft AD

O AWS Directory Service cria um Active Directory totalmente gerenciado na Nuvem AWS. Ao criar um diretório do AWS Managed Microsoft AD, o AWS Directory Service cria dois controladores de domínio e servidores do Domain Name System (DNS) em seu nome. Os servidores do diretório são criados em sub-redes diferentes em uma VPC. Essa redundância ajuda a garantir que o diretório permaneça acessível mesmo se ocorrer uma falha.

Ao criar um diretório do AWS Managed Microsoft AD, o AWS Directory Service executa as seguintes tarefas em seu nome:

- Configura um Active Directory dentro da VPC.

- Cria uma conta de administrador do diretório com o nome de usuário Admin e a senha especificada. Use essa conta para gerenciar seu diretório.

> **Note**
>
> Salve essa senha. O AWS Directory Service não a armazena. Você pode redefini-la, mas não recuperá-la.

- Cria um grupo de segurança para os controladores do diretório.

Quando você inicia o AWS Managed Microsoft AD, o AWS cria uma Unidade organizacional (UO) que contém todos os objetos do diretório. Essa OU, que tem o nome NetBIOS que você digitou quando criou o diretório, está localizada na raiz do domínio. A raiz do domínio é controlada e de propriedade da AWS.

A conta admin que foi criada com o diretório do AWS Managed Microsoft AD tem permissões para as atividades administrativas mais comuns de sua OU:

- Criar, atualizar ou excluir usuários
- Adicionar recursos ao domínio, como servidores de arquivos ou de impressão, e atribuir permissões para esses recursos aos usuários na UO
- Criar OUs adicionais e contêineres
- Delegar autoridade
- Restaurar objetos excluídos da Lixeira do Active Directory
- Execute os módulos AD e DNS do Windows PowerShell no Active Directory Web Service

A conta admin também tem direitos para executar as seguintes atividades em todo o domínio:

- Gerenciar configurações de DNS (adicionar, remover ou atualizar registros, zonas e encaminhadores)
- Visualizar logs de eventos de DNS
- Visualizar logs de eventos de segurança

Para criar o diretório, use a API AWS Management Console, AWS CLI ou AWS Directory Service. Certifique-se de abrir as portas de saída relevantes no security group de diretórios para que o diretório possa se comunicar com a instância de banco de dados Oracle.

Como criar um diretório com AWS Managed Microsoft AD

1. Faça login no AWS Management Console e abra o console do AWS Directory Service em https://console.aws.amazon.com/directoryservicev2/.
2. No painel de navegação, escolha Directories (Diretórios) e escolha Set up directory (Configurar diretório).
3. Escolha AWS Managed Microsoft AD. O AWS Managed Microsoft AD é a única opção que você pode usar atualmente com o Amazon RDS.
4. Insira as seguintes informações:

   Nome do DNS do diretório

   O nome completo do diretório, como **`corp.example.com`**.

   Nome de NetBIOS do diretório

   O nome curto do diretório, como **`CORP`**.

   Descrição do diretório

   (Opcional) Uma descrição do diretório.

   Senha do Admin

   A senha do administrador do diretório. O processo de criação do diretório cria uma conta de administrador com o nome de usuário Admin e essa senha.

   A senha do administrador do diretório e não pode incluir a palavra "admin". A senha diferencia letras maiúsculas de minúsculas e deve ter entre 8 e 64 caracteres. Ela também precisa conter pelo menos um caractere de três das quatro categorias a seguir:

   - Letras minúsculas (a–z)
   - Letras maiúsculas (A–Z)
   - Números (0–9)
   - Caracteres não alfanuméricos (~!@#$%^&*_-+=`|\(){}[]:;"'<>,.?/)

   Confirmar senha

   A senha do administrador digitada novamente.
5. Escolha Next (Próximo).
6. Insira as seguintes informações na seção Networking (Rede) e escolha Next (Próximo):

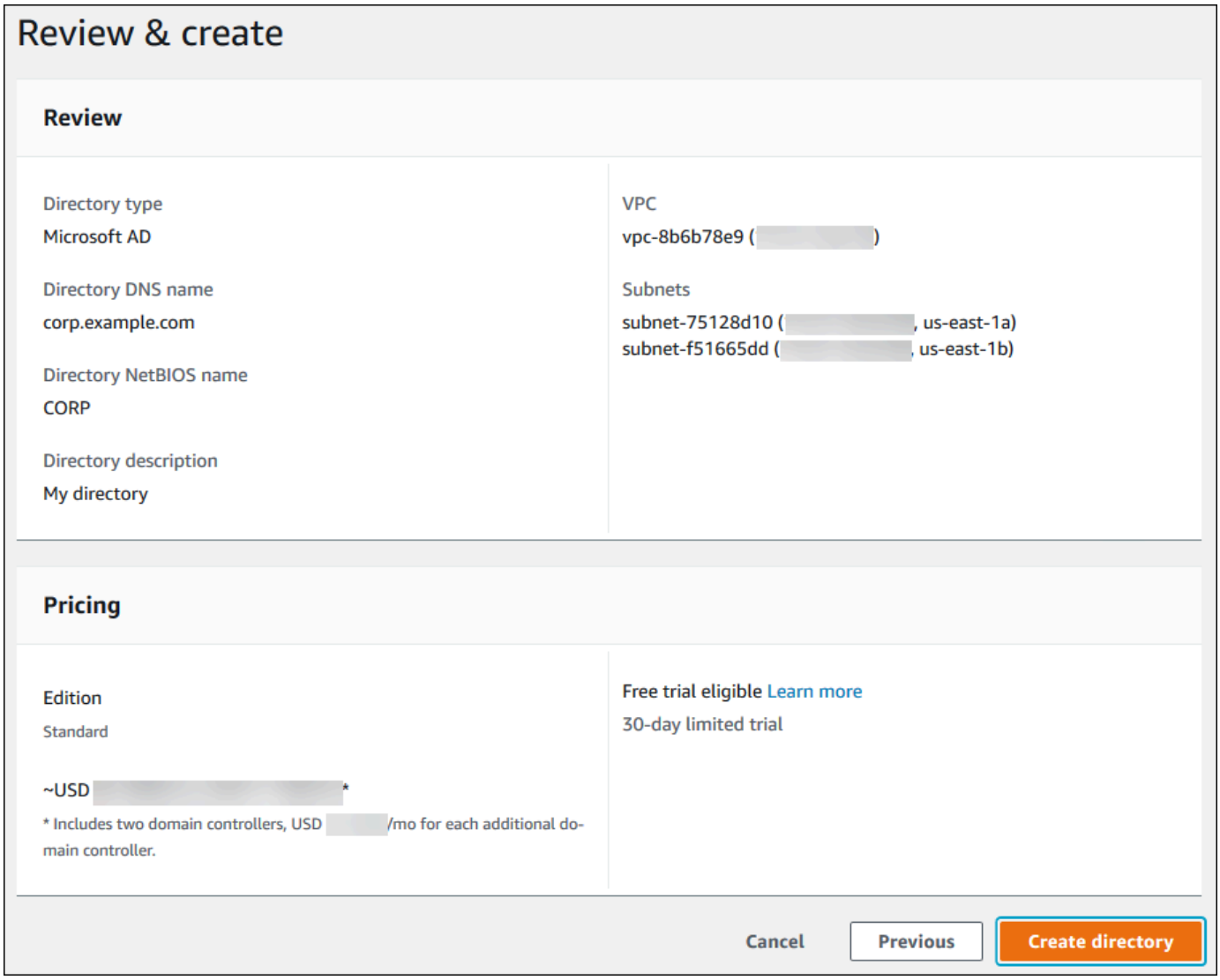
VPC

A VPC do diretório. Crie a instância de banco de dados Oracle nessa mesma VPC.

Sub-redes

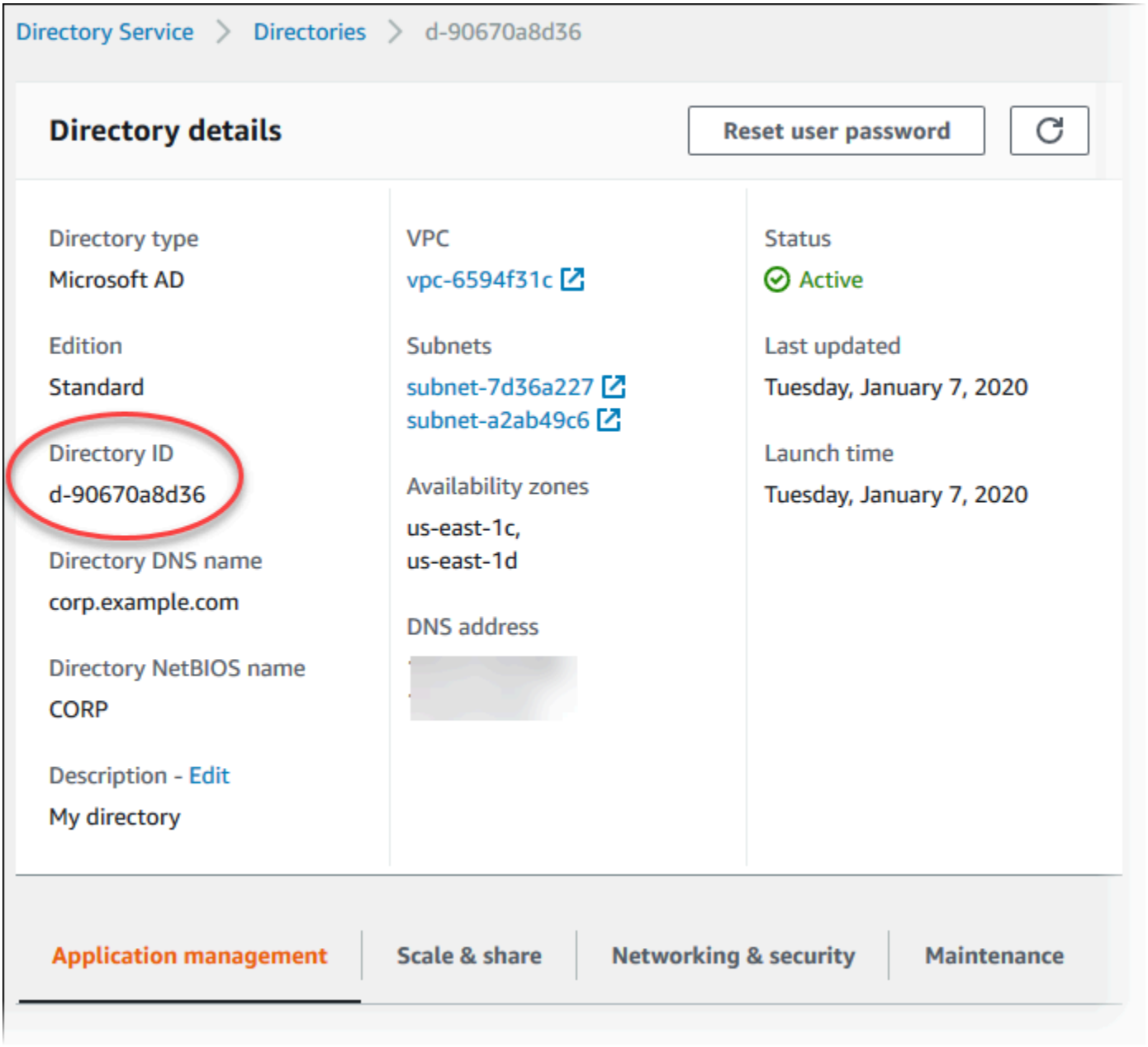
Sub-redes para os servidores do diretório. As duas sub-redes deve estar em diferentes zonas de disponibilidade.

7. Revise as informações do diretório e faça as alterações necessárias. Quando as informações estiverem corretas, selecione Create directory (Criar diretório).



A criação do diretório leva vários minutos. Depois que o diretório tiver sido criado com sucesso, o valor de Status muda para Active (Ativo).

Para ver informações sobre o diretório, selecione o nome do diretório na listagem de diretórios. Anote o valor do Directory ID (ID de diretório) pois você precisará desse valor ao criar ou modificar a instância de banco de dados Oracle.



Etapa 2: Criar uma confiança

Se você planeja usar somente o AWS Managed Microsoft AD, vá para Etapa 3: Configurar as permissões do IAM para o Amazon RDS.

Para habilitar a autenticação Kerberos usando seu Active Directory autogerenciado, você deve criar uma relação de confiança de floresta entre seu Active Directory autogerenciado e o AWS Managed

Microsoft AD criado na etapa anterior. A relação de confiança pode ser unidirecional, na qual o AWS Managed Microsoft AD confia no Active Directory autogerenciado. A confiança também pode ser bidirecional, onde os dois Active Directories confiam um no outro. Para ter mais informações sobre como configurar confianças de floresta usando o AWS Directory Service, consulte Quando criar um relacionamento de confiança no Guia de administração do AWS Directory Service.

Etapa 3: Configurar as permissões do IAM para o Amazon RDS

Para chamar o AWS Directory Service para você, o Amazon RDS precisa de um perfil do IAM que use a política gerenciada do IAM `AmazonRDSDirectoryServiceAccess`. Esta função permite que o Amazon RDS faça chamadas ao AWS Directory Service.

> **Note**
>
> Para o perfil permitir o acesso, o endpoint do AWS Security Token Service (AWS STS) deve estar ativado na Região da AWS correta de sua Conta da AWS. Os endpoints do AWS STS são ativados por padrão em todas as Regiões da AWS e você pode usá-los sem precisar tomar medidas adicionais. Para ter mais informações, consulte Ativar e desativar o AWS STS em uma Região da AWS no Guia do usuário do IAM.

Criar um perfil do IAM

Quando você cria uma instância de banco de dados usando o AWS Management Console e o usuário do console tem a permissão `iam:CreateRole`, o console cria essa `rds-directoryservice-kerberos-access-role` automaticamente. Caso contrário, é necessário criar a função do IAM manualmente. Ao criar um perfil do IAM, escolha `Directory Service` e associe a política gerenciada `AmazonRDSDirectoryServiceAccess` da AWS a ele.

Para ter mais informações sobre como criar funções do IAM para um serviço, consulte o tópico sobre como Criar uma função para delegar permissões a um serviço da AWS, no Guia do usuário do IAM.

> **Note**
>
> A função do IAM usada para a autenticação do Windows para RDS para o Microsoft SQL Server não pode ser usada para o RDS for Oracle.

Criar uma política de confiança do IAM manualmente

Se preferir, você poderá criar políticas de recursos com as permissões exigidas em vez de usar a política gerenciada do IAM `AmazonRDSDirectoryServiceAccess`. Especifique `directoryservice.rds.amazonaws.com` e `rds.amazonaws.com` como entidades principais.

Recomendamos o uso das chaves de contexto de condição global `aws:SourceArn` e `aws:SourceAccount` em políticas de recursos para limitar as permissões que o Amazon RDS concede a outro serviço para um recurso específico. A maneira mais eficaz de se proteger contra o problema do substituto confuso é usar a chave de contexto de condição global `aws:SourceArn` com o ARN completo do recurso do Amazon RDS. Para obter mais informações, consulte Prevenção do problema do substituto confuso entre serviços.

O exemplo a seguir mostra como é possível usar as chaves de contexto de condição global `aws:SourceArn` e `aws:SourceAccount` no Amazon RDS, a fim de evitar o problema do substituto confuso.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "directoryservice.rds.amazonaws.com",
          "rds.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "ArnLike": {
          "aws:SourceArn": "arn:aws:rds:us-east-1:123456789012:db:mydbinstance"
        },
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        }
      }
    }
```

```
  ]
}
```

Em relação a regiões de adesão, você também deve incluir uma entidade principal de serviço para essa região na forma de `directoryservice.rds.region_name.amazonaws.com`. Por exemplo, na região África (Cidade do Cabo), use a seguinte política de confiança:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "directoryservice.rds.amazonaws.com",
          "directoryservice.rds.af-south-1.amazonaws.com",
          "rds.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "ArnLike": {
          "aws:SourceArn": "arn:aws:rds:af-south-1:123456789012:db:mydbinstance"
        },
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        }
      }
    }
  ]
}
```

O perfil também deve ter a política do IAM a seguir.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "ds:DescribeDirectories",
        "ds:AuthorizeApplication",
        "ds:UnauthorizeApplication",
        "ds:GetAuthorizedApplicationDetails"
      ],
    "Effect": "Allow",
    "Resource": "*"
    }
  ]
}
```

Etapa 4: Criar e configurar usuários

É possível criar usuários com a ferramenta Usuários e computadores do Active Directory, que faz parte dos Serviços de Domínio Active Directory e das ferramentas do Active Directory Lightweight Directory Services. Nesse caso, os usuários são pessoas ou entidades individuais que têm acesso ao diretório.

Para criar usuários em um diretório do AWS Directory Service, é necessário estar conectado a uma instância do Amazon EC2 baseada no Windows que seja membro do diretório do AWS Directory Service. Ao mesmo tempo, é necessário estar conectado como um usuário que tenha privilégios para criar usuários. Para obter mais informações sobre como criar usuários no Microsoft Active Directory, consulte Gerenciar usuários e grupos em AWS Managed Microsoft AD no AWS Directory ServiceGuia de administração

Etapa 5: Ativar o tráfego entre VPCs entre o diretório e a instância de banco de dados

Se você planeja localizar o diretório e a instância de banco de dados na mesma VPC, ignore esta etapa e prossiga para Etapa 6: Criar ou modificar uma instância de banco de dados Oracle.

Se você planeja localizar o diretório e a instância de banco de dados em VPCs ou contas da AWS diferentes, configure o tráfego entre VPCs usando o emparelhamento de VPCs ou o AWS Transit

Gateway. O procedimento a seguir habilita o tráfego entre VPCs usando o emparelhamento de VPCs. Siga as instruções em O que é o emparelhamento de VPCs? no Guia de emparelhamento do Amazon Virtual Private Cloud.

Como habilitar o tráfego entre VPCs usando o emparelhamento de VPCs

1. Configure regras apropriadas de roteamento de VPC para garantir que o tráfego de rede possa fluir em ambos os sentidos.
2. Certifique-se de que o grupo de segurança da instância de banco de dados possa receber o tráfego de entrada do grupo de segurança do diretório. Para obter mais informações, consulte Práticas recomendadas para AWS Managed Microsoft AD no Guia deAWS Directory Service Administração.
3. Garanta que não há nenhuma regra na lista de controle de acesso (ACL) de rede para bloquear o tráfego.

Se uma conta diferente da AWS for proprietária do diretório, é necessário compartilhá-lo.

Como compartilhar o diretório entre contas da AWS.

1. Inicie o compartilhamento do diretório com a conta da AWS na qual a instância de banco de dados será criada seguindo as instruções em Tutorial: Compartilhar o diretório AWS Managed Microsoft AD para ingresso perfeito no domínio do EC2 no AWS Directory ServiceGuia de administração.
2. Faça login no console do AWS Directory Service usando a conta para a instância de banco de dados e garanta que o domínio tenha o status `SHARED` antes de prosseguir.
3. Enquanto estiver conectado ao console do AWS Directory Service usando a conta da instância de banco de dados, observe o valor do Directory ID (ID do diretório). Use esse ID do diretório para associar a instância de banco de dados ao domínio.

Etapa 6: Criar ou modificar uma instância de banco de dados Oracle

Crie ou modifique uma instância de banco de dados do Oracle para ser usada com o diretório. É possível usar o console, a CLI ou a API do RDS para associar uma instância de banco de dados a um diretório. Você pode fazer isso por meio de uma das seguintes maneiras:

- Crie uma instância de banco de dados do Oracle usando o console, o comando create-db-instance da CLI ou a operação de API CreateDBInstance do RDS.

Para obter instruções, consulte Criar uma instância de banco de dados do Amazon RDS.

- Modifique uma instância de banco de dados existente do Oracle usando o console, o comando modify-db-instance da CLI ou a operação de API ModifyDBInstance do RDS.

  Para obter instruções, consulte Modificar uma instância de banco de dados do Amazon RDS.

- Restaure uma instância de banco de dados de um snapshot de banco de dados usando o console, o comando restore-db-instance-from-db-snapshot da CLI ou a operação de API RestoreDBInstanceFromDBSnapshot do RDS.

  Para obter instruções, consulte Restaurar uma instância de banco de dados.

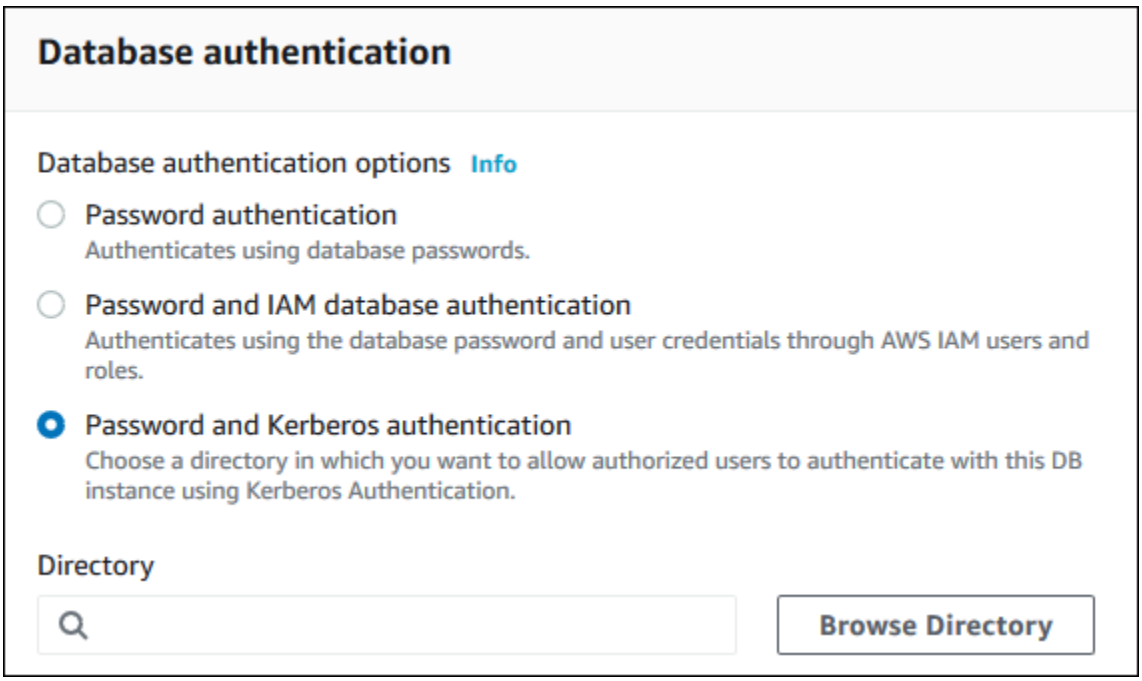
- Restaure uma instância de banco de dados para determinado momento usando o console, o comando restore-db-instance-to-point-in-time da CLI ou a operação de API RestoreDBInstanceToPointInTime do RDS.

  Para obter instruções, consulte Restaurar uma instância de banco de dados para um momento especificado no Amazon RDS.

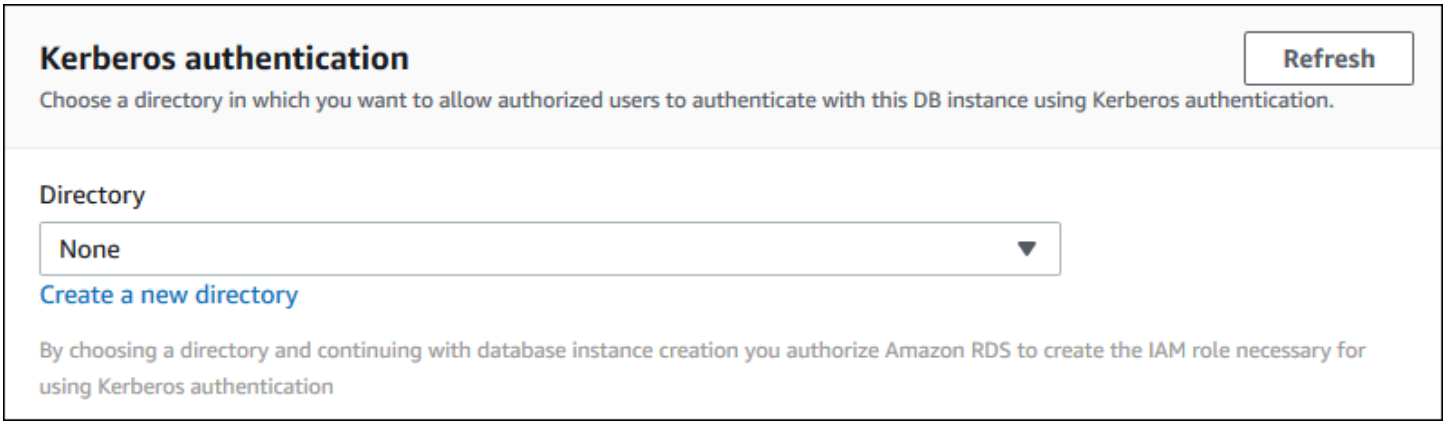
A autenticação Kerberos só é compatível com instâncias de banco de dados do Oracle em uma VPC. A instância de banco de dados pode estar na mesma VPC do diretório ou em uma VPC diferente. Ao criar ou modificar a instância de banco de dados, faça o seguinte:

- Forneça o identificador de domínio (identificador `d-*`) que foi gerado quando você criou seu diretório.
- Forneça o nome da função do IAM criada.
- Certifique-se de que o security group da instância de banco de dados possa receber tráfego de entrada do security group de diretórios e enviar tráfego de saída para o diretório.

Ao usar o console para criar uma instância de banco de dados, selecione Password and Kerberos authentication (Senha e autenticação Kerberos) na seção Database authentication (Autenticação de banco de dados). Escolha Browse Directory (Procurar diretório) e selecione o diretório ou escolha Create a new directory (Criar um diretório).



Ao usar o console para modificar ou restaurar uma instância de banco de dados, escolha o diretório na seção Kerberos authentication (Autenticação Kerberos) ou escolha Create a new directory (Criar um diretório).



Quando você usa a AWS CLI, são necessários os seguintes parâmetros para que a instância de banco de dados possa usar o diretório criado:

- Para o parâmetro `--domain`, use o identificador de domínio (identificador "d-*") gerado quando o diretório foi criado.
- Para o parâmetro `--domain-iam-role-name`, use a função criada que usa a política gerenciada `AmazonRDSDirectoryServiceAccess` do IAM.

Por exemplo, o comando da CLI a seguir modifica uma instância de banco de dados para usar um diretório.

Para Linux, macOS ou Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --domain d-ID \
    --domain-iam-role-name role-name
```

Para Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --domain d-ID ^
    --domain-iam-role-name role-name
```

Important

Se você modificar uma instância de banco de dados para permitir a autenticação Kerberos, reinicialize a instância de banco de dados após fazer a alteração.

Note

MANAGED_SERVICE_USER é uma conta de serviço cujo nome é gerado aleatoriamente pelo Directory Service para RDS. Durante a configuração de autenticação Kerberos, o RDS for Oracle cria um usuário com o mesmo nome e atribui a ele o privilégio `CREATE SESSION`. O usuário do Oracle DB é identificado externamente como *MANAGED_SERVICE_USER@EXAMPLE.COM*, em que *EXAMPLE.COM* é o nome de seu domínio. Periodicamente, o RDS usa as credenciais fornecidas pelo Serviço de Diretório para efetuar logon no banco de dados Oracle. Depois, o RDS destrói imediatamente o cache do tíquete.

Etapa 7: Criar logins do Oracle da autenticação Kerberos

Use as credenciais de usuário mestre do Amazon RDS para se conectar à instância de banco de dados do Oracle como você faz com qualquer outra instância de banco de dados. A instância

de banco de dados é ingressada no domínio AWS Managed Microsoft AD. Portanto, é possível provisionar logins e usuários do Oracle com base em usuários do Microsoft Active Directory em seu domínio. Para gerenciar permissões de banco de dados, conceda e revogue permissões padrão do Oracle para esses logins.

Para permitir que um usuário do Microsoft Active Directory se autentique com o Oracle

1. Conecte-se à instância de banco de dados Oracle usando suas credenciais de usuário Amazon RDS mestre.
2. Crie um usuário autenticado externamente no banco de dados Oracle.

   No exemplo a seguir, substitua *KRBUSER@CORP.EXAMPLE.COM* pelo nome de usuário e pelo nome de domínio.

   ```
   CREATE USER "KRBUSER@CORP.EXAMPLE.COM" IDENTIFIED EXTERNALLY;
   GRANT CREATE SESSION TO "KRBUSER@CORP.EXAMPLE.COM";
   ```

   Os usuários (humanos e aplicações) de seu domínio agora podem se conectar à instância de banco de dados Oracle em uma máquina cliente conectada ao domínio usando a autenticação Kerberos.

Etapa 8: Configurar um cliente Oracle

Para configurar um cliente Oracle, atenda aos seguintes requisitos:

- Crie um arquivo de configuração chamado krb5.conf (Linux) ou krb5.ini (Windows) para apontar para o domínio. Configure o cliente Oracle para usar este arquivo de configuração.
- Verifique se o tráfego pode fluir entre o host do cliente e o AWS Directory Service pela porta DNS 53 por TCP/UDP, portas Kerberos (88 e 464 para AWS Directory Service gerenciado) por TCP e porta LDAP 389 por TCP.
- Verifique se o tráfego pode fluir entre o host do cliente e a instância de banco de dados pela porta do banco de dados.

Este é o conteúdo de amostra para AWS Managed Microsoft AD.

```
[libdefaults]
 default_realm = EXAMPLE.COM
[realms]
```

```
 EXAMPLE.COM = {
  kdc = example.com
  admin_server = example.com
 }
[domain_realm]
 .example.com = CORP.EXAMPLE.COM
 example.com = CORP.EXAMPLE.COM
```

Veja a seguir um exemplo de conteúdo para o Microsoft AD on-premises: No arquivo krb5.conf ou krb5.ini, substitua *on-prem-ad-server-name* pelo nome do seu servidor AD local.

```
[libdefaults]
 default_realm = ONPREM.COM
[realms]
 AWSAD.COM = {
  kdc = awsad.com
  admin_server = awsad.com
 }
 ONPREM.COM = {
  kdc = on-prem-ad-server-name
  admin_server = on-prem-ad-server-name
 }
[domain_realm]
 .awsad.com = AWSAD.COM
 awsad.com= AWSAD.COM
 .onprem.com = ONPREM.COM
 onprem.com= ONPREM.COM
```

> **Note**
>
> Depois de configurar o arquivo krb5.ini ou krb5.conf, recomendamos que você reinicie o servidor.

Veja a seguir um exemplo de conteúdo sqlnet.ora para uma configuração do SQL*Plus:

```
SQLNET.AUTHENTICATION_SERVICES=(KERBEROS5PRE,KERBEROS5)
SQLNET.KERBEROS5_CONF=path_to_krb5.conf_file
```

Para obter um exemplo de uma configuração do SQL Developer, consulte o Documento 1609359.1 no suporte da Oracle.

Gerenciamento de uma instância de banco de dados em um domínio

É possível usar o console, a CLI ou a API do RDS para gerenciar a instância de banco de dados e suas relações com o Microsoft Active Directory. Por exemplo, é possível associar um Microsoft Active Directory para permitir a autenticação Kerberos. Também é possível desassociar um Microsoft Active Directory para desabilitar a autenticação Kerberos. Também é possível mover uma instância de banco de dados para ser autenticada externamente por um Microsoft Active Directory para outra.

Por exemplo, usando a CLI, é possível fazer o seguinte:

- Para tentar habilitar a autenticação Kerberos novamente para uma assinatura com falha, use o comando modify-db-instance da CLI e especifique o ID do diretório atual da assinatura para a opção `--domain`.
- Para desabilitar a autenticação Kerberos em uma instância de banco de dados, use o comando modify-db-instance da CLI e especifique `none` para a opção `--domain`.
- Para mover uma instância de banco de dados de um domínio para outro, use o comando modify-db-instance da CLI e especifique o identificador de domínio do novo domínio para a opção `--domain`.

Visualizar o status da associação ao domínio

Após criar ou modificar a instância de banco de dados, ela se tornará um membro do domínio. É possível visualizar o status da assinatura do domínio para a instância de banco de dados no console ou executando o comando describe-db-instances da CLI. O status da instância de banco de dados pode ser um dos seguintes:

- `kerberos-enabled` – a instância de banco de dados que tem a autenticação Kerberos habilitada.
- `enabling-kerberos`: a AWS está no processo de habilitar a autenticação Kerberos nessa instância de bancos de dados.
- `pending-enable-kerberos` – a habilitação da autenticação Kerberos está pendente nessa instância de banco de dados.
- `pending-maintenance-enable-kerberos`: a AWS tentará habilitar a autenticação Kerberos na instância de bancos de dados durante a próxima janela de manutenção programada.
- `pending-disable-kerberos` – a desabilitação da autenticação Kerberos está pendente nessa instância de banco de dados.

- `pending-maintenance-disable-kerberos`: a AWS tentará desabilitar a autenticação Kerberos na instância de banco de dados durante a próxima janela de manutenção programada.
- `enable-kerberos-failed`: um problema de configuração impediu que a AWS habilitasse a autenticação Kerberos na instância de banco de dados. Corrija o problema de configuração antes de emitir o comando novamente para modificar a instância de banco de dados.
- `disabling-kerberos`: a AWS está no processo de desabilitar a autenticação Kerberos nessa instância de bancos de dados.

Uma solicitação para habilitar a autenticação Kerberos pode falhar por conta de um novo problema de conectividade de rede ou de uma função do IAM incorreta. Se a tentativa de habilitar a autenticação Kerberos falhar ao criar ou modificar uma instância de banco de dados, verifique se você está usando a função do IAM correta. Depois, modifique a instância de banco de dados para ingressar no domínio.

> **Note**
>
> Somente a autenticação Kerberos com o Amazon RDS para Oracle envia tráfego para os servidores DNS do domínio. Todas as outras solicitações de DNS são tratadas como acesso à rede de saída nas instâncias de bancos de dados que executam o Oracle. Para obter mais informações sobre o acesso à rede de saída com o Amazon RDS para Oracle, consulte Configuração de um servidor DNS personalizado.

Chaves Kerberos de alternância forçada

Uma chave secreta é compartilhada entre AWS Managed Microsoft AD e o Amazon RDS para Oracle para a instância de banco de dados Oracle. Essa chave é alternada automaticamente a cada 45 dias. É possível usar o procedimento do Amazon RDS a seguir para forçar a rotação dessa chave.

```
SELECT rdsadmin.rdsadmin_kerberos_auth_tasks.rotate_kerberos_keytab AS TASK_ID FROM
 DUAL;
```

> **Note**
>
> Em uma configuração de réplica de leitura, esse procedimento está disponível apenas na instância de banco de dados de origem e não na réplica de leitura.

A instrução SELECT retorna o ID da tarefa em um tipo de dados VARCHAR2. Você pode visualizar o status de uma tarefa em andamento em um arquivo bdump. Os arquivos bdump estão localizados no diretório /rdsdbdata/log/trace. Cada nome de arquivo bdump está no seguinte formato.

```
dbtask-task-id.log
```

Você pode visualizar o resultado exibindo o arquivo de saída da tarefa.

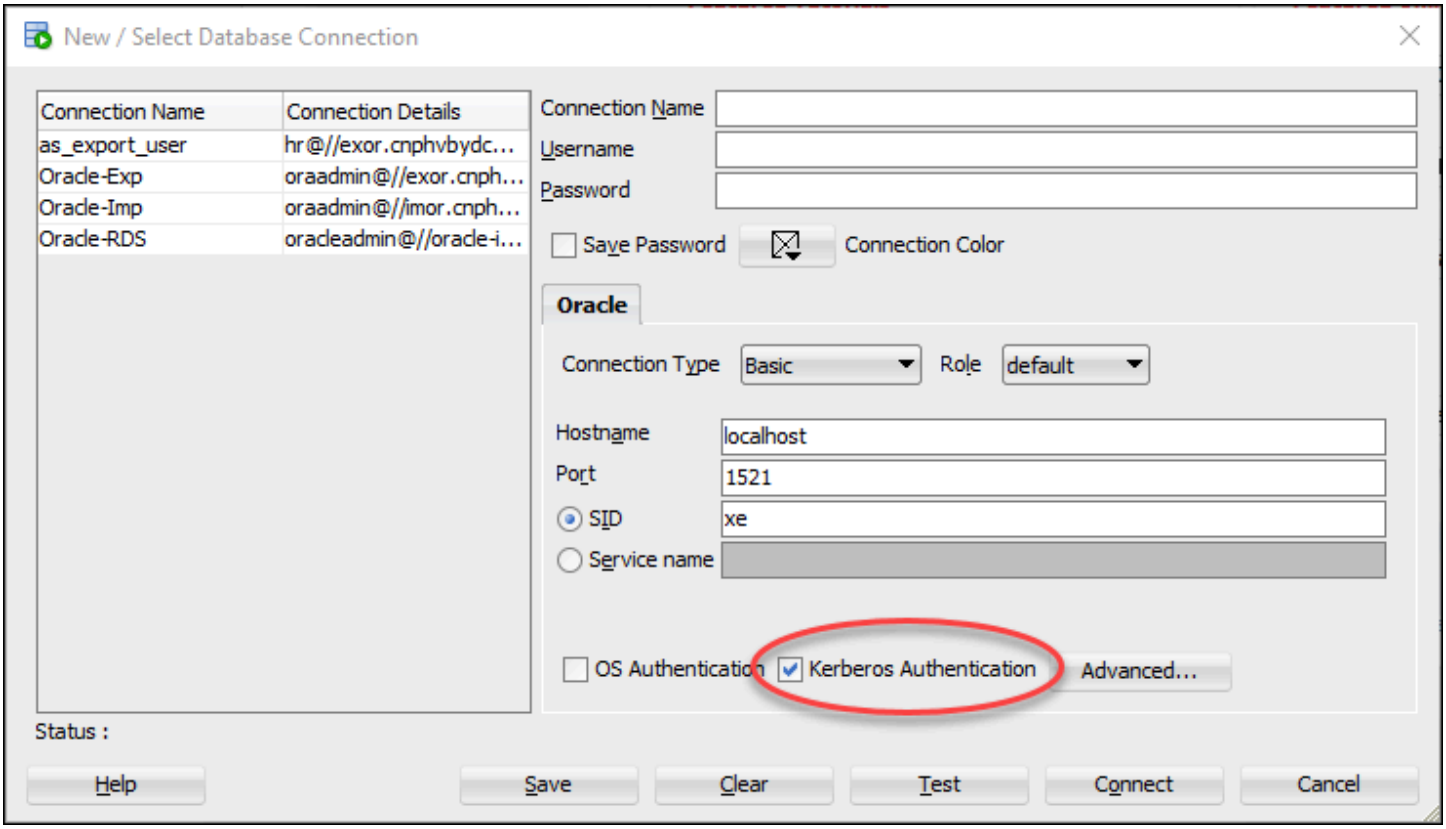
```
SELECT text FROM table(rdsadmin.rds_file_util.read_text_file('BDUMP','dbtask-task-
id.log'));
```

Substitua *task-id* pelo ID da tarefa retornado pelo procedimento.

> **Note**
>
> As tarefas são executadas de forma assíncrona.

Conectar-se ao Oracle com a autenticação Kerberos

Esta seção pressupõe que você tenha configurado o cliente Oracle conforme descrito em Etapa 8: Configurar um cliente Oracle. Para conectar-se ao banco de dados Oracle com a autenticação Kerberos, faça login usando o tipo de autenticação Kerberos. Por exemplo, após iniciar o Oracle SQL Developer, selecione Autenticação do Kerberos como o tipo de autenticação, conforme mostrado no exemplo a seguir.



Para conectar-se ao Oracle com a autenticação Kerberos com SQL*Plus:

1. Em um prompt de comando, execute o seguinte comando:

   ```
   kinit username
   ```

   Substitua *username* pelo nome de usuário e, no prompt, insira a senha armazenada no Microsoft Active Directory para o usuário.

2. Abra o SQL*Plus e conecte-se usando o nome de DNS e o número da porta da instância de banco de dados do Oracle.

   Para obter mais informações sobre como se conectar à instância de banco de dados no SQL*Plus, consulte Conectar-se à sua instância de banco de dados do Linux usando o SQL*Plus.

> **Tip**
>
> Se estiver usando um cache nativo do Windows, você também poderá definir o parâmetro `SQLNET.KERBEROS5_CC_NAME` para `OSMSFT://` ou `MSLSA` no arquivo sqlnet.ora para usar as credenciais armazenadas no Microsoft Active Directory.

Configurar o acesso UTL_HTTP usando certificados e uma carteira Oracle

O Amazon RDS comporta o acesso à rede de saída nas instâncias de banco de dados Oracle. Para conectar sua instância de banco de dados à rede, você pode usar os seguintes pacotes PL/SQL:

`UTL_HTTP`

Este pacote faz chamadas HTTP do SQL e PL/SQL. Você pode usá-lo para acessar dados na Internet com HTTP. Para obter mais informações, consulte UTL_HTTP na documentação da Oracle.

`UTL_TCP`

Este pacote fornece funcionalidade de acesso TCP/IP do lado do cliente em PL/SQL. Esse pacote é útil para aplicações PL/SQL que usam protocolos de Internet e e-mail. Para obter mais informações, consulte UTL_TCPna documentação da Oracle.

`UTL_SMTP`

Este pacote fornece interfaces aos comandos SMTP que permitem que um cliente despache e-mails para um servidor SMTP. Para obter mais informações, consulte UTL_SMTP na documentação da Oracle.

Ao concluir as tarefas a seguir, você pode configurar o `UTL_HTTP.REQUEST` para trabalhar com sites que exigem certificados de autenticação de cliente durante o handshake SSL. Você também pode configurar a autenticação por senha para acesso `UTL_HTTP` a sites modificando os comandos de geração da carteira Oracle e o procedimento `DBMS_NETWORK_ACL_ADMIN.APPEND_WALLET_ACE`. Para obter mais informações, consulte DBMS_NETWORK_ACL_ADMIN na documentação do Oracle Database.

> **Note**
>
> Você pode adaptar as seguintes tarefas para o UTL_SMTP, que permite enviar e-mails por SSL/TLS (incluindo o Amazon Simple Email Service).

Tópicos

- Considerações ao configurar o acesso UTL_HTTP
- Etapa 1: obter o certificado raiz de um site
- Etapa 2: criar uma carteira Oracle
- Etapa 3: baixar a carteira Oracle para sua instância do RDS for Oracle
- Etapa 4: conceder permissões de usuário para a carteira Oracle
- Etapa 5: configurar o acesso a um site da sua instância de banco de dados
- Etapa 6: testar conexões de sua instância de banco de dados para um site

Considerações ao configurar o acesso UTL_HTTP

Antes de configurar o acesso, pense no seguinte:

- É possível usar SMTP com a opção UTL_MAIL. Para obter mais informações, consulte Oracle UTL_MAIL.
- O nome do servidor de nomes de domínio (DNS) do host remoto pode ser um dos seguintes:
  - Resolvível publicamente.
  - O endpoint de uma instância de banco de dados do Amazon RDS.
  - Resolvível através de um servidor DNS personalizado. Para obter mais informações, consulte Configuração de um servidor DNS personalizado.
  - O nome DNS privado de uma instância do Amazon EC2 na mesma VPC ou em uma VPC emparelhada. Nesse caso, certifique-se de que o nome seja resolvível através de um servidor DNS personalizado. Como alternativa, para usar o DNS fornecido pela Amazon, você pode habilitar o atributo `enableDnsSupport` nas configurações da VPC e habilitar o suporte à resolução de DNS para a conexão de emparelhamento de VPC. Para obter mais informações, consulte Suporte ao DNS na VPC e Modificação da conexão de emparelhamento de VPC.
  - Para se conectar com segurança a recursos remotos de SSL/TLS, você pode criar e carregar carteiras Oracle personalizadas. Usando o recurso de Integração do Amazon S3 com o Amazon

RDS for Oracle, você pode baixar uma carteira do Amazon S3 em instâncias de banco de dados Oracle. Para obter informações sobre a integração do Amazon S3 para Oracle, consulte Integração do Amazon S3.

- Você poderá estabelecer links de banco de dados entre instâncias de bancos de dados Oracle em um endpoint SSL/TLS se a opção Oracle SSL estiver configurada para cada instância. Nenhuma outra configuração é necessária. Para obter mais informações, consulte Oracle Secure Sockets Layer.

Etapa 1: obter o certificado raiz de um site

Para que a instância de banco de dados do RDS para Oracle faça conexões seguras com um site, adicione o certificado de CA raiz. O Amazon RDS usa o certificado raiz para assinar o certificado do site para a carteira Oracle.

Você pode obter o certificado raiz de várias maneiras. Por exemplo, você pode fazer o seguinte:

1. Use um servidor Web para visitar o site protegido pelo certificado.
2. Baixe o certificado raiz usado para assinatura.

Para serviços da AWS, os certificados geralmente podem ser encontrados no Repositório de serviços de confiança da Amazon.

Etapa 2: criar uma carteira Oracle

Crie uma carteira Oracle que contenha os certificados do servidor Web e os certificados de autenticação do cliente. A instância Oracle do RDS usa o certificado do servidor Web para estabelecer uma conexão segura com o site. O site precisa do certificado do cliente para autenticar o usuário do banco de dados Oracle.

Talvez você queira configurar conexões seguras sem usar certificados de cliente para autenticação. Nesse caso, você pode ignorar as etapas do armazenamento de chaves Java no procedimento a seguir.

Para criar uma carteira Oracle

1. Coloque os certificados raiz e do cliente em um único diretório e, em seguida, mude para esse diretório.
2. Converta o certificado do cliente .p12 para o repositório de chaves Java.

> **Note**
>
> Se não estiver usando certificados de cliente para autenticação, ignore esta etapa.

O exemplo a seguir converte o certificado do cliente chamado *client_certificate.p12* para o repositório de chaves Java chamado *client_keystore.jks*. O repositório de chaves é então incluído na carteira Oracle. A senha do repositório de chaves é *P12PASSWORD*.

```
orapki wallet pkcs12_to_jks -wallet ./client_certificate.p12 -
jksKeyStoreLoc ./client_keystore.jks -jksKeyStorepwd P12PASSWORD
```

3. Crie um diretório para sua carteira Oracle diferente do diretório de certificados.

   O exemplo a seguir cria o diretório `/tmp/wallet`.

```
mkdir -p /tmp/wallet
```

4. Crie uma carteira Oracle no diretório da carteira.

   O exemplo a seguir define a senha da carteira Oracle como *P12PASSWORD*, que é a mesma senha usada pelo repositório de chaves Java em uma etapa anterior. Usar a mesma senha é conveniente, mas não é necessário. O parâmetro `-auto_login` ativa o recurso de login automático para que você não precise especificar uma senha toda vez que quiser acessá-la.

   > **Note**
   >
   > Especifique uma senha diferente do prompt mostrado aqui como prática recomendada de segurança.

```
orapki wallet create -wallet /tmp/wallet -pwd P12PASSWORD -auto_login
```

5. Adicione o repositório de chaves Java à sua carteira Oracle.

   > **Note**
   >
   > Se não estiver usando certificados de cliente para autenticação, ignore esta etapa.

O exemplo a seguir inclui o repositório de chaves *client_keystore.jks* para a carteira Oracle chamada */tmp/wallet*. Neste exemplo, você especifica a mesma senha para o repositório de chaves Java e para a carteira Oracle.

```
orapki wallet jks_to_pkcs12 -wallet /tmp/wallet -pwd P12PASSWORD -
keystore ./client_keystore.jks -jkspwd P12PASSWORD
```

6. Adicione o certificado raiz do seu site de destino à carteira Oracle.

   O exemplo a seguir inclui um certificado chamado *Root_CA.cer*.

   ```
   orapki wallet add -wallet /tmp/wallet -trusted_cert -cert ./Root_CA.cer -
   pwd P12PASSWORD
   ```

7. Inclua qualquer certificado intermediário.

   O exemplo a seguir inclui um certificado chamado *Intermediate.cer*. Repita essa etapa quantas vezes for necessário para carregar todos os certificados intermediários.

   ```
   orapki wallet add -wallet /tmp/wallet -trusted_cert -cert ./Intermediate.cer -
   pwd P12PASSWORD
   ```

8. Confirme se sua carteira Oracle recém-criada tem os certificados necessários.

   ```
   orapki wallet display -wallet /tmp/wallet -pwd P12PASSWORD
   ```

Etapa 3: baixar a carteira Oracle para sua instância do RDS for Oracle

Nesta etapa, você carrega sua carteira Oracle para o Simple Storage Service (Simple Storage Service (Amazon S3)) e, em seguida, baixa a carteira do Simple Storage Service (Amazon S3) para sua instância do RDS for Oracle.

Para baixar sua carteira Oracle em sua instância de banco de dados do RDS for Oracle

1. Conclua os pré-requisitos para a integração do Amazon S3 com o Oracle e adicione a opção `S3_INTEGRATION` à sua instância de banco de dados Oracle. Certifique-se de que a função do IAM para a opção tenha acesso ao bucket do Amazon S3 que você está usando.

Para obter mais informações, consulte Integração do Amazon S3.

2. Faça login na sua instância de banco de dados como o usuário primário e crie um diretório Oracle para manter a carteira Oracle.

   O exemplo a seguir cria um diretório Oracle chamado *WALLET_DIR*.

   ```
   EXEC rdsadmin.rdsadmin_util.create_directory('WALLET_DIR');
   ```

   Para obter mais informações, consulte Criar e eliminar diretórios no espaço de armazenamento de dados principal.

3. Carregue a carteira Oracle para o seu bucket do Simple Storage Service (Amazon S3).

   Você pode usar qualquer técnica de carregamento compatível.

4. Se estiver carregando novamente uma carteira Oracle, exclua a carteira existente. Caso contrário, vá para a próxima etapa.

   O exemplo a seguir remove a carteira existente chamada *cwallet.sso*.

   ```
   EXEC UTL_FILE.FREMOVE ('WALLET_DIR','cwallet.sso');
   ```

5. Baixe a carteira Oracle do seu bucket do Simple Storage Service (Simple Storage Service (Amazon S3)) para a instância de banco de dados Oracle.

   O exemplo a seguir baixa a carteira chamada *cwallet.sso* do bucket do Amazon S3 chamado *my_s3_bucket* para o diretório da instância de banco de dados chamado *WALLET_DIR*.

   ```
   SELECT rdsadmin.rdsadmin_s3_tasks.download_from_s3(
         p_bucket_name    =>  'my_s3_bucket',
         p_s3_prefix      =>  'cwallet.sso',
         p_directory_name =>  'WALLET_DIR')
      AS TASK_ID FROM DUAL;
   ```

6. (Opcional) Baixe uma carteira Oracle protegida por senha.

   Baixe esta carteira somente se quiser exigir uma senha para cada uso da carteira. O exemplo a seguir baixa a carteira protegida por senha *ewallet.p12*.

   ```
   SELECT rdsadmin.rdsadmin_s3_tasks.download_from_s3(
         p_bucket_name    =>  'my_s3_bucket',
   ```

```
      p_s3_prefix      =>  'ewallet.p12',
      p_directory_name =>  'WALLET_DIR')
   AS TASK_ID FROM DUAL;
```

7. Verifique o status da sua tarefa do banco de dados.

   Substitua o ID da tarefa retornado das etapas anteriores para *dbtask-1234567890123-4567.log* no exemplo a seguir.

```
SELECT TEXT FROM
 TABLE(rdsadmin.rds_file_util.read_text_file('BDUMP','dbtask-1234567890123-4567.log'));
```

8. Verifique o conteúdo do diretório que você está usando para armazenar a carteira Oracle.

```
SELECT * FROM TABLE(rdsadmin.rds_file_util.listdir(p_directory => 'WALLET_DIR'));
```

   Para obter mais informações, consulte Listagem de arquivos no diretório de uma instância de banco de dados.

Etapa 4: conceder permissões de usuário para a carteira Oracle

Você pode criar um novo usuário do banco de dados ou configurar um usuário existente. Em ambos os casos, você deve configurar o usuário para acessar a carteira Oracle para conexões seguras e autenticação do cliente usando certificados.

Para conceder permissões do usuário para a carteira Oracle

1. Faça login na instância de banco de dados do RDS for Oracle como usuário primário.
2. Se não quiser configurar um usuário do banco de dados existente, crie um novo usuário. Caso contrário, vá para a próxima etapa.

   O exemplo a seguir cria um usuário do banco de dados chamado *my-user*.

```
CREATE USER my-user IDENTIFIED BY my-user-pwd;
GRANT CONNECT TO my-user;
```

3. Conceda permissão ao usuário do banco de dados no diretório que contém sua carteira Oracle.

   O exemplo a seguir concede acesso de leitura ao usuário *my-user* no diretório *WALLET_DIR*.

```
GRANT READ ON DIRECTORY WALLET_DIR TO my-user;
```

4. Conceda permissão ao usuário do banco de dados para usar o pacote UTL_HTTP.

   O programa PL/SQL a seguir concede acesso UTL_HTTP ao usuário *my-user*.

```
BEGIN
  rdsadmin.rdsadmin_util.grant_sys_object('UTL_HTTP', UPPER('my-user'));
  END;
/
```

5. Conceda permissão ao usuário do banco de dados para usar o pacote UTL_FILE.

   O programa PL/SQL a seguir concede acesso UTL_FILE ao usuário *my-user*.

```
BEGIN
  rdsadmin.rdsadmin_util.grant_sys_object('UTL_FILE', UPPER('my-user'));
  END;
/
```

Etapa 5: configurar o acesso a um site da sua instância de banco de dados

Nesta etapa, você configura o usuário do banco de dados Oracle para que ele possa se conectar ao site de destino usando o UTL_HTTP, a carteira Oracle carregada e o certificado do cliente. Para obter mais informações, consulte Configuring Access Control to an Oracle Wallet (Configurar o controle de acesso a uma carteira Oracle) na documentação do Oracle Database.

Para configurar o acesso a um site da instância de banco de dados do RDS for Oracle

1. Faça login na instância de banco de dados do RDS for Oracle como usuário primário.
2. Crie uma entrada de controle de acesso ao host (ACE) para o usuário e o site de destino em uma porta segura.

   O exemplo a seguir configura *my-user* para acessar o *secret.encrypted-website.com* na porta segura 443.

```
BEGIN
  DBMS_NETWORK_ACL_ADMIN.APPEND_HOST_ACE(
    host       => 'secret.encrypted-website.com',
```

```
    lower_port => 443,
    upper_port => 443,
    ace        => xs$ace_type(privilege_list => xs$name_list('http'),
                              principal_name => 'my-user',
                              principal_type => xs_acl.ptype_db));
                           -- If the program unit results in PLS-00201, set
                           -- the principal_type parameter to 2 as follows:
                           -- principal_type => 2));
END;
/
```

> **Important**
>
> A unidade do programa anterior pode ocasionar o seguinte erro: `PLS-00201: identifier 'XS_ACL' must be declared`. Se esse erro for exibido, substitua a linha que atribui um valor a `principal_type` com a seguinte linha e, depois, execute novamente a unidade do programa:
>
> ```
> principal_type => 2));
> ```
>
> Para ter mais informações sobre constantes no pacote PL/SQL `XS_ACL`, consulte o Real Application Security Administrator's and Developer's Guide na documentação do Oracle Database.

Para obter mais informações, consulte Configuring Access Control for External Network Services (Configuração do controle de acesso para serviços de rede externos) na documentação do Oracle Database.

3. (Opcional) Crie uma ACE para seu usuário e site de destino na porta padrão.

   Talvez seja necessário usar a porta padrão se algumas páginas da Web forem exibidas da porta padrão do servidor Web (80) em vez da porta segura (443).

```
BEGIN
  DBMS_NETWORK_ACL_ADMIN.APPEND_HOST_ACE(
    host       => 'secret.encrypted-website.com',
    lower_port => 80,
    upper_port => 80,
    ace        => xs$ace_type(privilege_list => xs$name_list('http'),
```

```
                                principal_name => 'my-user',
                                principal_type => xs_acl.ptype_db));
                              -- If the program unit results in PLS-00201, set
                              -- the principal_type parameter to 2 as follows:
                              -- principal_type => 2));
END;
/
```

4. Confirme se as entradas de controle de acesso existem.

```
SET LINESIZE 150
COLUMN HOST FORMAT A40
COLUMN ACL FORMAT A50

SELECT HOST, LOWER_PORT, UPPER_PORT, ACL
  FROM DBA_NETWORK_ACLS
ORDER BY HOST;
```

5. Conceda permissão ao usuário do banco de dados para usar o pacote UTL_HTTP.

   O programa PL/SQL a seguir concede acesso UTL_HTTP ao usuário *my-user*.

```
BEGIN
  rdsadmin.rdsadmin_util.grant_sys_object('UTL_HTTP', UPPER('my-user'));
  END;
/
```

6. Confirme se existem listas de controle de acesso relacionadas.

```
SET LINESIZE 150
COLUMN ACL FORMAT A50
COLUMN PRINCIPAL FORMAT A20
COLUMN PRIVILEGE FORMAT A10

SELECT ACL, PRINCIPAL, PRIVILEGE, IS_GRANT,
       TO_CHAR(START_DATE, 'DD-MON-YYYY') AS START_DATE,
       TO_CHAR(END_DATE, 'DD-MON-YYYY') AS END_DATE
  FROM DBA_NETWORK_ACL_PRIVILEGES
ORDER BY ACL, PRINCIPAL, PRIVILEGE;
```

7. Conceda permissão ao usuário do banco de dados para usar certificados para autenticação de cliente e sua carteira Oracle para conexões.

> **Note**
>
> Se não estiver usando certificados de cliente para autenticação, ignore esta etapa.

```
DECLARE
  l_wallet_path all_directories.directory_path%type;
BEGIN
  SELECT DIRECTORY_PATH
    INTO l_wallet_path
    FROM ALL_DIRECTORIES
   WHERE UPPER(DIRECTORY_NAME)='WALLET_DIR';
  DBMS_NETWORK_ACL_ADMIN.APPEND_WALLET_ACE(
    wallet_path => 'file:/' || l_wallet_path,
    ace         =>  xs$ace_type(privilege_list => xs
$name_list('use_client_certificates'),
                                principal_name => 'my-user',
                                principal_type => xs_acl.ptype_db));
END;
/
```

Etapa 6: testar conexões de sua instância de banco de dados para um site

Nesta etapa, você configura o usuário do banco de dados para que ele possa se conectar ao site usando o UTL_HTTP, a carteira Oracle carregada e o certificado do cliente.

Para configurar o acesso a um site da instância de banco de dados do RDS for Oracle

1. Faça login na instância de banco de dados do RDS for Oracle como usuário do banco de dados com permissões UTL_HTTP.
2. Confirme se uma conexão com o site de destino pode resolver o endereço do host.

   O exemplo a seguir obtém o endereço do host de *secret.encrypted-website.com*.

   ```
   SELECT UTL_INADDR.GET_HOST_ADDRESS(host => 'secret.encrypted-website.com')
     FROM DUAL;
   ```
3. Teste uma conexão com falha.

A consulta a seguir falha porque UTL_HTTP requer a localização da carteira Oracle com os certificados.

```
SELECT UTL_HTTP.REQUEST('secret.encrypted-website.com') FROM DUAL;
```

4. Teste o acesso ao site usando UTL_HTTP.SET_WALLET e selecionando do DUAL.

```
DECLARE
  l_wallet_path all_directories.directory_path%type;
BEGIN
  SELECT DIRECTORY_PATH
    INTO l_wallet_path
    FROM ALL_DIRECTORIES
   WHERE UPPER(DIRECTORY_NAME)='WALLET_DIR';
  UTL_HTTP.SET_WALLET('file:/' || l_wallet_path);
END;
/

SELECT UTL_HTTP.REQUEST('secret.encrypted-website.com') FROM DUAL;
```

5. (Opcional) Teste o acesso ao site armazenando sua consulta em uma variável e usando EXECUTE IMMEDIATE.

```
DECLARE
  l_wallet_path all_directories.directory_path%type;
  v_webpage_sql VARCHAR2(1000);
  v_results     VARCHAR2(32767);
BEGIN
  SELECT DIRECTORY_PATH
    INTO l_wallet_path
    FROM ALL_DIRECTORIES
   WHERE UPPER(DIRECTORY_NAME)='WALLET_DIR';
  v_webpage_sql := 'SELECT UTL_HTTP.REQUEST(''secret.encrypted-website.com'', ''''',
 ''file:/' ||l_wallet_path||''''') FROM DUAL';
  DBMS_OUTPUT.PUT_LINE(v_webpage_sql);
  EXECUTE IMMEDIATE v_webpage_sql INTO v_results;
  DBMS_OUTPUT.PUT_LINE(v_results);
END;
/
```

6. (Opcional) Encontre a localização do sistema de arquivos do diretório da carteira Oracle.

```
SELECT * FROM TABLE(rdsadmin.rds_file_util.listdir(p_directory => 'WALLET_DIR'));
```

Use a saída do comando anterior para fazer uma solicitação HTTP. Por exemplo, se o diretório for *rdsdbdata/userdirs/01*, execute a consulta a seguir.

```
SELECT UTL_HTTP.REQUEST('https://secret.encrypted-website.com/', '',
 'file://rdsdbdata/userdirs/01')
FROM   DUAL;
```

Trabalhar com CDBs no RDS para Oracle

Na arquitetura multilocatário do Oracle, um banco de dados de contêiner (CDB) pode incluir bancos de dados conectáveis (PDBs) criados pelo cliente. Para obter mais informações sobre CDBs, consulte Introduction to the Multitenant Architecture na documentação do Oracle Database.

Tópicos

- Visão geral dos CDBs RDS para Oracle
- Configurar um CDB RDS para Oracle
- Fazer backup e restaurar um CDB
- Converter um não CDB RDS para Oracle em um CDB
- Converter a configuração de locatário único em multilocatário
- Adicionar um banco de dados de locatários do RDS for Oracle à sua instância de CDB
- Modificando um banco de dados de locatários do RDS para Oracle
- Excluindo um banco de dados de locatários do RDS for Oracle do seu CDB
- Visualizando detalhes do banco de dados do inquilino
- Fazer upgrade de um CDB

Visão geral dos CDBs RDS para Oracle

Você pode criar uma instância de banco de dados do RDS para Oracle como um banco de dados de contêiner (CDB) ao executar o Oracle Database 19c ou posterior. A partir do Oracle Database 21c, todos os bancos de dados são CDBs. Um CDB é diferente de um não CDB porque pode conter bancos de dados conectáveis (PDBs), que são chamados de bancos de dados de locatário no RDS para Oracle. Um PDB é uma coleção portátil de esquemas e objetos que aparece para uma aplicação como um banco de dados separado.

Você vai criar o banco de dados de locatários (PDB) inicial ao criar a instância de CDB. No RDS para Oracle, a aplicação cliente interage com um PDB e não com um CDB. Sua experiência com um PDB é majoritariamente idêntica à experiência com um não CDB.

Tópicos

- Configuração de multilocatário da arquitetura CDB
- Configuração de locatário único da arquitetura CDB
- Opções de criação e conversão em CDBs

- Contas de usuário e privilégios em um CDB
- Famílias de grupos de parâmetros em um CDB
- Limitações do RDS for CDBs Oracle

Configuração de multilocatário da arquitetura CDB

O RDS para Oracle é compatível com a configuração de multilocatário da arquitetura multilocatária da Oracle, também chamada de arquitetura CDB. Nessa configuração, a instância de CDB do RDS para Oracle pode conter bancos de dados de um a trinta locatários, dependendo da edição do banco de dados e de quaisquer licenças de opção necessárias. No banco de dados Oracle, um banco de dados de locatários é um PDB. Sua instância de banco de dados deve usar a versão 19.0.0.0.ru-2022-01.rur-2022.r1 ou posterior.

> **Note**
>
> A configuração do Amazon RDS é chamada de “multi-tenant”, em vez de “multitenant”, porque é um recurso da plataforma RDS, não apenas do mecanismo de banco de dados Oracle. Da mesma forma, o termo “locatário” do RDS refere-se a qualquer locatário em uma configuração do RDS, não apenas aos PDBs da Oracle. Na documentação do RDS, o termo “multilocatário da Oracle” (Oracle multitenant), sem hífen, refere-se exclusivamente à arquitetura CDB de banco de dados Oracle, que é compatível com implantações on-premises e do RDS.

Agora, é possível definir as seguintes configurações:

- Nome do banco de dados do locatário
- Nome de usuário mestre do banco de dados locatário
- Senha principal do banco de dados de locatário (integrada opcionalmente ao Secrets Manager)
- Conjunto de caracteres de banco de dados locatário
- Conjunto de caracteres nacionais do banco de dados do inquilino

O conjunto de caracteres do banco de dados locatário pode ser diferente do conjunto de caracteres do CDB. O mesmo se aplica ao conjunto de caracteres nacionais. Depois de criar seu banco de dados locatário inicial, você pode criar, modificar ou excluir bancos de dados locatário usando as APIs do RDS. O nome do CDB é padronizado como `RDSCDB` e não pode ser alterado. Para obter

mais informações, consulte Configurações para instâncias de banco de dados e Modificando um banco de dados de locatários do RDS para Oracle.

Configuração de locatário único da arquitetura CDB

O RDS para Oracle é compatível com a configuração legada da arquitetura multilocatário da Oracle chamada configuração de locatário único. Nessa configuração, uma instância CDB RDS para Oracle pode conter apenas um locatário (PDB). Não será possível criar mais PDBs posteriormente.

Opções de criação e conversão em CDBs

O Oracle Database 21c só é compatível com CDBs, enquanto o Oracle Database 19c é compatível com CDBs e não CDBs. Todas as instâncias do CDB RDS para Oracle são compatíveis com as configurações de multilocatário e locatário único.

Opções de criação, conversão e atualização para a arquitetura de banco de dados Oracle

A tabela a seguir mostra as diferentes opções de arquitetura para criação e atualização de bancos de dados RDS para Oracle.

Versão	Opções de criação de banco de dados	Opções de conversão de arquitetura	Destinos de upgrade da versão principal
Oracle Database 21c	Somente arquitetura CDB	N/D	N/D
Oracle Database 19c	Arquitetura CDB ou não CDB	Arquitetura não CDB para CDB (RU de abril de 2021 ou posterior)	CDB do Oracle Database 21c

Conforme mostrado na tabela anterior, não é possível atualizar de um não CDB para um CDB diretamente em uma nova versão principal. Mas você pode converter um não CDB Oracle Database 19c em um CDB Oracle Database 19c, depois fazer upgrade do CDB Oracle Database 19c para CDB Oracle Database 21c. Para ter mais informações, consulte Converter um não CDB RDS para Oracle em um CDB.

Opções de conversão para configurações de arquitetura CDB

A tabela a seguir mostra as diferentes opções para converter a configuração de arquitetura de uma instância de banco de dados RDS para Oracle.

Arquitetura e configuração atuais	Conversão para a configuração de locatário único da arquitetura CDB	Conversão para a configuração de multilocatário da arquitetura CDB	Conversão para a arquitetura não CDB
Não CDB	Compatível	Compatível*	N/D
CDB usando a configuração de locatário único	N/D	Compatível	Sem compatibilidade
CDB usando a configuração multilocatário	Sem compatibilidade	N/D	Sem compatibilidade

* Não é possível converter um não CDB na configuração de multilocatário em uma única operação. Quando você converte um não CDB em CDB, o CDB está na configuração de locatário único. Você pode converter a configuração de locatário único na configuração de multilocatário em uma operação separada.

Contas de usuário e privilégios em um CDB

Na arquitetura multilocatário do Oracle, todas as contas de usuário são usuários comuns ou usuários locais. Um usuário comum do CDB é um usuário de banco de dados cuja única identidade e senha são conhecidas na raiz CDB e em todos os PDB existentes e futuros. Em contraste, um usuário local existe somente em um único PDB.

O usuário principal do RDS é uma conta de usuário local no PDB, que você nomeia ao criar sua instância de banco de dados. Se você criar novas contas de usuário, esses usuários também serão usuários locais residentes no PDB. Você não pode usar nenhuma conta de usuário para criar novos PDBs ou modificar o estado do PDB existente.

O usuário `rdsadmin` é uma conta de usuário comum. Você pode executar pacotes do RDS para Oracle que existem nesta conta, mas não pode fazer login como `rdsadmin`. Para obter mais informações, consulte Sobre usuários comuns e usuários locais na documentação da Oracle.

Para usuários principais nas configurações multilocatário e de locatário único, você pode usar credenciais que são autogerenciadas ou gerenciadas pelo AWS Secrets Manager. Na configuração

de locatário único, você usa comandos da CLI em nível de instância, como `create-db-instance` para senhas principais gerenciadas. Na configuração multilocatário, você usa comandos do banco de dados de locatário, como `create-tenant-database` para senhas principais gerenciadas. Para ter mais informações sobre a integração do Secrets Manager, consulte Gerenciar a senha do usuário principal para um banco de dados de locatário do RDS para Oracle com o Secrets Manager.

Famílias de grupos de parâmetros em um CDB

Os CDBs têm suas próprias famílias de grupos de parâmetros e valores de parâmetros padrão. As famílias de grupos de parâmetros de CDB são as seguintes:

- oracle-ee-cdb-21
- oracle-se2-cdb-21
- oracle-ee-cdb-19
- oracle-se2-cdb-19

Limitações do RDS for CDBs Oracle

O RDS para Oracle é compatível com um subconjunto de atributos disponíveis em um CDB on-premises.

Limitações do CDB

As limitações a seguir se aplicam aos RDS para Oracle em nível de CDB:

- Não é possível se conectar a um CDB. Você sempre se conecta ao banco de dados (PDB) locatário em vez do CDB. Especifique o endpoint para o PDB assim como para um não CDB. A única diferença é que você especifica pdb_name para o nome do banco de dados, em que pdb_name é o nome que você escolheu para seu PDB.
- Não é possível converter um CDB na configuração de multilocatário em um CDB na configuração de locatário único. A conversão na configuração de multilocatário é unidirecional e irreversível.
- Não será possível habilitar ou converter na configuração de multilocatário se sua instância de banco de dados usar uma versão de banco de dados Oracle menor que 19.0.0.0.ru-2022-01.rur-2022.r1.
- Não é possível usar um CDB do RDS para Oracle com o ORDS 22 e posterior. Uma alternativa é usar uma versão anterior do ORDS ou usar um Oracle Database 19c não CDB.

- Não é possível usar o Oracle Data Guard na configuração multilocatária, mas é possível usá-lo na configuração de locatário único.
- Não é possível usar o Database Activity Streams em um CDB.
- Não é possível habilitar a auditoria em `CDB$ROOT`. É necessário habilitar a auditoria em cada PDB individualmente.

Limitações do banco de dados (PDB) locatário

As seguintes limitações se aplicam aos bancos de dados locatários na configuração de multilocatário do RDS para Oracle:

- Não é possível transferir as operações do banco de dados locatário para a janela de manutenção. Todas as alterações ocorrem imediatamente.
- Não é possível adicionar um banco de dados locatário a um CDB que usa a configuração de locatário único.
- Não é possível adicionar ou modificar vários bancos de dados locatários em uma única operação. Só é possível adicioná-los ou modificá-los um de cada vez.
- Não é possível alterar o nome de um banco de dados locatário para `CDB$ROOT` ou `PDB$SEED`.
- Não será possível excluir um banco de dados locatário se ele for o único locatário no CDB.
- Nem todos os tipos de classe de instância de banco de dados têm recursos suficientes para oferecer suporte a vários PDBs em uma instância de CDB RDS para Oracle. Um aumento na contagem de PDBs afeta a performance e a estabilidade das classes de instâncias menores e aumenta o tempo da maioria das operações em nível de instância, por exemplo, atualizações de banco de dados.
- Não é possível usar várias Contas da AWS para criar PDBs no mesmo CDB. Os PDBs devem ser de propriedade da mesma conta da instância de banco de dados em que os PDBs estão hospedados.
- Todos os PDBs em um CDB usam o mesmo endpoint e o mesmo receptor de banco de dados.
- As seguintes operações não são compatíveis no nível do PDB, mas sim no nível do CDB:
  - Backup e recuperação
  - Atualizações do banco de dados
  - Ações de manutenção
- Os seguintes atributos não são compatíveis no nível do PDB, mas sim no nível do CDB:
  - Grupos de opções (as opções são instaladas em todos os PDBs em sua instância de CDB)

- Grupos de parâmetros (todos os parâmetros são derivados do grupo de parâmetros associado à sua instância de CDB)

- As operações em nível de PDB que são compatíveis com a arquitetura de CDB on-premises, mas não são compatíveis com CDB RDS para Oracle, incluem o seguinte:

> **Note**
>
> A lista a seguir não é completa.

  - PDBs de aplicação
  - PDBs de proxy
  - Iniciar e encerrar um PDB
  - Desconectar e conectar PDBs

    Para inserir ou remover dados de um CDB, use as mesmas técnicas de um não CDB. Para obter mais informações sobre a migração de dados, consulte Importar dados para o Oracle no Amazon RDS.
  - Opções de configuração no nível do PDB

    O PDB herda as configurações de opções do grupo de opções do CDB. Para obter mais informações sobre as opções de configuração, consulte Grupos de parâmetros para Amazon RDS. Para ver as práticas recomendadas, consulte Trabalhar com grupos de parâmetros de banco de dados.
  - Configurar parâmetros em um PDB

    O PDB herda as configurações de parâmetros do CDB. Para obter mais informações sobre as opções de configuração, consulte Adição de opções a instâncias de banco de dados Oracle.
  - Configurar diferentes receptores para PDBs no mesmo CDB
  - Atributos do Oracle Flashback

Configurar um CDB RDS para Oracle

A configuração de um CDB é semelhante à configuração de um não CDB.

Tópicos

- Criar uma instância de CDB do RDS para Oracle
- Conectar-se a um PDB em seu CDB RDS para Oracle

Criar uma instância de CDB do RDS para Oracle

No RDS para Oracle, a criação de uma instância de CDB é quase idêntica à criação de uma instância não CDB. A diferença é que você escolhe a arquitetura multilocatário Oracle ao criar a instância de banco de dados e também escolhe a configuração de arquitetura: multilocatário ou locatário único. Se você criar tags ao criar um CDB na configuração de vários locatários, o RDS propagará as tags para o banco de dados inicial do inquilino. Para criar um CDB, use o AWS Management Console, a AWS CLI ou a API do RDS.

Console

Como criar uma instância de CDB

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No canto superior direito do console do Amazon RDS, escolha a Região da AWS na qual você deseja criar a instância de CDB.
3. No painel de navegação, escolha Databases (Bancos de dados).
4. Escolha Create database (Criar banco de dados).
5. Em Choose a database creation method (Selecionar um método de criação de banco de dados), selecione Standard Create (Criação padrão).
6. Em Engine options (Opções de mecanismo), escolha Oracle.
7. Em Tipo de gerenciamento de banco de dados, escolha Amazon RDS.
8. Em Configurações de arquitetura, escolha Arquitetura multilocatário.
9. Para a configuração da arquitetura, faça o seguinte:
   - Escolha a configuração multilocatário e avance para a próxima etapa.
   - Escolha a configuração de inquilino único e vá para a Etapa 11.
10. (Configuração multilocatária) Para as configurações do banco de dados do locatário, faça as seguintes alterações:
    - Em Nome do banco de dados inicial, insira o nome do PDB. O nome do PDB deve ser diferente do nome do CDB, cujo padrão é. `RDSCDB`

- Em Nome de usuário principal do banco de dados do locatário, insira o nome de usuário principal do seu PDB. Você não pode usar o nome de usuário principal para fazer login na raiz do CDB.
- Em Gerenciamento de credenciais, escolha uma das seguintes opções de gerenciamento de credenciais:
  - Gerenciado no AWS Secrets Manager

    A senha gerenciada é para o banco de dados de locatário inicial e não para a instância. Em Selecionar a chave de criptografia, escolha uma chave do KMS que o Secrets Manager cria ou uma chave que você criou.

    > **Note**
    >
    > Recomendamos o AWS Secrets Manager como a técnica mais segura para gerenciar credenciais. Aplicam-se cobranças adicionais. Para ter mais informações, consulte Gerenciamento de senhas com Amazon RDS e AWS Secrets Manager.

  - Autogerenciado

    Para especificar uma senha, desmarque a caixa de seleção Gerar uma senha automaticamente se ela estiver marcada. Insira a mesma senha em Senha primária e Confirmar senha.
- Para o conjunto de caracteres do banco de dados do locatário, escolha um conjunto de caracteres para o PDB. Você pode escolher um conjunto de caracteres do banco de dados do inquilino que seja diferente do conjunto de caracteres do CDB.

  O conjunto de caracteres PDB padrão é AL32UTF8. Se você escolher um conjunto de caracteres PDB não padrão, a criação do CDB poderá ser mais lenta.

> **Note**
>
> Não é possível especificar vários bancos de dados de locatário na operação de criação. O CDB tem um PDB quando é criado. Você pode adicionar PDBs a um CDB existente em uma operação separada.

11. (Configuração de locatário único) Escolha as configurações desejadas com base nas opções listadas em Configurações para instâncias de banco de dados:

- Na seção Settings (Configurações), abra Credential Settings (Configurações de credencial). Faça o seguinte:

  i. Em Nome do usuário principal, insira o nome de um usuário local em seu PDB. Você não pode usar o nome de usuário principal para fazer login na raiz do CDB.

  ii. Em Gerenciamento de credenciais, escolha uma das seguintes opções de gerenciamento de credenciais:

  - Gerenciado no AWS Secrets Manager

    Em Selecionar a chave de criptografia, escolha uma chave do KMS que o Secrets Manager cria ou uma chave que você criou.

    > **Note**
    >
    > Recomendamos o AWS Secrets Manager como a técnica mais segura para gerenciar credenciais. Aplicam-se cobranças adicionais. Para ter mais informações, consulte Gerenciamento de senhas com Amazon RDS e AWS Secrets Manager.

  - Autogerenciado

    Para especificar uma senha, desmarque a caixa de seleção Gerar uma senha automaticamente se ela estiver marcada. Insira a mesma senha em Senha primária e Confirmar senha.

12. Nas seções restantes, especifique suas configurações de instância de banco de dados. Para ter informações sobre cada configuração, consulte Configurações para instâncias de banco de dados.
13. Selecione Criar banco de dados.

AWS CLI

Para criar um CDB na configuração multilocatário, use o comando create-db-instance com os seguintes parâmetros:

- `--db-instance-identifier`
- `--db-instance-class`

- `--engine { oracle-ee-cdb | oracle-se2-cdb }`
- `--master-username`
- `--master-user-password` ou `--manage-master-user-password`
- `--multi-tenant`(para a configuração de inquilino único, não especifique `multi-tenant` ou especifique) `--no-multi-tenant`
- `--allocated-storage`
- `--backup-retention-period`

Para obter informações sobre cada configuração, consulte Configurações para instâncias de banco de dados.

O exemplo a seguir cria uma instância de banco de dados do RDS para Oracle chamada *my-cdb-inst*. Se você especificar `--no-multi-tenant` ou não especificar`--multi-tenant`, a configuração padrão do CDB será de inquilino único. O mecanismo é`oracle-ee-cdb`: um comando que especifica `oracle-ee` e `--multi-tenant` falha com um erro. O banco de dados inicial do inquilino é denominado *mypdb*.

Example

Para Linux, macOS ou Unix:

```
aws rds create-db-instance \
    --engine oracle-ee-cdb \
    --db-instance-identifier my-cdb-inst \
    --multi-tenant \
    --db-name mypdb \
    --allocated-storage 250 \
    --db-instance-class db.t3.large \
    --master-username pdb_admin \
    --manage-master-user-password \
    --backup-retention-period 3
```

Para Windows:

```
aws rds create-db-instance ^
    --engine oracle-ee-cdb ^
    --db-instance-identifier my-cdb-inst ^
    --multi-tenant ^
```

```
    --db-name mypdb ^
    --allocated-storage 250 ^
    --db-instance-class db.t3.large ^
    --master-username pdb_admin ^
    --manage-master-user-password \ ^
    --backup-retention-period 3
```

> **Note**
>
> Especifique uma senha diferente do prompt mostrado aqui como prática recomendada de segurança.

O comando gerará uma saída semelhante à seguinte: O nome do banco de dados, o conjunto de caracteres, o conjunto de caracteres nacionais, o usuário principal e o segredo do usuário principal não estão incluídos na saída. Você pode visualizar essas informações usando o comando CLI. `describe-tenant-databases`


```
{
    "DBInstance": {
        "DBInstanceIdentifier": "my-cdb-inst",
        "DBInstanceClass": "db.t3.large",
        "MultiTenant": true,
        "Engine": "oracle-ee-cdb",
        "DBResourceId": "db-ABCDEFGJIJKLMNOPQRSTUVWXYZ",
        "DBInstanceStatus": "creating",
        "AllocatedStorage": 250,
        "PreferredBackupWindow": "04:59-05:29",
        "BackupRetentionPeriod": 3,
        "DBSecurityGroups": [],
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sg-0a1bcd2e",
                "Status": "active"
            }
        ],
        "DBParameterGroups": [
            {
                "DBParameterGroupName": "default.oracle-ee-cdb-19",
                "ParameterApplyStatus": "in-sync"
            }
        ],
```

```
        "DBSubnetGroup": {
            "DBSubnetGroupName": "default",
            "DBSubnetGroupDescription": "default",
            "VpcId": "vpc-1234567a",
            "SubnetGroupStatus": "Complete",
            ...
```

API do RDS

Para criar uma instância de banco de dados utilizando a API do Amazon RDS, chame a operação CreateDBInstance com os seguintes parâmetros.

Para obter informações sobre cada configuração, consulte Configurações para instâncias de banco de dados.

Conectar-se a um PDB em seu CDB RDS para Oracle

Você pode usar um utilitário como o SQL*Plus para se conectar a um PDB. Para fazer o download do Oracle Instant Client, que inclui uma versão independente do SQL*Plus, consulte Downloads do Oracle Instant Client.

Para conectar o SQL*Plus ao PDB, são necessárias as seguintes informações:

- Nome do PDB
- Nome de usuário e senha do banco de dados
- Endpoint da sua instância de banco de dados
- Número da porta

Para obter informações sobre como encontrar as informações anteriores, consulte Encontrar o endpoint da instância de banco de dados do RDS para Oracle.

Example Como se conectar ao seu PDB usando o SQL*Plus

Nos exemplos a seguir, substitua o usuário principal por *master_user_name*. Além disso, substitua o endpoint da instância de banco de dados e inclua o número da porta e o Oracle SID. O valor do SID é o nome do PDB que você especificou ao criar a instância de banco de dados, não o identificador da instância de banco de dados.

Para Linux, macOS ou Unix:

```
sqlplus 'master_user_name@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=endpoint)
(PORT=port))(CONNECT_DATA=(SID=pdb_name)))'
```

Para Windows:

```
sqlplus master_user_name@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=endpoint)
(PORT=port))(CONNECT_DATA=(SID=pdb_name)))
```

Você deve ver saída semelhante ao seguinte:

```
SQL*Plus: Release 19.0.0.0.0 Production on Mon Aug 21 09:42:20 2021
```

Depois que você inserir a senha de usuário, o prompt do SQL será exibido.

```
SQL>
```

> **Note**
>
> A string de conexão de formato mais curto (Easy connect ou EZCONNECT), como `sqlplus username/password@LONGER-THAN-63-CHARS-RDS-ENDPOINT-HERE:1521/database-identifier`, pode encontrar um limite máximo de caracteres e não deve ser usada para conexão.

Fazer backup e restaurar um CDB

Você pode fazer backup e restaurar seu CDB usando os snapshots de banco de dados do RDS ou o Recovery Manager (RMAN).

Fazer backup e restaurar um CDB usando snapshots de bancos de dados

Os snapshots de banco de dados funcionam de forma semelhante nas arquiteturas CDB e não CDB. As principais diferenças são as seguintes.

- Ao restaurar um snapshot de banco de dados de um CDB, você não pode renomear o CDB. O nome do CDB é `RDSCDB` e não pode ser alterado.
- Ao restaurar um snapshot de banco de dados de um CDB, você não pode renomear o PDB. Você pode alterar o nome do PDB usando o comando modify-tenant-database.

- Para encontrar bancos de dados locatários em um snapshot, utilize o comando da CLI describe-db-snapshot-tenant-databases.
- Você não pode interagir diretamente com os bancos de dados locatários em um snapshot de CDB que usa a configuração da arquitetura de multilocatário. Se você restaurar o snapshot do banco de dados, restaurará todos os bancos de dados locatários.
- O RDS para Oracle copia implicitamente as tags de um banco de dados locatário no banco de dados locatário de um snapshot de banco de dados. Quando você restaura um banco de dados locatário, as tags aparecem no banco de dados restaurado.
- Se você restaurar um snapshot de banco de dados e especificar novas tags usando o parâmetro `--tags`, as novas tags substituirão todas as tags existentes.
- Se você tirar um DB snapshot de uma instância de CDB que tem tags e especificar`--copy-tags-to-snapshot`, o RDS para Oracle copia as tags dos bancos de dados locatário nos bancos de dados locatário no snapshot.

Para obter mais informações, consulte Considerações sobre o Oracle Database.

Fazer backup e restaurar um CDB usando o RMAN

Para saber como fazer backup e restaurar um CDB ou banco de dados locatário individual usando o RMAN, consulte. Realização de tarefas comuns do RMAN para instâncias de banco de dados Oracle

Converter um não CDB RDS para Oracle em um CDB

É possível alterar a arquitetura de um banco de dados Oracle da arquitetura tradicional não CDB para a arquitetura multilocatária, também chamada de arquitetura CDB, com o comando `modify-db-instance`. Na maioria dos casos, essa técnica é preferível à criação de um CDB e à importação de dados. A operação de conversão gera tempo de inatividade.

Ao fazer upgrade da versão do mecanismo de banco de dados, você não pode alterar a arquitetura do banco de dados na mesma operação. Portanto, para fazer upgrade de um não CDB Oracle Database 19c para um CDB Oracle Database 21c, primeiro é necessário converter o não CDB em CDB, depois fazer upgrade do CDB 19c para CDB 21c.

A operação de conversão de não CDB apresenta os seguintes requisitos:

- Você deve especificar `oracle-ee-cdb` ou `oracle-se2-cdb` para o tipo de mecanismo de banco de dados. Esses são os únicos valores compatíveis.

- Seu mecanismo de banco de dados deve usar o Oracle Database 19c com uma atualização de versão (RU) de abril de 2021 ou posterior.

A operação tem as seguintes limitações:

- Não é possível converter um CDB em um não CDB. Só é possível converter um não CDB em CDB.
- Não é possível converter um não CDB na configuração de locatário em uma única chamada. `modify-db-instance` Depois de converter um não CDB em CDB, seu CDB estará na configuração de locatário único. Para converter a configuração de inquilino único em configuração de vários locatários, execute novamente. `modify-db-instance` Para ter mais informações, consulte Converter a configuração de locatário único em multilocatário.
- Não é possível converter um banco de dados primário ou de réplica que tenha o Oracle Data Guard habilitado. Para converter um não-CDB que tenha réplicas de leitura, primeiro exclua todas as réplicas de leitura.
- Não é possível fazer upgrade da versão do mecanismo de banco de dados e converter um não CDB em CDB na mesma operação.

Antes de converter a não CDB, pense no seguinte:

- As considerações para grupos de opções e parâmetros são iguais àquelas do upgrade do mecanismo de banco de dados. Para ter mais informações, consulte Considerações sobre atualizações de bancos de dados Oracle.
- Você pode converter instâncias não CDB existentes que usam senhas principais gerenciadas em instâncias de locatário único em uma única operação. As instâncias de locatário único herdam as senhas gerenciadas.
- Se sua instância de banco de dados tiver a opção `OEMAGENT` instalada, uma prática recomendada é remover essa opção antes de converter a não CDB. Depois que a não CDB for convertida em CDB, reinstale a opção. Para ter mais informações, consulte Oracle Management Agent para Enterprise Manager Cloud Control.
- Durante o processo de conversão, o RDS redefine o tamanho do log redo online para o padrão de 128 M.

Console

Como converter um não CDB em CDB

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No canto superior direito do console do Amazon RDS, escolha a Região da AWS em que sua instância de banco de dados se encontra.
3. No painel de navegação, escolha Bancos de dados e selecione a instância de não CDB que deseja converter em instância de CDB.
4. Escolha Modificar.
5. Em Configurações de arquitetura, selecione Arquitetura multilocatário da Oracle. Após a conversão, seu CDB estará na configuração de locatário único.
6. (Opcional) Em Grupo de parâmetros de banco de dados, escolha um novo grupo de parâmetros para a instância de CDB. As mesmas considerações de grupos de parâmetros se aplicam tanto ao converter uma instância de banco de dados como ao fazer upgrade de uma instância de banco de dados. Para ter mais informações, consulte Considerações de grupos de parâmetros.
7. (Opcional) Em Grupo de opções, escolha um novo grupo de opções para sua instância de CDB. As mesmas considerações de grupos de opções se aplicam tanto ao converter uma instância de banco de dados como ao fazer upgrade de uma instância de banco de dados. Para ter mais informações, consulte Considerações de grupos de opções.
8. (Opcional) Em Gerenciamento de credenciais, escolha Gerenciado no AWS Secrets Manager ou Autogerenciado Para ter mais informações, consulte Gerenciar a senha do usuário principal para uma instância de banco de dados com o Secrets Manager.
9. Quando todas as alterações estiverem conforme o desejado, escolha Continue (Continuar) e verifique o resumo das modificações.
10. (Opcional) Escolha Apply immediately (Aplicar imediatamente) para aplicar as alterações imediatamente. Escolher essa opção pode causar uma inatividade em alguns casos. Para obter mais informações, consulte Usar a configuração de programação de modificações.
11. Na página de confirmação, revise suas alterações. Se estiverem corretas, escolha Modificar instância de banco de dados.

    Ou escolha Back (Voltar) para editar as alterações ou Cancel (Cancelar) para cancelar as alterações.

AWS CLI

Para converter o não CDB de sua instância de banco de dados em CDB, defina `--engine` como `oracle-ee-cdb` ou `oracle-se2-cdb` no comando modify-db-instance da AWS CLI. Para ter mais informações, consulte Configurações para instâncias de banco de dados.

O exemplo a seguir converte a instância de banco de dados chamada *my-non-cdb* e especifica um grupo de opções e um grupo de parâmetros personalizados. O comando também habilita o gerenciamento de senhas com o Secrets Manager.

Example

Para Linux, macOS ou Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier my-non-cdb \
    --engine oracle-ee-cdb \
    --option-group-name custom-option-group \
    --db-parameter-group-name custom-parameter-group \
    --manage-master-user-password
```

Para Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier my-non-cdb ^
    --engine oracle-ee-cdb ^
    --option-group-name custom-option-group ^
    --db-parameter-group-name custom-parameter-group ^
    --manage-master-user-password
```

API do RDS

Para converter um não CDB em CDB, especifique `Engine` na operação de API ModifyDBInstance do RDS.

Converter a configuração de locatário único em multilocatário

Você pode modificar a arquitetura de um CDB RDS para Oracle da configuração de locatário único para a configuração de multilocatário. Antes e depois da conversão, o CDB conterá um banco de dados de locatário único (PDB). As tags da instância de banco de dados se propagam para o banco de dados de locatário inicial criado durante a conversão.

Antes de começar, verifique se sua política do IAM tem permissão para criar um banco de dados de locatário. Durante a conversão, o RDS para Oracle migra os seguintes metadados para o novo banco de dados de locatários:

- Nome do usuário principal
- A senha principal gerenciada (se o CDB de origem se integrar ao Secrets Manager)
- Nome do banco de dados
- Conjunto de caracteres
- Conjunto de caracteres nacionais

Antes da conversão, você visualiza as informações anteriores usando o comando `describe-db-instances`. Após da conversão, você pode visualizar as informações usando o comando `describe-tenant-database`.

A conversão de locatário único em multilocatário tem as seguintes limitações:

- Portanto, não é possível voltar para a configuração de locatário único depois que você muda para a configuração multilocatário. A conversão é irreversível.
- Não é possível converter um banco de dados primário ou de réplica que tenha o Oracle Data Guard habilitado.
- Não é possível atualizar a versão do mecanismo de banco de dados e converter em uma configuração de multilocatário na mesma operação.
- Durante a conversão, não é possível habilitar ou desabilitar senhas de usuário principal gerenciadas.

Console

Para converter um CDB usando a configuração de locatário único na configuração de vários locatários

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No canto superior direito do console do Amazon RDS, escolha a Região da AWS em que sua instância de banco de dados se encontra.
3. No painel de navegação, escolha Bancos de dados e selecione a instância de não CDB que deseja converter em instância de CDB.

4. Escolha Modificar.
5. Em Configurações de arquitetura, selecione Arquitetura multilocatário da Oracle.
6. Em Configuração de arquitetura, selecione Configuração multilocatária.
7. (Opcional) Em Grupo de parâmetros de banco de dados, escolha um novo grupo de parâmetros para a instância de CDB. As mesmas considerações de grupos de parâmetros se aplicam tanto ao converter uma instância de banco de dados como ao fazer upgrade de uma instância de banco de dados.
8. (Opcional) Em Grupo de opções, escolha um novo grupo de opções para sua instância de CDB. As mesmas considerações de grupos de opções se aplicam tanto ao converter uma instância de banco de dados como ao fazer upgrade de uma instância de banco de dados.
9. Quando todas as alterações estiverem conforme o desejado, escolha Continue (Continuar) e verifique o resumo das modificações.
10. Escolha Apply immediately (Aplicar imediatamente). Essa opção é necessária quando você alterna para uma configuração de multilocatário. Essa opção pode causar inatividade em alguns casos.
11. Na página de confirmação, revise suas alterações. Se estiverem corretas, escolha Modificar instância de banco de dados.

    Ou escolha Back (Voltar) para editar as alterações ou Cancel (Cancelar) para cancelar as alterações.

AWS CLI

Para converter um CDB usando a configuração de locatário único na configuração de multilocatário, especifique `--multi-tenant` no comando AWS CLI modify-db-instance.

O exemplo a seguir converte a instância de banco de dados chamada `my-st-cdb` da configuração de inquilino único para a configuração de multilocatário. A opção `--apply-immediately` é obrigatória.

Example

Para Linux, macOS ou Unix:

```
aws rds modify-db-instance --region us-east-1\
    --db-instance-identifier my-st-cdb \
    --multi-tenant \
```

```
    --apply-immediately
```

Para Windows:

```
aws rds modify-db-instance --region us-east-1 ^
    --db-instance-identifier my-st-cdb ^
    --multi-tenant ^
    --apply-immediately
```

A saída é semelhante ao apontado abaixo.

```
{
    "DBInstance": {
        "DBInstanceIdentifier": "my-st-cdb",
        "DBInstanceClass": "db.r5.large",
        "MultiTenant": false,
        "Engine": "oracle-ee-cdb",
        "DBResourceId": "db-AB1CDE2FGHIJK34LMNOPRLXTXU",
        "DBInstanceStatus": "modifying",
        "MasterUsername": "admin",
        "DBName": "ORCL",
        ...
        "EngineVersion": "19.0.0.0.ru-2022-01.rur-2022-01.r1",
        "AutoMinorVersionUpgrade": true,
        "ReadReplicaDBInstanceIdentifiers": [],
        "LicenseModel": "bring-your-own-license",
        "OptionGroupMemberships": [
            {
                "OptionGroupName": "default:oracle-ee-cdb-19",
                "Status": "in-sync"
            }
        ],
        ...
        "PendingModifiedValues": {
            "MultiTenant": "true"
        }
    }
}
```

Adicionar um banco de dados de locatários do RDS for Oracle à sua instância de CDB

Na configuração multilocatária do RDS for Oracle, um banco de dados de inquilinos é um PDB. Para adicionar um banco de dados de inquilinos, certifique-se de que os seguintes pré-requisitos sejam atendidos:

- Seu CDB tem a configuração multilocatária ativada. Para ter mais informações, consulte Configuração de multilocatário da arquitetura CDB.
- Você tem as permissões necessárias do IAM para criar o banco de dados do inquilino.

Você pode adicionar um banco de dados de inquilinos usando a AWS Management ConsoleAWS CLI, a ou a API do RDS. Você não pode adicionar vários bancos de dados de inquilinos em uma única operação: você deve adicioná-los um por vez. Se o CDB tiver a retenção de backup ativada, o Amazon RDS fará backup da instância de banco de dados antes e depois de adicionar um novo banco de dados de inquilinos.

Console

Para adicionar um banco de dados de inquilinos à sua instância de banco de dados

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No canto superior direito do console do Amazon RDS, selecione a Região da AWS na qual você deseja criar o cluster de banco de dados.
3. No painel de navegação, escolha Bancos de dados.
4. Escolha a instância do CDB à qual você deseja adicionar um banco de dados de locatários. Sua instância de banco de dados deve usar a configuração multilocatária da arquitetura CDB.
5. Escolha Ações e, em seguida, Adicionar banco de dados de inquilinos.
6. Para Global database settings (Configurações de banco de dados globais), faça o seguinte:
    - Em Nome do banco de dados inicial, insira o nome do PDB.
    - Para o nome de usuário mestre do banco de dados do locatário, insira o nome do usuário mestre do seu PDB.
    - Escolha uma das seguintes opções de gerenciamento de credenciais:
        - Gerenciado no AWS Secrets Manager

Em Selecionar a chave de criptografia, escolha uma chave do KMS que o Secrets Manager cria ou uma chave que você criou.

> **Note**
>
> Recomendamos o AWS Secrets Manager como a técnica mais segura para gerenciar credenciais. Aplicam-se cobranças adicionais. Para ter mais informações, consulte Gerenciamento de senhas com Amazon RDS e AWS Secrets Manager.

- Autogerenciado

  Para especificar uma senha, desmarque a caixa de seleção Gerar uma senha automaticamente se ela estiver marcada. Insira a mesma senha em Senha primária e Confirmar senha.

- Em Configuração adicional, insira o nome do seu PDB em Nome do banco de dados inicial. Você não pode nomear o CDB, que tem o nome padrão `RDSCDB`.

- Para o conjunto de caracteres do banco de dados do locatário, escolha um conjunto de caracteres para o PDB. O padrão é AL32UTF8. Você pode escolher um conjunto de caracteres PDB diferente do conjunto de caracteres CDB.
- Para o conjunto de caracteres nacionais do banco de dados do inquilino, escolha um conjunto de caracteres nacional para o PDB. O padrão é AL32UTF8. O conjunto de caracteres nacionais especifica a codificação somente para colunas que usam o tipo de `NCHAR` dados (`NCHAR`,`NVARCHAR2`, e`NCLOB`) e não afeta os metadados do banco de dados.

Para obter mais informações sobre os arquivos de configurações, consulte Configurações para instâncias de banco de dados.

7. Escolha Adicionar inquilino.

AWS CLI

Para adicionar um banco de dados de inquilinos ao seu CDB com oAWS CLI, use o comando create-tenant-database com os seguintes parâmetros necessários:

- `--db-instance-identifier`
- `--tenant-db-name`

- `--master-username`
- `--master-user-password`

O exemplo a seguir cria um banco de dados de inquilinos chamado mypdb2 na instância RDS para Oracle CDB chamada my-cdb-inst. O conjunto de caracteres do PDB éUTF-16.

Example

Para Linux, macOS ou Unix:

```
aws rds create-tenant-database --region us-east-1 \
    --db-instance-identifier my-cdb-inst \
    --tenant-db-name mypdb2 \
    --master-username mypdb2-admin \
    --master-user-password mypdb2-pwd \
    --character-set-name UTF-16
```

Para Windows:

```
aws rds create-tenant-database --region us-east-1 \
    --db-instance-identifier my-cdb-inst ^
    --tenant-db-name mypdb2 ^
    --master-username mypdb2-admin ^
    --master-user-password mypdb2-pwd ^
    --character-set-name UTF-16
```

A saída será semelhante à seguinte.

```
...}
    "TenantDatabase" :
         {
            "DbiResourceId" : "db-abc123",
            "TenantDatabaseResourceId" : "tdb-bac567",
            "TenantDatabaseArn" : "arn:aws:rds:us-east-1:123456789012:db:my-cdb-
inst:mypdb2",
            "DBInstanceIdentifier" : "my-cdb-inst",
            "TenantDBName" : "mypdb2",
            "Status" : "creating",
            "MasterUsername" : "mypdb2",
```

```
            "CharacterSetName" : "UTF-16",
            ...
        }
}...
```

Modificando um banco de dados de locatários do RDS para Oracle

Você pode modificar somente o nome do PDB e a senha do usuário mestre de um banco de dados de inquilinos no seu CDB. Anote os seguintes requisitos e limitações para o :

- Para modificar as configurações de um banco de dados de inquilinos em sua instância de banco de dados, o banco de dados de inquilinos deve existir.
- Você não pode modificar vários bancos de dados de inquilinos em uma única operação. É possível modificar apenas um banco de dados de inquilinos de cada vez.
- Você não pode alterar o nome de um banco de dados de inquilinos para `CDB$ROOT` ou`PDB$SEED`.

Você pode modificar um cluster de banco de dados usando o AWS Management Console, a AWS CLI ou a API do RDS.

Console

Para modificar o nome do PDB ou a senha mestra de um banco de dados de inquilinos

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No canto superior direito do console do Amazon RDS, selecione a Região da AWS na qual você deseja criar o cluster de banco de dados.
3. No painel de navegação, escolha Databases (Bancos de dados).
4. Escolha o banco de dados do inquilino cujo nome do banco de dados ou senha do usuário mestre você deseja modificar.
5. Escolha Modificar.
6. Para Global database settings (Configurações de banco de dados globais), faça o seguinte:
   - Em Nome do banco de dados inicial, insira o nome do PDB.
   - Para Senha mestra do banco de dados do locatário, insira uma nova senha.
7. Escolha Modificar inquilino.

AWS CLI

Para modificar um banco de dados de inquilinos usando oAWS CLI, chame o comando modify-tenant-database com os seguintes parâmetros:

- `--db-instance-identifier` *value*
- `--tenant-db-name` *value*
- `[--new-tenant-db-name` *value*`]`
- `[--master-user-password` *value*`]`

O exemplo a seguir renomeia o banco de dados do inquilino `pdb1` para uma instância de banco `pdb-hr` de dados. `my-cdb-inst`

Example

Para Linux, macOS ou Unix:

```
aws rds modify-tenant-database --region us-east-1 \
    --db-instance-identifier my-cdb-inst \
    --tenant-db-name pdb1 \
    --new-tenant-db-name pdb-hr
```

Para Windows:

```
aws rds modify-tenant-database --region us-east-1 ^
    --db-instance-identifier my-cdb-inst ^
    --tenant-db-name pdb1 ^
    --new-tenant-db-name pdb-hr
```

O comando gerará uma saída semelhante à seguinte:

```
{
    "TenantDatabase" : {
        "DbiResourceId" : "db-abc123",
        "TenantDatabaseResourceId" : "tdb-bac567",
        "TenantDatabaseArn" : "arn:aws:rds:us-east-1:123456789012:db:my-cdb-inst:pdb1",
        "DBInstanceIdentifier" : "my-cdb-inst",
        "TenantDBName" : "pdb1",
        "Status" : "modifying",
        "MasterUsername" : "tenant-admin-user"
```

```
        "Port" : "6555",
        "CharacterSetName" : "UTF-16",
        "MaxAllocatedStorage" : "1000",
        "ParameterGroups": [
            {
                "ParameterGroupName": "pdb1-params",
                "ParameterApplyStatus": "in-sync"
            }
        ],
        "OptionGroupMemberships": [
            {
                "OptionGroupName": "pdb1-options",
                "Status": "in-sync"
            }
        ],
        "PendingModifiedValues": {
            "TenantDBName": "pdb-hr"
        }
    }
}
```

Excluindo um banco de dados de locatários do RDS for Oracle do seu CDB

Você pode excluir um banco de dados de inquilinos (PDB) usando oAWS Management Console, o ou a AWS CLI API do RDS. Considere os seguintes requisitos e limitações:

- O banco de dados do locatário e a instância de banco de dados devem existir.
- Para que a exclusão seja bem-sucedida, uma das seguintes situações deve existir:
  - O banco de dados do inquilino e a instância de banco de dados estão disponíveis.

    > **Note**
    >
    > Você pode tirar um instantâneo final, mas somente se o banco de dados do locatário e a instância de banco de dados estiverem em um estado disponível antes de você emitir o `delete-tenant-database` comando.

  - O banco de dados de inquilinos está sendo criado.
  - A instância de banco de dados está modificando o banco de dados do inquilino.
- É possível excluir vários bancos de dados de locatários em uma única operação.
- Não será possível excluir um banco de dados locatário se ele for o único locatário no CDB.

Console

Para excluir um banco de dados de inquilinos

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Bancos de dados e o cluster de banco de dados que você deseja excluir.
3. Em Actions, escolha Delete.
4. Para criar um snapshot de banco de dados final para a instância de banco de dados, escolha Create final snapshot? (Criar snapshot final?).
5. Se você optar por criar um snapshot final, insira o Final snapshot name (Nome do snapshot final).
6. Digite **`delete me`** na caixa.
7. Escolha Delete (Excluir).

AWS CLI

Para excluir um banco de dados de inquilinos usando oAWS CLI, chame o comando delete-tenant-database com os seguintes parâmetros:

- `--db-instance-identifier` *`value`*
- `--tenant-db-name` *`value`*
- `[--skip-final-snapshot | --no-skip-final-snapshot]`
- `[--final-snapshot-identifier` *`value`*`]`

O exemplo a seguir exclui o banco de dados do inquilino chamado pdb-test do CDB chamado my-cdb-inst. Por padrão, a operação cria um instantâneo final.

Example

Para Linux, macOS ou Unix:

```
aws rds delete-tenant-database --region us-east-1 \
    --db-instance-identifier my-cdb-inst \
    --tenant-db-name pdb-test \
    --final-snapshot-identifier final-snap-pdb-test
```

Para Windows:

```
aws rds delete-tenant-database --region us-east-1 ^
    --db-instance-identifier my-cdb-inst ^
    --tenant-db-name pdb-test ^
    --final-snapshot-identifier final-snap-pdb-test
```

O comando gerará uma saída semelhante à seguinte:

```
{
    "TenantDatabase" : {
        "DbiResourceId" : "db-abc123",
        "TenantDatabaseResourceId" : "tdb-bac456",
        "TenantDatabaseArn" : "arn:aws:rds:us-east-1:123456789012:db:my-cdb-inst:pdb-
test",
        "DBInstanceIdentifier" : "my-cdb-inst",
        "TenantDBName" : "pdb-test",
        "Status" : "deleting",
        "MasterUsername" : "pdb-test-admin"
        "Port" : "6555",
        "CharacterSetName" : "UTF-16",
        "MaxAllocatedStorage" : "1000",
        "ParameterGroups": [
            {
                "ParameterGroupName": "tenant-1-params",
                "ParameterApplyStatus": "in-sync"
            }
        ],
        "OptionGroupMemberships": [
            {
                "OptionGroupName": "tenant-1-options",
                "Status": "in-sync"
            }
        ]
    }
}
```

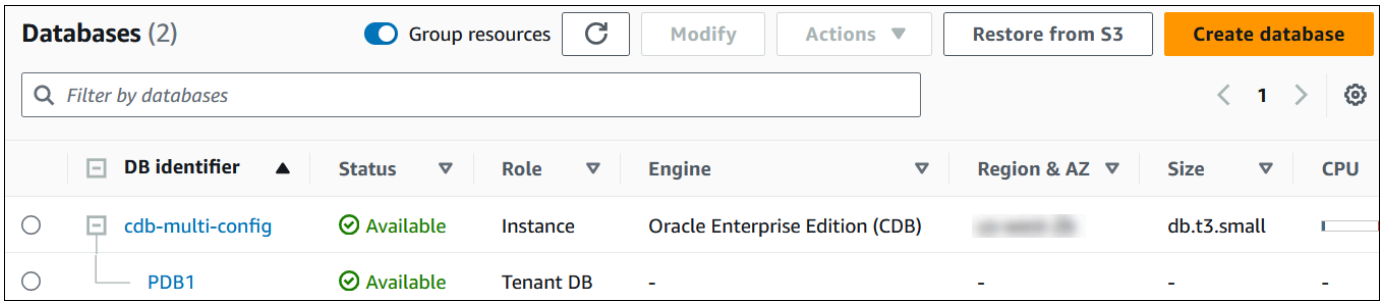
Visualizando detalhes do banco de dados do inquilino

Você pode visualizar detalhes sobre um banco de dados de inquilinos da mesma forma que você pode ver em um banco de dados que não seja CDB ou CDB.

Console

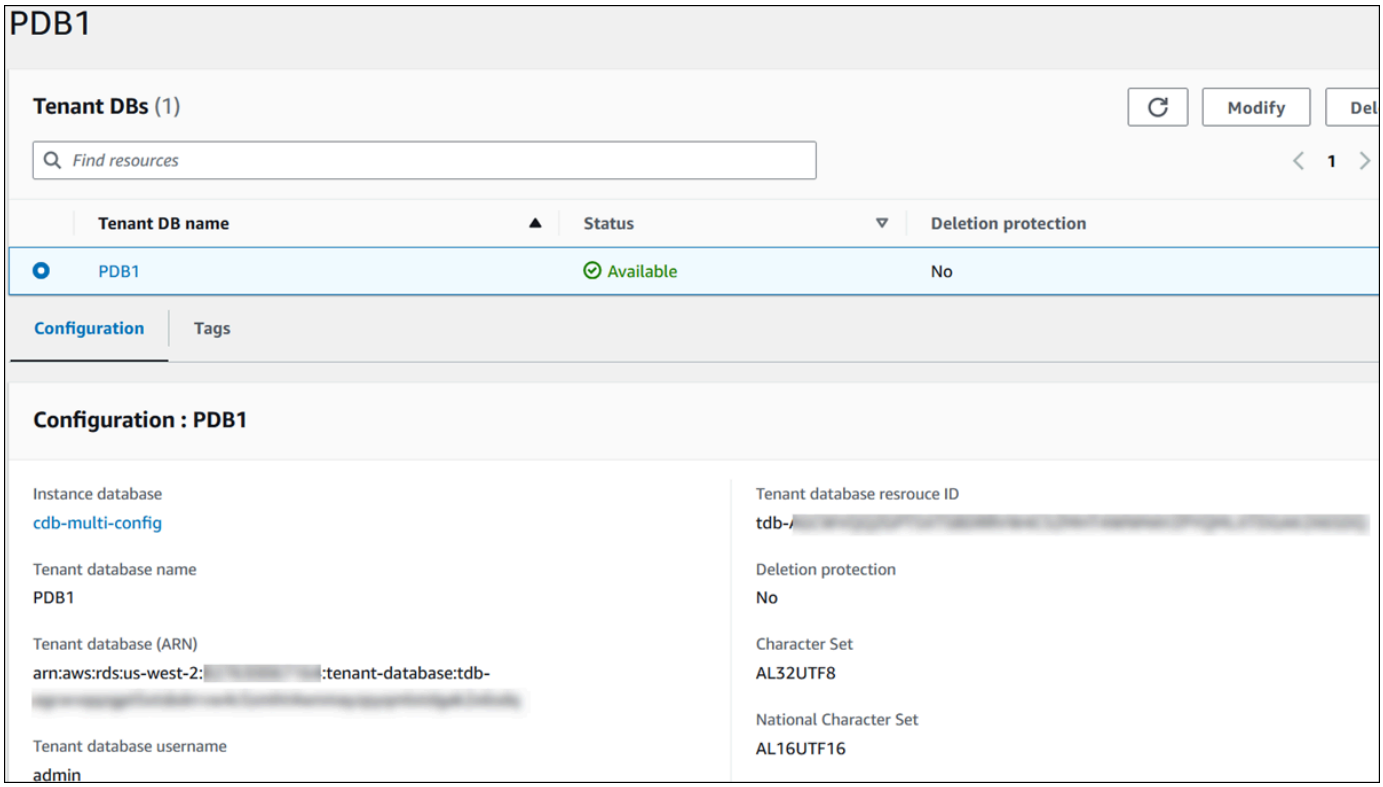
Para visualizar detalhes sobre um banco de dados de inquilinos

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No canto superior direito do console do Amazon RDS, escolha a Região da AWS em que sua instância de banco de dados se encontra.
3. No painel de navegação, escolha Databases (Bancos de dados).



Na imagem anterior, o banco de dados do único locatário (PDB) aparece como filho da instância de banco de dados.

4. O nome de um banco de dados hospedado em uma .



AWS CLI

Para ver detalhes sobre seus PDBs, use o AWS CLI comando describe-tenant-databases.

O exemplo a seguir descreve todos os bancos de dados de locatários na região especificada.

Example

Para Linux, macOS ou Unix:

```
aws rds describe-tenant-databases --region us-east-1
```

Para Windows:

```
aws rds describe-tenant-databases --region us-east-1
```

O comando gerará uma saída semelhante à seguinte:

```
    "TenantDatabases" : [
         {
            "DBInstanceIdentifier" : "my-cdb-inst",
            "TenantDBName" : "pdb-test",
            "Status" : "available",
            "MasterUsername" : "pdb-test-admin",
            "DbiResourceId" : "db-abc123",
            "TenantDatabaseResourceId" : "tdb-bac456",
            "TenantDatabaseArn" : "arn:aws:rds:us-east-1:123456789012:db:my-cdb-
inst:pdb-test",
            "CharacterSetName": "AL32UTF8",
            "NcharCharacterSetName": "AL16UTF16",
            "DeletionProtection": false,
            "PendingModifiedValues": {
                 "MasterUserPassword": "****"
            },
            "TagList": []
         },
         {

            "DBInstanceIdentifier" : "my-cdb-inst2",
            "TenantDBName" : "pdb-dev",
            "Status" : "modifying",
            "MasterUsername" : "masterrdsuser"
```

```
            "DbiResourceId" : "db-xyz789",
            "TenantDatabaseResourceId" : "tdb-ghp890",
            "TenantDatabaseArn" : "arn:aws:rds:us-east-1:123456789012:db:my-cdb-
inst2:pdb-dev",
            "CharacterSetName": "AL32UTF8",
            "NcharCharacterSetName": "AL16UTF16",
            "DeletionProtection": false,
            "PendingModifiedValues": {
                 "MasterUserPassword": "****"
            },
            "TagList": []
         },
         ... other truncated data
```

O exemplo a seguir descreve os bancos de dados de locatários na instância de banco de dados `my-cdb-inst` na região especificada.

Example

Para Linux, macOS ou Unix:

```
aws rds describe-tenant-databases --region us-east-1 \
    --db-instance-identifier my-cdb-inst
```

Para Windows:

```
aws rds describe-tenant-databases --region us-east-1 ^
    --db-instance-identifier my-cdb-inst
```

O comando gerará uma saída semelhante à seguinte:

```
{
    "TenantDatabase": {
        "TenantDatabaseCreateTime": "2023-10-19T23:55:30.046Z",
        "DBInstanceIdentifier": "my-cdb-inst",
        "TenantDBName": "pdb-hr",
        "Status": "creating",
        "MasterUsername": "tenant-admin-user",
        "DbiResourceId": "db-abc123",
        "TenantDatabaseResourceId": "tdb-bac567",
        "TenantDatabaseARN": "arn:aws:rds:us-west-2:579508833180:pdb-hr:tdb-
abcdefghi1jklmno2p3qrst4uvw5xy6zabc7defghi8jklmn90op",
```

```
        "CharacterSetName": "AL32UTF8",
        "NcharCharacterSetName": "AL16UTF16",
        "DeletionProtection": false,
        "PendingModifiedValues": {
            "MasterUserPassword": "****"
        },
        "TagList": [
            {
                "Key": "TEST",
                "Value": "testValue"
            }
        ]
    }
}
```

O exemplo a seguir descreve um banco de dados de pdb1 locatários `my-cdb-inst` na região Leste dos EUA (Norte da Virgínia).

Example

Para Linux, macOS ou Unix:

```
aws rds describe-tenant-databases --region us-east-1 \
--db-instance-identifier my-cdb-inst \
--tenant-db-name pdb1
```

Para Windows:

```
aws rds describe-tenant-databases --region us-east-1 ^
--db-instance-identifier my-cdb-inst ^
--tenant-db-name pdb1
```

O comando gerará uma saída semelhante à seguinte:

```
{
    "TenantDatabases" : [
        {
            "DbiResourceId" : "db-abc123",
            "TenantDatabaseResourceId" : "tdb-bac567",
            "TenantDatabaseArn" : "arn:aws:rds:us-east-1:123456789012:db:my-cdb-
inst:pdb1"
            "DBInstanceIdentifier" : "my-cdb-inst",
```

```
            "TenantDBName" : "pdb1",
            "Status" : "ACTIVE",
            "MasterUsername" : "masterawsuser"
            "Port" : "1234",
            "CharacterSetName": "UTF-8",
            "ParameterGroups": [
                {
                    "ParameterGroupName": "tenant-custom-pg",
                    "ParameterApplyStatus": "in-sync"
                }
            ],
            {
            "OptionGroupMemberships": [
                {
                    "OptionGroupName": "tenant-custom-og",
                    "Status": "in-sync"
                }
            ]
        }
    ]
}
```

Fazer upgrade de um CDB

Você pode atualizar um CDB para uma versão diferente do Oracle Database. Por exemplo, você pode fazer upgrade de um CDB Oracle Database 19c para CDB Oracle Database 21c. Não é possível alterar a arquitetura do banco de dados durante um upgrade. Portanto, não é possível fazer upgrade de um não CDB para CDB ou fazer upgrade de um CDB para não CDB.

O procedimento para fazer upgrade de um CDB para CDB é igual àquele para fazer upgrade de um não CDB para não CDB. Para obter mais informações, consulte Fazer upgrade do mecanismo de banco de dados RDS para Oracle.

Administração da sua instância de banco de dados Oracle

A seguir estão as tarefas de gerenciamento comuns que você realiza com uma instância de banco de dados do Amazon RDS. Algumas tarefas são as mesmas para todas as instâncias de banco de dados do RDS. Outras tarefas são específicas do RDS for Oracle.

As tarefas a seguir são comuns a todos os bancos de dados do RDS, mas o Oracle tem considerações especiais. Por exemplo, conecte-se a um banco de dados Oracle usando os clientes Oracle SQL*Plus e SQL Developer.

Área de tarefa	Documentação relevante
Classes de instância, armazenamento e PIOPS Se você estiver criando uma instância de produção, saiba como funcionam as classes de instância, os tipos de armazenamento e as IOPS provisionadas no Amazon RDS.	Classes de instância de banco de dados do RDS para Oracle Tipos de armazenamento do Amazon RDS
Implantações multi-AZ Uma instância de banco de dados de produção deve usar implantações multi-AZ. As implantações Multi-AZ oferecem maior disponibilidade, durabilidade de dados e tolerância a falhas para instâncias de banco de dados.	Configurar e gerenciar uma implantação multi-AZ para o Amazon RDS
Amazon VPC Se a sua conta AWS tiver uma nuvem privada virtual (VPC) padrão, a instância de banco de dados será criada automatic amente dentro da VPC padrão. Se a sua conta não tiver uma VPC padrão e você desejar a instância de banco de dados em uma VPC, crie os grupos de VPC e sub-redes antes de criar a instância.	Trabalhar com uma instância de banco de dados em uma VPC
Grupos de segurança Por padrão, as instâncias de banco de dados usam um firewall que impede o acesso. Crie um grupo de segurança com os	Controlar acesso com grupos de segurança

Área de tarefa	Documentação relevante
endereços IP corretos e uma configuração de rede para acessar a instância de banco de dados.	
Grupos de parâmetros Se a sua instância de banco de dados exigir parâmetros de banco de dados específicos, crie um grupo de parâmetros antes de criar essa instância de banco de dados.	Grupos de parâmetros para Amazon RDS
Grupos de opções Se a sua instância de banco de dados exigir opções de banco de dados específicas, crie um grupo de opções antes de criar a instância de banco de dados.	Adição de opções a instâncias de banco de dados Oracle
Conexão à instância de banco de dados Após criar um grupo de segurança e associá-lo a uma instância de banco de dados, será possível se conectar à instância de banco de dados usando qualquer aplicativo de cliente SQL padrão, como o Oracle SQL*Plus.	Conectar-se a uma instância de banco de dados Oracle
Backup e restauração Você pode configurar a instância de banco de dados para fazer backups automáticos ou snapshots manuais e, depois, restaurar instâncias a partir dos backups ou snapshots.	Backup, restauração e exportação de dados
Monitoramento Você pode monitorar uma instância de banco de dados Oracle usando as métricas, eventos e o monitoramento aprimorado do Amazon RDS do CloudWatch.	Visualizar métricas no console do Amazon RDS Visualizar eventos do Amazon RDS
Arquivos de log Você pode acessar os arquivos de log de sua instância de banco de dados Oracle.	Monitorar arquivos de log do Amazon RDS

A seguir, você pode encontrar uma descrição para implantações específicas do Amazon RDS de tarefas comuns de DBA para o RDS Oracle. Para oferecer uma experiência de serviço gerenciada, o Amazon RDS não fornece acesso ao shell para as instâncias de banco de dados. Além disso, o RDS restringe o acesso a determinados procedimentos e tabelas do sistema que exigem privilégios avançados. Em muitas das tarefas, você executa o pacote `rdsadmin`, que é uma ferramenta específica do Amazon RDS que permite administrar seu banco de dados.

Veja a seguir tarefas do DBA comuns para instâncias de bancos de dados executando o Oracle:

- Tarefas do sistema

Desconectar uma sessão	Método do Amazon RDS: `rdsadmin.rdsadmin_util.disconnect` Método do Oracle: `alter system disconnect session`
Encerrar uma sessão	Método do Amazon RDS: `rdsadmin.rdsadmin_util.kill` Método do Oracle: `alter system kill session`
Como cancelar uma instrução SQL em uma sessão	Método do Amazon RDS: `rdsadmin.rdsadmin_util.cancel` Método do Oracle: `alter system cancel sql`
Habilitar e desabilitar sessões restritas	Método do Amazon RDS: `rdsadmin.rdsadmin_util.restricted_session` Método do Oracle: `alter system enable restricted session`
Descarregar o grupo compartilhado	Método do Amazon RDS: `rdsadmin.rdsadmin_util.flush_shared_pool` Método do Oracle: `alter system flush shared_pool`
Descarregar o cache do buffer	Método do Amazon RDS: `rdsadmin.rdsadmin_util.flush_buffer_cache` Método do Oracle: `alter system flush buffer_cache`

Concessão de privilégios SELECT ou EXECUTE a objetos SYS	Método do Amazon RDS: `rdsadmin.rdsadmin_util.grant_sys_object` Método do Oracle: `grant`
Revogar privilégios SELECT ou EXECUTE em objetos SYS	Método do Amazon RDS: `rdsadmin.rdsadmin_util.revoke_sys_object` Método do Oracle: `revoke`
Gerenciar visualizações RDS_X$ para instâncias de banco de dados Oracle	Método do Amazon RDS: `rdsadmin.rdsadmin_util.create_sys_x$_view` Método do Oracle: `CREATE VIEW`
Concessão de privilégios a usuários não mestres	Método do Amazon RDS: `grant`
Criação de funções personalizadas para verificar senhas	Método do Amazon RDS: `rdsadmin.rdsadmin_password_verify.create_verify_function` Método do Amazon RDS: `rdsadmin.rdsadmin_password_verify.create_passthrough_verify_fcn`
Configuração de um servidor DNS personalizado	—
Listar eventos de diagnóstico do sistema permitidos	Método do Amazon RDS: `rdsadmin.rdsadmin_util.list_allowed_system_events` Método do Oracle: —
Definir eventos de diagnóstico do sistema	Método do Amazon RDS: `rdsadmin.rdsadmin_util.set_allowed_system_events` Método do Oracle: `ALTER SYSTEM SET EVENTS 'set_event_clause'`

Listar eventos de diagnóstico do sistema definidos	Método do Amazon RDS: `rdsadmin.rdsadmin_util.list_set_system_events` Método do Oracle: `ALTER SESSION SET EVENTS 'IMMEDIATE EVENTDUMP(SYSTEM)'`
Desconfigurar eventos de diagnóstico do sistema	Método do Amazon RDS: `rdsadmin.rdsadmin_util.unset_system_event` Método do Oracle: `ALTER SYSTEM SET EVENTS 'unset_event_clause'`

- Tarefas de banco de dados

Alteração do nome global de um banco de dados	Método do Amazon RDS: `rdsadmin.rdsadmin_util.rename_global_name` Método do Oracle: `alter database rename`
Criação e uso de espaços de tabela	Método do Amazon RDS: `create tablespace` Método do Oracle: `alter database`
Definição do espaço de tabela padrão	Método do Amazon RDS: `rdsadmin.rdsadmin_util.alter_default_tablespace` Método do Oracle: `alter database default tablespace`
Definição do espaço de tabela temporário padrão	Método do Amazon RDS: `rdsadmin.rdsadmin_util.alter_default_temp_tablespace` Método do Oracle: `alter database default temporary tablespace`

Criar um espaço de tabela temporário no armazenamento de instância	Método do Amazon RDS: `rdsadmin.rdsadmin_util.create_inst_store_tmp_tblspace` Método do Oracle: `create temporary tablespace`
Verificação de um banco de dados	Método do Amazon RDS: `rdsadmin.rdsadmin_util.checkpoint` Método do Oracle: `alter system checkpoint`
Definição da recuperação distribuída	Método do Amazon RDS: `rdsadmin.rdsadmin_util.enable_distr_recovery` Método do Oracle: `alter system enable distributed recovery`
Definição do fuso horário do banco de dados	Método do Amazon RDS: `rdsadmin.rdsadmin_util.alter_db_time_zone` Método do Oracle: `alter database set time_zone`
Trabalhar com tabelas externas do Oracle	—
Gerar relatórios de performance com o AWR (Automatic Workload Repository)	Método do Amazon RDS: procedimentos `rdsadmin.rdsadmin_diagnostic_util` Método do Oracle: pacote `dbms_workload_repository`
Ajuste de vínculos de banco de dados para uso com instâncias de bancos de dados em uma VPC	—
Definir a edição padrão para uma instância de banco de dados	Método do Amazon RDS: `rdsadmin.rdsadmin_util.alter_default_edition` Método do Oracle: `alter database default edition`

Habilitar a auditoria para a tabela SYS.AUD$	Método do Amazon RDS: `rdsadmin.rdsadmin_master_util.audit_all_sys_aud_table` Método do Oracle: `audit`
Desabilitar a auditoria para a tabela SYS.AUD$	Método do Amazon RDS: `rdsadmin.rdsadmin_master_util.noaudit_all_sys_aud_table` Método do Oracle: `noaudit`
Limpar criações de índice online interrompidas	Método do Amazon RDS: `rdsadmin.rdsadmin_dbms_repair.online_index_clean` Método do Oracle: `dbms_repair.online_index_clean`
Ignorar blocos corrompidos	Método do Amazon RDS: diversos procedimentos `rdsadmin.rdsadmin_dbms_repair` Método do Oracle: pacote `dbms_repair`
Redimensionar espaços de tabela, arquivos de dados e arquivos temporários	Método do Amazon RDS: procedimentos `rdsadmin.rdsadmin_util.resize_temp_tablespace`, `rdsadmin.rdsadmin_util.resize_tempfile` ou `rdsadmin.rdsadmin_util.autoextend_tempfile` Procedimento `rdsadmin.rdsadmin_util.resize_datafile` ou `rdsadmin.rdsadmin_util.autoextend_datafile` Método do Oracle: —
Limpar a lixeira	Método do Amazon RDS: `EXEC rdsadmin.rdsadmin_util.purge_dba_recyclebin` Método do Oracle: `purge dba_recyclebin`

Definir os valores padrão exibidos para a redação completa	Método do Amazon RDS: `EXEC rdsadmin.rdsadmin_util.dbms_redact_upd_full_rdct_val` Método do Oracle: `exec dbms_redact.UPDATE_FULL_REDACTION_VALUES`

- Tarefas de log

Definição do registro em log forçado	Método do Amazon RDS: `rdsadmin.rdsadmin_util.force_logging` Método do Oracle: `alter database force logging`
Definição do registro em log complementar	Método do Amazon RDS: `rdsadmin.rdsadmin_util.alter_supplemental_logging` Método do Oracle: `alter database add supplemental log`
Alternância de arquivos de log online	Método do Amazon RDS: `rdsadmin.rdsadmin_util.switch_logfile` Método do Oracle: `alter system switch logfile`
Adição de logs redo online	Método do Amazon RDS: `rdsadmin.rdsadmin_util.add_logfile`

Descarte de logs redo online	Método do Amazon RDS: `rdsadmin.rdsadmin_util.drop_logfile`
Redimensionamento de logs redo online	—
Retenção de logs redo arquivados	Método do Amazon RDS: `rdsadmin.rdsadmin_util.set_configuration`
Baixar logs de refazimento arquivados do Amazon S3	Método do Amazon RDS: `rdsadmin.rdsadmin_archive_log_download.download_log_with_seqnum` Método do Amazon RDS: `rdsadmin.rdsadmin_archive_log_download.download_logs_in_seqnum_range`
Acessando logs de refazimento online e arquivados	Método do Amazon RDS: `rdsadmin.rdsadmin_master_util.create_archivelog_dir` Método do Amazon RDS: `rdsadmin.rdsadmin_master_util.create_onlinelog_dir`

- Tarefas do RMAN

Validar arquivos de banco de dados no RDS para Oracle	Método do Amazon RDS: `rdsadmin_rman_util . procedure` Método do Oracle: `RMAN VALIDATE`
Habilitar e desabilitar o controle de alterações de bloco	Método do Amazon RDS: `rdsadmin_rman_util . procedure` Método do Oracle: `ALTER DATABASE`
Verificação cruzada de logs redo	Método do Amazon RDS: `rdsadmin_rman_util .crosscheck_archivelog` Método do Oracle: `RMAN BACKUP`
Fazer backup de logs redo armazenados	Método do Amazon RDS: `rdsadmin_rman_util . procedure` Método do Oracle: `RMAN BACKUP`
Realizar um backup de banco de dados completo	Método do Amazon RDS: `rdsadmin_rman_util .backup_database_full` Método do Oracle: `RMAN BACKUP`

Realizar um backup de banco de dados incremental	Método do Amazon RDS: `rdsadmin_rman_util.backup_database_incremental` Método do Oracle: `RMAN BACKUP`
Backup de um tablespace	Método do Amazon RDS: `rdsadmin_rman_util.backup_database_tablespace` Método do Oracle: `RMAN BACKUP`

- Tarefas do Oracle Scheduler

Modificação de trabalhos DBMS_SCHEDULER	Método do Amazon RDS: `dbms_scheduler.set_attribute` Método do Oracle: `dbms_scheduler.set_attribute`
Modificando janelas de manutenção de AutoTask	Método do Amazon RDS: `dbms_scheduler.set_attribute` Método do Oracle: `dbms_scheduler.set_attribute`

Definir o fuso horário para trabalhos do Oracle Scheduler	Método do Amazon RDS: `dbms_scheduler.set_scheduler_attribute` Método do Oracle: `dbms_scheduler.set_scheduler_attribute`
Desativar trabalhos do Oracle Scheduler de propriedade do SYS	Método do Amazon RDS: `rdsadmin.rdsadmin_dbms_scheduler.disable` Método do Oracle: `dbms_scheduler.disable`
Ativar trabalhos do Oracle Scheduler de propriedade do SYS	Método do Amazon RDS: `rdsadmin.rdsadmin_dbms_scheduler.enable` Método do Oracle: `dbms_scheduler.enable`
Modificar o intervalo de repetição do Oracle Scheduler para trabalhos do tipo CALENDAR	Método do Amazon RDS: `rdsadmin.rdsadmin_dbms_scheduler.set_attribute` Método do Oracle: `dbms_scheduler.set_attribute`

Modificar o intervalo de repetição do Oracle Scheduler para trabalhos do tipo NAMED	Método do Amazon RDS: `rdsadmin.rdsadmin_dbms_scheduler.set_attribute` Método do Oracle: `dbms_scheduler.set_attribute`
Desativar a confirmação automática para a criação de trabalhos do Oracle Scheduler	Método do Amazon RDS: `rdsadmin.rdsadmin_dbms_scheduler.set_no_commit_flag` Método do Oracle: `dbms_isched.set_no_commit_flag`

- Diagnosticar problemas

Listar incidentes	Método do Amazon RDS: `rdsadmin.rdsadmin_adrci_util.list_adrci_incidents` Método do Oracle: comando ADRCI `show incident`
Listar problemas	Método do Amazon RDS: `rdsadmin.rdsadmin_adrci_util.list_adrci_problem` Método do Oracle: comando ADRCI `show problem`

Criar pacotes de incidente	Método do Amazon RDS: `rdsadmin.rdsadmin_adrci_util.create_adrci_package` Método do Oracle: comando ADRCI `ips create package`
Mostrar arquivos de rastreamento	Método do Amazon RDS: `rdsadmin.rdsadmin_adrci_util.show_adrci_tracefile` Método do Oracle: comando ADRCI `show tracefile`

- Outras tarefas

Criar e eliminar diretórios no espaço de armazenamento de dados principal	Método do Amazon RDS: `rdsadmin.rdsadmin_util.create_directory` Método do Oracle: `CREATE DIRECTORY` Método do Amazon RDS: `rdsadmin.rdsadmin_util.drop_directory` Método do Oracle: `DROP DIRECTORY`

Listagem de arquivos no diretório de uma instância de banco de dados	Método do Amazon RDS: `rdsadmin.rds_file_util.listdir` Método do Oracle: —
Leitura de arquivos no diretório de uma instância de banco de dados	Método do Amazon RDS: `rdsadmin.rds_file_util.read_text_file` Método do Oracle: —
Acessar arquivos do Opatch	Método do Amazon RDS: `rdsadmin.rds_file_util.read_text_file` ou `rdsadmin.tracefile_listing` Método do Oracle: `opatch`
Definição de parâmetros para tarefas do advisor	Método do Amazon RDS: `rdsadmin.rdsadmin_util.advisor_task_set_parameter` Método Oracle: vários procedimentos de pacotes armazenados
Desativação de AUTO_STATS_ADVISOR_TASK	Método do Amazon RDS: `rdsadmin.rdsadmin_util.advisor_task_drop` Método do Oracle: —

Reativação de AUTO_STATS_ADVISOR_TASK	Método do Amazon RDS: `rdsadmin.rdsadmin_util.dbms_stats_init` Método do Oracle: —

Você também pode usar procedimentos do Amazon RDS para integração do Amazon S3 com o Oracle e para executar tarefas de banco de dados do OEM Management Agent. Para ter mais informações, consulte Integração do Amazon S3 e Como executar tarefas de banco de dados com o Management Agent.

Realização de tarefas comuns do sistema para instâncias de banco de dados Oracle

Veja a seguir como executar determinadas tarefas comuns de DBA relacionadas ao sistema nas instâncias de bancos de dados do Amazon RDS que executam o Oracle. Para oferecer uma experiência de serviço gerenciado, o Amazon RDS não fornece acesso ao shell para instâncias de bancos de dados e restringe o acesso a determinados procedimentos e tabelas do sistema que exigem privilégios avançados.

Tópicos

- Desconectar uma sessão
- Encerrar uma sessão
- Como cancelar uma instrução SQL em uma sessão
- Habilitar e desabilitar sessões restritas
- Descarregar o grupo compartilhado
- Concessão de privilégios SELECT ou EXECUTE a objetos SYS
- Revogar privilégios SELECT ou EXECUTE em objetos SYS
- Gerenciar visualizações RDS_X$ para instâncias de banco de dados Oracle
- Concessão de privilégios a usuários não mestres
- Criação de funções personalizadas para verificar senhas
- Configuração de um servidor DNS personalizado

- Configurar e anular configuração de eventos de diagnóstico do sistema

Desconectar uma sessão

Para desconectar a sessão atual finalizando o processo de servidor dedicado, use o procedimento `rdsadmin.rdsadmin_util.disconnect` do Amazon RDS. O procedimento `disconnect` tem os seguintes parâmetros.

Nome do parâmetro	Tipo de dados	Padrão	Obrigatório	Descrição
`sid`	número	—	Sim	O identificador da sessão.
`serial`	número	—	Sim	O número de série da sessão.
`method`	varchar	'IMMEDIATE'	Não	Os valores válidos são `'IMMEDIATE'` ou `'POST_TRANSACTION'`.

O exemplo a seguir desconecta uma sessão.

```
begin
    rdsadmin.rdsadmin_util.disconnect(
        sid    => sid,
        serial => serial_number);
end;
/
```

Para obter o identificador da sessão e o número de série da sessão, consulte a visualização `V$SESSION`. O seguinte exemplo obtém todas as sessões para o usuário `AWSUSER`.

```
SELECT SID, SERIAL#, STATUS FROM V$SESSION WHERE USERNAME = 'AWSUSER';
```

O banco de dados deve estar aberto para usar esse método. Para obter mais informações sobre como desconectar uma sessão, consulte ALTER SYSTEM (Alterar sistema) na documentação do Oracle.

Encerrar uma sessão

Para encerrar uma sessão, use o procedimento do Amazon RDS `rdsadmin.rdsadmin_util.kill`. O procedimento `kill` tem os seguintes parâmetros.

Nome do parâmetro	Tipo de dados	Padrão	Obrigatório	Descrição
`sid`	número	—	Sim	O identificador da sessão.
`serial`	número	—	Sim	O número de série da sessão.
`method`	varchar	nulo	Não	Os valores válidos são `'IMMEDIATE'` ou `'PROCESS'`. Se você especificar `IMMEDIATE`, terá o mesmo resultado que o produzido ao executar a seguinte instrução: `ALTER SYSTEM KILL SESSION 'sid,serial#' IMMEDIATE` Se você especificar `PROCESS`, encerrará os processos associados a uma sessão. Especifique `PROCESS` apenas se o encerramento da sessão com `IMMEDIATE` foi bem-sucedido.

Para obter o identificador da sessão e o número de série da sessão, consulte a visualização V`$SESSION`. O seguinte exemplo obtém todas as sessões para o usuário *`AWSUSER`*.

```
SELECT SID, SERIAL#, STATUS FROM V$SESSION WHERE USERNAME = 'AWSUSER';
```

O exemplo a seguir encerra uma sessão.

```
BEGIN
    rdsadmin.rdsadmin_util.kill(
        sid    => sid,
        serial => serial_number,
        method => 'IMMEDIATE');
END;
/
```

No exemplo a seguir, os processos associados a uma sessão são encerrados.

```
BEGIN
    rdsadmin.rdsadmin_util.kill(
        sid    => sid,
        serial => serial_number,
        method => 'PROCESS');
END;
/
```

Como cancelar uma instrução SQL em uma sessão

Para cancelar uma instrução SQL em uma sessão, use o procedimento `rdsadmin.rdsadmin_util.cancel` do Amazon RDS.

> **Note**
>
> Esse procedimento é compatível com o Oracle Database 19c (19.0.0) e com todas as versões principais e secundárias posteriores do RDS for Oracle.

O procedimento `cancel` tem os seguintes parâmetros.

Nome do parâmetro	Tipo de dados	Padrão	Obrigatório	Descrição
`sid`	número	—	Sim	O identificador da sessão.

Nome do parâmetro	Tipo de dados	Padrão	Obrigatório	Descrição
`serial`	número	—	Sim	O número de série da sessão.
`sql_id`	varchar2	nulo	Não	O identificador SQL da instrução SQL

O exemplo a seguir canela uma instrução SQL em uma sessão.

```
begin
    rdsadmin.rdsadmin_util.cancel(
        sid    => sid,
        serial => serial_number,
        sql_id => sql_id);
end;
/
```

Para obter o identificador da sessão, o número de série da sessão e o identificador SQL de uma instrução SQL, consulte a visualização `V$SESSION`. O exemplo a seguir obtém todos os identificadores SQL e de sessão para o usuário `AWSUSER`.

```
select SID, SERIAL#, SQL_ID, STATUS from V$SESSION where USERNAME = 'AWSUSER';
```

Habilitar e desabilitar sessões restritas

Para habilitar e desabilitar sessões restritas, use o procedimento `rdsadmin.rdsadmin_util.restricted_session` do Amazon RDS. O procedimento `restricted_session` tem os seguintes parâmetros.

Nome do parâmetro	Tipo de dados	Padrão	Sim	Descrição
`p_enable`	booliano	true	Não	Defina como `true` para habilitar sessões restritas , `false` para desabilitar sessões restritas.

O exemplo a seguir mostra como habilitar e desabilitar sessões restritas.

```
/* Verify that the database is currently unrestricted. */

SELECT LOGINS FROM V$INSTANCE;

LOGINS
-------
ALLOWED

/* Enable restricted sessions */

EXEC rdsadmin.rdsadmin_util.restricted_session(p_enable => true);


/* Verify that the database is now restricted. */

SELECT LOGINS FROM V$INSTANCE;

LOGINS
----------
RESTRICTED


/* Disable restricted sessions */

EXEC rdsadmin.rdsadmin_util.restricted_session(p_enable => false);


/* Verify that the database is now unrestricted again. */

SELECT LOGINS FROM V$INSTANCE;

LOGINS
-------
ALLOWED
```

Descarregar o grupo compartilhado

Para descarregar o grupo compartilhado, use o procedimento `rdsadmin.rdsadmin_util.flush_shared_pool` do Amazon RDS. O procedimento `flush_shared_pool` não tem parâmetros.

O exemplo a seguir libera o pool compartilhado.

```
EXEC rdsadmin.rdsadmin_util.flush_shared_pool;
```

Descarregar o cache do buffer

Para descarregar o cache do buffer, use o procedimento `rdsadmin.rdsadmin_util.flush_buffer_cache` do Amazon RDS. O procedimento `flush_buffer_cache` não tem parâmetros.

O exemplo a seguir libera o cache do buffer.

```
EXEC rdsadmin.rdsadmin_util.flush_buffer_cache;
```

Limpar o cache flash inteligente do banco de dados

Para limpar o cache do flash inteligente do banco de dados, use o procedimento `rdsadmin.rdsadmin_util.flush_flash_cache` do Amazon RDS. O procedimento `flush_flash_cache` não tem parâmetros. O exemplo a seguir libera o cache do flash inteligente do banco de dados.

```
EXEC rdsadmin.rdsadmin_util.flush_flash_cache;
```

Para obter mais informações sobre o uso do cache flash inteligente do banco de dados com o RDS para Oracle, consulte Armazenar dados temporários em um armazenamento de instância do RDS for Oracle.

Concessão de privilégios SELECT ou EXECUTE a objetos SYS

Normalmente, você transfere privilégios usando funções, que podem conter muitos objetos. Para conceder privilégios a um único objeto, use o procedimento `rdsadmin.rdsadmin_util.grant_sys_object` do Amazon RDS. O procedimento concede apenas privilégios já concedidos ao usuário mestre por uma função ou por uma concessão direta.

O procedimento `grant_sys_object` tem os seguintes parâmetros.

> **Important**
>
> Para todos os valores de parâmetro, use letras maiúsculas, a menos que você tenha criado o usuário com um identificador que diferencia letras maiúsculas de minúsculas. Por exemplo,

se você executar `CREATE USER myuser` ou `CREATE USER MYUSER`, o dicionário de dados armazenará `MYUSER`. No entanto, se você usar aspas duplas em `CREATE USER "MyUser"`, o dicionário de dados armazenará `MyUser`.

Nome do parâmetro	Tipo de dados	Padrão	Obrigatório	Descrição
`p_obj_name`	varchar2	—	Sim	O nome do objeto ao qual conceder privilégios. O objeto pode ser um diretório, uma função, um pacote, um procedimento, uma sequência, uma tabela ou uma visualização. Os nomes dos objetos devem ser expressos exatamente como eles aparecem em `DBA_OBJECTS`. Como a maioria dos objetos do sistema é definida em maiúsculas, recomendamos que você tente isso primeiro.
`p_grantee`	varchar2	—	Sim	O nome do objeto para o qual conceder privilégios. O objeto pode ser uma esquema ou um função.
`p_privilege`	varchar2	nulo	Sim	—
`p_grant_option`	booliano	false	Não	Defina como `true` para usar a opção de concessão.

O exemplo a seguir concede privilégios de seleção em um objeto chamado V_$SESSION a um usuário chamado USER1.

```
begin
    rdsadmin.rdsadmin_util.grant_sys_object(
        p_obj_name  => 'V_$SESSION',
        p_grantee   => 'USER1',
        p_privilege => 'SELECT');
end;
/
```

O exemplo a seguir concede privilégios de seleção em um objeto chamado V_$SESSION a um usuário chamado USER1 com a opção de concessão.

```
begin
    rdsadmin.rdsadmin_util.grant_sys_object(
        p_obj_name     => 'V_$SESSION',
        p_grantee      => 'USER1',
        p_privilege    => 'SELECT',
        p_grant_option => true);
end;
/
```

Para poder conceder privilégios a um objeto, sua conta deve ter esses privilégios concedidos a ela diretamente com a opção de concessão ou por meio de uma função concedida usando `with admin option`. No caso mais comum, você pode querer conceder SELECT em uma visualização de DBA que foi concedida à função SELECT_CATALOG_ROLE. Se essa função ainda não estiver diretamente concedida ao seu usuário usando `with admin option`, você não poderá transferir o privilégio. Se você tiver privilégios de DBA, poderá conceder a função diretamente a outro usuário.

O exemplo a seguir concede SELECT_CATALOG_ROLE e EXECUTE_CATALOG_ROLE a USER1. Com o `with admin option` é usado, USER1 agora pode conceder acesso a objetos SYS que foram concedidos a SELECT_CATALOG_ROLE.

```
GRANT SELECT_CATALOG_ROLE TO USER1 WITH ADMIN OPTION;
GRANT EXECUTE_CATALOG_ROLE to USER1 WITH ADMIN OPTION;
```

Objetos já concedidos a PUBLIC não precisam ser novamente concedidos. Se você usar o procedimento `grant_sys_object` para conceder acesso novamente, a chamada de procedimento será bem-sucedida.

Revogar privilégios SELECT ou EXECUTE em objetos SYS

Para revogar os privilégios de um único objeto, use o procedimento `rdsadmin.rdsadmin_util.revoke_sys_object` do Amazon RDS. O procedimento revoga apenas os privilégios já concedido à conta mestra por uma função ou por uma concessão direta.

O procedimento `revoke_sys_object` tem os seguintes parâmetros.

Nome do parâmetro	Tipo de dados	Padrão	Obrigatório	Descrição
`p_obj_name`	varchar2	—	Sim	O nome do objeto ao qual revogar privilégios. O objeto pode ser um diretório, uma função, um pacote, um procedimento, uma sequência, uma tabela ou uma visualização. Os nomes dos objetos devem ser expressos exatamente e como eles aparecem em `DBA_OBJECTS`. A maioria dos objetos do sistema é definida em maiúsculas, então recomendamos que você tente isso primeiro.
`p_revokee`	varchar2	—	Sim	O nome do objeto ao qual revogar privilégios. O objeto pode ser uma esquema ou um função.
`p_privilege`	varchar2	nulo	Sim	—

O exemplo a seguir revoga privilégios de seleção em um objeto denominado V_$SESSION de um usuário denominado USER1.

```
begin
    rdsadmin.rdsadmin_util.revoke_sys_object(
        p_obj_name  => 'V_$SESSION',
        p_revokee   => 'USER1',
        p_privilege => 'SELECT');
end;
/
```

Gerenciar visualizações RDS_X$ para instâncias de banco de dados Oracle

Talvez seja necessário acessar tabelas SYS.X$ fixas, que só podem ser acessadas por SYS. Para criar visualizações SYS.RDS_X$ em tabelas X$ elegíveis, use os procedimentos no pacote rdsadmin.rdsadmin_util. O usuário principal recebe automaticamente o privilégio SELECT … WITH GRANT OPTION nas visualizações RDS_X$.

Os procedimentos rdsadmin.rdsadmin_util estão disponíveis nas seguintes versões do banco de dados:

- 21.0.0.0.ru-2023-10.rur-2023-10.r1 e versões posteriores do Oracle Database 21c
- 19.0.0.0.ru-2023-10.rur-2023-10.r1 e versões posteriores do Oracle Database 19c

> **Important**
>
> Internamente, o pacote rdsadmin.rdsadmin_util cria visualizações em tabelas X$. As tabelas X$ são objetos internos do sistema que não estão descritos na documentação do Oracle Database. Recomendamos testar visualizações específicas no banco de dados que não é de produção e somente criar visualizações no banco de dados de produção com a orientação do Oracle Support.

Listar tabelas fixas X$ elegíveis para uso em visualizações RDS_X$

Para listar tabelas X$ que são elegíveis para uso em visualizações RDS_X$, use o procedimento rdsadmin.rdsadmin_util.list_allowed_sys_x$_views do RDS. Este procedimento não aceita parâmetros. As declarações a seguir listam todas as tabelas X$ elegíveis (exemplo de saída incluído).

```
SQL> SET SERVEROUTPUT ON
SQL> SELECT * FROM TABLE(rdsadmin.rdsadmin_util.list_allowed_sys_x$_views);

'X$BH'
'X$K2GTE'
'X$KCBWBPD'
'X$KCBWDS'
'X$KGLLK'
'X$KGLOB'
'X$KGLPN'
'X$KSLHOT'
'X$KSMSP'
'X$KSPPCV'
'X$KSPPI'
'X$KSPPSV'
'X$KSQEQ'
'X$KSQRS'
'X$KTUXE'
'X$KQRFP'
```

A lista das tabelas X$ elegíveis pode mudar ao longo do tempo. Para garantir que a lista de tabelas fixas X$ elegíveis esteja atualizada, execute `list_allowed_sys_x$_views` periodicamente.

Criar visualizações SYS.RDS_X$

Para criar uma visualização RDS_X$ em uma tabela X$ elegível, use o procedimento `rdsadmin.rdsadmin_util.create_sys_x$_view` do RDS. É possível criar visualizações apenas para as tabelas listadas na saída de `rdsadmin.rdsadmin_util.list_allowed_sys_x$_views`. O procedimento `create_sys_x$_view` tem os parâmetros a seguir.

Nome do parâmetro	Tipo de dados	Padrão	Obrigatório	Descrição
`p_x$_tbl`	varchar2	Null	Sim	O nome de uma tabela X$ válida. O valor deve ser uma das tabelas X$ relatadas por `list_allowed_sys_x$_views` .
`p_force_creation`	Booleano	`FALSE`	Não	Um valor que indica se é necessário forçar a

Nome do parâmetro	Tipo de dados	Padrão	Obrigatório	Descrição
				criação de uma visualização RDS_X$ que já existe para uma tabela X$. Por padrão, o RDS não criará uma visualização se ela já existir. Para forçar a criação, defina esse parâmetro como TRUE.

O exemplo a seguir cria a visualização SYS.RDS_X$KGLOB na tabela X$KGLOB. O formato do nome da visualização é RDS_*X$tablename*.

```
SQL> SET SERVEROUTPUT ON
SQL> EXEC rdsadmin.rdsadmin_util.create_sys_x$_view('X$KGLOB');

PL/SQL procedure successfully completed.
```

A consulta do dicionário de dados a seguir lista a visualização SYS.RDS_X$KGLOB e mostra o status. O usuário principal recebe automaticamente o privilégio SELECT ... WITH GRANT OPTION nessa visualização.

```
SQL> SET SERVEROUTPUT ON
SQL> COL OWNER FORMAT A30
SQL> COL OBJECT_NAME FORMAT A30
SQL> COL STATUS FORMAT A30
SQL> SET LINESIZE 200
SQL> SELECT OWNER, OBJECT_NAME, STATUS
FROM DBA_OBJECTS
WHERE OWNER = 'SYS' AND OBJECT_NAME = 'RDS_X$KGLOB';

OWNER                          OBJECT_NAME                    STATUS
------------------------------ ------------------------------
 ------------------------------
SYS                            RDS_X$KGLOB                    VALID
```

> **Important**
>
> Não é garantido que as tabelas X$ permaneçam as mesmas antes e depois de uma atualização. O RDS para Oracle descarta e recria as visualizações RDS_X$ nas tabelas X$ durante uma atualização do mecanismo. Depois, ele concede o privilégio `SELECT ... WITH GRANT OPTION` ao usuário principal. Depois de uma atualização, conceda privilégios aos usuários do banco de dados conforme necessário nas visualizações RDS_X$ correspondentes.

Listar visualizações SYS.RDS_X$

Para listar as visualizações RDS_X$ existentes, use o procedimento `rdsadmin.rdsadmin_util.list_created_sys_x$_views` do RDS. O procedimento lista somente as visualizações que foram criadas pelo procedimento `create_sys_x$_view`. O exemplo a seguir lista as tabelas X$ que têm visualizações RDS_X$ correspondentes (exemplo de saída incluído).

```
SQL> SET SERVEROUTPUT ON
SQL> COL XD_TBL_NAME FORMAT A30
SQL> COL STATUS FORMAT A30
SQL> SET LINESIZE 200
SQL> SELECT * FROM TABLE(rdsadmin.rdsadmin_util.list_created_sys_x$_views);

XD_TBL_NAME                    STATUS
------------------------------ ------------------------------
X$BH                           VALID
X$K2GTE                        VALID
X$KCBWBPD                      VALID

3 rows selected.
```

Descartar visualizações RDS_X$

Para descartar uma visualização SYS.RDS_X$, use o procedimento `rdsadmin.rdsadmin_util.drop_sys_x$_view` do RDS. É possível descartar apenas as visualizações na saída de `rdsadmin.rdsadmin_util.list_allowed_sys_x$_views`. O procedimento `drop_sys_x$_view` aceita o parâmetro a seguir.

Nome do parâmetro	Tipo de dados	Padrão	Obrigatório	Descrição
p_x$_tbl	varchar2	Null	Sim	O nome de uma tabela X$ fixa válida. O valor deve ser uma das tabelas fixas X$ relatadas por list_created_sys_x$_views .

O exemplo a seguir cria a visualização RDS_X$KGLOB criada na tabela X$KGLOB.

```
SQL> SET SERVEROUTPUT ON
SQL> EXEC rdsadmin.rdsadmin_util.drop_sys_x$_view('X$KGLOB');

PL/SQL procedure successfully completed.
```

O exemplo a seguir mostra que a visualização SYS.RDS_X$KGLOB foi descartada (exemplo de saída incluído).

```
SQL> SET SERVEROUTPUT ON
SQL> COL OWNER FORMAT A30
SQL> COL OBJECT_NAME FORMAT A30
SQL> COL STATUS FORMAT A30
SQL> SET LINESIZE 200
SQL> SELECT OWNER, OBJECT_NAME, STATUS
FROM DBA_OBJECTS
WHERE OWNER = 'SYS' AND OBJECT_NAME = 'RDS_X$KGLOB';

no rows selected
```

Concessão de privilégios a usuários não mestres

Você pode conceder privilégios de seleção para muitos objetos no esquema SYS, usando a função SELECT_CATALOG_ROLE. A função SELECT_CATALOG_ROLE concede aos usuários privilégios SELECT em visualizações de dicionários de dados. O exemplo a seguir concede a função SELECT_CATALOG_ROLE a um usuário chamado user1.

```
GRANT SELECT_CATALOG_ROLE TO user1;
```

É possível conceder privilégios EXECUTE para muitos objetos no esquema SYS usando a função EXECUTE_CATALOG_ROLE. A função EXECUTE_CATALOG_ROLE concede aos usuários privilégios EXECUTE para pacotes e procedimentos no dicionário de dados. O exemplo a seguir concede a função EXECUTE_CATALOG_ROLE a um usuário chamado user1.

```
GRANT EXECUTE_CATALOG_ROLE TO user1;
```

O exemplo a seguir obtém as permissões permitidas pelas funções SELECT_CATALOG_ROLE e EXECUTE_CATALOG_ROLE.

```
  SELECT *
    FROM ROLE_TAB_PRIVS
   WHERE ROLE IN ('SELECT_CATALOG_ROLE','EXECUTE_CATALOG_ROLE')
ORDER BY ROLE, TABLE_NAME ASC;
```

O exemplo a seguir cria um usuário não mestre chamado user1, concede o privilégio CREATE SESSION e concede o privilégio SELECT em um banco de dados chamado sh.sales.

```
CREATE USER user1 IDENTIFIED BY PASSWORD;
GRANT CREATE SESSION TO user1;
GRANT SELECT ON sh.sales TO user1;
```

Criação de funções personalizadas para verificar senhas

Você pode criar uma função de verificação de senha personalizada das seguintes maneiras:

- Para usar a lógica de verificação padrão e armazenar a função no esquema SYS, utilize o procedimento create_verify_function.
- Para usar a lógica de verificação padrão ou evitar armazenar a função no esquema SYS, utilize o procedimento create_passthrough_verify_fcn.

O procedimento create_verify_function

Você pode criar uma função personalizada para verificar senhas usando o procedimento do Amazon RDS rdsadmin.rdsadmin_password_verify.create_verify_function. O procedimento create_verify_function é compatível com todas as versões do RDS para Oracle.

O procedimento `create_verify_function` tem os seguintes parâmetros.

Nome do parâmetro	Tipo de dados	Padrão	Obrigatório	Descrição
`p_verify_function_name`	varchar2	—	Sim	O nome da sua função personalizada. Essa função é criada por você no esquema SYS. Você atribui essa função a perfis de usuário.
`p_min_length`	número	8	Não	O número mínimo de caracteres necessários.
`p_max_length`	número	256	Não	O número máximo de caracteres permitidos.
`p_min_letters`	número	1	Não	O número mínimo de letras necessárias.
`p_min_uppercase`	número	0	Não	O número mínimo de letras maiúsculas necessárias.
`p_min_lowercase`	número	0	Não	O número mínimo de letras minúsculas necessárias.
`p_min_digits`	número	1	Não	O número mínimo de dígitos necessários.
`p_min_special`	número	0	Não	O número mínimo de caracteres especiais necessários.
`p_min_different_chars`	número	3	Não	O número mínimo de caracteres diferentes

Nome do parâmetro	Tipo de dados	Padrão	Obrigatório	Descrição
				exigido entre a senha antiga e nova.
p_disallow_username	booliano	true	Não	Defina como true para não permitir o nome de usuário na senha.
p_disallow_reverse	booliano	true	Não	Defina como true para não permitir o inverso do nome de usuário na senha.
p_disallow_db_name	booliano	true	Não	Defina como true para não permitir o nome do banco de dados ou do servidor na senha.
p_disallow_simple_strings	booliano	true	Não	Defina como true para não permitir strings simples como senha.
p_disallow_whitespace	booliano	false	Não	Defina como true para não permitir caracteres de espaço em branco na senha.
p_disallow_at_sign	booliano	false	Não	Defina como true para não permitir o caractere @ na senha.

Você pode criar várias funções de verificação de senha.

Existem restrições quanto ao nome da sua função personalizada. Sua função personalizada não pode ter o mesmo nome que um objeto de sistema existente. O nome pode ter no máximo 30

caracteres. Além disso, o nome deve incluir uma das seguintes strings: PASSWORD, VERIFY, COMPLEXITY, ENFORCE ou STRENGTH.

O exemplo a seguir cria uma função chamada CUSTOM_PASSWORD_FUNCTION. A função requer que uma senha tenha pelo menos 12 caracteres, 2 caracteres em maiúsculas, 1 dígito e 1 caractere especial e que a senha não permita o caractere @.

```
begin
    rdsadmin.rdsadmin_password_verify.create_verify_function(
        p_verify_function_name => 'CUSTOM_PASSWORD_FUNCTION',
        p_min_length           => 12,
        p_min_uppercase        => 2,
        p_min_digits           => 1,
        p_min_special          => 1,
        p_disallow_at_sign     => true);
end;
/
```

Para ver o texto da sua função de verificação, consulte DBA_SOURCE. O exemplo a seguir obtém o texto de uma função de senha personalizada chamada CUSTOM_PASSWORD_FUNCTION.

```
COL TEXT FORMAT a150

  SELECT TEXT
    FROM DBA_SOURCE
   WHERE OWNER = 'SYS'
     AND NAME = 'CUSTOM_PASSWORD_FUNCTION'
ORDER BY LINE;
```

Para associar sua função de verificação a um perfil de usuário, use `alter profile`. O seguinte exemplo associa uma função de verificação ao perfil de usuário DEFAULT.

```
ALTER PROFILE DEFAULT LIMIT PASSWORD_VERIFY_FUNCTION CUSTOM_PASSWORD_FUNCTION;
```

Para ver quais perfis de usuários estão associados a quais funções de verificação, consulte DBA_PROFILES. O exemplo a seguir obtém os perfis associados à função de verificação personalizada chamada CUSTOM_PASSWORD_FUNCTION.

```
SELECT * FROM DBA_PROFILES WHERE RESOURCE_NAME = 'PASSWORD' AND LIMIT =
 'CUSTOM_PASSWORD_FUNCTION';
```

```
PROFILE                    RESOURCE_NAME                    RESOURCE  LIMIT
-------------------------  -------------------------------  --------
 ------------------------
DEFAULT                    PASSWORD_VERIFY_FUNCTION         PASSWORD
 CUSTOM_PASSWORD_FUNCTION
```

O exemplo a seguir obtém todos os perfis e as funções de verificação de senha às quais eles estão associados.

```
SELECT * FROM DBA_PROFILES WHERE RESOURCE_NAME = 'PASSWORD_VERIFY_FUNCTION';

PROFILE                    RESOURCE_NAME                    RESOURCE  LIMIT
-------------------------  -------------------------------  --------
 ------------------------
DEFAULT                    PASSWORD_VERIFY_FUNCTION         PASSWORD
 CUSTOM_PASSWORD_FUNCTION
RDSADMIN                   PASSWORD_VERIFY_FUNCTION         PASSWORD  NULL
```

O procedimento create_passthrough_verify_fcn

O procedimento `create_passthrough_verify_fcn` é compatível com todas as versões do RDS para Oracle.

Você pode criar uma função personalizada para verificar senhas usando o procedimento do Amazon RDS `rdsadmin.rdsadmin_password_verify.create_passthrough_verify_fcn`. O procedimento `create_passthrough_verify_fcn` tem os seguintes parâmetros.

Nome do parâmetro	Tipo de dados	Padrão	Obrigatório	Descrição
`p_verify_function_name`	varchar2	—	Sim	O nome da sua função de verificação personalizada. Esta é uma função de wrapper criada por você no esquema SYS e não contém lógica de verificação. Você atribui

Nome do parâmetro	Tipo de dados	Padrão	Obrigatório	Descrição
				essa função a perfis de usuário.
`p_target_owner`	varchar2	—	Sim	O proprietário do esquema da sua função de verificação personalizada.
`p_target_function_name`	varchar2	—	Sim	O nome da sua função personalizada existente que contém a lógica de verificação. Sua função personalizada deve retornar um valor booliano. Sua função deverá retornar `true` se a senha é válida e `false` se a senha for inválida.

O exemplo a seguir cria uma função de verificação de senha que usa a lógica da função chamada `PASSWORD_LOGIC_EXTRA_STRONG`.

```
begin
    rdsadmin.rdsadmin_password_verify.create_passthrough_verify_fcn(
        p_verify_function_name => 'CUSTOM_PASSWORD_FUNCTION',
        p_target_owner         => 'TEST_USER',
        p_target_function_name => 'PASSWORD_LOGIC_EXTRA_STRONG');
end;
/
```

Para associar a função de verificação a um perfil de usuário, use `alter profile`. O seguinte exemplo associa a função de verificação ao perfil de usuário `DEFAULT`.

```
ALTER PROFILE DEFAULT LIMIT PASSWORD_VERIFY_FUNCTION CUSTOM_PASSWORD_FUNCTION;
```

Configuração de um servidor DNS personalizado

O Amazon RDS oferece suporte para acesso à rede de saída nas suas instâncias de bancos de dados que executam o Oracle. Para obter mais informações sobre acesso à rede de saída, inclusive pré-requisitos, consulte Configurar o acesso UTL_HTTP usando certificados e uma carteira Oracle.

O Amazon RDS Oracle permite a resolução do Serviço de Nomes de Domínio (DNS) de um servidor DNS personalizado que pertence ao cliente. Você pode resolver apenas nomes de domínio totalmente qualificados a partir da sua instância de banco de dados do Amazon RDS por meio do seu servidor DNS personalizado.

Após a configuração do seu servidor de nomes DNS personalizado, demora até 30 minutos para propagar as alterações na sua instância de banco de dados. Depois que as alterações são propagadas para sua a instância de banco de dados, todo o tráfego de rede de saída que requer uma pesquisa de DNS consulta o servidor DNS por meio da porta 53.

Para configurar um servidor DNS personalizado para a instância de banco de dados do Amazon RDS for Oracle, faça o seguinte:

- Nas opções de DHCP anexadas à nuvem privada virtual (VPC), defina a opção `domain-name-servers` como o endereço IP do seu servidor de nomes DNS. Para obter mais informações, consulte Conjuntos de opções de DHCP.

  > **Note**
  >
  > A opção `domain-name-servers` aceita até quatro valores, mas sua instância de banco de dados do Amazon RDS usa apenas o primeiro valor.

- Certifique-se de que seu servidor DNS possa resolver todas as consultas de pesquisa, incluindo nomes DNS públicos, nomes DNS privados do Amazon EC2 e nomes DNS específicos do cliente. Se o tráfego de rede de saída contiver pesquisas de DNS que o seu servidor DNS não pode manipular, este deverá ter provedores de DNS upstream configurados.
- Configure seu servidor DNS para produzir respostas de protocolo UDP de 512 bytes ou menos.
- Configure seu servidor DNS para produzir respostas de protocolo TCP de 1024 bytes ou menos.
- Configure seu servidor DNS para permitir tráfego de entrada das suas instâncias de banco de dados do Amazon RDS pela porta 53. Se o seu servidor DNS estiver em uma Amazon VPC, ela deverá ter um grupo de segurança que contenha regras de entrada que permitam tráfego UDP e TCP na porta 53. Se o seu servidor DNS não estiver em uma Amazon VPC, ele deverá ter uma

lista de permissões de firewall apropriada para permitir o tráfego de entrada UDP e TCP na porta 53.

Para obter mais informações, consulte Grupos de segurança para sua VPC e Adição e remoção de regras.

- Configurar a VPC da sua instância de banco de dados do Amazon RDS para permitir o tráfego de saída por meio da porta 53. Sua VPC deve ter um security group que contém regras de saída que permitem o tráfego TCP e UDP na porta 53.

  Para obter mais informações, consulte Grupos de segurança para sua VPC e Adição e remoção de regras.
- O caminho de roteamento entre a instância de banco de dados do Amazon RDS e o servidor DNS deve ser configurado corretamente para permitir o tráfego DNS.
  - Se a instância de banco de dados do Amazon RDS e o servidor DNS não estiverem na mesma VPC, uma conexão de emparelhamento precisará ser configurada entre eles. Para obter mais informações, consulte O que é emparelhamento de VPC?

Configurar e anular configuração de eventos de diagnóstico do sistema

Para configurar e anular a configuração de eventos de diagnóstico no nível da sessão, você pode usar a instrução Oracle SQL `ALTER SESSION SET EVENTS`. No entanto, para definir eventos no nível do sistema, você não pode usar o Oracle SQL. Em vez disso, use os procedimentos de evento do sistema no pacote `rdsadmin.rdsadmin_util`. Os procedimentos de eventos do sistema estão disponíveis nas seguintes versões do mecanismo:

- Todas as versões do Oracle Database 21c
- 19.0.0.0.ru-2020-10.rur-2020-10.r1 e versões posteriores do Oracle Database 19c

  Consulte mais informações em Version 19.0.0.0.ru-2020-10.rur-2020-10.r1 em Notas de lançamento do Amazon RDS para Oracle.

> **Important**
>
> Internamente, o pacote `rdsadmin.rdsadmin_util` define eventos usando a instrução `ALTER SYSTEM SET EVENTS`. Essa instrução `ALTER SYSTEM` não está incluída na documentação do Oracle Database. Alguns eventos de diagnóstico do sistema podem gerar grandes quantidades de informações de monitoramento, causar contenção ou afetar a

disponibilidade do banco de dados. Recomendamos que você teste eventos de diagnóstico específicos em seu banco de dados de não produção e defina apenas eventos em seu banco de dados de produção sob orientação do Oracle Support.

Listar eventos de diagnóstico do sistema permitidos

Para listar os eventos do sistema que você pode definir, use o procedimento `rdsadmin.rdsadmin_util.list_allowed_system_events` do Amazon RDS. Este procedimento não aceita parâmetros.

O exemplo a seguir lista todos os eventos do sistema que você pode definir.

```
SET SERVEROUTPUT ON
EXEC rdsadmin.rdsadmin_util.list_allowed_system_events;
```

A saída de exemplo a seguir lista os números de eventos e suas descrições. Use os procedimentos `set_system_event` do Amazon RDS para definir esses eventos e `unset_system_event` para desconfigurá-los.

```
604   - error occurred at recursive SQL level
942   - table or view does not exist
1401  - inserted value too large for column
1403  - no data found
1410  - invalid ROWID
1422  - exact fetch returns more than requested number of rows
1426  - numeric overflow
1427  - single-row subquery returns more than one row
1476  - divisor is equal to zero
1483  - invalid length for DATE or NUMBER bind variable
1489  - result of string concatenation is too long
1652  - unable to extend temp segment by  in tablespace
1858  - a non-numeric character was found where a numeric was expected
4031  - unable to allocate  bytes of shared memory ("","","","")
6502  - PL/SQL: numeric or value error
10027 - Specify Deadlock Trace Information to be Dumped
10046 - enable SQL statement timing
10053 - CBO Enable optimizer trace
10173 - Dynamic Sampling time-out error
10442 - enable trace of kst for ORA-01555 diagnostics
12008 - error in materialized view refresh path
```

```
12012 - error on auto execute of job
12504 - TNS:listener was not given the SERVICE_NAME in CONNECT_DATA
14400 - inserted partition key does not map to any partition
31693 - Table data object  failed to load/unload and is being skipped due to error:
```

> **Note**
>
> A lista dos eventos do sistema permitidos pode mudar ao longo do tempo. Para se certificar de que você tem a lista mais recente de eventos qualificados, use `rdsadmin.rdsadmin_util.list_allowed_system_events`.

Definir eventos de diagnóstico do sistema

Para definir um evento do sistema, use o procedimento `rdsadmin.rdsadmin_util.set_system_event` do Amazon RDS. Você só pode definir eventos listados na saída do `rdsadmin.rdsadmin_util.list_allowed_system_events`. O procedimento `set_system_event` tem os parâmetros a seguir.

Nome do parâmetro	Tipo de dados	Padrão	Obrigatório	Descrição
`p_event`	número	—	Sim	O número do evento do sistema. O valor deve ser um dos números de eventos relatados por `list_allowed_system_events` .
`p_level`	número	—	Sim	O nível do evento. Consulte a documentação do Oracle Database ou o Oracle Support para obter descrições de diferentes valores de nível.

O procedimento `set_system_event` constrói e executa as instruções `ALTER SYSTEM SET EVENTS` necessárias de acordo com os seguintes princípios:

- O tipo de evento (`context` ou `errorstack`) é determinado automaticamente.
- Uma instrução no formulário `ALTER SYSTEM SET EVENTS 'event LEVEL event_level'` define os eventos de contexto. Esta notação é equivalente a `ALTER SYSTEM SET EVENTS 'event TRACE NAME CONTEXT FOREVER, LEVEL event_level'`.
- Uma instrução no formulário `ALTER SYSTEM SET EVENTS 'event ERRORSTACK (event_level)'` define os eventos de pilha de erros. Esta notação é equivalente a `ALTER SYSTEM SET EVENTS 'event TRACE NAME ERRORSTACK LEVEL event_level'`.

O exemplo a seguir define o evento 942 no nível 3 e o evento 10442 no nível 10. A amostra de saída está incluída.

```
SQL> SET SERVEROUTPUT ON
SQL> EXEC rdsadmin.rdsadmin_util.set_system_event(942,3);
Setting system event 942 with: alter system set events '942 errorstack (3)'

PL/SQL procedure successfully completed.

SQL> EXEC rdsadmin.rdsadmin_util.set_system_event(10442,10);
Setting system event 10442 with: alter system set events '10442 level 10'

PL/SQL procedure successfully completed.
```

Listar eventos de diagnóstico do sistema definidos

Para listar os eventos do sistema que estão definidos no momento, use o procedimento `rdsadmin.rdsadmin_util.list_set_system_events` do Amazon RDS. Este procedimento relata apenas os eventos definidos no nível do sistema por `set_system_event`.

O exemplo a seguir lista os eventos ativos do sistema.

```
SET SERVEROUTPUT ON
EXEC rdsadmin.rdsadmin_util.list_set_system_events;
```

A saída de exemplo a seguir mostra a lista de eventos, o tipo de evento, o nível no qual os eventos estão definidos no momento e a hora em que o evento foi definido.

```
942 errorstack (3) - set at 2020-11-03 11:42:27
10442 level 10 - set at 2020-11-03 11:42:41

PL/SQL procedure successfully completed.
```

Desconfigurar eventos de diagnóstico do sistema

Para desconfigurar um evento do sistema, use o procedimento `rdsadmin.rdsadmin_util.unset_system_event` do Amazon RDS. Você só pode desconfigurar eventos listados na saída do `rdsadmin.rdsadmin_util.list_allowed_system_events`. O procedimento `unset_system_event` aceita o seguinte parâmetro.

Nome do parâmetro	Tipo de dados	Padrão	Obrigatório	Descrição
`p_event`	número	—	Sim	O número do evento do sistema. O valor deve ser um dos números de eventos relatados por `list_allowed_system_events` .

O exemplo a seguir desconfigura os eventos 942 e 10442. A amostra de saída está incluída.

```
SQL> SET SERVEROUTPUT ON
SQL> EXEC rdsadmin.rdsadmin_util.unset_system_event(942);
Unsetting system event 942 with: alter system set events '942 off'

PL/SQL procedure successfully completed.

SQL> EXEC rdsadmin.rdsadmin_util.unset_system_event(10442);
Unsetting system event 10442 with: alter system set events '10442 off'

PL/SQL procedure successfully completed.
```

Realização de tarefas comuns de banco de dados para instâncias de banco de dados

Veja a seguir como executar determinadas tarefas comuns de DBA relacionadas aos bancos de dados nas instâncias de bancos de dados do Amazon RDS que executam o Oracle. Para oferecer uma experiência de serviço gerenciada, o Amazon RDS não fornece acesso ao shell para as instâncias de banco de dados. O Amazon RDS também restringe o acesso a alguns procedimentos e tabelas do sistema que exigem privilégios avançados.

Tópicos

- Alteração do nome global de um banco de dados
- Usar espaços de tabela
- Usar arquivos temporários
- Verificação de um banco de dados
- Definição da recuperação distribuída
- Definição do fuso horário do banco de dados
- Trabalhar com tabelas externas do Oracle
- Gerar relatórios de performance com o AWR (Automatic Workload Repository)
- Ajuste de vínculos de banco de dados para uso com instâncias de bancos de dados em uma VPC
- Definir a edição padrão para uma instância de banco de dados
- Habilitar a auditoria para a tabela SYS.AUD$
- Desabilitar a auditoria para a tabela SYS.AUD$
- Limpar criações de índice online interrompidas
- Ignorar blocos corrompidos
- Redimensionar espaços de tabela, arquivos de dados e arquivos temporários
- Limpar a lixeira
- Definir os valores padrão exibidos para a redação completa

Alteração do nome global de um banco de dados

Para alterar o nome global de um banco de dados, use o procedimento `rdsadmin.rdsadmin_util.rename_global_name` do Amazon RDS. O procedimento `rename_global_name` tem os seguintes parâmetros.

Nome do parâmetro	Tipo de dados	Padrão	Obrigatório	Descrição
p_new_global_name	varchar2	—	Sim	O novo nome global do banco de dados.

O banco de dados deve estar aberto para que a alteração de nome ocorra. Para ter mais informações sobre como alterar o nome global de um banco de dados, consulte ALTER DATABASE na documentação do Oracle.

O exemplo a seguir altera o nome global de um banco de dados para new_global_name.

```
EXEC rdsadmin.rdsadmin_util.rename_global_name(p_new_global_name => 'new_global_name');
```

Usar espaços de tabela

É possível usar espaços de tabela com o RDS para Oracle, que é uma unidade lógica de armazenamento e armazena dados de banco de dados.

Tópicos

- Criação e uso de espaços de tabela
- Definição do espaço de tabela padrão
- Definição do espaço de tabela temporário padrão
- Criar um espaço de tabela temporário no armazenamento de instância

Criação e uso de espaços de tabela

O Amazon RDS só oferece suporte a arquivos gerenciados Oracle (OMF) para arquivos de dados, arquivos de log e arquivos de controle. Ai criar arquivos de dados e arquivos de log, você não pode especificar os nomes dos arquivos físicos.

Por padrão, se você não especificar o tamanho de um arquivo de dados, os espaços de tabela serão criados com um tamanho padrão de AUTOEXTEND ON e sem um tamanho máximo. No exemplo a seguir, o tablespace *users1* é autoextensível.

```
CREATE TABLESPACE users1;
```

Devido a essas configurações padrão, espaços de tabela podem crescer ao ponto de consumirem todo o armazenamento alocado. Recomendamos que você especifique um tamanho máximo apropriado em espaços de tabela permanentes e temporários e que monitore cuidadosamente o uso do espaço.

O exemplo a seguir cria um espaço de tabela chamado *users2* com um tamanho inicial de 1 gigabyte. Como o tamanho do arquivo de dados é especificado, mas AUTOEXTEND ON não é especificado, o espaço de tabela não é autoextensível.

```
CREATE TABLESPACE users2 DATAFILE SIZE 1G;
```

O exemplo a seguir cria um espaço de tabela chamado *users3* com um tamanho inicial de 1 gigabyte, extensão automática ativada e um tamanho máximo de 10 gigabytes.

```
CREATE TABLESPACE users3 DATAFILE SIZE 1G AUTOEXTEND ON MAXSIZE 10G;
```

O exemplo a seguir cria um espaço de tabela temporário chamado *temp01*:

```
CREATE TEMPORARY TABLESPACE temp01;
```

É possível redimensionar um espaço de tabela de arquivos grandes usando ALTER TABLESPACE. Você pode especificar o tamanho em quilobytes (K), megabytes (M), gigabytes (G) ou terabytes (T). O exemplo a seguir redimensiona um espaço de tabela de arquivos grandes chamado *users_bf* para 200 MB.

```
ALTER TABLESPACE users_bf RESIZE 200M;
```

O exemplo a seguir acrescenta um arquivo de dados adicional a um espaço de tabela de arquivos pequenos chamado *users_sf*.

```
ALTER TABLESPACE users_sf ADD DATAFILE SIZE 100000M AUTOEXTEND ON NEXT 250m
 MAXSIZE UNLIMITED;
```

Definição do espaço de tabela padrão

Para definir o espaço padrão tabela padrão, use o procedimento `rdsadmin.rdsadmin_util.alter_default_tablespace` do Amazon RDS. O procedimento `alter_default_tablespace` tem os seguintes parâmetros.

Nome do parâmetro	Tipo de dados	Padrão	Obrigatório	Descrição
`tablespace_name`	varchar	—	Sim	O nome do espaço de tabela padrão.

O exemplo a seguir define o espaço de tabela padrão como *`users2`*:

```
EXEC rdsadmin.rdsadmin_util.alter_default_tablespace(tablespace_name => 'users2');
```

Definição do espaço de tabela temporário padrão

Para definir o espaço de tabela temporário padrão, use o procedimento `rdsadmin.rdsadmin_util.alter_default_temp_tablespace` do Amazon RDS. O procedimento `alter_default_temp_tablespace` tem os seguintes parâmetros.

Nome do parâmetro	Tipo de dados	Padrão	Obrigatório	Descrição
`tablespace_name`	varchar	—	Sim	O nome do espaço de tabela temporário padrão.

O exemplo a seguir define o espaço de tabela temporário padrão como *`temp01`*.

```
EXEC rdsadmin.rdsadmin_util.alter_default_temp_tablespace(tablespace_name => 'temp01');
```

Criar um espaço de tabela temporário no armazenamento de instância

Para criar um espaço de tabela temporário no armazenamento de instância, use o procedimento `rdsadmin.rdsadmin_util.create_inst_store_tmp_tblspace` do Amazon RDS. O procedimento `create_inst_store_tmp_tblspace` tem os seguintes parâmetros.

Nome do parâmetro	Tipo de dados	Padrão	Obrigatório	Descrição
p_tablespace_name	varchar	—	Sim	O nome do espaço de tabela temporário.

O exemplo a seguir cria o espaço de tabela temporário *temp01* no armazenamento de instância.

```
EXEC rdsadmin.rdsadmin_util.create_inst_store_tmp_tblspace(p_tablespace_name =>
 'temp01');
```

> **Important**
>
> Quando você executa `rdsadmin_util.create_inst_store_tmp_tblspace`, o espaço de tabela temporário recém-criado não é automaticamente definido como o espaço de tabela temporário padrão. Para defini-lo como padrão, consulte Definição do espaço de tabela temporário padrão.

Para ter mais informações, consulte Armazenar dados temporários em um armazenamento de instância do RDS for Oracle.

Usar arquivos temporários

Adicionar um arquivo temporário ao armazenamento de instância em uma réplica de leitura

Quando você cria um espaço de tabela temporário em uma instância de banco de dados principal, a réplica de leitura não cria arquivos temporários. Suponha que exista um espaço de tabela temporário vazio em sua réplica de leitura por um dos seguintes motivos:

- Você removeu um arquivo temporário do espaço de tabela em sua réplica de leitura. Para ter mais informações, consulte Descartar arquivos temporários em uma réplica de leitura.
- Você criou um espaço de tabela temporário na instância de banco de dados primária. Nesse caso, o RDS para Oracle sincroniza os metadados com a réplica de leitura.

Você pode adicionar um arquivo temporário ao espaço de tabela temporário vazio e armazená-lo no armazenamento de instância. Para criar um arquivo temporário no armazenamento de instância,

use o procedimento `rdsadmin.rdsadmin_util.add_inst_store_tempfile` do Amazon RDS. Você pode usar esse procedimento somente em uma réplica de leitura. O procedimento tem os seguintes parâmetros.

Nome do parâmetro	Tipo de dados	Padrão	Obrigatório	Descrição
`p_tablespace_name`	varchar	—	Sim	O nome do espaço da tabela temporário em sua réplica de leitura.

No exemplo a seguir, o espaço de tabela temporário vazio *`temp01`* existe na réplica de leitura. Execute o comando a seguir para criar um arquivo temporário para esse espaço de tabela e armazená-lo no armazenamento de instância.

```
EXEC rdsadmin.rdsadmin_util.add_inst_store_tempfile(p_tablespace_name => 'temp01');
```

Para ter mais informações, consulte Armazenar dados temporários em um armazenamento de instância do RDS for Oracle.

Descartar arquivos temporários em uma réplica de leitura

Não é possível descartar um espaço de tabela temporário em uma réplica de leitura. Você pode alterar o armazenamento de arquivos temporários em uma réplica de leitura do Amazon EBS para o armazenamento de instância ou do armazenamento de instância para o Amazon EBS. Para conseguir isso, faça o seguinte:

1. Descarte os arquivos temporários atuais no espaço de tabela temporário na réplica de leitura.
2. Crie arquivos temporários em outro armazenamento.

Para descartar os arquivos temporários, use o procedimento `rdsadmin.rdsadmin_util.drop_replica_tempfiles` do Amazon RDS. Você pode usar esse procedimento somente em réplicas de leitura. O procedimento `drop_replica_tempfiles` tem os seguintes parâmetros.

Nome do parâmetro	Tipo de dados	Padrão	Obrigatório	Descrição
p_tablespace_name	varchar	—	Sim	O nome do espaço da tabela temporário em sua réplica de leitura.

Suponha que um espaço de tabela temporário chamado *temp01* resida no armazenamento de instância da réplica de leitura. Descarte todos os arquivos temporários nesse espaço de tabela executando o comando a seguir.

```
EXEC rdsadmin.rdsadmin_util.drop_replica_tempfiles(p_tablespace_name => 'temp01');
```

Para ter mais informações, consulte Armazenar dados temporários em um armazenamento de instância do RDS for Oracle.

Verificação de um banco de dados

Para verificar um banco de dados, use o procedimento `rdsadmin.rdsadmin_util.checkpoint` do Amazon RDS. O procedimento `checkpoint` não tem parâmetros.

O exemplo a seguir verifica o banco de dados.

```
EXEC rdsadmin.rdsadmin_util.checkpoint;
```

Definição da recuperação distribuída

Para definir a recuperação distribuída, use os procedimentos `rdsadmin.rdsadmin_util.enable_distr_recovery` e `disable_distr_recovery` do Amazon RDS. Os procedimentos não têm parâmetros.

O exemplo a seguir habilita a recuperação distribuída.

```
EXEC rdsadmin.rdsadmin_util.enable_distr_recovery;
```

O exemplo a seguir desabilita a recuperação distribuída.

```
EXEC rdsadmin.rdsadmin_util.disable_distr_recovery;
```

Definição do fuso horário do banco de dados

Você pode definir o fuso horário do banco de dados Oracle do Amazon RDS das seguintes maneiras:

- A opção `Timezone`

  A opção `Timezone` altera o fuso horário no nível do host e afeta todas as colunas e valores de data, como `SYSDATE`. Para ter mais informações, consulte Fuso horário da Oracle.
- O procedimento do Amazon RDS `rdsadmin.rdsadmin_util.alter_db_time_zone`

  O procedimento `alter_db_time_zone` altera o fuso horário para apenas alguns tipos de dados e não altera `SYSDATE`. Existem restrições adicionais na definição do fuso horário, listadas na documentação do Oracle.

> **Note**
>
> Você também pode definir o fuso horário padrão para o Oracle Scheduler. Para ter mais informações, consulte Definir o fuso horário para trabalhos do Oracle Scheduler.

O procedimento `alter_db_time_zone` tem os seguintes parâmetros.

Nome do parâmetro	Tipo de dados	Padrão	Obrigatório	Descrição
`p_new_tz`	varchar2	—	Sim	O novo fuso horário como uma região nomeada ou um deslocamento absoluto do Tempo Universal Coordenado (UTC). Os deslocamentos válidos variam de -12:00 a +14:00.

O exemplo a seguir altera o fuso horário para UTC mais 3 horas.

```
EXEC rdsadmin.rdsadmin_util.alter_db_time_zone(p_new_tz => '+3:00');
```

O exemplo a seguir altera o fuso horário para o fuso horário de África/Algiers.

```
EXEC rdsadmin.rdsadmin_util.alter_db_time_zone(p_new_tz => 'Africa/Algiers');
```

Depois de alterar o fuso horário usando o procedimento `alter_db_time_zone`, reinicie a instância de banco de dados para que a alteração entre em vigor. Para ter mais informações, consulte Reinicializar uma instância de banco de dados . Para obter informações sobre como atualizar fusos horários, consulte Considerações sobre fuso horário.

Trabalhar com tabelas externas do Oracle

Tabelas externas do Oracle são tabelas com dados que não estão no banco de dados. Em vez disso, os dados estão em arquivos externos que o banco de dados pode acessar. Usando tabelas externas, você pode acessar dados sem carregá-los no banco de dados. Para ter mais informações sobre tabelas externas, consulte Managing External Tables (Gerenciar tabelas externas) na documentação do Oracle.

Com o Amazon RDS, você pode armazenar arquivos de tabela externos em objetos de diretório. Você pode criar um objeto de diretório ou pode usar um objeto predefinido no banco de dados Oracle, como o diretório DATA_PUMP_DIR. Para obter informações sobre como criar os objetos de diretório, consulte Criar e eliminar diretórios no espaço de armazenamento de dados principal. Você pode consultar a visualização ALL_DIRECTORIES a fim de listar os objetos de diretório para a instância de banco de dados Oracle do Amazon RDS.

> **Note**
>
> Os objetos de diretório apontam para o espaço de armazenamento físico de dados principal (volume do Amazon EBS) usado pela instância. O espaço usado, juntamente com arquivos de dados, logs redo, auditoria, rastreamento e outros arquivos, conta em relação ao armazenamento alocado.

Você pode mover um arquivo de dados externo de um banco de dados Oracle para outro usando o pacote DBMS_FILE_TRANSFER ou o pacote UTL_FILE. O arquivo de dados externo é movido de um diretório no banco de dados de origem para o diretório especificado no banco de dados de

destino. Para informações sobre como usar DBMS_FILE_TRANSFER, consulte Importar usando o Oracle Data Pump.

Depois de mover o arquivo de dados externo, você poderá criar uma tabela externa com ele. O seguinte exemplo cria uma tabela externa que usa o arquivo `emp_xt_file1.txt` no diretório USER_DIR1.

```
CREATE TABLE emp_xt (
  emp_id      NUMBER,
  first_name  VARCHAR2(50),
  last_name   VARCHAR2(50),
  user_name   VARCHAR2(20)
)
ORGANIZATION EXTERNAL (
  TYPE ORACLE_LOADER
  DEFAULT DIRECTORY USER_DIR1
  ACCESS PARAMETERS (
    RECORDS DELIMITED BY NEWLINE
    FIELDS TERMINATED BY ','
    MISSING FIELD VALUES ARE NULL
    (emp_id,first_name,last_name,user_name)
  )
  LOCATION ('emp_xt_file1.txt')
)
PARALLEL
REJECT LIMIT UNLIMITED;
```

Suponhamos que você queira migrar dados que estejam em uma instância de banco de dados Oracle do Amazon RDS para um arquivo de dados externo. Nesse caso, você pode preencher o arquivo de dados externo criando uma tabela externa e selecionando os dados da tabela no banco de dados. Por exemplo, a instrução SQL a seguir cria a tabela externa `orders_xt` consultando a tabela `orders` no banco de dados.

```
CREATE TABLE orders_xt
  ORGANIZATION EXTERNAL
   (
     TYPE ORACLE_DATAPUMP
     DEFAULT DIRECTORY DATA_PUMP_DIR
     LOCATION ('orders_xt.dmp')
   )
   AS SELECT * FROM orders;
```

Neste exemplo, os dados são preenchidos no arquivo `orders_xt.dmp` no diretório DATA_PUMP_DIR.

Gerar relatórios de performance com o AWR (Automatic Workload Repository)

Para reunir dados de performance e gerar relatórios, a Oracle recomenda o AWR (Automatic Workload Repository). O AWR requer o Oracle Database Enterprise Edition e uma licença para os pacotes de Diagnóstico e Ajuste. Para ativar o AWR, defina o parâmetro de inicialização `CONTROL_MANAGEMENT_PACK_ACCESS` como `DIAGNOSTIC` ou `DIAGNOSTIC+TUNING`.

Trabalhar com relatórios do AWR no RDS

Para gerar relatórios do AWR, é possível executar scripts como `awrrpt.sql`. Esses scripts são instalados no servidor host do banco de dados. No Amazon RDS, não é possível ter acesso direto ao host. No entanto, é possível obter cópias de scripts SQL de outra instalação do Oracle Database.

Também é possível usar o AWR executando procedimentos no pacote PL/SQL `SYS.DBMS_WORKLOAD_REPOSITORY`. É possível usar esse pacote para gerenciar linhas de base e snapshots e também para exibir relatórios do ASH e do AWR. Por exemplo, para gerar um relatório do AWR em formato de texto, execute o procedimento `DBMS_WORKLOAD_REPOSITORY.AWR_REPORT_TEXT`. No entanto, não é possível acessar esses relatórios da AWR a partir do AWS Management Console.

Ao trabalhar com o AWR, recomendamos usar os procedimentos `rdsadmin.rdsadmin_diagnostic_util`. É possível usar esses procedimentos para gerar o seguinte:

- Relatórios do AWR
- Relatórios do ASH (Active Session History)
- Relatórios do ADDM (Automatic Database Diagnostic Monitor)
- Arquivos de despejo do Oracle Data Pump Export de dados do AWR

Os procedimentos `rdsadmin_diagnostic_util` salvam os relatórios no sistema de arquivos da instância de banco de dados. É possível acessar esses relatórios no console. Também é possível acessar relatórios usando os procedimentos `rdsadmin.rds_file_util` e acessar relatórios copiados para o Amazon S3 usando a opção Integração do S3. Para ter mais informações, consulte Leitura de arquivos no diretório de uma instância de banco de dados e Integração do Amazon S3.

É possível usar os procedimentos `rdsadmin_diagnostic_util` nas seguintes versões de mecanismo de banco de dados do Amazon RDS for Oracle:

- Todas as versões do Oracle Database 21c
- 19.0.0.0.ru-2020-04.rur-2020-04.r1 e versões posteriores do Oracle Database 19c

Para conferir um blog que explica como trabalhar com relatórios de diagnóstico em cenários de replicação, consulte Gerar relatórios do AWR para réplicas de leitura do Amazon RDS para Oracle.

Parâmetros comuns para o pacote do utilitário de diagnóstico

Normalmente, você usa os seguintes parâmetros ao gerenciar o AWR e o ADDM com o pacote `rdsadmin_diagnostic_util`.

Parâmetro	Tipo de dados	Padrã	Obriga io	Descrição
`begin_snap_id`	`NUMBEI`	—	Sim	O ID do snapshot inicial.
`end_snap_id`	`NUMBEI`	—	Sim	O ID do snapshot final.
`dump_directory`	`VARCHA`	`BDUMI`	Não	O diretório no qual gravar o relatório ou para o qual exportar o arquivo. Se você especificar um diretório não padrão, o usuário que executa os procedimentos `rdsadmin_diagnostic_util` deverá ter permissões de gravação para o diretório.
`p_tag`	`VARCHA`	—	Não	Uma string que pode ser usada para distinguir entre backups para indicar a finalidade ou o uso de backups, como `incremental` ou `daily`. Você pode especificar até 30 caracteres. Os caracteres válidos são `a-z`, `A-Z`, `0-9`, um sublinhado (_), um traço (-), e um ponto (.). A etiqueta não diferencia maiúsculas de minúsculas. O RMAN sempre armazena etiquetas em maiúsculas, independentemente de como são inseridas.

Parâmetro	Tipo de dados	Padrã	Obriga io	Descrição
				As etiquetas não precisam ser exclusivas, portanto, vários backups podem ter a mesma etiqueta. Se você não especificar uma etiqueta, o RMAN atribuirá uma etiqueta padrão automaticamente usando o formato `TAG`*`YYYYMMDDTHHMMSS`* , em que *`YYYY`* é o ano, *`MM`* é o mês, *`DD`* é o dia, *`HH`* é a hora (no formato de 24 horas), *`MM`* são os minutos e *`SS`* os segundos. A data e a hora indicam quando o RMAN iniciou o backup. Por exemplo, um backup com a etiqueta padrão `TAG20190927T214517` indica um backup iniciado em 27-09-2019 às 21:45:17. O parâmetro `p_tag` é compatível com as seguintes versões de mecanismo de banco de dados Amazon RDS for Oracle: • Oracle Database 21c (21.0.0) • Oracle Database 19c (19.0.0) usando 19.0.0.0.ru-2021-10.rur-2021-10.r1 e posteriores
`report_type`	`VARCHA`	`HTML`	Não	O formato do relatório. Os valores válidos são `TEXT` e `HTML`.
`dbid`	`NUMBEI`	—	Não	Um identificador de banco de dados (DBID) válido mostrado na visualização `DBA_HIST_DATABASE_INSTANCE` do Oracle. Se esse parâmetro não for especificado, o RDS usará o DBID atual, que é mostrado na visualização `V$DATABASE.DBID` .

Normalmente, você usa os seguintes parâmetros ao gerenciar o ASH com o pacote rdsadmin_diagnostic_util.

Parâmetro	Tipo de dados	Padrão	Obrigatório	Descrição
`begin_time`	`DATE`	—	Sim	A hora de início da análise do ASH.
`end_time`	`DATE`	—	Sim	A hora final da análise do ASH.
`slot_width`	`NUMBER`	`0`	Não	A duração dos slots (em segundos) usados na seção "Atividade superior" do relatório do ASH. Se esse parâmetro não for especificado, o intervalo de tempo entre `begin_time` e `end_time` não usará mais de 10 slots.
`sid`	`NUMBER`	Nulo	Não	O ID da sessão.
`sql_id`	`VARCHAR2`	Nulo	Não	O ID do SQL.
`wait_class`	`VARCHAR2`	Nulo	Não	O nome da classe de espera.
`service_hash`	`NUMBER`	Nulo	Não	O hash do nome do serviço.
`module_name`	`VARCHAR2`	Nulo	Não	O nome do módulo.
`action_name`	`VARCHAR2`	Nulo	Não	O nome da ação.
`client_id`	`VARCHAR2`	Nulo	Não	O ID específico do aplicativo da sessão do banco de dados.
`plsql_entry`	`VARCHAR2`	Nulo	Não	O ponto de entrada PL/SQL.

Gerar um relatório do AWR

Para gerar um relatório do AWR, use o procedimento `rdsadmin.rdsadmin_diagnostic_util.awr_report`.

O exemplo a seguir gera um relatório do AWR para o intervalo de snapshot 101–106. O arquivo de texto de saída é chamado de `awrrpt_101_106.txt`. É possível acessar esse relatório no AWS Management Console.

```
EXEC rdsadmin.rdsadmin_diagnostic_util.awr_report(101,106,'TEXT');
```

O exemplo a seguir gera um relatório HTML para o intervalo de snapshot 63–65. O arquivo HTML de saída é chamado de `awrrpt_63_65.html`. O procedimento grava o relatório no diretório de banco de dados não padrão chamado de `AWR_RPT_DUMP`.

```
EXEC rdsadmin.rdsadmin_diagnostic_util.awr_report(63,65,'HTML','AWR_RPT_DUMP');
```

Extrair dados do AWR em um arquivo de despejo

Para extrair dados do AWR em um arquivo de despejo, use o procedimento `rdsadmin.rdsadmin_diagnostic_util.awr_extract`.

O exemplo a seguir extrai o intervalo de snapshot 101–106. O arquivo de despejo de saída é chamado de `awrextract_101_106.dmp`. É possível acessar esse arquivo por meio do console.

```
EXEC rdsadmin.rdsadmin_diagnostic_util.awr_extract(101,106);
```

O exemplo a seguir extrai o intervalo de snapshot 63–65. O arquivo de despejo de saída é chamado de `awrextract_63_65.dmp`. O arquivo é armazenado no diretório de banco de dados não padrão chamado de `AWR_RPT_DUMP`.

```
EXEC rdsadmin.rdsadmin_diagnostic_util.awr_extract(63,65,'AWR_RPT_DUMP');
```

Gerar um relatório do ADDM

Para gerar um relatório do ADDM, use o procedimento `rdsadmin.rdsadmin_diagnostic_util.addm_report`.

O exemplo a seguir gera um relatório do ADDM para o intervalo de snapshot 101–106. O arquivo de texto de saída é chamado de `addmrpt_101_106.txt`. É possível acessar o relatório por meio do console.

```
EXEC rdsadmin.rdsadmin_diagnostic_util.addm_report(101,106);
```

O exemplo a seguir gera um relatório do ADDM para o intervalo de snapshot 63–65. O arquivo de texto de saída é chamado de `addmrpt_63_65.txt`. O arquivo é armazenado no diretório de banco de dados não padrão chamado de `ADDM_RPT_DUMP`.

```
EXEC rdsadmin.rdsadmin_diagnostic_util.addm_report(63,65,'ADDM_RPT_DUMP');
```

Gerar um relatório do ASH

Para gerar um relatório do ASH, use o procedimento `rdsadmin.rdsadmin_diagnostic_util.ash_report`.

O exemplo a seguir gera um relatório do ASH que inclui os dados de 14 minutos atrás até a hora atual. O nome do arquivo de saída usa o formato `ashrpt`*`begin_time`*`end_time`.`txt`, onde *`begin_time`* e *`end_time`* usam o formato `YYYYMMDDHH24MISS`. É possível acessar o arquivo por meio do console.

```
BEGIN
    rdsadmin.rdsadmin_diagnostic_util.ash_report(
        begin_time     =>     SYSDATE-14/1440,
        end_time       =>     SYSDATE,
        report_type    =>     'TEXT');
END;
/
```

O exemplo a seguir gera um relatório do ASH que inclui os dados de 18 de novembro de 2019 às 18:07 a 18 de novembro de 2019 às 18h15. O nome do relatório HTML de saída é `ashrpt_20190918180700_20190918181500.html`. O relatório é armazenado no diretório de banco de dados não padrão chamado de `AWR_RPT_DUMP`.

```
BEGIN
    rdsadmin.rdsadmin_diagnostic_util.ash_report(
        begin_time     =>    TO_DATE('2019-09-18 18:07:00', 'YYYY-MM-DD HH24:MI:SS'),
        end_time       =>    TO_DATE('2019-09-18 18:15:00', 'YYYY-MM-DD HH24:MI:SS'),
        report_type    =>    'html',
```

```
        dump_directory =>    'AWR_RPT_DUMP');
END;
/
```

Acessar relatórios do AWR no console ou na CLI

Para acessar relatórios do AWR ou exportar arquivos de despejo, é possível usar o AWS Management Console ou a AWS CLI. Para ter mais informações, consulte Como baixar um arquivo de log de banco de dados.

Ajuste de vínculos de banco de dados para uso com instâncias de bancos de dados em uma VPC

Para usar links de banco de dados Oracle com instâncias de bancos de dados do Amazon RDS na mesma nuvem privada virtual (VPC) ou em VPCs emparelhadas, as duas instâncias de bancos de dados devem ter uma rota válida entre elas. Verifique a rota válida entre as instâncias de bancos de dados usando suas tabelas de roteamento da VPC e a lista de controle de acesso (ACL) à rede.

O grupo de segurança de cada instância de banco de dados deve permitir a entrada e a saída da outra instância de banco de dados. As regras de entrada e saída podem se referir a security groups da mesma VPC ou de uma VPC emparelhada. Para ter mais informações, consulte Atualização dos seus grupos de segurança para referenciar grupos de segurança na VPC emparelhada.

Se você tiver configurado um servidor DNS personalizado usando os conjuntos de opções DHCP na sua VPC, seu servidor DNS personalizado deverá ser capaz de resolver o nome do destino do vínculo de banco de dados. Para ter mais informações, consulte Configuração de um servidor DNS personalizado.

Para ter mais informações sobre como usar vínculos de banco de dados com o Oracle Data Pump, consulte Importar usando o Oracle Data Pump.

Definir a edição padrão para uma instância de banco de dados

Você pode redefinir objetos de banco de dados em um ambiente privado chamado de edição. Você pode usar a redefinição com base na edição para atualizar os objetos de banco de dados de um aplicativo com tempo de inatividade mínimo.

Você pode definir a edição padrão de uma instância de banco de dados Oracle do Amazon RDS usando o procedimento do Amazon RDS `rdsadmin.rdsadmin_util.alter_default_edition`.

O exemplo a seguir define a edição padrão para a instância de banco de dados Oracle do Amazon RDS como `RELEASE_V1`.

```
EXEC rdsadmin.rdsadmin_util.alter_default_edition('RELEASE_V1');
```

O exemplo a seguir define a edição padrão para a instância de banco de dados Oracle do Amazon RDS como o padrão Oracle.

```
EXEC rdsadmin.rdsadmin_util.alter_default_edition('ORA$BASE');
```

Para ter mais informações sobre a redefinição com base na edição Oracle, consulte About editions and edition-based redefinition na documentação da Oracle.

Habilitar a auditoria para a tabela SYS.AUD$

Para habilitar a auditoria na tabela de trilha de auditoria de banco de dados `SYS.AUD$`, use o procedimento `rdsadmin.rdsadmin_master_util.audit_all_sys_aud_table` do Amazon RDS. A única propriedade de auditoria com suporte é `ALL`. Não é possível auditar ou não auditar instruções ou operações individuais.

A habilitação da auditoria tem suporte para instâncias de banco de dados Oracle que executam as seguintes versões:

- Oracle Database 21c (21.0.0)
- Oracle Database 19c (19.0.0)

O procedimento `audit_all_sys_aud_table` tem os seguintes parâmetros.

Nome do parâmetro	Tipo de dados	Padrão	Obrigatório	Descrição
`p_by_access`	booliano	true	Não	Defina como `true` para auditar `BY ACCESS`. Defina como `false` para auditar `BY SESSION`.

A consulta a seguir retorna a configuração de auditoria atual para SYS.AUD$ para um banco de dados:

```
SELECT * FROM DBA_OBJ_AUDIT_OPTS WHERE OWNER='SYS' AND OBJECT_NAME='AUD$';
```

Os comandos a seguir habilitam a auditoria de ALL em SYS.AUD$ BY ACCESS.

```
EXEC rdsadmin.rdsadmin_master_util.audit_all_sys_aud_table;

EXEC rdsadmin.rdsadmin_master_util.audit_all_sys_aud_table(p_by_access => true);
```

O comando a seguir habilita a auditoria de ALL em SYS.AUD$ BY SESSION.

```
EXEC rdsadmin.rdsadmin_master_util.audit_all_sys_aud_table(p_by_access => false);
```

Para ter mais informações, consulte AUDIT (traditional auditing) (Auditoria [auditoria tradicional]) na documentação do Oracle.

Desabilitar a auditoria para a tabela SYS.AUD$

Para desabilitar a auditoria na tabela de trilha de auditoria de banco de dados SYS.AUD$, use o procedimento `rdsadmin.rdsadmin_master_util.noaudit_all_sys_aud_table` do Amazon RDS. Este procedimento não usa parâmetros.

A consulta a seguir retorna a configuração de auditoria atual para SYS.AUD$ para um banco de dados:

```
SELECT * FROM DBA_OBJ_AUDIT_OPTS WHERE OWNER='SYS' AND OBJECT_NAME='AUD$';
```

O comando a seguir desabilita a auditoria de ALL em SYS.AUD$.

```
EXEC rdsadmin.rdsadmin_master_util.noaudit_all_sys_aud_table;
```

Para ter mais informações, consulte NOAUDIT (traditional auditing) (Sem auditoria [auditoria tradicional]) na documentação do Oracle.

Limpar criações de índice online interrompidas

Para limpar criações de índice online com falha, use o procedimento `rdsadmin.rdsadmin_dbms_repair.online_index_clean` do Amazon RDS.

O procedimento `online_index_clean` tem os seguintes parâmetros.

Nome do parâmetro	Tipo de dados	Padrão	Obrigatório	Descrição
`object_id`	binary_integer	`ALL_INDEX_ID`	Não	O ID do objeto do índice. Normalmente, é possível usar o ID do objeto do texto do erro ORA-08104.
`wait_for_lock`	binary_integer	`rdsadmin.rdsadmin_dbms_repair.lock_wait`	Não	Especifique `rdsadmin.rdsadmin_dbms_repair.lock_wait` , o padrão, para tentar obter um bloqueio no objeto subjacente e tentar novamente até que um limite interno seja atingido se o bloqueio falhar. Especifique `rdsadmin.rdsadmin_dbms_repair.lock_nowait` para tentar obter um bloqueio no objeto subjacente, mas não tentar novamente se o bloqueio falhar.

O exemplo a seguir limpa uma criação de índice online com falha:

```
declare
  is_clean boolean;
begin
  is_clean := rdsadmin.rdsadmin_dbms_repair.online_index_clean(
```

```
    object_id     => 1234567890,
    wait_for_lock => rdsadmin.rdsadmin_dbms_repair.lock_nowait
  );
end;
/
```

Para ter mais informações, consulte ONLINE_INDEX_CLEAN function (Função ONLINE_INDEX_CLEAN) na documentação do Oracle.

Ignorar blocos corrompidos

Para ignorar blocos corrompidos durante as verificações de índices e tabelas, use o pacote `rdsadmin.rdsadmin_dbms_repair`.

Os procedimentos a seguir encapsulam a funcionalidade do procedimento `sys.dbms_repair.admin_table` e não usam parâmetros:

- `rdsadmin.rdsadmin_dbms_repair.create_repair_table`
- `rdsadmin.rdsadmin_dbms_repair.create_orphan_keys_table`
- `rdsadmin.rdsadmin_dbms_repair.drop_repair_table`
- `rdsadmin.rdsadmin_dbms_repair.drop_orphan_keys_table`
- `rdsadmin.rdsadmin_dbms_repair.purge_repair_table`
- `rdsadmin.rdsadmin_dbms_repair.purge_orphan_keys_table`

Os seguintes procedimentos usam o mesmo parâmetro que seus equivalentes no pacote `DBMS_REPAIR` para bancos de dados Oracle:

- `rdsadmin.rdsadmin_dbms_repair.check_object`
- `rdsadmin.rdsadmin_dbms_repair.dump_orphan_keys`
- `rdsadmin.rdsadmin_dbms_repair.fix_corrupt_blocks`
- `rdsadmin.rdsadmin_dbms_repair.rebuild_freelists`
- `rdsadmin.rdsadmin_dbms_repair.segment_fix_status`
- `rdsadmin.rdsadmin_dbms_repair.skip_corrupt_blocks`

Para ter mais informações sobre como lidar com a corrupção de banco de dados, consulte DBMS_REPARE na documentação do Oracle.

Example Responder a blocos corrompidos

Este exemplo mostra o fluxo de trabalho básico para responder a blocos corrompidos. Suas etapas dependerão do local e da natureza da corrupção do bloco.

> **Important**
>
> Antes de tentar reparar blocos corrompidos, revise atentamente a documentação DBMS_REPARE.

Como ignorar blocos corrompidos durante as verificações de índices e tabelas

1. Execute os procedimentos a seguir para criar tabelas de reparo se elas ainda não existirem.

```
EXEC rdsadmin.rdsadmin_dbms_repair.create_repair_table;
EXEC rdsadmin.rdsadmin_dbms_repair.create_orphan_keys_table;
```

2. Execute os procedimentos a seguir para verificar registros existentes e depurá-los, caso seja apropriado.

```
SELECT COUNT(*) FROM SYS.REPAIR_TABLE;
SELECT COUNT(*) FROM SYS.ORPHAN_KEY_TABLE;
SELECT COUNT(*) FROM SYS.DBA_REPAIR_TABLE;
SELECT COUNT(*) FROM SYS.DBA_ORPHAN_KEY_TABLE;

EXEC rdsadmin.rdsadmin_dbms_repair.purge_repair_table;
EXEC rdsadmin.rdsadmin_dbms_repair.purge_orphan_keys_table;
```

3. Execute o procedimento a seguir para verificar blocos corrompidos.

```
SET SERVEROUTPUT ON
DECLARE v_num_corrupt INT;
BEGIN
  v_num_corrupt := 0;
  rdsadmin.rdsadmin_dbms_repair.check_object (
    schema_name => '&corruptionOwner',
    object_name => '&corruptionTable',
    corrupt_count =>  v_num_corrupt
  );
  dbms_output.put_line('number corrupt: '||to_char(v_num_corrupt));
END;
```

```
/

COL CORRUPT_DESCRIPTION FORMAT a30
COL REPAIR_DESCRIPTION FORMAT a30

SELECT OBJECT_NAME, BLOCK_ID, CORRUPT_TYPE, MARKED_CORRUPT,
       CORRUPT_DESCRIPTION, REPAIR_DESCRIPTION
FROM   SYS.REPAIR_TABLE;

SELECT SKIP_CORRUPT
FROM   DBA_TABLES
WHERE  OWNER = '&corruptionOwner'
AND    TABLE_NAME = '&corruptionTable';
```

4. Use o procedimento `skip_corrupt_blocks` para habilitar ou desabilitar a ação de ignorar as tabelas afetadas. Dependendo da situação, também pode ser necessário extrair dados para uma nova tabela e, depois, descartar a tabela que contém o bloco corrompido.

   Execute o procedimento a seguir para permitir que sejam ignoradas as tabelas corrompidas.

   ```
   begin
     rdsadmin.rdsadmin_dbms_repair.skip_corrupt_blocks (
       schema_name => '&corruptionOwner',
       object_name => '&corruptionTable',
       object_type => rdsadmin.rdsadmin_dbms_repair.table_object,
       flags => rdsadmin.rdsadmin_dbms_repair.skip_flag);
   end;
   /
   select skip_corrupt from dba_tables where owner = '&corruptionOwner' and table_name
    = '&corruptionTable';
   ```

   Execute o procedimento a seguir para não permitir que sejam ignoradas as tabelas corrompidas.

   ```
   begin
     rdsadmin.rdsadmin_dbms_repair.skip_corrupt_blocks (
       schema_name => '&corruptionOwner',
       object_name => '&corruptionTable',
       object_type => rdsadmin.rdsadmin_dbms_repair.table_object,
       flags => rdsadmin.rdsadmin_dbms_repair.noskip_flag);
   end;
   /
   ```

```
select skip_corrupt from dba_tables where owner = '&corruptionOwner' and table_name
 = '&corruptionTable';
```

5. Quando tiver concluído todos os trabalhos de reparação, execute os procedimentos a seguir para descartar as tabelas reparadas.

```
EXEC rdsadmin.rdsadmin_dbms_repair.drop_repair_table;
EXEC rdsadmin.rdsadmin_dbms_repair.drop_orphan_keys_table;
```

Redimensionar espaços de tabela, arquivos de dados e arquivos temporários

Por padrão, espaços de tabela Oracle são criados com a opção de extensão automática ativada e sem um tamanho máximo. Devido a essas configurações padrão, os espaços de tabela às vezes podem aumentar demais. Recomendamos que você especifique um tamanho máximo apropriado em espaços de tabela permanentes e temporários e que monitore cuidadosamente o uso do espaço.

Redimensionar espaços de tabela permanentes

Para redimensionar um espaço de tabela permanente em uma instância de banco de dados do RDS para Oracle, use qualquer um dos seguintes procedimentos do Amazon RDS:

- `rdsadmin.rdsadmin_util.resize_datafile`
- `rdsadmin.rdsadmin_util.autoextend_datafile`

O procedimento `resize_datafile` tem os seguintes parâmetros.

Nome do parâmetro	Tipo de dados	Padrão	Obrigatório	Descrição
`p_data_file_id`	número	—	Sim	O identificador do arquivo de dados a ser redimensionado.
`p_size`	varchar2	—	Sim	O tamanho do arquivo de dados. Especifique o tamanho em bytes (o padrão), em quilobytes

Nome do parâmetro	Tipo de dados	Padrão	Obrigatório	Descrição
				(K), em megabytes (M) ou em gigabytes (G).

O procedimento `autoextend_datafile` tem os seguintes parâmetros.

Nome do parâmetro	Tipo de dados	Padrão	Obrigatório	Descrição
`p_data_file_id`	número	—	Sim	O identificador do arquivo de dados a ser redimensionado.
`p_autoextend_state`	varchar2	—	Sim	O estado do recurso de extensão automática. Especifique `ON` para estender o arquivo de dados automaticamente e `OFF` para desativar a extensão automática.
`p_next`	varchar2	—	Não	O tamanho do próximo incremento do arquivo de dados. Especifique o tamanho em bytes (o padrão), em quilobytes (K), em megabytes (M) ou em gigabytes (G).
`p_maxsize`	varchar2	—	Não	O espaço máximo em disco permitido para extensão automática. Especifique o tamanho em bytes (o padrão), em quilobytes (K), em

Nome do parâmetro	Tipo de dados	Padrão	Obrigatório	Descrição
				megabytes (M) ou em gigabytes (G). Você pode especificar `UNLIMITED` para remover o limite de tamanho do arquivo.

O exemplo a seguir redimensiona o arquivo de dados de 4 MB a 500 MB.

```
EXEC rdsadmin.rdsadmin_util.resize_datafile(4,'500M');
```

O exemplo a seguir desativa a opção de extensão automática do arquivo de dados 4. Ele também ativa a extensão automática para o arquivo de dados 5, com um incremento de 128 MB e sem tamanho máximo.

```
EXEC rdsadmin.rdsadmin_util.autoextend_datafile(4,'OFF');
EXEC rdsadmin.rdsadmin_util.autoextend_datafile(5,'ON','128M','UNLIMITED');
```

Redimensionar espaços de tabela temporários

Para redimensionar espaços de tabela temporários em uma instância do RDS para Oracle, inclusive uma réplica de leitura, use qualquer um dos seguintes procedimentos do Amazon RDS:

- `rdsadmin.rdsadmin_util.resize_temp_tablespace`
- `rdsadmin.rdsadmin_util.resize_tempfile`
- `rdsadmin.rdsadmin_util.autoextend_tempfile`

O procedimento `resize_temp_tablespace` tem os seguintes parâmetros.

Nome do parâmetro	Tipo de dados	Padrão	Obrigatório	Descrição
`p_temp_tablespace_name`	varchar2	—	Sim	O nome do espaço de tabela temporário a ser redimensionado.

Nome do parâmetro	Tipo de dados	Padrão	Obrigatório	Descrição
p_size	varchar2	—	Sim	O nome do espaço de tabela. Especifique o tamanho em bytes (o padrão), em quilobytes (K), em megabytes (M) ou em gigabytes (G).

O procedimento resize_tempfile tem os seguintes parâmetros.

Nome do parâmetro	Tipo de dados	Padrão	Obrigatório	Descrição
p_temp_file_id	número	—	Sim	O identificador do arquivo temporário a ser redimensionado.
p_size	varchar2	—	Sim	O tamanho do arquivo temporário. Especifique o tamanho em bytes (o padrão), em quilobytes (K), em megabytes (M) ou em gigabytes (G).

O procedimento autoextend_tempfile tem os seguintes parâmetros.

Nome do parâmetro	Tipo de dados	Padrão	Obrigatório	Descrição
p_temp_file_id	número	—	Sim	O identificador do arquivo temporário a ser redimensionado.

Nome do parâmetro	Tipo de dados	Padrão	Obrigatório	Descrição
p_autoextend_state	varchar2	—	Sim	O estado do recurso de extensão automática. Especifique ON para estender o arquivo temporário automaticamente e OFF para desativar a extensão automática.
p_next	varchar2	—	Não	O tamanho do próximo incremento do arquivo temporário. Especifique o tamanho em bytes (o padrão), em quilobytes (K), em megabytes (M) ou em gigabytes (G).
p_maxsize	varchar2	—	Não	O espaço máximo em disco permitido para extensão automática. Especifique o tamanho em bytes (o padrão), em quilobytes (K), em megabytes (M) ou em gigabytes (G). Você pode especificar UNLIMITED para remover o limite de tamanho do arquivo.

Os exemplos a seguir redimensionam um espaço de tabela temporário chamado TEMP para o tamanho de 4 GB.

```
EXEC rdsadmin.rdsadmin_util.resize_temp_tablespace('TEMP','4G');
```

```
EXEC rdsadmin.rdsadmin_util.resize_temp_tablespace('TEMP','4096000000');
```

O exemplo a seguir redimensiona um espaço de tabela temporário com base no arquivo temporário com o identificador de arquivo 1 para o tamanho de 2 MB.

```
EXEC rdsadmin.rdsadmin_util.resize_tempfile(1,'2M');
```

O exemplo a seguir desativa a opção de extensão automática do arquivo temporário 1. Ele também define o tamanho máximo de extensão automática do arquivo temporário de 2 a 10 GB, com um incremento de 100 MB.

```
EXEC rdsadmin.rdsadmin_util.autoextend_tempfile(1,'OFF');
EXEC rdsadmin.rdsadmin_util.autoextend_tempfile(2,'ON','100M','10G');
```

Para ter mais informações sobre réplicas de leitura para instâncias de banco de dados Oracle, consulte Trabalhar com réplicas de leitura do Amazon RDS para Oracle.

Limpar a lixeira

Quando você remove uma tabela, o banco de dados Oracle não remove imediatamente o espaço de armazenamento. O banco de dados renomeia a tabela e coloca ela e todos os objetos associados em uma lixeira. A limpeza da lixeira remove esses itens e libera o espaço de armazenamento.

Para limpar toda a lixeira, use o procedimento `rdsadmin.rdsadmin_util.purge_dba_recyclebin` do Amazon RDS. No entanto, este procedimento não pode limpar a lixeira de objetos `SYS` e `RDSADMIN`. Se você precisar limpar esses objetos, entre em contato com o AWSSupport.

O exemplo a seguir limpa toda a lixeira.

```
EXEC rdsadmin.rdsadmin_util.purge_dba_recyclebin;
```

Definir os valores padrão exibidos para a redação completa

Para alterar os valores exibidos padrão para a redação completa na instância Oracle do Amazon RDS, use o procedimento `rdsadmin.rdsadmin_util.dbms_redact_upd_full_rdct_val` do Amazon RDS. Observe que você cria uma política de redação com o pacote `DBMS_REDACT` do PL/SQL, conforme explicado na documentação do Oracle Database. O

procedimento dbms_redact_upd_full_rdct_val especifica os caracteres a serem exibidos para diferentes tipos de dados afetados por uma política existente.

O procedimento dbms_redact_upd_full_rdct_val tem os seguintes parâmetros.

Nome do parâmetro	Tipo de dados	Padrão	Obrigatório	Descrição
p_number_val	número	Null	Não	Modifica o valor padrão para colunas do tipo de dados NUMBER.
p_binfloat_val	binary_float	Null	Não	Modifica o valor padrão para colunas do tipo de dados BINARY_FLOAT .
p_bindouble_val	binary_double	Null	Não	Modifica o valor padrão para colunas do tipo de dados BINARY_DOUBLE .
p_char_val	char	Null	Não	Modifica o valor padrão para colunas do tipo de dados CHAR.
p_varchar_val	varchar2	Null	Não	Modifica o valor padrão para colunas do tipo de dados VARCHAR2.
p_nchar_val	nchar	Null	Não	Modifica o valor padrão para colunas do tipo de dados NCHAR.
p_nvarchar_val	nvarchar2	Null	Não	Modifica o valor padrão para colunas do tipo de dados NVARCHAR2 .

Nome do parâmetro	Tipo de dados	Padrão	Obrigatório	Descrição
p_date_val	date	Null	Não	Modifica o valor padrão para colunas do tipo de dados DATE.
p_ts_val	timestamp	Null	Não	Modifica o valor padrão para colunas do tipo de dados TIMESTAMP .
p_tswtz_val	timestamp with time zone	Null	Não	Modifica o valor padrão para colunas do tipo de dados TIMESTAMP WITH TIME ZONE.
p_blob_val	blob	Null	Não	Modifica o valor padrão para colunas do tipo de dados BLOB.
p_clob_val	clob	Null	Não	Modifica o valor padrão para colunas do tipo de dados CLOB.
p_nclob_val	nclob	Null	Não	Modifica o valor padrão para colunas do tipo de dados NCLOB.

O exemplo a seguir altera o valor editado padrão para * referente ao tipo de dados CHAR:

```
EXEC rdsadmin.rdsadmin_util.dbms_redact_upd_full_rdct_val(p_char_val => '*');
```

O exemplo a seguir altera os valores editados padrão para tipos de dados NUMBER, DATE e CHAR:

```
BEGIN
rdsadmin.rdsadmin_util.dbms_redact_upd_full_rdct_val(
    p_number_val=>1,
    p_date_val=>to_date('1900-01-01','YYYY-MM-DD'),
```

```
    p_varchar_val=>'X');
END;
/
```

Depois de alterar os valores padrão para a redação completa com o procedimento `dbms_redact_upd_full_rdct_val`, reinicialize a instância de banco de dados para que a alteração seja implementada. Para ter mais informações, consulte Reinicializar uma instância de banco de dados .

Realização de tarefas comuns relacionadas ao log para instâncias de banco de dados Oracle

Veja a seguir como executar determinadas tarefas comuns de DBA relacionadas ao registro em log nas instâncias de bancos de dados do Amazon RDS que executam o Oracle. Para oferecer uma experiência de serviço gerenciado, o Amazon RDS não fornece acesso ao shell para instâncias de bancos de dados e restringe o acesso a determinados procedimentos e tabelas do sistema que exigem privilégios avançados.

Para obter mais informações, consulte Arquivos de log do banco de dados do Amazon RDS para Oracle.

Tópicos

- Definição do registro em log forçado
- Definição do registro em log complementar
- Alternância de arquivos de log online
- Adição de logs redo online
- Descarte de logs redo online
- Redimensionamento de logs redo online
- Retenção de logs redo arquivados
- Acessando logs de refazimento online e arquivados
- Baixar logs de refazimento arquivados do Amazon S3

Definição do registro em log forçado

No modo de registro em log forçado, o Oracle registra todas as alterações no banco de dados, exceto as alterações em espaços de tabela temporários e em segmentos temporários (as cláusulas

NOLOGGING são ignoradas). Para obter mais informações, consulte Specifying FORCE LOGGING mode na documentação da Oracle.

Para definir o registro em log forçado, use o procedimento `rdsadmin.rdsadmin_util.force_logging` do Amazon RDS. O procedimento `force_logging` tem os seguintes parâmetros.

Nome do parâmetro	Tipo de dados	Padrão	Sim	Descrição
`p_enable`	booliano	true	Não	Defina como `true` para colocar o banco de dados no modo de registro em log forçado, `false` para remover o banco de dados do modo de registro em log forçado.

O exemplo a seguir coloca o banco de dados no modo de registro em log forçado.

```
EXEC rdsadmin.rdsadmin_util.force_logging(p_enable => true);
```

Definição do registro em log complementar

Se você habilitar o registro em log suplementar, o LogMiner terá as informações necessárias para oferecer suporte a linhas encadeadas e tabelas clusterizadas. Para obter mais informações, consulte Supplemental logging na documentação da Oracle.

Por padrão, o banco de dados Oracle não habilita o registro em log complementar.
Para habilitar e desabilitar o registro em log complementar, use o procedimento `rdsadmin.rdsadmin_util.alter_supplemental_logging` do Amazon RDS. Para obter mais informações sobre como o Amazon RDS gerencia a retenção de logs redo arquivados para instâncias de bancos de dados Oracle, consulte Retenção de logs redo arquivados.

O procedimento `alter_supplemental_logging` tem os seguintes parâmetros.

Nome do parâmetro	Tipo de dados	Padrão	Obrigatório	Descrição
`p_action`	varchar2	—	Sim	`'ADD'` para adicionar o registro em log complementar, `'DROP'` para descartar o registro em log complementar.
`p_type`	varchar2	nulo	Não	O tipo de registro em log complementar. Os valores válidos são `'ALL'`, `'FOREIGN KEY'`, `'PRIMARY KEY'`, `'UNIQUE'` ou `PROCEDURAL` .

O exemplo a seguir habilita o registro em log complementar.

```
begin
    rdsadmin.rdsadmin_util.alter_supplemental_logging(
        p_action => 'ADD');
end;
/
```

O exemplo a seguir habilita o registro em log complementar para todas as colunas de tamanho máximo com comprimento fixo.

```
begin
    rdsadmin.rdsadmin_util.alter_supplemental_logging(
        p_action => 'ADD',
        p_type   => 'ALL');
end;
/
```

O exemplo a seguir habilita o registro em log complementar para colunas de chave primária.

```
begin
```

```
    rdsadmin.rdsadmin_util.alter_supplemental_logging(
        p_action => 'ADD',
        p_type   => 'PRIMARY KEY');
end;
/
```

Alternância de arquivos de log online

Para alternar arquivos de log, use o procedimento `rdsadmin.rdsadmin_util.switch_logfile` do Amazon RDS. O procedimento `switch_logfile` não tem parâmetros.

O exemplo a seguir alterna arquivos de log.

```
EXEC rdsadmin.rdsadmin_util.switch_logfile;
```

Adição de logs redo online

Uma instância de banco de dados do Amazon RDS executando o Oracle começa com quatro logs redo online, cada um com 128 MB. Para adicionar mais logs redo, use o procedimento `rdsadmin.rdsadmin_util.add_logfile` do Amazon RDS.

O procedimento `add_logfile` tem os seguintes parâmetros.

> **Note**
>
> Os parâmetros são mutuamente exclusivos.

Nome do parâmetro	Tipo de dados	Padrão	Obrigatório	Descrição
`bytes`	positivo	nulo	Não	O tamanho do arquivo de log em bytes. Use esse parâmetro somente se o tamanho do log for inferior a 2.147.483.648 bytes (2 GiB). Caso contrário

Nome do parâmetro	Tipo de dados	Padrão	Obrigatório	Descrição
				, o RDS emitirá um erro. Entretanto, para tamanhos de log acima desse valor em bytes, use o parâmetro `p_size`.
`p_size`	varchar2	—	Sim	O tamanho do arquivo de log em quilobytes (K), megabytes (M) ou gigabytes (G).

O comando a seguir adiciona um arquivo de log de 100 MB.

```
EXEC rdsadmin.rdsadmin_util.add_logfile(p_size => '100M');
```

Descarte de logs redo online

Para descartar logs redo, use o procedimento `rdsadmin.rdsadmin_util.drop_logfile` do Amazon RDS. O procedimento `drop_logfile` tem os seguintes parâmetros.

Nome do parâmetro	Tipo de dados	Padrão	Obrigatório	Descrição
`grp`	positivo	—	Sim	O número do grupo do log.

O exemplo a seguir descarta o log com o número de grupo 3.

```
EXEC rdsadmin.rdsadmin_util.drop_logfile(grp => 3);
```

Você só pode descartar logs que tenham um status de não utilizado ou inativo. O exemplo a seguir obtém os status dos logs.

```
SELECT GROUP#, STATUS FROM V$LOG;
```

```
GROUP#     STATUS
---------- ----------------
1          CURRENT
2          INACTIVE
3          INACTIVE
4          UNUSED
```

Redimensionamento de logs redo online

Uma instância de banco de dados do Amazon RDS executando o Oracle começa com quatro logs redo online, cada um com 128 MB. O exemplo a seguir mostra como você pode usar procedimentos do Amazon RDS para redimensionar seus logs de 128 MB cada para 512 MB cada.

```
/* Query V$LOG to see the logs.          */
/* You start with 4 logs of 128 MB each. */

SELECT GROUP#, BYTES, STATUS FROM V$LOG;

GROUP#     BYTES      STATUS
---------- ---------- ----------------
1          134217728  INACTIVE
2          134217728  CURRENT
3          134217728  INACTIVE
4          134217728  INACTIVE


/* Add four new logs that are each 512 MB */

EXEC rdsadmin.rdsadmin_util.add_logfile(bytes => 536870912);
EXEC rdsadmin.rdsadmin_util.add_logfile(bytes => 536870912);
EXEC rdsadmin.rdsadmin_util.add_logfile(bytes => 536870912);
EXEC rdsadmin.rdsadmin_util.add_logfile(bytes => 536870912);


/* Query V$LOG to see the logs. */
/* Now there are 8 logs.        */

SELECT GROUP#, BYTES, STATUS FROM V$LOG;

GROUP#     BYTES      STATUS
---------- ---------- ----------------
1          134217728  INACTIVE
```

```
2          134217728  CURRENT
3          134217728  INACTIVE
4          134217728  INACTIVE
5          536870912  UNUSED
6          536870912  UNUSED
7          536870912  UNUSED
8          536870912  UNUSED


/* Drop each inactive log using the group number. */

EXEC rdsadmin.rdsadmin_util.drop_logfile(grp => 1);
EXEC rdsadmin.rdsadmin_util.drop_logfile(grp => 3);
EXEC rdsadmin.rdsadmin_util.drop_logfile(grp => 4);


/* Query V$LOG to see the logs. */
/* Now there are 5 logs.        */

select GROUP#, BYTES, STATUS from V$LOG;

GROUP#     BYTES      STATUS
---------- ---------- ----------------
2          134217728  CURRENT
5          536870912  UNUSED
6          536870912  UNUSED
7          536870912  UNUSED
8          536870912  UNUSED


/* Switch logs so that group 2 is no longer current. */

EXEC rdsadmin.rdsadmin_util.switch_logfile;


/* Query V$LOG to see the logs.        */
/* Now one of the new logs is current. */

SQL>SELECT GROUP#, BYTES, STATUS FROM V$LOG;

GROUP#     BYTES      STATUS
---------- ---------- ----------------
2          134217728  ACTIVE
5          536870912  CURRENT
```

```
6          536870912  UNUSED
7          536870912  UNUSED
8          536870912  UNUSED


/* If the status of log 2 is still "ACTIVE", issue a checkpoint to clear it to
 "INACTIVE".  */

EXEC rdsadmin.rdsadmin_util.checkpoint;


/* Query V$LOG to see the logs.            */
/* Now the final original log is inactive. */

select GROUP#, BYTES, STATUS from V$LOG;

GROUP#     BYTES      STATUS
---------- ---------- ----------------
2          134217728  INACTIVE
5          536870912  CURRENT
6          536870912  UNUSED
7          536870912  UNUSED
8          536870912  UNUSED


# Drop the final inactive log.

EXEC rdsadmin.rdsadmin_util.drop_logfile(grp => 2);


/* Query V$LOG to see the logs.    */
/* Now there are four 512 MB logs. */

SELECT GROUP#, BYTES, STATUS FROM V$LOG;

GROUP#     BYTES      STATUS
---------- ---------- ----------------
5          536870912  CURRENT
6          536870912  UNUSED
7          536870912  UNUSED
8          536870912  UNUSED
```

Retenção de logs redo arquivados

Você pode reter logs de refazimento arquivados localmente emsua instância de banco de dados para uso com produtos como o Oracle LogMiner (`DBMS_LOGMNR`). Depois de ter retido os redo logs, você pode usar o LogMiner para analisar esses logs. Para obter mais informações, consulte Using LogMiner to analyze redo log files na documentação da Oracle.

Para reter logs redo arquivados, use o procedimento `rdsadmin.rdsadmin_util.set_configuration` do Amazon RDS. Se você usar esse procedimento em uma instância primária no Oracle Data Guard, o RDS alterará a configuração de retenção do log de arquivamento na instância primária e nas réplicas de leitura abertas, mas não nas réplicas montadas. O RDS retém os últimos logs de redo arquivados em réplicas montadas por um curto período. O RDS exclui automaticamente os logs mais antigos baixados para as réplicas montadas.

O procedimento `set_configuration` tem os seguintes parâmetros.

Nome do parâmetro	Tipo de dados	Padrão	Obrigatório	Descrição
`name`	varchar	—	Sim	O nome da configuração a ser atualizada. Para alterar as horas de retenção do log de redo arquivado, defina o nome como `archivelog retention hours`.
`value`	varchar	—	Sim	O valor para a configuração. Defina o valor como o número de horas para reter os logs.

O exemplo a seguir retém 24 horas de logs redo.

```
begin
    rdsadmin.rdsadmin_util.set_configuration(
        name  => 'archivelog retention hours',
```

```
        value => '24');
end;
/
commit;
```

> **Note**
>
> A confirmação é necessária para que a alteração entre em vigor.

Para visualizar por quanto tempo os logs redo arquivados são mantidos para a instância de banco de dados, use o procedimento `rdsadmin.rdsadmin_util.show_configuration` do Amazon RDS.

O seguinte exemplo mostra o tempo de retenção do log.

```
set serveroutput on
EXEC rdsadmin.rdsadmin_util.show_configuration;
```

A saída mostra a definição atual de `archivelog retention hours`. A seguinte saída mostra que logs redo arquivados são mantidos por 48 horas.

```
NAME:archivelog retention hours
VALUE:48
DESCRIPTION:ArchiveLog expiration specifies the duration in hours before archive/redo
 log files are automatically deleted.
```

Como os logs redo arquivados são retidos na sua instância de banco de dados, certifique-se de que essa instância tenha armazenamento alocado suficiente para os logs retidos. Para determinar quanto espaço sua instância de banco de dados usou nas últimas X horas, você pode executar a consulta a seguir, substituindo X pelo número de horas.

```
SELECT SUM(BLOCKS * BLOCK_SIZE) bytes
  FROM V$ARCHIVED_LOG
 WHERE FIRST_TIME >= SYSDATE-(X/24) AND DEST_ID=1;
```

O RDS for Oracle apenas gera logs de refazimento arquivados se o período de retenção de backup de sua instância de banco de dados for maior que zero. Por padrão, o período de retenção de backup é maior que zero.

Quando o período de retenção de logs arquivados expira, o RDS for Oracle remove os logs de refazimento arquivados de sua instância de banco de dados. Para oferecer suporte à restauração point-in-time de sua instância de banco de dados, o Amazon RDS retém os logs de refazimento arquivados fora de sua instância de banco de dados com base no período de retenção de backup. Para modificar o período de retenção de backup, consulte Modificar uma instância de banco de dados do Amazon RDS.

> **Note**
>
> Em alguns casos, você pode estar usando o JDBC no Linux para baixar logs redo arquivados e fazer testes com tempos de latência longos e redefinições de conexão. Nesses casos, os problemas podem ser causados pela configuração padrão do gerador de números aleatórios no seu cliente Java. Recomendamos que você configure seus drivers JDBC para usar um gerador de números aleatórios sem bloqueio.

Acessando logs de refazimento online e arquivados

Talvez você queira acessar seus arquivos de log redo online e arquivados para exploração com ferramentas externas, como o GoldenGate, o Attunity e o Informatica, entre outras. Para acessar esses arquivos, faça o seguinte:

1. Crie objetos de diretório que forneçam acesso somente leitura aos caminhos de arquivos físicos.

   Utilizar `rdsadmin.rdsadmin_master_util.create_archivelog_dir` e `rdsadmin.rdsadmin_master_util.create_onlinelog_dir`.

2. Leia os arquivos usando PL/SQL.

   É possível ler os arquivos usando PL/SQL. Para obter mais informações sobre como ler arquivos a partir de objetos de diretório, consulte Listagem de arquivos no diretório de uma instância de banco de dados e Leitura de arquivos no diretório de uma instância de banco de dados.

O acesso a logs de transação é compatível com as seguintes versões:

- Oracle Database 21c
- Oracle Database 19c

O código a seguir cria diretórios que fornecem acesso somente leitura aos seus arquivos de log redo online e arquivados:

> **Important**
>
> Esse código também revoga o privilégio `DROP ANY DIRECTORY`.

```
EXEC rdsadmin.rdsadmin_master_util.create_archivelog_dir;
EXEC rdsadmin.rdsadmin_master_util.create_onlinelog_dir;
```

O código a seguir descarta os diretórios para seus arquivos de log redo online e arquivados.

```
EXEC rdsadmin.rdsadmin_master_util.drop_archivelog_dir;
EXEC rdsadmin.rdsadmin_master_util.drop_onlinelog_dir;
```

O código a seguir concede e revoga o privilégio `DROP ANY DIRECTORY`.

```
EXEC rdsadmin.rdsadmin_master_util.revoke_drop_any_directory;
EXEC rdsadmin.rdsadmin_master_util.grant_drop_any_directory;
```

Baixar logs de refazimento arquivados do Amazon S3

É possível baixar logs de refazimento arquivados em sua instância de banco de dados usando o pacote `rdsadmin.rdsadmin_archive_log_download`. Se os logs de refazimento arquivados não estiverem mais em sua instância de banco de dados, você poderá baixá-los novamente do Amazon S3. Então você pode extrair os logs ou usá-los para recuperar ou replicar seu banco de dados.

> **Note**
>
> Não é possível baixar logs de redo arquivados em instâncias de réplica de leitura.

Baixar logs de refazimento arquivados: etapas básicas

A disponibilidade dos logs de refazimento arquivados depende das seguintes políticas de retenção:

- Política de retenção de backup: os logs dentro dessa política estão disponíveis no Amazon S3. Os logs fora dessa política são removidos.
- Política de retenção de logs arquivados: os logs dentro dessa política estão disponíveis em sua instância de banco de dados. Os logs fora dessa política são removidos.

Se os logs não estiverem em sua instância, mas estiverem protegidos pelo período de retenção de backup, use `rdsadmin.rdsadmin_archive_log_download` para baixar novamente. O RDS for Oracle salva os logs no diretório `/rdsdbdata/log/arch` em sua instância de banco de dados.

Para baixar logs de refazimento arquivados do Amazon S3

1. Configure seu período de retenção para garantir que os redo logs arquivados que foram baixados sejam retidos pelo tempo em que você precisar deles. Não se esqueça de `COMMIT` a alteração.

   O RDS retém os logs baixados de acordo com a política de retenção de logs arquivados, a partir do momento em que os logs foram baixados. Para saber como definir a política de retenção, consulte Retenção de logs redo arquivados.
2. Aguarde até 5 minutos para que a alteração da política de retenção de logs arquivados seja implementada.
3. Baixe logs de refazimento arquivados do Amazon S3 usando `rdsadmin.rdsadmin_archive_log_download`.

   Para obter mais informações, consulte Baixar um único log de refazimento arquivado e Baixar uma série de logs de refazimento arquivados.

   > **Note**
   >
   > O RDS verifica automaticamente o armazenamento disponível antes de baixar. Se os logs solicitados consumirem uma alta porcentagem de espaço, você receberá um alerta.
4. Confirme se o download dos logs do Amazon S3 foi concluído corretamente.

   Você pode visualizar o status de uma tarefa de download em um arquivo bdump. Os arquivos bdump têm o nome de caminho `/rdsdbdata/log/trace/dbtask-`*`task-id`*`.log`. Na etapa de download anterior, execute uma instrução `SELECT` que retorna o ID da tarefa em um tipo de dados `VARCHAR2`. Para obter mais informações, veja exemplos semelhantes em Monitorar o status de uma transferência de arquivo.

Baixar um único log de refazimento arquivado

Para baixar um único log de refazimento arquivado para o diretório `/rdsdbdata/log/arch`, use `rdsadmin.rdsadmin_archive_log_download.download_log_with_seqnum`. Esse procedimento tem o seguinte parâmetro.

Nome do parâmetro	Tipo de dados	Padrão	Obrigatório	Descrição
`seqnum`	número	—	Sim	O número de sequência do log de refazimento arquivado.

O exemplo a seguir baixa o log com o número de sequência 20.

```
SELECT rdsadmin.rdsadmin_archive_log_download.download_log_with_seqnum(seqnum => 20)
       AS TASK_ID
FROM   DUAL;
```

Baixar uma série de logs de refazimento arquivados

Para baixar uma série de logs de refazimento arquivados para o diretório `/rdsdbdata/log/arch`, use `download_logs_in_seqnum_range`. Você só pode baixar até 300 logs por solicitação. O procedimento `download_logs_in_seqnum_range` tem os seguintes parâmetros.

Nome do parâmetro	Tipo de dados	Padrão	Obrigatório	Descrição
`start_seq`	número	—	Sim	O número de sequência inicial da série.
`end_seq`	número	—	Sim	O número de sequência final da série.

O exemplo a seguir baixa os logs da sequência 50 a 100.

```
SELECT rdsadmin.rdsadmin_archive_log_download.download_logs_in_seqnum_range(start_seq
 => 50, end_seq => 100)
```

```
       AS TASK_ID
FROM   DUAL;
```

Realização de tarefas comuns do RMAN para instâncias de banco de dados Oracle

Na seção a seguir, você pode encontrar como você pode realizar tarefas de DBA do Oracle Recovery Manager (RMAN) em suas instâncias de bancos de dados do Amazon RDS executando o Oracle. Para oferecer uma experiência de serviço gerenciada, o Amazon RDS não fornece acesso ao shell para as instâncias de banco de dados. Ele também restringe o acesso a determinados procedimentos e tabelas do sistema que exigem privilégios avançados.

Use o pacote `rdsadmin.rdsadmin_rman_util` do Amazon RDS para realizar backups do RMAN do banco de dados do Amazon RDS para Oracle em um disco. O pacote `rdsadmin.rdsadmin_rman_util` é compatível com backups de arquivos de banco de dados completos e incrementais, backups de espaços de tabelas e backups de redo logs arquivados.

Após a conclusão de um backup do RMAN, é possível copiar os arquivos de backup do host da instância de banco de dados do Amazon RDS para Oracle. Você pode fazer isso com a finalidade de restaurar para um host não-RDS ou para armazenamento de backups a longo prazo. Por exemplo, você pode copiar os arquivos de backup para um bucket do Amazon S3. Para obter mais informações, consulte como usar Integração do Amazon S3.

Os arquivos de backup para backups do RMAN permanecerão no host da instância de banco de dados do Amazon RDS até que você os remova manualmente. Você pode usar o procedimento `UTL_FILE.FREMOVE` do Oracle para remover arquivos de um diretório. Para receber mais informações, consulte FREMOVE procedure na documentação da Oracle.

Você não pode usar o RMAN para restaurar o RDS para instâncias de banco de dados Oracle. No entanto, é possível usar o RMAN para restaurar um backup em uma instância on-premises ou do Amazon EC2. Para receber mais informações, consulte o artigo de blog Restaurar uma instância do Amazon RDS para Oracle para uma instância autogerenciada.

> **Note**
>
> Para fazer o backup e a restauração para outra instância de banco de dados do Amazon RDS para Oracle, você pode continuar usando os atributos de backup e restauração do

Amazon RDS. Para ter mais informações, consulte Backup, restauração e exportação de dados.

Tópicos

- Pré-requisitos para backups do RMAN
- Parâmetros comuns para procedimentos do RMAN
- Validar arquivos de banco de dados no RDS para Oracle
- Habilitar e desabilitar o controle de alterações de bloco
- Verificação cruzada de logs redo
- Fazer backup de logs redo armazenados
- Realizar um backup de banco de dados completo
- Realizar um backup completo de um banco de dados de locatários
- Realizar um backup de banco de dados incremental
- Realizar um backup incremental de um banco de dados de locatários
- Backup de um tablespace
- Backup de um arquivo de controle
- Realizar a recuperação de mídia de blocos

Pré-requisitos para backups do RMAN

Antes de fazer backup do banco de dados usando o pacote `rdsadmin.rdsadmin_rman_util`, atenda aos seguintes pré-requisitos:

- Assegure-se de que o banco de dados RDS para Oracle esteja no modo `ARCHIVELOG`. Para habilitar esse modo, defina o período de retenção de backup como um valor diferente de zero.
- Ao fazer o backup de redo logs arquivados ou realizar um backup completo ou incremental que inclua redo logs arquivados, a retenção de redo logs deve estar definida como um valor diferente de zero. Os redo logs arquivados são necessários para tornar os arquivos do banco de dados consistentes durante a recuperação. Para ter mais informações, consulte Retenção de logs redo arquivados.
- Assegure-se de que sua instância de banco de dados tenha espaço livre suficiente para armazenar os backups. Ao fazer backup do banco de dados, especifique um objeto de diretório Oracle como um parâmetro na chamada de procedimento. O RMAN coloca os arquivos no

diretório especificado. Você pode usar o diretórios padrão, como `DATA_PUMP_DIR`, ou criar um novo diretório. Para ter mais informações, consulte Criar e eliminar diretórios no espaço de armazenamento de dados principal.

Você pode monitorar o espaço livre atual em uma instância do RDS para Oracle usando a métrica `FreeStorageSpace` do CloudWatch. Recomendamos que o espaço livre exceda o tamanho atual do banco de dados, embora o RMAN faça backup somente de blocos formatados e seja compatível com a compactação.

Parâmetros comuns para procedimentos do RMAN

Você pode usar procedimentos no pacote `rdsadmin.rdsadmin_rman_util` do Amazon RDS para realizar tarefas com o RMAN. Vários parâmetros são comuns aos procedimentos no pacote. O pacote tem os seguintes parâmetros comuns.

Nome do parâmetro	Tipo de dados	Valores válidos	Padrã	Obrigat io	Descrição
`p_directory_name`	varchar	Um nome de diretório de banco de dados válido.	—	Sim	O nome do diretório para conter os arquivos de backup.
`p_label`	varchar	`a-z`, `A-Z`, `0-9`, `'_'`, `'-'`, `'.'`	—	Não	Uma string exclusiva incluída nos nomes dos arquivos de backup. **Note** O limite é de 30 caracteres.
`p_owner`	varchar	Um proprietá rio válido do diretório especific	—	Sim	O proprietário do diretório para conter os arquivos de backup.

Nome do parâmetro	Tipo de dados	Valores válidos	Padrã	Obrigat io	Descrição
		ado em `p_directory_name` .			

Nome do parâmetro	Tipo de dados	Valores válidos	Padrã	Obrigat io	Descrição
p_tag	varchar	a-z, A-Z, 0-9, '_', '-', '.'	NULL	Não	Uma string que pode ser usada para distinguir entre backups para indicar a finalidade ou o uso de backups, como backups diários, semanais ou incrementais. O limite é de 30 caracteres. A etiqueta não diferencia maiúsculas de minúsculas. As etiquetas são sempre armazenadas em letras maiúsculas, independentemente de como são inseridas. As etiquetas não precisam ser exclusivas, portanto, vários backups podem ter a mesma etiqueta. Se você não especificar uma etiqueta, o RMAN atribuirá uma etiqueta padrão automaticamente usando o formato TAG*YYYYMMDDTHHMMSS* , em que *YYYY* é o ano, *MM* é o mês, *DD* é o dia, *HH* é a hora (no formato de 24 horas), *MM* são os minutos e *SS* os segundos. A data e a hora referem-se a quando o RMAN iniciou o backup. Por exemplo, um backup pode receber uma etiqueta TAG20190927T214517 para um backup iniciado em 27/9/2019, às 21h45m17s. O parâmetro p_tag é compatível com as seguintes versões de mecanismo de

Nome do parâmetro	Tipo de dados	Valores válidos	Padrão	Obrigatório	Descrição
					banco de dados do Amazon RDS para Oracle: • Oracle Database 21c (21.0.0) • Oracle Database 19c (19.0.0) usando 19.0.0.0.ru-2021-10.rur-2021-10.r1 e posteriores
p_compress	boolean	TRUE, FALSE	FALSE	Não	Especifique TRUE para habilitar a compactação de backup BASIC. Especifique FALSE para desabilitar a compactação de backup BASIC.
p_include_archive_logs	boolean	TRUE, FALSE	FALSE	Não	Especifique TRUE para incluir logs redo arquivados no backup. Especifique FALSE para excluir logs redo arquivados do backup. Se você incluir logs redo arquivados no backup, defina a retenção para uma hora ou mais usando o procedimento rdsadmin.rdsadmin_util.set_configuration . Além disso, chame o procedimento rdsadmin.rdsadmin_rman_util.crosscheck_archivelog imediatamente antes de executar o backup. Caso contrário, o backup poderá falhar devido à ausência de arquivos de log redo que foram excluídos por procedimentos de gerenciamento do Amazon RDS.

Nome do parâmetro	Tipo de dados	Valores válidos	Padrã	Obrigat io	Descrição
`p_include_controlfile`	boolian	`TRUE`, `FALSE`	`FALSE`	Não	Especifique `TRUE` para incluir o arquivo de controle no backup. Especifique `FALSE` para excluir o arquivo de controle do backup.
`p_optimize`	boolian	`TRUE`, `FALSE`	`TRUE`	Não	Especifique `TRUE` para habilitar a otimização de backup, se logs redo arquivados estiverem incluídos, para reduzir o tamanho do backup. Especifique `FALSE` para desabilitar a otimização de backup.
`p_parallel`	número	Um inteiro válido entre 1 e 254 para Oracle Database Enterprise Edition (EE) 1 para outras edições do Oracle Database	1	Não	Número de canais.

Nome do parâmetro	Tipo de dados	Valores válidos	Padrã	Obrigat io	Descrição
p_rman_to_dbms_output	boolian	TRUE, FALSE	FALSE	Não	Quando TRUE, a saída RMAN é enviada para o pacote DBMS_OUTPUT além de um arquivo no diretório BDUMP. No SQL*Plus, use SET SERVEROUTPUT ON para ver a saída. Quando FALSE, a saída RMAN é enviada somente para um arquivo no diretório BDUMP.
p_section_size_mb	número	Um inteiro válido	NULL	Não	O tamanho da seção em megabytes (MB). Valida em paralelo dividindo cada arquivo no tamanho da seção especificado. Quando NULL, o parâmetro é ignorado.
p_validation_type	varchar	'PHYSICAL', 'PHYSICAL+LOGICAL'	'PHYS'	Não	O nível de detecção da corrupção. Especifique 'PHYSICAL' para verificar se há danos físicos. Um exemplo de dano físico é um bloco com uma diferença no cabeçalho e no rodapé. Especifique 'PHYSICAL+LOGICAL' para verificar se há inconsistências lógicas além de danos físicos. Um exemplo de dano lógico é um bloco corrompido.

Validar arquivos de banco de dados no RDS para Oracle

É possível usar o pacote do Amazon RDS `rdsadmin.rdsadmin_rman_util` para validar os arquivos de banco de dados do Amazon RDS para Oracle, como arquivos de dados, espaços de tabela, arquivos de controle ou arquivos de parâmetro do servidor (SPFILEs).

Para obter mais informações sobre a validação do RMAN, consulte Validating database files and backups (Validar arquivos e backups de bancos de dados) e VALIDATE (Validar) na documentação do Oracle.

Tópicos

- Validar um banco de dados
- Validando um banco de dados de inquilinos
- Validar um espaço de tabela
- Validar um arquivo de controle
- Validar um SPFILE
- Validar um arquivo de dados Oracle

Validar um banco de dados

Para validar todos os arquivos relevantes usados por um banco de dados Oracle no RDS para Oracle, use o procedimento `rdsadmin.rdsadmin_rman_util.validate_database` do Amazon RDS.

Este procedimento usa os seguintes parâmetros comuns para tarefas RMAN:

- `p_validation_type`
- `p_parallel`
- `p_section_size_mb`
- `p_rman_to_dbms_output`

Para ter mais informações, consulte Parâmetros comuns para procedimentos do RMAN.

O seguinte exemplo valida o banco de dados usando os valores padrão dos parâmetros.

```
EXEC rdsadmin.rdsadmin_rman_util.validate_database;
```

O seguinte exemplo valida o banco de dados usando os valores especificado dos parâmetros.

```
BEGIN
    rdsadmin.rdsadmin_rman_util.validate_database(
        p_validation_type     => 'PHYSICAL+LOGICAL',
        p_parallel            => 4,
        p_section_size_mb     => 10,
        p_rman_to_dbms_output => FALSE);
END;
/
```

Quando o parâmetro `p_rman_to_dbms_output` é definido como `FALSE`, a saída RMAN é gravada em um arquivo no diretório `BDUMP`.

Para visualizar os arquivos no diretório `BDUMP`, execute o seguinte comando `SELECT`.

```
SELECT * FROM table(rdsadmin.rds_file_util.listdir('BDUMP')) order by mtime;
```

Para visualizar o conteúdo de um arquivo no diretório `BDUMP`, execute o seguinte comando `SELECT`.

```
SELECT text FROM table(rdsadmin.rds_file_util.read_text_file('BDUMP','rds-rman-
validate-nnn.txt'));
```

Substitua o nome do arquivo pelo nome do arquivo que você deseja visualizar.

Validando um banco de dados de inquilinos

Para validar os arquivos de dados do banco de dados do inquilino em um banco de dados de contêiner (CDB), use o procedimento do Amazon RDS. `rdsadmin.rdsadmin_rman_util.validate_tenant`

Este procedimento se aplica somente ao banco de dados do locatário atual e usa os seguintes parâmetros comuns para tarefas RMAN:

- `p_validation_type`
- `p_parallel`
- `p_section_size_mb`
- `p_rman_to_dbms_output`

Para ter mais informações, consulte Parâmetros comuns para procedimentos do RMAN. Esse procedimento não tem suporte para as seguintes versões de mecanismo de banco de dados do for Oracle:

- Oracle Database 21c (21.0.0)
- Oracle Database 19c (19.0.0)

O seguinte exemplo valida a instância de banco de dados usando os valores padrão dos parâmetros.

```
EXEC rdsadmin.rdsadmin_rman_util.validate_tenant;
```

O seguinte exemplo valida a instância de banco de dados usando os valores especificado dos parâmetros.

```
BEGIN
    rdsadmin.rdsadmin_rman_util.validate_tenant(
        p_validation_type     => 'PHYSICAL+LOGICAL',
        p_parallel            => 4,
        p_section_size_mb     => 10,
        p_rman_to_dbms_output => FALSE);
END;
/
```

Quando o parâmetro `p_rman_to_dbms_output` é definido como `FALSE`, a saída RMAN é gravada em um arquivo no diretório `BDUMP`.

Para visualizar os arquivos no diretório `BDUMP`, execute o seguinte comando `SELECT`.

```
SELECT * FROM table(rdsadmin.rds_file_util.listdir('BDUMP')) order by mtime;
```

Para visualizar o conteúdo de um arquivo no diretório `BDUMP`, execute o seguinte comando `SELECT`.

```
SELECT text FROM table(rdsadmin.rds_file_util.read_text_file('BDUMP','rds-rman-
validate-nnn.txt'));
```

Substitua o nome do arquivo pelo nome do arquivo que você deseja visualizar.

Validar um espaço de tabela

Para validar os arquivos associados a um espaço de tabela, use o procedimento `rdsadmin.rdsadmin_rman_util.validate_tablespace` do Amazon RDS.

Este procedimento usa os seguintes parâmetros comuns para tarefas RMAN:

- `p_validation_type`
- `p_parallel`
- `p_section_size_mb`
- `p_rman_to_dbms_output`

Para obter mais informações, consulte Parâmetros comuns para procedimentos do RMAN.

Esse procedimento também usa o seguinte parâmetro adicional.

Nome do parâmetro	Tipo de dados	Valores válidos	Padrão	Obrigatório	Descrição
`p_tablespace_name`	varchar2	Um nome de espaço de tabela válido	—	Sim	O nome do espaço de tabela.

Validar um arquivo de controle

Para validar somente o arquivo de controle usado por uma instância de banco de dados Oracle do Amazon RDS, use o procedimento `rdsadmin.rdsadmin_rman_util.validate_current_controlfile` do Amazon RDS.

Esse procedimento usa o seguinte parâmetro comum para tarefas RMAN:

- `p_validation_type`
- `p_rman_to_dbms_output`

Para obter mais informações, consulte Parâmetros comuns para procedimentos do RMAN.

Validar um SPFILE

Para validar somente o arquivo de parâmetro do servidor (SPFILE) usado por uma instância de banco de dados Oracle do Amazon RDS, use o procedimento `rdsadmin.rdsadmin_rman_util.validate_spfile` do Amazon RDS.

Esse procedimento usa o seguinte parâmetro comum para tarefas RMAN:

- `p_validation_type`
- `p_rman_to_dbms_output`

Para ter mais informações, consulte Parâmetros comuns para procedimentos do RMAN.

Validar um arquivo de dados Oracle

Para validar um arquivo de dados, use o procedimento `rdsadmin.rdsadmin_rman_util.validate_datafile` do Amazon RDS.

Este procedimento usa os seguintes parâmetros comuns para tarefas RMAN:

- `p_validation_type`
- `p_parallel`
- `p_section_size_mb`
- `p_rman_to_dbms_output`

Para obter mais informações, consulte Parâmetros comuns para procedimentos do RMAN.

Esse procedimento também usa os seguintes parâmetros adicionais.

Nome do parâmetro	Tipo de dados	Valores válidos	Padrão	Obrigatório	Descrição
`p_datafile`	varchar2	Um número de ID datafile válido ou um nome datafile	—	Sim	O número do ID do arquivo de dados (de `v$datafile.file#`) ou o nome completo do arquivo de dados,

Nome do parâmetro	Tipo de dados	Valores válidos	Padrão	Obrigatório	Descrição
		válido incluindo o caminho completo			incluindo o caminho (de `v$datafile.name`).
`p_from_block`	número	Um inteiro válido	`NULL`	Não	O número do bloco em que a validação começa dentro do arquivo de dados. Quando é `NULL`, `1` é usado.
`p_to_block`	número	Um inteiro válido	`NULL`	Não	O número do bloco em que a validação termina dentro do arquivo de dados. Quando é `NULL`, o bloco máximo no arquivo de dados é usado.

Habilitar e desabilitar o controle de alterações de bloco

Os registros de monitoramento de alterações de blocos alterou os blocos em um arquivo de monitoramento. Essa técnica pode melhorar a performance de backups incrementais RMAN. Para obter mais informações, consulte Using Block Change Tracking to Improve Incremental Backup Performance (Uso do monitoramento de alterações em bloco para aumentar a performance do backup incremental) na documentação do Oracle Database.

Os recursos do RMAN não são comportados em uma réplica de leitura. No entanto, como parte de sua estratégia de alta disponibilidade, você pode optar por habilitar o rastreamento de blocos em uma réplica somente leitura usando o procedimento `rdsadmin.rdsadmin_rman_util.enable_block_change_tracking`. Se você promover essa

réplica somente leitura a uma instância de banco de dados de origem, o rastreamento de alterações de blocos será habilitado para a nova instância de origem. Assim, a instância pode se beneficiar de backups incrementais rápidos.

Os procedimentos de monitoramento de alterações de blocos são compatíveis na Enterprise Edition somente com as seguintes versões do mecanismo de banco de dados:

- Oracle Database 21c (21.0.0)
- Oracle Database 19c (19.0.0)

> **Note**
>
> Em um CDB de locatário único, as operações a seguir funcionam, mas nenhum mecanismo visível para o cliente pode detectar o status atual das operações. Consulte também Limitações do RDS for CDBs Oracle.

Para habilitar o controle de alterações de bloco para uma instância de banco de dados usando o procedimento `rdsadmin.rdsadmin_rman_util.enable_block_change_tracking` do Amazon RDS. Para desativar o monitoramento de alterações de bloco, use `disable_block_change_tracking`. Esses procedimentos não usam parâmetros.

Para determinar se o controle de alterações de bloco está habilitado para a sua instância de banco de dados, execute a consulta a seguir.

```
SELECT STATUS, FILENAME FROM V$BLOCK_CHANGE_TRACKING;
```

O exemplo a seguir habilita o controle de alterações de bloco para uma instância de banco de dados.

```
EXEC rdsadmin.rdsadmin_rman_util.enable_block_change_tracking;
```

O exemplo a seguir desabilita o controle de alterações de bloco para uma instância de banco de dados.

```
EXEC rdsadmin.rdsadmin_rman_util.disable_block_change_tracking;
```

Verificação cruzada de logs redo

Você pode fazer a verificação cruzada de logs redo arquivados usando o procedimento `rdsadmin.rdsadmin_rman_util.crosscheck_archivelog` do Amazon RDS.

É possível usar esse procedimento para fazer uma verificação cruzada dos logs redo arquivados registrados no arquivo de controle e, opcionalmente, excluir os registros de logs expirados. Quando o RMAN faz um backup, ele cria um registro no arquivo de controle. Com o tempo, esses registros aumentam o tamanho do arquivo de controle. Recomendamos que você remova registros expirados periodicamente.

> **Note**
>
> Os backups padrão do Amazon RDS não usam o RMAN e, portanto, não criam registros no arquivo de controle.

Esse procedimento usa o parâmetro comum `p_rman_to_dbms_output` para tarefas do RMAN.

Para obter mais informações, consulte Parâmetros comuns para procedimentos do RMAN.

Esse procedimento também usa o seguinte parâmetro adicional.

Nome do parâmetro	Tipo de dados	Valores válidos	Padrão	Obrigatório	Descrição
`p_delete_expired`	booliano	`TRUE`, `FALSE`	`TRUE`	Não	Quando `TRUE`, exclui os registros de logs redo arquivados expirados do arquivo de controle. Quando `FALSE`, retém os registros de logs redo arquivados expirados no arquivo de controle.

Esse procedimento não tem suporte para as seguintes versões de mecanismo de banco de dados do Amazon RDS para Oracle:

- Oracle Database 21c (21.0.0)
- Oracle Database 19c (19.0.0)

O exemplo a seguir marca os registros de log redo arquivados no arquivo de controle como expirados, mas não exclui os registros.

```
BEGIN
    rdsadmin.rdsadmin_rman_util.crosscheck_archivelog(
        p_delete_expired      => FALSE,
        p_rman_to_dbms_output => FALSE);
END;
/
```

O exemplo a seguir exclui os registros d logs redo arquivados expirados do arquivo de controle.

```
BEGIN
    rdsadmin.rdsadmin_rman_util.crosscheck_archivelog(
        p_delete_expired      => TRUE,
        p_rman_to_dbms_output => FALSE);
END;
/
```

Fazer backup de logs redo armazenados

Você pode usar o pacote `rdsadmin.rdsadmin_rman_util` do Amazon RDS para fazer backup de logs redo arquivados para uma instância de banco de dados do Amazon RDS Oracle.

Os procedimentos para fazer backup de logs redo arquivados têm suporte para as seguintes versões de mecanismo de banco de dados do Amazon RDS para Oracle:

- Oracle Database 21c (21.0.0)
- Oracle Database 19c (19.0.0)

Tópicos

- Fazer backup de todos os logs redo arquivados

- Fazer backup de um log redo arquivado de um intervalo de datas
- Fazer backup de um log redo arquivado de um intervalo de SCN
- Fazer backup de um log redo arquivado de um intervalo de números de sequência

Fazer backup de todos os logs redo arquivados

Para fazer backup de todos os logs redo arquivados para uma instância de banco de dados Oracle do Amazon RDS, use o procedimento `rdsadmin.rdsadmin_rman_util.backup_archivelog_all` do Amazon RDS.

Este procedimento usa os seguintes parâmetros comuns para tarefas RMAN:

- `p_owner`
- `p_directory_name`
- `p_label`
- `p_parallel`
- `p_compress`
- `p_rman_to_dbms_output`
- `p_tag`

Para obter mais informações, consulte Parâmetros comuns para procedimentos do RMAN.

O exemplo a seguir faz backup de todos os logs redo arquivados para a instância de banco de dados.

```
BEGIN
    rdsadmin.rdsadmin_rman_util.backup_archivelog_all(
        p_owner               => 'SYS',
        p_directory_name      => 'MYDIRECTORY',
        p_parallel            => 4,
        p_tag                 => 'MY_LOG_BACKUP',
        p_rman_to_dbms_output => FALSE);
END;
/
```

Fazer backup de um log redo arquivado de um intervalo de datas

Para fazer backup de logs redo arquivados específicos para uma instância de banco de dados Oracle do Amazon RDS especificando um intervalo de datas, use o procedimento `rdsadmin.rdsadmin_rman_util.backup_archivelog_date` do Amazon RDS. O intervalo de datas especifica quais logs redo arquivados devem ser incluídos no backup.

Este procedimento usa os seguintes parâmetros comuns para tarefas RMAN:

- `p_owner`
- `p_directory_name`
- `p_label`
- `p_parallel`
- `p_compress`
- `p_rman_to_dbms_output`
- `p_tag`

Para obter mais informações, consulte Parâmetros comuns para procedimentos do RMAN.

Esse procedimento também usa os seguintes parâmetros adicionais.

Nome do parâmetro	Tipo de dados	Valores válidos	Padrão	Obrigatório	Descrição
`p_from_date`	data	Uma data entre `start_dat` e e `next_date` de um log redo arquivado que existe no disco. O valor	—	Sim	A data de início dos backups de log arquivados.

Nome do parâmetro	Tipo de dados	Valores válidos	Padrão	Obrigatório	Descrição
		deve ser menor que ou igual ao valor especificado para `p_to_date` .			
`p_to_date`	data	Uma data entre `start_date` e `next_date` de um log redo arquivado que existe no disco. O valor deve ser maior que ou igual ao valor especificado para `p_from_date` .	—	Sim	A data de término dos backups de logs arquivados.

O exemplo a seguir faz backup de logs redo arquivados no intervalo de datas para a instância de banco de dados.

```
BEGIN
    rdsadmin.rdsadmin_rman_util.backup_archivelog_date(
        p_owner               => 'SYS',
        p_directory_name      => 'MYDIRECTORY',
        p_from_date           => '03/01/2019 00:00:00',
        p_to_date             => '03/02/2019 00:00:00',
        p_parallel            => 4,
        p_tag                 => 'MY_LOG_BACKUP',
        p_rman_to_dbms_output => FALSE);
END;
/
```

Fazer backup de um log redo arquivado de um intervalo de SCN

Para fazer backup de logs redo arquivados específicos para uma instância de banco de dados do Oracle do Amazon RDS, especificando um intervalo de SCNs (números de alteração de sistema), use o procedimento `rdsadmin.rdsadmin_rman_util.backup_archivelog_scn` do Amazon RDS. O intervalo de SCNs especifica quais logs redo arquivados devem ser incluídos no backup.

Este procedimento usa os seguintes parâmetros comuns para tarefas RMAN:

- `p_owner`
- `p_directory_name`
- `p_label`
- `p_parallel`
- `p_compress`
- `p_rman_to_dbms_output`
- `p_tag`

Para obter mais informações, consulte Parâmetros comuns para procedimentos do RMAN.

Esse procedimento também usa os seguintes parâmetros adicionais.

Nome do parâmetro	Tipo de dados	Valores válidos	Padrão	Obrigatório	Descrição
`p_from_scn`	número	Um SCN de um log redo arquivado que existe no disco. O valor deve ser menor que ou igual ao valor especificado para `p_to_scn`.	—	Sim	O SCN inicial para os backups de logs arquivados.
`p_to_scn`	número	Um SCN de um log redo arquivado que existe no disco. O valor deve ser maior que ou igual ao valor especificado para	—	Sim	O SCN final para os backups de logs arquivados.

Nome do parâmetro	Tipo de dados	Valores válidos	Padrão	Obrigatório	Descrição
		p_from_scn .			

O exemplo a seguir faz backup de logs redo arquivados no intervalo de SCNs para a instância de banco de dados.

```
BEGIN
    rdsadmin.rdsadmin_rman_util.backup_archivelog_scn(
        p_owner                => 'SYS',
        p_directory_name       => 'MYDIRECTORY',
        p_from_scn             => 1533835,
        p_to_scn               => 1892447,
        p_parallel             => 4,
        p_tag                  => 'MY_LOG_BACKUP',
        p_rman_to_dbms_output => FALSE);
END;
/
```

Fazer backup de um log redo arquivado de um intervalo de números de sequência

Para fazer backup de logs redo arquivados específicos para uma instância de banco de dados Oracle do Amazon RDS, especificando um intervalo de números de sequência, use o procedimento `rdsadmin.rdsadmin_rman_util.backup_archivelog_sequence` do Amazon RDS. O intervalo de números de sequência especifica quais logs redo arquivados devem ser incluídos no backup.

Este procedimento usa os seguintes parâmetros comuns para tarefas RMAN:

- `p_owner`
- `p_directory_name`
- `p_label`
- `p_parallel`
- `p_compress`

- `p_rman_to_dbms_output`
- `p_tag`

Para obter mais informações, consulte Parâmetros comuns para procedimentos do RMAN.

Esse procedimento também usa os seguintes parâmetros adicionais.

Nome do parâmetro	Tipo de dados	Valores válidos	Padrão	Obrigatório	Descrição
`p_from_sequence`	número	Um número de sequência de um log redo arquivado que existe no disco. O valor deve ser menor que ou igual ao valor especificado para `p_to_sequence` .	—	Sim	O número de sequência inicial dos backups de log arquivados.
`p_to_sequence`	número	Um número de sequência de um log redo	—	Sim	O número de sequência final dos backups de log arquivados.

Nome do parâmetro	Tipo de dados	Valores válidos	Padrão	Obrigatório	Descrição
		arquivado que existe no disco. O valor deve ser maior que ou igual ao valor especificado para `p_from_sequence`.			

O exemplo a seguir faz backup de logs redo arquivados no intervalo de números de sequência para a instância de banco de dados.

```
BEGIN
    rdsadmin.rdsadmin_rman_util.backup_archivelog_sequence(
        p_owner               => 'SYS',
        p_directory_name      => 'MYDIRECTORY',
        p_from_sequence       => 11160,
        p_to_sequence         => 11160,
        p_parallel            => 4,
        p_tag                 => 'MY_LOG_BACKUP',
        p_rman_to_dbms_output => FALSE);
END;
/
```

Realizar um backup de banco de dados completo

Você pode realizar um backup de todos os blocos de arquivos de dados incluídos no backup usando o procedimento `rdsadmin.rdsadmin_rman_util.backup_database_full` do Amazon RDS.

Este procedimento usa os seguintes parâmetros comuns para tarefas RMAN:

- p_owner
- p_directory_name
- p_label
- p_parallel
- p_section_size_mb
- p_include_archive_logs
- p_optimize
- p_compress
- p_rman_to_dbms_output
- p_tag

Para obter mais informações, consulte Parâmetros comuns para procedimentos do RMAN.

Esse procedimento não tem suporte para as seguintes versões de mecanismo de banco de dados do Amazon RDS para Oracle:

- Oracle Database 21c (21.0.0)
- Oracle Database 19c (19.0.0)

O exemplo a seguir realiza um backup completo da instância de banco de dados usando os valores especificados para os parâmetros.

```
BEGIN
    rdsadmin.rdsadmin_rman_util.backup_database_full(
        p_owner               => 'SYS',
        p_directory_name      => 'MYDIRECTORY',
        p_parallel            => 4,
        p_section_size_mb     => 10,
        p_tag                 => 'FULL_DB_BACKUP',
        p_rman_to_dbms_output => FALSE);
END;
/
```

Realizar um backup completo de um banco de dados de locatários

Você pode realizar um backup de todos os blocos de dados incluídos em um banco de dados de locatário em um banco de dados de contêiner (CDB). Use o procedimento do Amazon RDS

`rdsadmin.rdsadmin_rman_util.backup_tenant_full`. Este procedimento se aplica somente ao backup do banco de dados atual e usa os seguintes parâmetros comuns para tarefas RMAN:

- `p_owner`
- `p_directory_name`
- `p_label`
- `p_parallel`
- `p_section_size_mb`
- `p_include_archive_logs`
- `p_optimize`
- `p_compress`
- `p_rman_to_dbms_output`
- `p_tag`

Para ter mais informações, consulte Parâmetros comuns para procedimentos do RMAN.

O parâmetro `rdsadmin_rman_util.backup_tenant_full` é compatível com as seguintes versões de mecanismo de banco de dados Amazon RDS for Oracle:

- Oracle Database 21c (21.0.0)
- Oracle Database 19c (19.0.0)

O exemplo a seguir realiza um backup completo da instância de banco de dados usando os valores especificados para os parâmetros.

```
BEGIN
    rdsadmin.rdsadmin_rman_util.backup_tenant_full(
        p_owner               => 'SYS',
        p_directory_name      => 'MYDIRECTORY',
        p_parallel            => 4,
        p_section_size_mb     => 10,
        p_tag                 => 'FULL_TENANT_DB_BACKUP',
        p_rman_to_dbms_output => FALSE);
END;
/
```

Realizar um backup de banco de dados incremental

Você pode realizar um backup incremental da sua instância de banco de dados usando o procedimento `rdsadmin.rdsadmin_rman_util.backup_database_incremental` do Amazon RDS.

Para obter mais informações sobre backups incrementais, consulte Incremental backups (Backups incrementais) na documentação do Oracle.

Este procedimento usa os seguintes parâmetros comuns para tarefas RMAN:

- `p_owner`
- `p_directory_name`
- `p_label`
- `p_parallel`
- `p_section_size_mb`
- `p_include_archive_logs`
- `p_include_controlfile`
- `p_optimize`
- `p_compress`
- `p_rman_to_dbms_output`
- `p_tag`

Para obter mais informações, consulte Parâmetros comuns para procedimentos do RMAN.

Esse procedimento não tem suporte para as seguintes versões de mecanismo de banco de dados do Amazon RDS para Oracle:

- Oracle Database 21c (21.0.0)
- Oracle Database 19c (19.0.0)

Esse procedimento também usa o seguinte parâmetro adicional.

Nome do parâmetro	Tipo de dados	Valores válidos	Padrão	Obrigatório	Descrição
p_level	número	0, 1	0	Não	Especifique 0 para habilitar um backup incremental completo. Especifique 1 para habilitar um backup incremental não cumulativo.

O exemplo a seguir realiza um backup incremental da instância de banco de dados usando os valores especificados para os parâmetros.

```
BEGIN
    rdsadmin.rdsadmin_rman_util.backup_database_incremental(
        p_owner               => 'SYS',
        p_directory_name      => 'MYDIRECTORY',
        p_level               => 1,
        p_parallel            => 4,
        p_section_size_mb     => 10,
        p_tag                 => 'MY_INCREMENTAL_BACKUP',
        p_rman_to_dbms_output => FALSE);
END;
/
```

Realizar um backup incremental de um banco de dados de locatários

Você pode realizar um backup incremental do banco de dados do inquilino atual no seu CDB. Use o procedimento do Amazon RDS `rdsadmin.rdsadmin_rman_util.backup_tenant_incremental`.

Para obter mais informações sobre backups incrementais, consulte Incremental backups (Backups incrementais) na documentação do Oracle.

Este procedimento se aplica somente ao banco de dados do locatário atual e usa os seguintes parâmetros comuns para tarefas RMAN:

- p_owner
- p_directory_name
- p_label
- p_parallel
- p_section_size_mb
- p_include_archive_logs
- p_include_controlfile
- p_optimize
- p_compress
- p_rman_to_dbms_output
- p_tag

Para ter mais informações, consulte Parâmetros comuns para procedimentos do RMAN.

Esse procedimento não tem suporte para as seguintes versões de mecanismo de banco de dados do Amazon RDS para Oracle:

- Oracle Database 21c (21.0.0)
- Oracle Database 19c (19.0.0)

Esse procedimento também usa o seguinte parâmetro adicional.

Nome do parâmetro	Tipo de dados	Valores válidos	Padrão	Obrigatório	Descrição
p_level	número	0, 1	0	Não	Especifique 0 para habilitar um backup incremental completo. Especifique 1 para habilitar um backup incremental não cumulativo.

O exemplo a seguir realiza um backup incremental da instância de banco de dados usando os valores especificados para os parâmetros.

```
BEGIN
    rdsadmin.rdsadmin_rman_util.backup_tenant_incremental(
        p_owner               => 'SYS',
        p_directory_name      => 'MYDIRECTORY',
        p_level               => 1,
        p_parallel            => 4,
        p_section_size_mb     => 10,
        p_tag                 => 'MY_INCREMENTAL_BACKUP',
        p_rman_to_dbms_output => FALSE);
END;
/
```

Backup de um tablespace

Você pode realizar um backup de um espaço de tabela usando o procedimento do Amazon RDS `rdsadmin.rdsadmin_rman_util.backup_tablespace`.

Este procedimento usa os seguintes parâmetros comuns para tarefas RMAN:

- `p_owner`
- `p_directory_name`
- `p_label`
- `p_parallel`
- `p_section_size_mb`
- `p_include_archive_logs`
- `p_include_controlfile`
- `p_optimize`
- `p_compress`
- `p_rman_to_dbms_output`
- `p_tag`

Para obter mais informações, consulte Parâmetros comuns para procedimentos do RMAN.

Esse procedimento também usa o seguinte parâmetro adicional.

Nome do parâmetro	Tipo de dados	Valores válidos	Padrão	Obrigatório	Descrição
`p_tablespace_name`	varchar2	Um nome de espaço de tabela válido.	—	Sim	O nome do espaço de tabela para backup.

Esse procedimento não tem suporte para as seguintes versões de mecanismo de banco de dados do Amazon RDS para Oracle:

- Oracle Database 21c (21.0.0)
- Oracle Database 19c (19.0.0)

O exemplo a seguir realiza um backup de espaço de tabela usando os valores especificados para os parâmetros.

```
BEGIN
    rdsadmin.rdsadmin_rman_util.backup_tablespace(
        p_owner               => 'SYS',
        p_directory_name      => 'MYDIRECTORY',
        p_tablespace_name     => 'MYTABLESPACE',
        p_parallel            => 4,
        p_section_size_mb     => 10,
        p_tag                 => 'MYTABLESPACE_BACKUP',
        p_rman_to_dbms_output => FALSE);
END;
/
```

Backup de um arquivo de controle

Você pode fazer backup de um arquivo de controle usando o procedimento `rdsadmin.rdsadmin_rman_util.backup_current_controlfile` do Amazon RDS.

Este procedimento usa os seguintes parâmetros comuns para tarefas RMAN:

- `p_owner`

- `p_directory_name`
- `p_label`
- `p_compress`
- `p_rman_to_dbms_output`
- `p_tag`

Para obter mais informações, consulte Parâmetros comuns para procedimentos do RMAN.

Esse procedimento não tem suporte para as seguintes versões de mecanismo de banco de dados do Amazon RDS para Oracle:

- Oracle Database 21c (21.0.0)
- Oracle Database 19c (19.0.0)

O exemplo a seguir realiza um backup de um arquivo de controle usando os valores especificados para os parâmetros.

```
BEGIN
    rdsadmin.rdsadmin_rman_util.backup_current_controlfile(
        p_owner               => 'SYS',
        p_directory_name      => 'MYDIRECTORY',
        p_tag                 => 'CONTROL_FILE_BACKUP',
        p_rman_to_dbms_output => FALSE);
END;
/
```

Realizar a recuperação de mídia de blocos

É possível recuperar blocos de dados individuais, conhecidos como recuperação de mídia de blocos, usando os procedimentos `rdsadmin.rdsadmin_rman_util.recover_datafile_block` do Amazon RDS. É possível usar esse procedimento sobrecarregado para recuperar um ou vários blocos de dados.

Esse procedimento usa o seguinte parâmetro comum para tarefas RMAN:

- `p_rman_to_dbms_output`

Para ter mais informações, consulte Parâmetros comuns para procedimentos do RMAN.

Esse procedimento usa os parâmetros adicionais a seguir.

Nome do parâmetro	Tipo de dados	Valores válidos	Padrão	Obrigatório	Descrição
`p_datafile`	`NUMBER`	Um número de ID válido do arquivo de dados.	—	Sim	O arquivo de dados contendo os blocos corrompidos. Especifique o arquivo de dados de uma das seguintes formas: • O número de ID do arquivo de dados, localizado em `V$DATAFILE.FILE#`. • O nome completo do arquivo de dados, incluindo o caminho, localizado em `V$DATAFILE.NAME`.
`p_block`	`NUMBER`	Um número inteiro válido.	—	Sim	O número de um bloco individual a ser recuperado. Os seguintes parâmetros são mutuamente exclusivos: • `p_block` • `p_from_block` e `p_to_block`
`p_from_block`	`NUMBER`	Um número inteiro válido.	—	Sim	O primeiro número de bloco em um intervalo de blocos a serem recuperados.

Nome do parâmetro	Tipo de dados	Valores válidos	Padrão	Obrigatório	Descrição
					Os seguintes parâmetros são mutuamente exclusivos: • `p_block` • `p_from_block` e `p_to_block`
`p_to_block`	`NUMBER`	Um número inteiro válido.	—	Sim	O último número de bloco em um intervalo de blocos a serem recuperados. Os seguintes parâmetros são mutuamente exclusivos: • `p_block` • `p_from_block` e `p_to_block`

Esse procedimento não tem suporte para as seguintes versões de mecanismo de banco de dados do Amazon RDS para Oracle:

- Oracle Database 21c (21.0.0)
- Oracle Database 19c (19.0.0)

O exemplo a seguir recupera o bloco 100 no arquivo de dados 5.

```
BEGIN
    rdsadmin.rdsadmin_rman_util.recover_datafile_block(
        p_datafile            => 5,
        p_block               => 100,
        p_rman_to_dbms_output => TRUE);
END;
```

```
/
```

O exemplo a seguir recupera os blocos 100 a 150 no arquivo de dados 5.

```
BEGIN
    rdsadmin.rdsadmin_rman_util.recover_datafile_block(
        p_datafile            => 5,
        p_from_block          => 100,
        p_to_block            => 150,
        p_rman_to_dbms_output => TRUE);
END;
/
```

Realização de tarefas comuns de agendamento para instâncias de banco de dados Oracle

Alguns trabalhos pertencentes ao SYS podem interferir nas operações normais de banco de dados. Nesses casos, o Oracle Support recomenda que você modifique a programação. Se você precisar habilitar ou desabilitar trabalhos do SYS, teste a operação em jobs programados em um ambiente de teste antes de implementá-la em um ambiente de produção. Para realizar tarefas para trabalhos do Oracle Scheduler pertencentes ao SYS, use o pacote `rdsadmin.rdsadmin_dbms_scheduler` do Amazon RDS.

Os procedimentos `rdsadmin.rdsadmin_dbms_scheduler` são compatíveis com as versões de mecanismo de banco de dados do Amazon RDS para Oracle na tabela a seguir. Ao usar esse pacote, você pode especificar os trabalhos do SYS listados na tabela.

Versão do banco de dados	Trabalhos habilitados por padrão	Trabalhos desabilitados por padrão

Oracle Database 19c	`BSLN_MAINTAIN_STATS_JOB`
	`CLEANUP_NON_EXIST_OBJ`
	`CLEANUP_ONLINE_IND_BUILD`
	`CLEANUP_ONLINE_PMO`
	`CLEANUP_TAB_IOT_PMO`
	`CLEANUP_TRANSIENT_PKG`
	`CLEANUP_TRANSIENT_TYPE`
	`DRA_REEVALUATE_OPEN_FAILU`
	`RES`
	`FILE_SIZE_UPD`
		`FGR$AUTOPURGE_JOB`
		`FILE_WATCHER`
		`HM_CREATE_OFFLINE_DICTIONARY`
		`LOAD_OPATCH_INVENTORY`
		`ORA$PREPLUGIN_BACKUP_JOB`
		`XMLDB_NFS_CLEANUP_JOB`

Versão do banco de dados	Trabalhos habilitados por padrão	Trabalhos desabilitados por padrão
	ORA$AUTOTASK_CLEAN PMO_DEFERRED_GIDX_MAINT_JOB PURGE_LOG RSE$CLEAN_RECOVERABLE_SCRIPT SM$CLEAN_AUTO_SPLIT_MERGE	
Oracle Database 21c	BSLN_MAINTAIN_STATS_JOB CLEANUP_NON_EXIST_OBJ CLEANUP_ONLINE_IND_BUILD CLEANUP_ONLINE_PMO CLEANUP_TAB_IOT_PMO CLEANUP_TRANSIENT_PKG CLEANUP_TRANSIENT_TYPE DRA_REEVALUATE_OPEN_FAILURES FILE_SIZE_UPD ORA$AUTOTASK_CLEAN PMO_DEFERRED_GIDX_MAINT_JOB PURGE_LOG	FGR$AUTOPURGE_JOB FILE_WATCHER HM_CREATE_OFFLINE_DICTIONARY LOAD_OPATCH_INVENTORY ORA$PREPLUGIN_BACKUP_JOB ORA$_ATSK_AUTOSTS XMLDB_NFS_CLEANUP_JOB

Parâmetros comuns para os procedimentos do Oracle Scheduler

Para realizar tarefas com o Oracle Scheduler, use os procedimentos no pacote `rdsadmin.rdsadmin_dbms_scheduler` do Amazon RDS. Vários parâmetros são comuns aos procedimentos no pacote. O pacote tem os seguintes parâmetros comuns.

Nome do parâmetro	Tipo de dados	Valores válidos	Padrão	Obrigatório	Descrição
`name`	varchar2	Os procedimentos listados	—	Sim	O nome do trabalho a ser modificado.

Nome do parâmetro	Tipo de dados	Valores válidos	Padrão	Obrigatório	Descrição
		na tabela em Realização de tarefas comuns de agendamento para instâncias de banco de dados Oracle			
`attribute`	varchar2	`'REPEAT_INTERVAL'` `_NAME'`	–	Sim	Atributo a ser modificado. Para modificar o intervalo de repetição do trabalho, especifique `'REPEAT_INTERVAL'` . Para modificar o nome da programação do trabalho, especifique `'SCHEDULE_NAME'` .

Nome do parâmetro	Tipo de dados	Valores válidos	Padrão	Obrigatório	Descrição
`value`	varchar2	Um intervalo de programação ou nome de programação válidos, dependendo do atributo usado.	–	Sim	O novo valor do atributo.

Modificação de trabalhos DBMS_SCHEDULER

Para modificar determinados componentes do Oracle Scheduler, use o procedimento `dbms_scheduler.set_attribute` do Oracle. Para obter mais informações, consulte DBMS_SCHEDULER e Procedimento SET_ATTRIBUTE na documentação da Oracle.

Ao trabalhar com instâncias de bancos de dados do Amazon RDS, anteponha o nome do esquema SYS ao nome do objeto. O exemplo a seguir define o atributo de plano de recurso para o objeto da janela de segunda-feira.

```
BEGIN
    DBMS_SCHEDULER.SET_ATTRIBUTE(
        name      => 'SYS.MONDAY_WINDOW',
        attribute => 'RESOURCE_PLAN',
        value     => 'resource_plan_1');
END;
/
```

Modificando janelas de manutenção de AutoTask

As instâncias do Amazon RDS for Oracle são criadas com configurações padrão para janelas de manutenção. Tarefas de manutenção automatizadas, como coleta de estatísticas do otimizador, são executadas durante essas janelas. Por padrão, as janelas de manutenção ativam o Oracle Database Resource Manager.

Para modificar a janela, use o pacote `DBMS_SCHEDULER`. Talvez seja necessário modificar as configurações da janela de manutenção pelos seguintes motivos:

- Você deseja que os trabalhos de manutenção sejam executados em um momento diferente, com configurações diferentes ou não. Por exemplo, talvez você queira modificar a duração da janela ou alterar o tempo e o intervalo de repetição.
- Você quer evitar o impacto sobre a performance causado pela habilitação do Gerenciador de recursos durante a manutenção. Por exemplo, se o plano de manutenção padrão for especificado e se a janela de manutenção for aberta enquanto o banco de dados estiver sob carga, você poderá observar eventos de espera como `resmgr:cpu quantum`. Este evento de espera está relacionado ao Gerenciador de recursos de banco de dados. Você tem as seguintes opções:
  - Certifique-se de que as janelas de manutenção estejam ativas durante horários fora do pico de utilização da sua instância de banco de dados.
  - Desabilite o plano de manutenção padrão definindo o atributo `resource_plan` como uma string vazia.
  - Defina o parâmetro `resource_manager_plan` no grupo de parâmetros do banco de dados como `FORCE:`. Se sua instância usa o Enterprise Edition, essa configuração impede que os planos do Gerenciador de recursos de banco de dados sejam ativados.

Para modificar as configurações da janela de manutenção

1. Conecte ao banco de dados usando um cliente Oracle SQL.
2. Consulte a configuração atual para uma janela do programador.

   O exemplo a seguir consulta a configuração para `MONDAY_WINDOW`.

```
SELECT ENABLED, RESOURCE_PLAN, DURATION, REPEAT_INTERVAL
FROM   DBA_SCHEDULER_WINDOWS
WHERE  WINDOW_NAME='MONDAY_WINDOW';
```

A saída a seguir mostra que a janela está usando os valores padrão.

```
ENABLED         RESOURCE_PLAN                  DURATION         REPEAT_INTERVAL
--------------- ------------------------------ ---------------
 ------------------------------
TRUE            DEFAULT_MAINTENANCE_PLAN       +000 04:00:00
 freq=daily;byday=MON;byhour=22
                                                                 ;byminute=0;
 bysecond=0
```

3. Modifique a janela usando o pacote DBMS_SCHEDULER.

   O exemplo a seguir define o plano de recursos como nulo para que o Gerenciador de recursos não seja executado durante a janela de manutenção.

```
BEGIN
  -- disable the window to make changes
  DBMS_SCHEDULER.DISABLE(name=>'"SYS"."MONDAY_WINDOW"',force=>TRUE);

  -- specify the empty string to use no plan
  DBMS_SCHEDULER.SET_ATTRIBUTE(name=>'"SYS"."MONDAY_WINDOW"',
 attribute=>'RESOURCE_PLAN', value=>'');

  -- re-enable the window
  DBMS_SCHEDULER.ENABLE(name=>'"SYS"."MONDAY_WINDOW"');
END;
/
```

   O exemplo a seguir define a duração máxima da janela como 2 horas.

```
BEGIN
  DBMS_SCHEDULER.DISABLE(name=>'"SYS"."MONDAY_WINDOW"',force=>TRUE);
  DBMS_SCHEDULER.SET_ATTRIBUTE(name=>'"SYS"."MONDAY_WINDOW"',
 attribute=>'DURATION', value=>'0 2:00:00');
  DBMS_SCHEDULER.ENABLE(name=>'"SYS"."MONDAY_WINDOW"');
END;
/
```

   O exemplo a seguir define o intervalo de repetição para todas as segundas-feiras às 10 horas da manhã.

```
BEGIN
  DBMS_SCHEDULER.DISABLE(name=>'"SYS"."MONDAY_WINDOW"',force=>TRUE);
  DBMS_SCHEDULER.SET_ATTRIBUTE(name=>'"SYS"."MONDAY_WINDOW"',
 attribute=>'REPEAT_INTERVAL',
 value=>'freq=daily;byday=MON;byhour=10;byminute=0;bysecond=0');
  DBMS_SCHEDULER.ENABLE(name=>'"SYS"."MONDAY_WINDOW"');
END;
/
```

Definir o fuso horário para trabalhos do Oracle Scheduler

Para modificar o fuso horário do Oracle Scheduler, você pode usar o procedimento do Oracl `dbms_scheduler.set_scheduler_attribute`. Para obter mais informações sobre o pacote `dbms_scheduler`, consulte DBMS_SCHEDULER e SET_SCHEDULER_ATTRIBUTE na documentação do Oracle.

Como modificar a configuração atual de fuso horário

1. Conecte-se ao banco de dados usando um cliente como o SQL Developer. Para obter mais informações, consulte Conectar-se à sua instância de banco de dados usando o Oracle SQL Developer.
2. Defina o fuso horário padrão da seguinte maneira, substituindo seu fuso horário por *`time_zone_name`*.

```
BEGIN
  DBMS_SCHEDULER.SET_SCHEDULER_ATTRIBUTE(
    attribute => 'default_timezone',
    value => 'time_zone_name'
  );
END;
/
```

No exemplo a seguir, altere o fuso horário para Ásia/Xangai.

Comece consultando o fuso horário atual, como mostrado a seguir.

```
SELECT VALUE FROM DBA_SCHEDULER_GLOBAL_ATTRIBUTE WHERE
 ATTRIBUTE_NAME='DEFAULT_TIMEZONE';
```

A saída mostra que o fuso horário atual é ETC/UTC.

```
VALUE
-------
Etc/UTC
```

Depois, você define o fuso horário como Ásia/Xangai.

```
BEGIN
  DBMS_SCHEDULER.SET_SCHEDULER_ATTRIBUTE(
    attribute => 'default_timezone',
    value => 'Asia/Shanghai'
  );
END;
/
```

Para obter mais informações sobre como alterar o fuso horário do sistema, consulte Fuso horário da Oracle.

Desativar trabalhos do Oracle Scheduler de propriedade do SYS

Para desativar um trabalho do Oracle Scheduler pertencente ao SYS, use o procedimento `rdsadmin.rdsadmin_dbms_scheduler.disable`.

Esse procedimento usa o parâmetro comum `name` para tarefas do Oracle Scheduler. Para obter mais informações, consulte Parâmetros comuns para os procedimentos do Oracle Scheduler.

O exemplo a seguir desabilita o trabalho `SYS.CLEANUP_ONLINE_IND_BUILD` do Oracle Scheduler.

```
BEGIN
   rdsadmin.rdsadmin_dbms_scheduler.disable('SYS.CLEANUP_ONLINE_IND_BUILD');
END;
/
```

Ativar trabalhos do Oracle Scheduler de propriedade do SYS

Para ativar um trabalho do Oracle Scheduler pertencente ao SYS, use o procedimento `rdsadmin.rdsadmin_dbms_scheduler.enable`.

Esse procedimento usa o parâmetro comum `name` para tarefas do Oracle Scheduler. Para obter mais informações, consulte Parâmetros comuns para os procedimentos do Oracle Scheduler.

O exemplo a seguir habilita o trabalho `SYS.CLEANUP_ONLINE_IND_BUILD` do Oracle Scheduler.

```
BEGIN
   rdsadmin.rdsadmin_dbms_scheduler.enable('SYS.CLEANUP_ONLINE_IND_BUILD');
END;
/
```

Modificar o intervalo de repetição do Oracle Scheduler para trabalhos do tipo CALENDAR

Para modificar o intervalo de repetição de um trabalho do Oracle Scheduler pertencente ao SYS do tipo `CALENDAR`, use o procedimento `rdsadmin.rdsadmin_dbms_scheduler.disable`.

Este procedimento usa os seguintes parâmetros comuns para tarefas do Oracle Scheduler:

- `name`
- `attribute`
- `value`

Para obter mais informações, consulte Parâmetros comuns para os procedimentos do Oracle Scheduler.

O exemplo a seguir modifica o intervalo de repetição do trabalho `SYS.CLEANUP_ONLINE_IND_BUILD` do Oracle Scheduler.

```
BEGIN
     rdsadmin.rdsadmin_dbms_scheduler.set_attribute(
          name      => 'SYS.CLEANUP_ONLINE_IND_BUILD',
          attribute => 'repeat_interval',
          value     => 'freq=daily;byday=FRI,SAT;byhour=20;byminute=0;bysecond=0');
END;
/
```

Modificar o intervalo de repetição do Oracle Scheduler para trabalhos do tipo NAMED

Alguns trabalhos do Oracle Scheduler usam um nome de programação em vez de um intervalo. Para esse tipo de trabalho, é necessário criar outra programação nomeada no esquema do usuário mestre. Use o procedimento `sys.dbms_scheduler.create_schedule` padrão do Oracle

para fazer isso. Além disso, use o `rdsadmin.rdsadmin_dbms_scheduler.set_attribute procedure` para atribuir a nova programação nomeada ao trabalho.

Este procedimento usa o seguinte parâmetro comum para tarefas do Oracle Scheduler:

- `name`
- `attribute`
- `value`

Para obter mais informações, consulte Parâmetros comuns para os procedimentos do Oracle Scheduler.

O exemplo a seguir modifica o intervalo de repetição do trabalho `SYS.BSLN_MAINTAIN_STATS_JOB` do Oracle Scheduler.

```
BEGIN
     DBMS_SCHEDULER.CREATE_SCHEDULE (
          schedule_name   => 'rds_master_user.new_schedule',
          start_date      => SYSTIMESTAMP,
          repeat_interval =>
 'freq=daily;byday=MON,TUE,WED,THU,FRI;byhour=0;byminute=0;bysecond=0',
          end_date        => NULL,
          comments        => 'Repeats daily forever');
END;
/

BEGIN
     rdsadmin.rdsadmin_dbms_scheduler.set_attribute (
          name      => 'SYS.BSLN_MAINTAIN_STATS_JOB',
          attribute => 'schedule_name',
          value     => 'rds_master_user.new_schedule');
END;
/
```

Desativar a confirmação automática para a criação de trabalhos do Oracle Scheduler

Quando o `DBMS_SCHEDULER.CREATE_JOB` cria trabalhos do Oracle Scheduler, ele os cria imediatamente e confirma as alterações. Talvez seja necessário incorporar a criação de trabalhos do Oracle Scheduler na transação do usuário para fazer o seguinte:

- Reverter o trabalho do Oracle Schedule quando a transação do usuário é revertida.

- Criar o trabalho do Oracle Scheduler quando a transação principal do usuário é confirmada.

Você pode usar o procedimento `rdsadmin.rdsadmin_dbms_scheduler.set_no_commit_flag` para ativar esse comportamento. Este procedimento não usa parâmetros. Você pode usar esse procedimento nas seguintes versões do RDS para Oracle:

- 21.0.0.0.ru-2022-07.rur-2022-07.r1 e posteriores
- 19.0.0.0.ru-2022-07.rur-2022-07.r1 e posteriores

O exemplo a seguir desativa a confirmação automática do Oracle Scheduler, cria um trabalho do Oracle Scheduler e, depois, reverte a transação. Como a confirmação automática está desativada, o banco de dados também reverte a criação do trabalho do Oracle Scheduler.

```
BEGIN
  rdsadmin.rdsadmin_dbms_scheduler.set_no_commit_flag;
  DBMS_SCHEDULER.CREATE_JOB(job_name   => 'EMPTY_JOB',
                            job_type   => 'PLSQL_BLOCK',
                            job_action => 'begin null; end;',
                            auto_drop  => false);
  ROLLBACK;
END;
/

PL/SQL procedure successfully completed.

SELECT * FROM DBA_SCHEDULER_JOBS WHERE JOB_NAME='EMPTY_JOB';

no rows selected
```

Diagnosticar problemas com instâncias de banco de dados do RDS para Oracle

O Oracle Database inclui uma infraestrutura de diagnóstico de falhas que pode ser usada para investigar problemas do banco de dados. Na terminologia do Oracle, um problema é um erro crítico, como um erro de código ou uma corrupção de dados. Um incidente é a ocorrência de um problema. Se o mesmo erro ocorrer três vezes, a infraestrutura mostrará três incidentes desse problema.

Para ter mais informações, consulte Diagnosing and resolving problems (Diagnosticar e resolver problemas) na documentação do Oracle Database.

O utilitário ADRCI (Automatic Diagnostic Repository Command Interpreter) é uma ferramenta da linha de comando Oracle usada para gerenciar dados de diagnóstico. Por exemplo, é possível usar essa ferramenta para investigar problemas e empacotar dados de diagnóstico. Um pacote de incidente inclui dados de diagnóstico de um incidente ou de todos os incidentes que fazem referência a um problema específico. Você pode fazer upload de um pacote de incidente, que é implementado como um arquivo .zip, para o suporte do Oracle.

Para oferecer uma experiência de serviço gerenciada, o Amazon RDS não fornece acesso ao shell para ADRCI. Use o pacote `rdsadmin.rdsadmin_adrci_util` do Amazon RDS para realizar tarefas de diagnóstico da instância de banco de dados do RDS para Oracle.

Usando as funções em `rdsadmin_adrci_util`, é possível listar e empacotar problemas e incidentes, além de mostrar arquivos de rastreamento. Todas as funções retornam um ID de tarefa. Esse ID faz parte do nome do arquivo de log que contém a saída de ADRCI, como em `dbtask-`*`task_id`*`.log`. O arquivo de log reside no diretório BDUMP. É possível baixar o arquivo de log seguindo o procedimento descrito em Como baixar um arquivo de log de banco de dados.

Parâmetros comuns para procedimentos de diagnóstico

Para executar tarefas de diagnóstico, use funções no pacote `rdsadmin.rdsadmin_adrci_util` do Amazon RDS. O pacote tem os seguintes parâmetros comuns.

Nome do parâmetro	Tipo de dados	Valores válidos	Padrão	Obrigatório	Descrição
`incident_id`	número	Um ID de incidente válido ou nulo	Nulo	Não	Se o valor for nulo, a função mostrará todos os incidentes. Se o valor não for nulo e representar um ID de incidente válido, a função mostrará o incidente especificado.

Nome do parâmetro	Tipo de dados	Valores válidos	Padrão	Obrigatório	Descrição
`problem_id`	número	Um ID de problema válido ou nulo	Nulo	Não	Se o valor for nulo, a função mostrará todos os problemas. Se o valor não for nulo e representar um ID de problema válido, a função mostrará o problema especificado.
`last`	número	Um inteiro válido maior que 0 ou nulo	Nulo	Não	Se o valor for nulo, a função exibirá no máximo 50 itens. Se o valor não for nulo, a função exibirá o número especificado.

Listar incidentes

Para listar incidentes de diagnóstico para o Oracle, use a função `rdsadmin.rdsadmin_adrci_util.list_adrci_incidents` do Amazon RDS. Você pode listar incidentes no modo básico ou detalhado. Por padrão, a função lista os 50 incidentes mais recentes.

Essa função usa os seguintes parâmetros comuns:

- `incident_id`
- `problem_id`
- `last`

Se você especificar `incident_id` e `problem_id`, `incident_id` substituirá `problem_id`. Para ter mais informações, consulte Parâmetros comuns para procedimentos de diagnóstico.

Essa função usa o parâmetro adicional a seguir.

Nome do parâmetro	Tipo de dados	Valores válidos	Padrão	Obrigatório	Descrição
`detail`	booliano	`TRUE` ou `FALSE`	`FALSE`	Não	Se for `TRUE`, a função listará incidentes no modo detalhado. Se for `FALSE`, a função listará incidentes no modo básico.

Para listar todos os incidentes, consulte a função `rdsadmin.rdsadmin_adrci_util.list_adrci_incidents` sem argumentos. A consulta retorna o ID da tarefa.

```
SQL> SELECT rdsadmin.rdsadmin_adrci_util.list_adrci_incidents AS task_id FROM DUAL;

TASK_ID
------------------
1590786706158-3126
```

Ou chame a função `rdsadmin.rdsadmin_adrci_util.list_adrci_incidents` sem argumentos e armazene a saída em uma variável de cliente SQL. Você pode usar a variável em outras instruções.

```
SQL> VAR task_id VARCHAR2(80);
SQL> EXEC :task_id := rdsadmin.rdsadmin_adrci_util.list_adrci_incidents;

PL/SQL procedure successfully completed.
```

Para ler o arquivo de log, chame o procedimento `rdsadmin.rds_file_util.read_text_file` do Amazon RDS. Forneça o ID da tarefa como parte do nome do arquivo. A saída a seguir mostra três incidentes: 53523, 53522 e 53521.

```
SQL> SELECT * FROM TABLE(rdsadmin.rds_file_util.read_text_file('BDUMP',
 'dbtask-'||:task_id||'.log'));

TEXT
```

```
--------------------------------------------------------------------------------
2020-05-29 21:11:46.193 UTC [INFO ] Listing ADRCI incidents.
2020-05-29 21:11:46.256 UTC [INFO ]
ADR Home = /rdsdbdata/log/diag/rdbms/orcl_a/ORCL:
*************************************************************************
INCIDENT_ID PROBLEM_KEY                                                 CREATE_TIME
----------- ------------------------------------------------------------
 ----------------------------------------
53523       ORA 700 [EVENT_CREATED_INCIDENT] [942] [SIMULATED_ERROR_003 2020-05-29
 20:15:20.928000 +00:00
53522       ORA 700 [EVENT_CREATED_INCIDENT] [942] [SIMULATED_ERROR_002 2020-05-29
 20:15:15.247000 +00:00
53521       ORA 700 [EVENT_CREATED_INCIDENT] [942] [SIMULATED_ERROR_001 2020-05-29
 20:15:06.047000 +00:00
3 rows fetched


2020-05-29 21:11:46.256 UTC [INFO ] The ADRCI incidents were successfully listed.
2020-05-29 21:11:46.256 UTC [INFO ] The task finished successfully.

14 rows selected.
```

Para listar determinado incidente, especifique seu ID usando o parâmetro `incident_id`. No exemplo a seguir, o arquivo de log é consultado somente para o incidente 53523.

```
SQL> EXEC :task_id :=
 rdsadmin.rdsadmin_adrci_util.list_adrci_incidents(incident_id=>53523);

PL/SQL procedure successfully completed.

SQL> SELECT * FROM TABLE(rdsadmin.rds_file_util.read_text_file('BDUMP',
 'dbtask-'||:task_id||'.log'));

TEXT
--------------------------------------------------------------------------------
2020-05-29 21:15:25.358 UTC [INFO ] Listing ADRCI incidents.
2020-05-29 21:15:25.426 UTC [INFO ]
ADR Home = /rdsdbdata/log/diag/rdbms/orcl_a/ORCL:
*************************************************************************
INCIDENT_ID          PROBLEM_KEY
 CREATE_TIME
-------------------- ------------------------------------------------------------
 ---------------------------------
```

```
53523                ORA 700 [EVENT_CREATED_INCIDENT] [942] [SIMULATED_ERROR_003
 2020-05-29 20:15:20.928000 +00:00
1 rows fetched


2020-05-29 21:15:25.427 UTC [INFO ] The ADRCI incidents were successfully listed.
2020-05-29 21:15:25.427 UTC [INFO ] The task finished successfully.

12 rows selected.
```

Listar problemas

Para listar problemas de diagnóstico para o Oracle, use a função `rdsadmin.rdsadmin_adrci_util.list_adrci_problems` do Amazon RDS.

Por padrão, a função lista os 50 problemas mais recentes.

Essa função usa os parâmetros comuns `problem_id` e `last`. Para ter mais informações, consulte Parâmetros comuns para procedimentos de diagnóstico.

Para obter o ID da tarefa para todos os problemas, chame a função `rdsadmin.rdsadmin_adrci_util.list_adrci_problems` sem argumentos e armazene a saída em uma variável de cliente SQL.

```
SQL> EXEC :task_id := rdsadmin.rdsadmin_adrci_util.list_adrci_problems;

PL/SQL procedure successfully completed.
```

Para ler o arquivo de log, chame a função `rdsadmin.rds_file_util.read_text_file`, fornecendo o ID da tarefa como parte do nome do arquivo. Na saída a seguir, o arquivo de log mostra três problemas: 1, 2 e 3.

```
SQL> SELECT * FROM TABLE(rdsadmin.rds_file_util.read_text_file('BDUMP',
 'dbtask-'||:task_id||'.log'));

TEXT
--------------------------------------------------------------------------------
2020-05-29 21:18:50.764 UTC [INFO ] Listing ADRCI problems.
2020-05-29 21:18:50.829 UTC [INFO ]
ADR Home = /rdsdbdata/log/diag/rdbms/orcl_a/ORCL:
*************************************************************************
```

```
PROBLEM_ID   PROBLEM_KEY                                                 LAST_INCIDENT
       LASTINC_TIME
---------- ---------------------------------------------------------- -------------
 -------------------------------
2          ORA 700 [EVENT_CREATED_INCIDENT] [942] [SIMULATED_ERROR_003 53523
 2020-05-29 20:15:20.928000 +00:00
3          ORA 700 [EVENT_CREATED_INCIDENT] [942] [SIMULATED_ERROR_002 53522
 2020-05-29 20:15:15.247000 +00:00
1          ORA 700 [EVENT_CREATED_INCIDENT] [942] [SIMULATED_ERROR_001 53521
 2020-05-29 20:15:06.047000 +00:00
3 rows fetched


2020-05-29 21:18:50.829 UTC [INFO ] The ADRCI problems were successfully listed.
2020-05-29 21:18:50.829 UTC [INFO ] The task finished successfully.

14 rows selected.
```

No exemplo a seguir, somente o problema 3 é listado.

```
SQL> EXEC :task_id := rdsadmin.rdsadmin_adrci_util.list_adrci_problems(problem_id=>3);

PL/SQL procedure successfully completed.
```

Para ler o arquivo de log do problema 3, chame `rdsadmin.rds_file_util.read_text_file`. Forneça o ID da tarefa como parte do nome do arquivo.

```
SQL> SELECT * FROM TABLE(rdsadmin.rds_file_util.read_text_file('BDUMP',
 'dbtask-'||:task_id||'.log'));

TEXT
--------------------------------------------------------------------------------
2020-05-29 21:19:42.533 UTC [INFO ] Listing ADRCI problems.
2020-05-29 21:19:42.599 UTC [INFO ]
ADR Home = /rdsdbdata/log/diag/rdbms/orcl_a/ORCL:
*************************************************************************
PROBLEM_ID PROBLEM_KEY                                                 LAST_INCIDENT
 LASTINC_TIME
---------- ---------------------------------------------------------- -------------
 -------------------------------
3          ORA 700 [EVENT_CREATED_INCIDENT] [942] [SIMULATED_ERROR_002 53522
 2020-05-29 20:15:15.247000 +00:00
1 rows fetched
```

```
2020-05-29 21:19:42.599 UTC [INFO ] The ADRCI problems were successfully listed.
2020-05-29 21:19:42.599 UTC [INFO ] The task finished successfully.

12 rows selected.
```

Criar pacotes de incidente

É possível criar pacotes de incidente usando a função do Amazon RDS. A saída é um arquivo .zip que você pode fornecer ao suporte do Oracle.

Essa função usa os seguintes parâmetros comuns:

- `problem_id`
- `incident_id`

Especifique um dos parâmetros anteriores. Se você especificar ambos os parâmetros, `incident_id` substituirá `problem_id`. Para ter mais informações, consulte Parâmetros comuns para procedimentos de diagnóstico.

Para criar um pacote para um incidente específico, chame a função `rdsadmin.rdsadmin_adrci_util.create_adrci_package` do Amazon RDS com o parâmetro `incident_id`. O exemplo a seguir cria um pacote para o incidente 53523.

```
SQL> EXEC :task_id :=
 rdsadmin.rdsadmin_adrci_util.create_adrci_package(incident_id=>53523);

PL/SQL procedure successfully completed.
```

Para ler o arquivo de log, chame `rdsadmin.rds_file_util.read_text_file`. Você pode fornecer o ID da tarefa como parte do nome do arquivo. A saída mostra que você gerou o pacote de incidente `ORA700EVE_20200529212043_COM_1.zip`.

```
SQL> SELECT * FROM TABLE(rdsadmin.rds_file_util.read_text_file('BDUMP',
 'dbtask-'||:task_id||'.log'));

TEXT
--------------------------------------------------------------------------------
2020-05-29 21:20:43.031 UTC [INFO ] The ADRCI package is being created.
```

```
2020-05-29 21:20:47.641 UTC [INFO ] Generated package 1 in file /rdsdbdata/log/trace/
ORA700EVE_20200529212043_COM_1.zip, mode complete
2020-05-29 21:20:47.642 UTC [INFO ] The ADRCI package was successfully created.
2020-05-29 21:20:47.642 UTC [INFO ] The task finished successfully.
```

Para empacotar dados de diagnóstico de determinado problema, especifique seu ID usando o parâmetro `problem_id`. No exemplo a seguir, os dados são empacotados apenas para o problema 3.

```
SQL> EXEC :task_id := rdsadmin.rdsadmin_adrci_util.create_adrci_package(problem_id=>3);

PL/SQL procedure successfully completed.
```

Para ler a saída da tarefa, chame `rdsadmin.rds_file_util.read_text_file`, fornecendo o ID da tarefa como parte do nome do arquivo. A saída mostra que você gerou o pacote de incidente `ORA700EVE_20200529212111_COM_1.zip`.

```
SQL> SELECT * FROM TABLE(rdsadmin.rds_file_util.read_text_file('BDUMP',
 'dbtask-'||:task_id||'.log'));

TEXT
--------------------------------------------------------------------------------
2020-05-29 21:21:11.050 UTC [INFO ] The ADRCI package is being created.
2020-05-29 21:21:15.646 UTC [INFO ] Generated package 2 in file /rdsdbdata/log/trace/
ORA700EVE_20200529212111_COM_1.zip, mode complete
2020-05-29 21:21:15.646 UTC [INFO ] The ADRCI package was successfully created.
2020-05-29 21:21:15.646 UTC [INFO ] The task finished successfully.
```

Também é possível baixar o arquivo de log. Para ter mais informações, consulte Como baixar um arquivo de log de banco de dados.

Mostrar arquivos de rastreamento

Você pode usar a função do Amazon RDS `rdsadmin.rdsadmin_adrci_util.show_adrci_tracefile` para listar arquivos de rastreamento no diretório de rastreamento e todos os diretórios de incidentes na página inicial atual do ADR. Você também pode mostrar o conteúdo dos arquivos de rastreamento e dos arquivos de rastreamento de incidentes.

Essa função usa o parâmetro a seguir.

Nome do parâmetro	Tipo de dados	Valores válidos	Padrão	Obrigatório	Descrição
`filename`	varchar2	Um nome de arquivo de rastreamento válido	Nulo	Não	Se o valor for nulo, a função mostrará todos os arquivos de rastreamento. Se não for nulo, a função mostrará o arquivo especificado.

Para mostrar o arquivo de rastreamento, chame a função `rdsadmin.rdsadmin_adrci_util.show_adrci_tracefile` do Amazon RDS.

```
SQL> EXEC :task_id := rdsadmin.rdsadmin_adrci_util.show_adrci_tracefile;

PL/SQL procedure successfully completed.
```

Para listar os nomes dos arquivos de rastreamento, chame o procedimento `rdsadmin.rds_file_util.read_text_file` do Amazon RDS, fornecendo o ID da tarefa como parte do nome do arquivo.

```
SQL> SELECT * FROM TABLE(rdsadmin.rds_file_util.read_text_file('BDUMP',
 'dbtask-'||:task_id||'.log')) WHERE TEXT LIKE '%/alert_%';

TEXT
--------------------------------------------------------------------
     diag/rdbms/orcl_a/ORCL/trace/alert_ORCL.log.2020-05-28
     diag/rdbms/orcl_a/ORCL/trace/alert_ORCL.log.2020-05-27
     diag/rdbms/orcl_a/ORCL/trace/alert_ORCL.log.2020-05-26
     diag/rdbms/orcl_a/ORCL/trace/alert_ORCL.log.2020-05-25
     diag/rdbms/orcl_a/ORCL/trace/alert_ORCL.log.2020-05-24
     diag/rdbms/orcl_a/ORCL/trace/alert_ORCL.log.2020-05-23
     diag/rdbms/orcl_a/ORCL/trace/alert_ORCL.log.2020-05-22
     diag/rdbms/orcl_a/ORCL/trace/alert_ORCL.log.2020-05-21
     diag/rdbms/orcl_a/ORCL/trace/alert_ORCL.log

9 rows selected.
```

No exemplo a seguir, é gerada uma saída para `alert_ORCL.log`.

```
SQL> EXEC :task_id := rdsadmin.rdsadmin_adrci_util.show_adrci_tracefile('diag/rdbms/
orcl_a/ORCL/trace/alert_ORCL.log');

PL/SQL procedure successfully completed.
```

Para ler o arquivo de log, chame `rdsadmin.rds_file_util.read_text_file`. Forneça o ID da tarefa como parte do nome do arquivo. A saída mostra as primeiras 10 linhas de alert_ORCL.log.

```
SQL> SELECT * FROM TABLE(rdsadmin.rds_file_util.read_text_file('BDUMP',
 'dbtask-'||:task_id||'.log')) WHERE ROWNUM <= 10;

TEXT
--------------------------------------------------------------------------------
2020-05-29 21:24:02.083 UTC [INFO ] The trace files are being displayed.
2020-05-29 21:24:02.128 UTC [INFO ] Thu May 28 23:59:10 2020
Thread 1 advanced to log sequence 2048 (LGWR switch)
  Current log# 3 seq# 2048 mem# 0: /rdsdbdata/db/ORCL_A/onlinelog/o1_mf_3_hbl2p8xs_.log
Thu May 28 23:59:10 2020
Archived Log entry 2037 added for thread 1 sequence 2047 ID 0x5d62ce43 dest 1:
Fri May 29 00:04:10 2020
Thread 1 advanced to log sequence 2049 (LGWR switch)
  Current log# 4 seq# 2049 mem# 0: /rdsdbdata/db/ORCL_A/onlinelog/o1_mf_4_hbl2qgmh_.log
Fri May 29 00:04:10 2020

10 rows selected.
```

Também é possível baixar o arquivo de log. Para ter mais informações, consulte Como baixar um arquivo de log de banco de dados.

Realização de tarefas comuns diversas para instâncias de banco de dados Oracle

Veja a seguir como executar diversas tarefas de DBA nas instâncias de banco de dados do Amazon RDS que executam o Oracle. Para oferecer uma experiência de serviço gerenciado, o Amazon RDS não fornece acesso ao shell para instâncias de bancos de dados e restringe o acesso a determinados procedimentos e tabelas do sistema que exigem privilégios avançados.

Tópicos

- Criar e eliminar diretórios no espaço de armazenamento de dados principal
- Listagem de arquivos no diretório de uma instância de banco de dados
- Leitura de arquivos no diretório de uma instância de banco de dados
- Acessar arquivos do Opatch
- Gerenciando de tarefas do advisor
- Transportar espaços para tabela

Criar e eliminar diretórios no espaço de armazenamento de dados principal

Para criar diretórios, use o procedimento `rdsadmin.rdsadmin_util.create_directory` do Amazon RDS. É possível criar até 10.000 diretórios, todas localizados no seu espaço de armazenamento físico de dados principal. Para criar diretórios, use o procedimento `rdsadmin.rdsadmin_util.drop_directory` do Amazon RDS.

Os procedimentos `create_directory` e `drop_directory` têm o parâmetro necessário a seguir.

Nome do parâmetro	Tipo de dados	Padrão	Obrigatório	Descrição
`p_directory_name`	VARCHAR2	—	Sim	O nome do diretório.

O exemplo a seguir cria um novo diretório chamado `PRODUCT_DESCRIPTIONS`.

```
EXEC rdsadmin.rdsadmin_util.create_directory(p_directory_name =>
 'product_descriptions');
```

O dicionário de dados armazena o nome do diretório em maiúsculas. Você pode listar os diretórios consultando `DBA_DIRECTORIES`. O sistema escolhe automaticamente o nome do caminho do host real. O exemplo a seguir obtém o caminho do diretório para o diretório chamado `PRODUCT_DESCRIPTIONS`:

```
SELECT DIRECTORY_PATH
  FROM DBA_DIRECTORIES
 WHERE DIRECTORY_NAME='PRODUCT_DESCRIPTIONS';
```

```
DIRECTORY_PATH
---------------------------------------
/rdsdbdata/userdirs/01
```

O nome de usuário mestre para a instância de banco de dados tem privilégios de leitura e gravação no novo diretório e pode conceder acesso a outros usuários. Privilégios `EXECUTE` não estão disponíveis para diretórios em uma instância de banco de dados. Os diretórios são criados no seu espaço de armazenamento físico de dados principal e consumirão espaço e largura de banda de E/S.

O exemplo a seguir elimina o diretório chamado `PRODUCT_DESCRIPTIONS`.

```
EXEC rdsadmin.rdsadmin_util.drop_directory(p_directory_name => 'product_descriptions');
```

> **Note**
>
> Também é possível descartar um diretório usando o comando do Oracle SQ `DROP DIRECTORY`.

Descartar um diretório não remove seu conteúdo. Como o procedimento `rdsadmin.rdsadmin_util.create_directory` pode reutilizar nomes de caminhos, os arquivos em diretórios eliminados podem aparecer em um diretório recém-criado. Antes de eliminar um diretório, recomendamos que você use `UTL_FILE.FREMOVE` para remover arquivos do diretório. Para obter mais informações, consulte Procedimento FREMOVE na documentação da Oracle.

Listagem de arquivos no diretório de uma instância de banco de dados

Para listar os arquivos em um diretório, use o procedimento `rdsadmin.rds_file_util.listdir` do Amazon RDS. Esse procedimento não é compatível com uma réplica da Oracle. O procedimento `listdir` tem os seguintes parâmetros.

Nome do parâmetro	Tipo de dados	Padrão	Obrigatório	Descrição
`p_directory`	varchar2	—	Sim	O nome do diretório a ser listado.

O exemplo a seguir concede privilégios de leitura/gravação no diretório PRODUCT_DESCRIPTIONS para o usuário `rdsadmin` e, em seguida, lista os arquivos neste diretório.

```
GRANT READ,WRITE ON DIRECTORY PRODUCT_DESCRIPTIONS TO rdsadmin;
SELECT * FROM TABLE(rdsadmin.rds_file_util.listdir(p_directory =>
 'PRODUCT_DESCRIPTIONS'));
```

Leitura de arquivos no diretório de uma instância de banco de dados

Para ler um arquivo de texto, use o procedimento `rdsadmin.rds_file_util.read_text_file` do Amazon RDS. O procedimento `read_text_file` tem os seguintes parâmetros.

Nome do parâmetro	Tipo de dados	Padrão	Obrigatório	Descrição
`p_directory`	varchar2	—	Sim	O nome do diretório que contém o arquivo.
`p_filename`	varchar2	—	Sim	O nome do arquivo a ser lido.

O exemplo a seguir cria o arquivo `rice.txt` no diretório PRODUCT_DESCRIPTIONS.

```
declare
  fh sys.utl_file.file_type;
begin
  fh := utl_file.fopen(location=>'PRODUCT_DESCRIPTIONS', filename=>'rice.txt',
 open_mode=>'w');
  utl_file.put(file=>fh, buffer=>'AnyCompany brown rice, 15 lbs');
  utl_file.fclose(file=>fh);
end;
/
```

O exemplo a seguir lê o arquivo `rice.txt` do diretório PRODUCT_DESCRIPTIONS.

```
SELECT * FROM TABLE
    (rdsadmin.rds_file_util.read_text_file(
        p_directory => 'PRODUCT_DESCRIPTIONS',
```

```
        p_filename  => 'rice.txt'));
```

Acessar arquivos do Opatch

O Opatch é um utilitário do Oracle que permite a aplicação e a reversão de patches para o software do Oracle. O mecanismo do Oracle para determinar quais patches foram aplicados a um banco de dados é o comando `opatch lsinventory`. Para abrir solicitações de serviço para clientes de Traga sua própria licença (BYOL – Bring Your Own Licence), o suporte do Oracle solicita o arquivo `lsinventory` e, às vezes, o arquivo `lsinventory_detail` gerado pelo Opatch.

Para oferecer uma experiência de serviço gerenciada, o Amazon RDS não fornece acesso ao shell para o Opatch. Em vez disso, o `lsinventory-`*`dbv`*`.txt` no diretório BDUMP contém as informações de patch relacionadas à versão atual do mecanismo. Quando você executa um upgrade principal ou secundário, o Amazon RDS atualiza `lsinventory-`*`dbv`*`.txt` em até uma hora após a aplicação do patch. Para verificar os patches aplicados, leia `lsinventory-`*`dbv`*`.txt`. Essa ação é semelhante a executar o comando `opatch lsinventory`.

> **Note**
>
> Os exemplos nesta seção pressupõem que o diretório BDUMP seja chamado `BDUMP`. Em uma réplica de leitura, o nome do diretório BDUMP é diferente. Para saber como obter o nome BDUMP consultando `V$DATABASE.DB_UNIQUE_NAME` em uma réplica de leitura, consulte Listar arquivos.

Os arquivos de inventário usam a convenção de nomenclatura do Amazon RDS `lsinventory-`*`dbv`*`.txt` e `lsinventory_detail-`*`dbv`*`.txt`, onde *`dbv`* é o nome completo da versão do banco de dados. O arquivo `lsinventory-`*`dbv`*`.txt` está disponível em todas as versões do banco de dados. O `lsinventory_detail-`*`dbv`*`.txt` correspondente está disponível em 19.0.0.0, ru-2020-01.rur-2020-01.r1 ou posterior.

Por exemplo, se a versão do seu banco de dados for 19.0.0.0.ru-2021-07.rur-2021-07.r1, os arquivos de inventário terão os nomes a seguir.

```
lsinventory-19.0.0.0.ru-2021-07.rur-2021-07.r1.txt
lsinventory_detail-19.0.0.0.ru-2021-07.rur-2021-07.r1.txt
```

Baixe os arquivos que correspondem à versão atual do mecanismo de banco de dados.

Console

Como baixar um arquivo de inventário usando o console

1. Abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Databases (Bancos de dados).
3. Escolha o nome da instância de banco de dados que contém o arquivo de log que você deseja visualizar.
4. Escolha a guia Logs & events (Logs e eventos).
5. Role para baixo até a seção Logs.
6. Na seção Logs, procure por `lsinventory`.
7. Escolha o arquivo que você deseja acessar e selecione Download (Baixar).

SQL

Para ler o `lsinventory-`*`dbv`*`.txt` em um cliente SQL, é possível usar uma instrução `SELECT`. Para esta técnica, use uma das seguintes funções `rdsadmin`: `rdsadmin.rds_file_util.read_text_file` ou `rdsadmin.tracefile_listing`.

Na consulta de exemplo a seguir, substitua *dbv* pela versão do seu banco de dados Oracle. Por exemplo, a versão do seu banco de dados pode ser 19.0.0.0.ru-2020-04.rur-2020-04.r1.

```
SELECT text
FROM   TABLE(rdsadmin.rds_file_util.read_text_file('BDUMP', 'lsinventory-dbv.txt'));
```

PL/SQL

Para ler o `lsinventory-`*`dbv`*`.txt` em um cliente SQL, é possível escrever um programa PL/SQL. Esse programa usa `utl_file` para ler o arquivo e `dbms_output` para imprimi-lo. Esses são pacotes fornecidos pelo Oracle.

No programa de exemplo a seguir, substitua *dbv* pela versão do seu banco de dados Oracle. Por exemplo, a versão do seu banco de dados pode ser 19.0.0.0.ru-2020-04.rur-2020-04.r1.

```
SET SERVEROUTPUT ON
DECLARE
  v_file              SYS.UTL_FILE.FILE_TYPE;
  v_line              VARCHAR2(1000);
  v_oracle_home_type  VARCHAR2(1000);
```

```
  c_directory         VARCHAR2(30) := 'BDUMP';
  c_output_file       VARCHAR2(30) := 'lsinventory-dbv.txt';
BEGIN
  v_file := SYS.UTL_FILE.FOPEN(c_directory, c_output_file, 'r');
  LOOP
    BEGIN
      SYS.UTL_FILE.GET_LINE(v_file, v_line,1000);
      DBMS_OUTPUT.PUT_LINE(v_line);
    EXCEPTION
      WHEN no_data_found THEN
        EXIT;
    END;
  END LOOP;
END;
/
```

Ou consulte `rdsadmin.tracefile_listing` e transfira a saída para um arquivo. O exemplo a seguir transfere a saída para `/tmp/tracefile.txt`.

```
SPOOL /tmp/tracefile.txt
SELECT *
FROM   rdsadmin.tracefile_listing
WHERE  FILENAME LIKE 'lsinventory%';
SPOOL OFF;
```

Gerenciando de tarefas do advisor

O banco de dados Oracle inclui vários advisors. Cada advisor suporta tarefas automatizadas e manuais. Você pode usar procedimentos no pacote `rdsadmin.rdsadmin_util` para gerenciar algumas tarefas do advisor.

Os procedimentos de tarefas do advisor estão disponíveis nas seguintes versões do mecanismo:

- Oracle Database 21c (21.0.0)
- Versão 19.0.0.0.ru-2021-01.rur-2021-01.r1 e versões posteriores do Oracle Database 19c

  Para obter mais informações, consulte Versão 19.0.0.0.ru-2021-01.rur-2021-01.r1 em Notas de lançamento do Amazon RDS for Oracle.

Tópicos

- Definição de parâmetros para tarefas do advisor

- Desativação de AUTO_STATS_ADVISOR_TASK
- Reativação de AUTO_STATS_ADVISOR_TASK

Definição de parâmetros para tarefas do advisor

Para definir parâmetros para algumas tarefas do advisor, use o procedimento do Amazon RDS `rdsadmin.rdsadmin_util.advisor_task_set_parameter`. O procedimento `advisor_task_set_parameter` tem os seguintes parâmetros.

Nome do parâmetro	Tipo de dados	Padrão	Obrigatório	Descrição
`p_task_name`	varchar2	—	Sim	O nome da tarefa do advisor cujos parâmetros você deseja alterar. Os valores a seguir são válidos: • `AUTO_STATS_ADVISOR_TASK` • `INDIVIDUAL_STATS_ADVISOR_TASK` • `SYS_AUTO_SPM_EVOLVE_TASK` • `SYS_AUTO_SQL_TUNING_TASK`

`p_parameter`	varchar2	—	Sim	O nome do parâmetro da tarefa. Para localizar parâmetros válidos para uma tarefa do advisor, execute a seguinte consulta. Substitua *`p_task_name`* com um valor válido para `p_task_name` :
				`COL PARAMETER_NAME FORMAT a30`
				`COL PARAMETER_VALUE FORMAT a30`
				`SELECT PARAMETER_NAME, PARAMETER_VALUE`
				`FROM DBA_ADVISOR_PARAMETERS`
				`WHERE TASK_NAME='` *`p_task_name`* `'`
				`AND PARAMETER_VALUE != 'UNUSED'`
				`ORDER BY PARAMETER_NAME;`

`p_value`	varchar2	—	Sim	O valor de um parâmetro de tarefa. Para localizar valores válidos para parâmetros de

Nome do parâmetro	Tipo de dados	Padrão	Obrigatório	Descrição

				tarefa, execute a seguinte consulta. Substitua *p_task_name* com um valor válido para p_task_name :
				`COL PARAMETER_NAME FORMAT a30`
				`COL PARAMETER_VALUE FORMAT a30`
				`SELECT PARAMETER_NAME, PARAMETER_VALUE`
				`FROM DBA_ADVISOR_PARAMETERS`
				`WHERE TASK_NAME='  p_task_name '`
				`AND PARAMETER_VALUE != 'UNUSED'`
				`ORDER BY PARAMETER_NAME;`

Os seguintes programas PL/SQL definem ACCEPT_PLANS como FALSE para SYS_AUTO_SPM_EVOLVE_TASK. A tarefa automatizada do Gerenciamento de Plano SQL verifica os planos e gera um relatório de suas descobertas, mas não evolui os planos automaticamente. Você pode usar um relatório para identificar novas listas de referência do plano SQL e aceitá-las manualmente.

```
BEGIN
  rdsadmin.rdsadmin_util.advisor_task_set_parameter(
    p_task_name => 'SYS_AUTO_SPM_EVOLVE_TASK',
    p_parameter => 'ACCEPT_PLANS',
    p_value     => 'FALSE');
END;
```

Os seguintes programas PL/SQL definem EXECUTION_DAYS_TO_EXPIRE como 10 para AUTO_STATS_ADVISOR_TASK. A tarefa predefinida AUTO_STATS_ADVISOR_TASK é executada automaticamente na janela de manutenção uma vez por dia. O exemplo define o período de retenção para a execução da tarefa como 10 dias.

```
BEGIN
  rdsadmin.rdsadmin_util.advisor_task_set_parameter(
    p_task_name => 'AUTO_STATS_ADVISOR_TASK',
    p_parameter => 'EXECUTION_DAYS_TO_EXPIRE',
    p_value     => '10');
```

```
END;
```

Desativação de AUTO_STATS_ADVISOR_TASK

Para desativar AUTO_STATS_ADVISOR_TASK, use o procedimento do Amazon RDS `rdsadmin.rdsadmin_util.advisor_task_drop`. O procedimento `advisor_task_drop` aceita o parâmetro a seguir.

Nome do parâmetro	Tipo de dados	Padrão	Obrigatório	Descrição
`p_task_name`	varchar2	—	Sim	O nome da tarefa do advisor a ser desabilitada. O único valor válido é `AUTO_STATS_ADVISOR_TASK` .

Execute o seguinte comando: AUTO_STATS_ADVISOR_TASK.

```
EXEC rdsadmin.rdsadmin_util.advisor_task_drop('AUTO_STATS_ADVISOR_TASK')
```

Você pode reabilitar AUTO_STATS_ADVISOR_TASK usando `rdsadmin.rdsadmin_util.dbms_stats_init`.

Reativação de AUTO_STATS_ADVISOR_TASK

Para reativar AUTO_STATS_ADVISOR_TASK, use o procedimento do Amazon RDS `rdsadmin.rdsadmin_util.dbms_stats_init`. O procedimento `dbms_stats_init` não recebe parâmetros.

O comando a seguir reativa AUTO_STATS_ADVISOR_TASK.

```
EXEC rdsadmin.rdsadmin_util.dbms_stats_init()
```

Transportar espaços para tabela

Use o pacote `rdsadmin.rdsadmin_transport_util` do Amazon RDS para copiar um conjunto de espaços para tabela de um banco de dados Oracle on-premises para uma instância de banco de dados do RDS para Oracle. No nível físico, o atributo de espaço para tabelas copia

incrementalmente os arquivos de dados de origem e os arquivos de metadados para a instância de destino. Você pode transferir os arquivos usando o Amazon EFS ou o Amazon S3. Para ter mais informações, consulte Migrar usando espaços de tabela transportáveis da Oracle.

Tópicos

- Importar espaços para tabela transportados para a instância de banco de dados
- Importar metadados de espaços para tabela transportáveis para a instância de banco de dados
- Listar arquivos órfãos após a importação de um espaço para tabela
- Excluir arquivos de dados órfãos após a importação de espaços para tabela

Importar espaços para tabela transportados para a instância de banco de dados

Use o procedimento `rdsadmin.rdsadmin_transport_util.import_xtts_tablespaces` para restaurar espaços para tabela que você exportou anteriormente de uma instância de banco de dados de origem. Na fase de transporte, você fará backup dos espaços para tabela somente leitura, exportará os metadados do Data Pump, transferirá esses arquivos para a instância de banco de dados de destino e, depois, importará os espaços para tabela. Para ter mais informações, consulte Fase 4: Transportar os espaços de tabela.

Sintaxe

```
FUNCTION import_xtts_tablespaces(
    p_tablespace_list IN CLOB,
    p_directory_name  IN VARCHAR2,
    p_platform_id     IN NUMBER DEFAULT 13,
    p_parallel        IN INTEGER DEFAULT 0) RETURN VARCHAR2;
```

Parâmetros

Nome do parâmetro	Tipo de dados	Padrão	Obrigatório	Descrição
`p_tablespace_list`	`CLOB`	—	Sim	A lista de espaços para tabela a serem importados.

Nome do parâmetro	Tipo de dados	Padrão	Obrigatório	Descrição
p_directory_name	VARCHAR2	—	Sim	O diretório que contém os backups de espaços para tabela.
p_platform_id	NUMBER	13	Não	Forneça um ID de plataforma que corresponda ao especificado durante a fase de backup. Para encontrar uma lista de plataformas, consulte V$TRANSPORTABLE_PLATFORM . A plataforma padrão é Linux x86 de 64 bits, que é pouco comum.
p_parallel	INTEGER	0	Não	O grau de paralelismo. Por padrão, o paralelismo está desabilitado.

Exemplos

O exemplo a seguir importa os espaços de tabela *TBS1*, *TBS2* e *TBS3* do diretório *DATA_PUMP_DIR*. A plataforma de origem é de sistemas baseados em AIX (64 bits), que tem o ID da plataforma de 6. É possível encontrar os IDs da plataforma consultando V$TRANSPORTABLE_PLATFORM.

```
VAR task_id CLOB

BEGIN
  :task_id:=rdsadmin.rdsadmin_transport_util.import_xtts_tablespaces(
        'TBS1,TBS2,TBS3',
        'DATA_PUMP_DIR',
        p_platform_id => 6);
END;
/
```

```
PRINT task_id
```

Importar metadados de espaços para tabela transportáveis para a instância de banco de dados

Use o procedimento `rdsadmin.rdsadmin_transport_util.import_xtts_metadata` para importar os metadados de espaços para tabela transportáveis para a instância de banco de dados do RDS para Oracle. Durante a operação, o status da importação de metadados é mostrado na tabela `rdsadmin.rds_xtts_operation_info`. Para ter mais informações, consulte Etapa 5: Importar os metadados de espaços de tabela na instância de banco de dados de destino.

Sintaxe

```
PROCEDURE import_xtts_metadata(
    p_datapump_metadata_file IN SYS.DBA_DATA_FILES.FILE_NAME%TYPE,
    p_directory_name         IN VARCHAR2,
    p_exclude_stats          IN BOOLEAN DEFAULT FALSE,
    p_remap_tablespace_list  IN CLOB DEFAULT NULL,
    p_remap_user_list        IN CLOB DEFAULT NULL);
```

Parâmetros

Nome do parâmetro	Tipo de dados	Padrão	Obrigatório	Descrição
`p_datapump_metadata_file`	`SYS.DBA_DATA_FILES.FILE_NAME%TYPE`	—	Sim	O nome do arquivo Oracle Data Pump que contém os metadados de seus espaços para tabela transportáveis.
`p_directory_name`	`VARCHAR2`	—	Sim	O diretório que contém o arquivo Data Pump.
`p_exclude_stats`	`BOOLEAN`	`FALSE`	Não	Sinalizador que indica se as estatísticas devem ser excluídas.

Nome do parâmetro	Tipo de dados	Padrão	Obrigatório	Descrição
`p_remap_tablespace_list`	`CLOB`	NULL	Não	Uma lista de espaços para tabela a serem associados novamente durante a importação de metadados. Use o formato *`from_tbs`*`:`*`to_tbs`*. Por exemplo, especifique `users:user_data`.
`p_remap_user_list`	`CLOB`	NULL	Não	Uma lista de esquemas de usuário a serem novamente associados durante a importação de metadados. Use o formato *`from_schema_name`*`:`*`to_schema_name`*. Por exemplo, especifique `hr:human_resources`.

Exemplos

O exemplo importa os metadados de espaço para tabelas do arquivo *`xttdump.dmp`*, que está localizado no diretório *`DATA_PUMP_DIR`*.

```
BEGIN
  rdsadmin.rdsadmin_transport_util.import_xtts_metadata('xttdump.dmp','DATA_PUMP_DIR');
END;
```

```
/
```

Listar arquivos órfãos após a importação de um espaço para tabela

Use o procedimento `rdsadmin.rdsadmin_transport_util.list_xtts_orphan_files` para listar arquivos de dados que ficaram órfãos após a importação de espaços para tabela. Depois de identificar os arquivos de dados, você pode excluí-los chamando `rdsadmin.rdsadmin_transport_util.cleanup_incomplete_xtts_import`.

Sintaxe

```
FUNCTION list_xtts_orphan_files RETURN xtts_orphan_files_list_t PIPELINED;
```

Exemplos

O exemplo a seguir executa o procedimento `rdsadmin.rdsadmin_transport_util.list_xtts_orphan_files`. A saída mostra dois arquivos de dados que estão órfãos.

```
SQL> SELECT * FROM TABLE(rdsadmin.rdsadmin_transport_util.list_xtts_orphan_files);

FILENAME       FILESIZE
-------------- ---------
datafile_7.dbf 104865792
datafile_8.dbf 104865792
```

Excluir arquivos de dados órfãos após a importação de espaços para tabela

Use o procedimento `rdsadmin.rdsadmin_transport_util.list_xtts_orphan_files` para excluir arquivos de dados que ficaram órfãos após a importação de espaços para tabela. A execução desse comando gera um arquivo de log que usa o formato do nome `rds-xtts-delete_xtts_orphaned_files-`*`YYYY-MM-DD.HH24-MI-SS.FF`*`.log` no diretório BDUMP. Use o procedimento `rdsadmin.rdsadmin_transport_util.cleanup_incomplete_xtts_import` para encontrar os arquivos órfãos. Você pode ler o arquivo de log chamando o procedimento `rdsadmin.rds_file_util.read_text_file`. Para ter mais informações, consulte Fase 6: Limpar os arquivos restantes.

Sintaxe

```
PROCEDURE cleanup_incomplete_xtts_import(
```

```
    p_directory_name IN VARCHAR2);
```

Parâmetros

Nome do parâmetro	Tipo de dados	Padrão	Obrigatório	Descrição
p_directory_name	VARCHAR2	—	Sim	O diretório que contém os arquivos de dados órfãos.

Exemplos

O exemplo a seguir exclui os arquivos de dados órfãos no *DATA_PUMP_DIR*.

```
BEGIN
  rdsadmin.rdsadmin_transport_util.cleanup_incomplete_xtts_import('DATA_PUMP_DIR');
END;
/
```

O exemplo a seguir lê o arquivo de log gerado pelo comando anterior.

```
SELECT *
FROM TABLE(rdsadmin.rds_file_util.read_text_file(
       p_directory => 'BDUMP',
       p_filename  => 'rds-xtts-
delete_xtts_orphaned_files-2023-06-01.09-33-11.868894000.log'));

TEXT
--------------------------------------------------------------------------------
orphan transported datafile datafile_7.dbf deleted.
orphan transported datafile datafile_8.dbf deleted.
```

Configurar recursos avançados do RDS para Oracle

O RDS para Oracle é compatível com vários recursos avançados, inclusive o HugePages, um armazenamento de instância e tipos de dados estendidos.

Tópicos

- Armazenar dados temporários em um armazenamento de instância do RDS for Oracle
- Ativar o HugePages para uma instância do RDS para Oracle
- Ativar tipos de dados estendidos no RDS para Oracle

Armazenar dados temporários em um armazenamento de instância do RDS for Oracle

Use um armazenamento de instância para os espaços de tabela temporários e o Database Smart Flash Cache (o cache flash) em classes de instância de banco de dados do RDS for Oracle compatíveis.

Tópicos

- Visão geral do armazenamento de instância do RDS for Oracle
- Ativar um armazenamento de instância do RDS for Oracle
- Configurar um armazenamento de instância do RDS for Oracle
- Trabalhar com um armazenamento de instância em uma réplica de leitura da Oracle
- Configurar um grupo de espaços de tabela temporários em um armazenamento de instância e no Amazon EBS
- Remover um armazenamento de instância do RDS for Oracle

Visão geral do armazenamento de instância do RDS for Oracle

Um armazenamento de instância fornece armazenamento temporário em nível de bloco para a instância de banco de dados do RDS for Oracle. Você pode usar um armazenamento de instância para armazenamento temporário de informações que mudam com frequência.

Um armazenamento de instância é baseado em dispositivos Non-Volatile Memory Express (NVMe) anexados fisicamente ao computador host. Esse armazenamento é otimizado para baixa latência, performance de E/S aleatória e throughput de leitura sequencial.

O tamanho do armazenamento de instância varia de acordo com o tipo de instância de banco de dados. Para obter mais informações sobre o armazenamento de instância, consulte Armazenamento de instância do Amazon EC2 no Amazon Elastic Compute Cloud Manual do usuário para instâncias do Linux.

Tópicos

- Tipos de dados no armazenamento de instância do RDS for Oracle
- Benefícios do armazenamento de instância do RDS for Oracle
- Classes de instância compatíveis com o armazenamento de instância do RDS for Oracle
- Versões de mecanismo compatíveis com o armazenamento de instância do RDS for Oracle
- Regiões da AWS compatíveis com o armazenamento de instância do RDS for Oracle
- Custo do armazenamento de instância do RDS para Oracle

Tipos de dados no armazenamento de instância do RDS for Oracle

Você pode colocar os seguintes tipos de dados temporários do RDS for Oracle em um armazenamento de instância:

Um espaço de tabela temporário

O Oracle Database usa espaços de tabela temporários para armazenar resultados de consultas intermediárias que não cabem na memória. Consultas maiores podem gerar grandes quantidades de dados intermediários que precisam ser armazenados temporariamente em cache, mas não precisam ser mantidos. Especificamente, um espaço de tabela temporário é útil para classificações, agregações de hash e uniões. Se sua instância de banco de dados do RDS for Oracle usar a Enterprise Edition ou a Standard Edition 2, você poderá colocar um espaço de tabela temporário em um armazenamento de instância.

O cache flash

O cache flash melhora a performance das leituras aleatórias de bloco único no caminho convencional. Uma prática recomendada é dimensionar o cache para acomodar a maior parte do conjunto de dados ativo. Se sua instância de banco de dados do RDS for Oracle usar a Enterprise Edition, você poderá colocar o cache flash em um armazenamento de instância.

Por padrão, um armazenamento de instância é configurado para um espaço de tabela temporário, mas não para o cache flash. Não é possível colocar arquivos de dados do Oracle e arquivos de log do banco de dados em um armazenamento de instância.

Benefícios do armazenamento de instância do RDS for Oracle

Você pode considerar o uso de um armazenamento de instância para armazenar arquivos e caches temporários que você pode perder. Se você quiser melhorar a performance do banco de dados ou se uma workload crescente estiver causando problemas de performance para seu armazenamento do Amazon EBS, considere escalar para uma classe de instância que seja compatível com um armazenamento de instância.

Ao colocar o espaço de tabela temporário e o cache flash em um armazenamento de instância, você obtém os seguintes benefícios:

- Latências de leitura mais baixas
- Throughput maior
- Menor carga nos volumes do Amazon EBS
- Custos mais baixos de armazenamento e snapshot devido à redução da carga do Amazon EBS
- Menos necessidade de provisionar IOPS alta, possivelmente reduzindo o custo geral

Ao colocar o espaço de tabela temporário no armazenamento de instância, você aumenta imediatamente a performance das consultas que usam espaço temporário. Quando você coloca o cache flash no armazenamento de instância, as leituras de blocos em cache normalmente têm uma latência muito menor do que as leituras do Amazon EBS. O cache flash precisa ser “aquecido” antes de oferecer benefícios de performance. O cache se aquece sozinho porque o banco de dados grava blocos no cache flash à medida que eles saem do cache do buffer do banco de dados.

> **Note**
>
> Em alguns casos, o cache flash causa sobrecarga de performance devido ao gerenciamento do cache. Antes de ativar o cache flash em um ambiente de produção, recomendamos que você analise sua workload e teste o cache em um ambiente de teste.

Classes de instância compatíveis com o armazenamento de instância do RDS for Oracle

O Amazon RDS é compatível com o armazenamento de instância para as seguintes classes de instância de banco de dados:

- db.m5d
- db.r5d
- db.x2idn
- db.x2iedn

O RDS para Oracle é compatível com as classes de instância de banco de dados anteriores somente para o modelo de licenciamento BYOL. Para ter mais informações, consulte Classes de instância de banco de dados compatíveis do RDS para Oracle e Traga a sua própria licença (BYOL) para EE e SE2.

Para ver o armazenamento total de instâncias para os tipos compatíveis de instância de banco de dados, execute o comando a seguir na CLI da AWS.

Example

```
aws ec2 describe-instance-types \
  --filters "Name=instance-type,Values=*5d.*large*" \
  --query "InstanceTypes[?contains(InstanceType,'m5d')||contains(InstanceType,'r5d')]
[InstanceType, InstanceStorageInfo.TotalSizeInGB]" \
  --output table
```

O comando anterior retorna o tamanho bruto do dispositivo para o armazenamento de instância. O RDS for Oracle usa uma pequena parte desse espaço para configuração. O espaço no armazenamento de instância que está disponível para espaços de tabela temporários ou para o cache flash é um pouco menor.

Versões de mecanismo compatíveis com o armazenamento de instância do RDS for Oracle

O armazenamento de instância é compatível com as seguintes versões de mecanismo do RDS for Oracle:

- Versões 21.0.0.0.ru-2022-01.rur-2022-01.r1 ou Oracle Database 21c posterior
- 19.0.0.0.ru-2021-10.rur-2021-10.r1 ou versões posteriores do Oracle Database 19c

Regiões da AWS compatíveis com o armazenamento de instância do RDS for Oracle

O armazenamento de instância está disponível em todas as Regiões da AWS onde um ou mais desses tipos de instância são compatíveis. Para obter mais informações sobre as classes de instância db.m5d e db.r5d, consulte Classes de instâncias de banco de dados . Para obter mais informações sobre as classes de instância compatíveis com o Amazon RDS para Oracle, consulte Classes de instância de banco de dados do RDS para Oracle.

Custo do armazenamento de instância do RDS para Oracle

O custo do armazenamento de instância é incorporado ao custo do armazenamento de instância ativado nas instâncias. Você não gera custos adicionais ao habilitar um armazenamento de instância em uma instância de banco de dados do RDS for Oracle. Para obter mais informações sobre instâncias ativadas em armazenamento de instância, consulte Classes de instância compatíveis com o armazenamento de instância do RDS for Oracle.

Ativar um armazenamento de instância do RDS for Oracle

Para ativar o armazenamento de instância para dados temporários do RDS for Oracle, siga um destes procedimentos:

- Crie uma instância de banco de dados do RDS for Oracle utilizando uma classe de instância compatível. Para obter mais informações, consulte Criar uma instância de banco de dados do Amazon RDS.
- Modifique uma instância de banco de dados do RDS for Oracle para utilizar uma classe de instância compatível. Para ter mais informações, consulte Modificar uma instância de banco de dados do Amazon RDS.

Configurar um armazenamento de instância do RDS for Oracle

Por padrão, 100% do espaço de armazenamento de instância é alocado ao espaço de tabela temporário. Para configurar o armazenamento de instância para alocar espaço para o cache flash e o espaço de tabela temporário, defina os seguintes parâmetros no grupo de parâmetros de sua instância:

db_flash_cache_size={DBInstanceStore*{0,2,4,6,8,10}/10}

Esse parâmetro especifica a quantidade de espaço de armazenamento alocado para o cache flash. Esse parâmetro é válido apenas para o Oracle Database Enterprise Edition. O valor

padrão é `{DBInstanceStore*0/10}`. Se você definir um valor diferente de zero para `db_flash_cache_size`, sua instância do RDS for Oracle ativará o cache flash após a reinicialização da instância.

rds.instance_store_temp_size={DBInstanceStore*{0,2,4,6,8,10}/10}

Esse parâmetro especifica a quantidade de espaço de armazenamento alocado para o espaço de tabela temporário. O valor padrão é `{DBInstanceStore*10/10}`. Esse parâmetro pode ser modificado para o Oracle Database Enterprise Edition e somente leitura para Standard Edition 2. Se você definir um valor diferente de zero para `rds.instance_store_temp_size`, o Amazon RDS alocará espaço no armazenamento de instância para o espaço de tabela temporário.

Você pode definir os parâmetros `db_flash_cache_size` e `rds.instance_store_temp_size` para instâncias de banco de dados que não usam um armazenamento de instância. Nesse caso, as duas configurações são avaliadas como `0`, o que desativa o recurso. Nesse caso, você pode usar o mesmo grupo de parâmetros para diferentes tamanhos de instância e para instâncias que não usam um armazenamento de instância. Se você modificar esses parâmetros, reinicialize as instâncias associadas para que as alterações tenham efeito.

> **Important**
>
> Se você alocar espaço para um espaço de tabela temporário, o Amazon RDS não criará o espaço de tabela temporário automaticamente. Para saber como criar o espaço de tabela temporário no armazenamento de instância, consulte Criar um espaço de tabela temporário no armazenamento de instância.

O valor combinado dos parâmetros anteriores não deve exceder 10/10 ou 100%. A tabela a seguir ilustra as configurações de parâmetros válidas e inválidas.

configuração db_flash_cache_size	configuração rds.instance_store_temp_size	Explicação
db_flash_cache_size={DBInstanceStore*0/10}	rds.instance_store_temp_size={DBInstanceStore*10/10}	Essa é uma configuração válida para todas as edições

configuração db_flash_cache_size	configuração rds.instance_store_temp_size	Explicação
		do Oracle Database. O Amazon RDS aloca 100% do espaço de armazenamento de instância ao espaço de tabela temporário. Esse é o padrão.
db_flash_cache_size={DBInstanceStore*10/10}	rds.instance_store_temp_size={DBInstanceStore*0/10}	Essa é uma configuração válida apenas para o Oracle Database Enterprise Edition. O Amazon RDS aloca 100% do espaço de armazenamento de instância ao cache flash.

configuração db_flash_cache_size	configuração rds.instance_store_temp_size	Explicação
db_flash_cache_size={DBInstanceStore*2/10}	rds.instance_store_temp_size={DBInstanceStore*8/10}	Essa é uma configuração válida apenas para o Oracle Database Enterprise Edition. O Amazon RDS aloca 20% do espaço de armazenamento de instância para o cache flash e 80% do espaço de armazenamento da instância para o espaço de tabela temporário.

configuração db_flash_cache_size	configuração rds.instance_store_temp_size	Explicação
db_flash_cache_size={DBInstanceStore*6/10}	rds.instance_store_temp_size={DBInstanceStore*4/10}	Essa é uma configuração válida apenas para o Oracle Database Enterprise Edition. O Amazon RDS aloca 60% do espaço de armazenamento de instância para o cache flash e 40% do espaço de armazenamento da instância para o espaço de tabela temporário.

configuração db_flash_cache_size	configuração rds.instance_store_temp_size	Explicação
db_flash_cache_size={DBInstanceStore*2/10}	rds.instance_store_temp_size={DBInstanceStore*4/10}	Essa é uma configuração válida apenas para o Oracle Database Enterprise Edition. O Amazon RDS aloca 20% do espaço de armazenamento de instância para o cache flash e 40% do espaço de armazenamento da instância para o espaço de tabela temporário.

configuração db_flash_cache_size	configuração rds.instance_store_temp_size	Explicação
db_flash_cache_size={DBInstanceStore*8/10}	rds.instance_store_temp_size={DBInstanceStore*8/10}	Essa é uma configuração inválida porque a porcentagem combinada do espaço de armazenamento da instância excede 100%. Nesses casos, o Amazon RDS falha na tentativa.

Considerações ao alterar o tipo de instância de banco de dados

Se você alterar o tipo de instância de banco de dados, isso poderá afetar a configuração do cache flash ou do espaço de tabela temporário no armazenamento de instância. Considere as seguintes modificações e seus efeitos:

Aumente ou reduza a escala verticalmente da instância de banco de dados que é compatível com o armazenamento de instância.

Os valores a seguir aumentam ou diminuem proporcionalmente ao novo tamanho do armazenamento de instância:

- O novo tamanho do cache flash.
- O espaço alocado para os espaços de tabela temporários que residem no armazenamento de instância.

Por exemplo, a configuração `db_flash_cache_size={DBInstanceStore*6/10}` em uma instância db.m5d.4xlarge fornece cerca de 340 GB de espaço em cache flash. Se você aumentar a escala verticalmente do tipo de instância para db.m5d.8xlarge, o espaço do cache flash aumentará para cerca de 680 GB.

Modifique uma instância de banco de dados que não use um armazenamento de instância para uma instância que usa um armazenamento de instância.

Se `db_flash_cache_size` for definido como um valor maior que `0`, o cache flash será configurado. Se `rds.instance_store_temp_size` for definido como um valor maior que `0`, o espaço de armazenamento da instância será alocado para uso por um espaço de tabela temporário. O RDS for Oracle não move arquivos temporários para o armazenamento de instância automaticamente. Para obter informações sobre como usar o espaço alocado, consulte Criar um espaço de tabela temporário no armazenamento de instância ou Adicionar um arquivo temporário ao armazenamento de instância em uma réplica de leitura.

Modifique uma instância de banco de dados que não use um armazenamento de instância para uma instância que usa um armazenamento de instância.

Nesse caso, o RDS for Oracle remove o cache flash. O RDS recria o arquivo temporário atualmente localizado no armazenamento de instância em um volume do Amazon EBS. O tamanho máximo do novo arquivo temporário é o tamanho anterior do parâmetro `rds.instance_store_temp_size`.

Trabalhar com um armazenamento de instância em uma réplica de leitura da Oracle

As réplicas de leitura são compatíveis com o cache flash e aos espaços de tabela temporários em um armazenamento de instância. Embora o cache flash funcione da mesma forma que na instância de banco de dados primária, observe as seguintes diferenças para espaços de tabela temporários:

- Não é possível criar um espaço de tabela temporário em uma réplica de leitura. Se você criar um espaço de tabela temporário na instância primária, o RDS for Oracle replicará as informações do espaço de tabela sem arquivos temporários. Para adicionar um novo arquivo temporário, use uma das seguintes técnicas:
  - Use o procedimento do Amazon RDS `rdsadmin.rdsadmin_util.add_inst_store_tempfile`. O RDS for Oracle cria um arquivo temporário no armazenamento de instância em sua réplica de leitura e o adiciona ao espaço de tabela temporário especificado.
  - Execute o comando `ALTER TABLESPACE … ADD TEMPFILE`. O RDS for Oracle coloca o arquivo temporário no armazenamento do Amazon EBS.

> **Note**
>
> Os tamanhos dos arquivos temporários e os tipos de armazenamento podem ser diferentes na instância de banco de dados primária e na réplica de leitura.

- Você pode gerenciar a configuração de espaço de tabela temporário padrão somente na instância de banco de dados primária. O RDS for Oracle replica a configuração para todas as réplicas de leitura.
- Você pode configurar os grupos de espaços de tabela temporários somente na instância de banco de dados primária. O RDS for Oracle replica a configuração para todas as réplicas de leitura.

Configurar um grupo de espaços de tabela temporários em um armazenamento de instância e no Amazon EBS

Você pode configurar um grupo de espaços de tabela temporários para incluir espaços de tabela temporários em um armazenamento de instância e no Amazon EBS. Essa técnica é útil quando você deseja mais armazenamento temporário do que o permitido pela configuração máxima de `rds.instance_store_temp_size`.

Quando você configura um grupo de espaços de tabela temporários em um armazenamento de instância e no Amazon EBS, os dois espaços de tabela têm características de performance significativamente diferentes. O Oracle Database escolhe o espaço de tabela para atender às consultas com base em um algoritmo interno. Portanto, consultas semelhantes podem variar em performance.

Normalmente, você cria um espaço de tabela temporário no armazenamento de instância da seguinte forma:

1. Crie um espaço de tabela temporário no armazenamento de instância.
2. Defina o novo espaço de tabela como o espaço de tabela temporário padrão do banco de dados.

Se o tamanho do espaço de tabela no armazenamento de instância for insuficiente, você poderá criar armazenamento temporário adicional da seguinte forma:

1. Atribua o espaço de tabela temporário no armazenamento de instância a um grupo de espaços de tabela temporários.

2. Crie um espaço de tabela temporário no Amazon EBS, caso não exista.
3. Atribua o espaço de tabela temporário no Amazon EBS ao mesmo grupo de espaços de tabela que inclui o espaço de tabela do armazenamento de instância.
4. Defina o grupo de espaços de tabela como o espaço de tabela temporário padrão.

O exemplo a seguir pressupõe que o tamanho do espaço de tabela temporário no armazenamento de instância não atenda aos requisitos de sua aplicação. O exemplo cria o espaço de tabela temporário `temp_in_inst_store` no armazenamento de instância, o atribui ao grupo de espaços de tabela `temp_group`, adiciona o espaço de tabela existente do Amazon EBS chamado `temp_in_ebs` a esse grupo e define esse grupo como o espaço de tabela temporário padrão.

```
SQL> EXEC rdsadmin.rdsadmin_util.create_inst_store_tmp_tblspace('temp_in_inst_store');

PL/SQL procedure successfully completed.

SQL> ALTER TABLESPACE temp_in_inst_store TABLESPACE GROUP temp_group;

Tablespace altered.

SQL> ALTER TABLESPACE temp_in_ebs TABLESPACE GROUP temp_group;

Tablespace altered.

SQL> EXEC rdsadmin.rdsadmin_util.alter_default_temp_tablespace('temp_group');

PL/SQL procedure successfully completed.

SQL> SELECT * FROM DBA_TABLESPACE_GROUPS;

GROUP_NAME                     TABLESPACE_NAME
------------------------------ ------------------------------
TEMP_GROUP                     TEMP_IN_EBS
TEMP_GROUP                     TEMP_IN_INST_STORE

SQL> SELECT PROPERTY_VALUE FROM DATABASE_PROPERTIES WHERE
 PROPERTY_NAME='DEFAULT_TEMP_TABLESPACE';

PROPERTY_VALUE
--------------
TEMP_GROUP
```

Remover um armazenamento de instância do RDS for Oracle

Para remover o armazenamento de instância, modifique sua instância de banco de dados do RDS for Oracle para usar um tipo de instância que não seja compatível com o armazenamento de instância, como db.m5 ou db.r5.

Ativar o HugePages para uma instância do RDS para Oracle

O Amazon RDS para Oracle é compatível com HugePages do kernel do Linux para aumentar a escalabilidade do banco de dados. O uso de HugePages resulta em tabelas de página menores e menos tempo de CPU gasto no gerenciamento de memória, aumentando a performance de instâncias de banco de dados grandes. Para obter mais informações, consulte Visão geral sobre HugePages na documentação da Oracle.

Você pode usar HugePages com todas as versões e edições do RDS para Oracle compatíveis.

O parâmetro `use_large_pages` controla se o recurso HugePages permanece ativado para uma instância de banco de dados. As configurações possíveis desse parâmetro são `ONLY`, `FALSE` e `{DBInstanceClassHugePagesDefault}`. O parâmetro `use_large_pages` é definido como `{DBInstanceClassHugePagesDefault}` no parameter group de banco de dados padrão para Oracle.

Para controlar se o recurso HugePages permanece ativado para uma instância de banco de dados automaticamente, você pode usar a variável de fórmula `DBInstanceClassHugePagesDefault` em grupos de parâmetros. O valor é determinado da seguinte forma:

- Para as classes de instância do banco de dados mencionadas na tabela a seguir, `DBInstanceClassHugePagesDefault` sempre é avaliado como `FALSE` por padrão, e `use_large_pages` é avaliado como `FALSE`. Será possível ativar o recurso HugePages manualmente para essas classes de instância de banco de dados se estas tiverem pelo menos 14 GiB de memória.
- Para classes de instância de banco de dados não mencionadas na tabela a seguir, se a classe tiver menos de 14 GiB de memória, `DBInstanceClassHugePagesDefault` sempre será avaliado como `FALSE`. Além disso, o `use_large_pages` será avaliado como `FALSE`
- Para classes de instância de banco de dados não mencionadas na tabela a seguir, se a classe da instância tiver pelo menos 14 GiB de memória e menos de 100 GiB de memória, `DBInstanceClassHugePagesDefault` será avaliado como `TRUE` por padrão. Além disso, o `use_large_pages` será avaliado como `ONLY` É possível desativar o recurso HugePages manualmente definindo `use_large_pages` como `FALSE`.

- Para classes de instância de banco de dados não mencionadas na tabela abaixo, se a classe da instância tiver pelo menos 100 GiB de memória, `DBInstanceClassHugePagesDefault` sempre é avaliado como `TRUE`. Além disso, o `use_large_pages` será avaliado como `ONLY` e o recurso HugePages não pode ser desabilitado.

O recurso HugePages não é ativado por padrão para as classes de instância de banco de dados a seguir.

Família de classe de instância de banco de dados	Classes de instância de banco de dados com o recurso HugePages não ativado por padrão
db.m5	db.m5.large
db.m4	db.m4.large, db.m4.xlarge, db.m4.2xlarge, db.m4.4xlarge, db.m4.10xlarge
db.t3	db.t3.micro, db.t3.small, db.t3.medium, db.t3.large

Para mais informações sobre classes de instância de banco de dados, consulte Especificações de hardware para classes de instância de banco de dados .

Para ativar o recurso HugePages para instâncias de banco de dados novas ou existentes manualmente, defina o parâmetro `use_large_pages` como `ONLY`. Não é possível usar HugePages com o Oracle Automatic Memory Management (AMM). Se você definir o parâmetro `use_large_pages` como `ONLY`, também deverá definir `memory_target` e `memory_max_target` como `0`. Para obter mais informações sobre como definir parâmetros de banco de dados para a sua instância de banco de dados, consulte Grupos de parâmetros para Amazon RDS.

Também é possível definir os parâmetros `sga_target`, `sga_max_size` e `pga_aggregate_target`. Quando você definir os parâmetros de memória da área global do sistema (SGA) e da área global do programa (PGA), adicione os valores juntos. Subtraia esse total da sua memória de instância disponível (`DBInstanceClassMemory`) para determinar a memória livre além da alocação de HugePages. Você deve deixar pelo menos 2 GiB de memória livre ou 10% da memória total da instância disponível, o que for menor.

Depois de configurar seus parâmetros, você deverá reiniciar sua instância de banco de dados para que as alterações entrem em vigor. Para obter mais informações, consulte Reinicializar uma instância de banco de dados .

> **Note**
>
> A instância de banco de dados Oracle adiará as alterações nos parâmetros de inicialização relacionados ao SGA até que você reinicialize a instância sem failover. No console do Amazon RDS, escolha Reinicializar, mas não escolha Reinicializar com failover. Na AWS CLI, chame o comando `reboot-db-instance` com o parâmetro `--no-force-failover`. A instância de banco de dados não processa os parâmetros relacionados ao SGA durante o failover ou durante outras operações de manutenção que fazem com que a instância seja reiniciada.

Essa é uma configuração de parâmetro de exemplo de HugePages que permite páginas enormes manualmente. Você deve definir os valores para atender às suas necessidades.

```
memory_target            = 0
memory_max_target        = 0
pga_aggregate_target     = {DBInstanceClassMemory*1/8}
sga_target               = {DBInstanceClassMemory*3/4}
sga_max_size             = {DBInstanceClassMemory*3/4}
use_large_pages          = ONLY
```

Suponhamos que os valores de parâmetros estejam definidos em um parameter group.

```
memory_target            = IF({DBInstanceClassHugePagesDefault}, 0,
 {DBInstanceClassMemory*3/4})
memory_max_target        = IF({DBInstanceClassHugePagesDefault}, 0,
 {DBInstanceClassMemory*3/4})
pga_aggregate_target     = IF({DBInstanceClassHugePagesDefault},
 {DBInstanceClassMemory*1/8}, 0)
sga_target               = IF({DBInstanceClassHugePagesDefault},
 {DBInstanceClassMemory*3/4}, 0)
sga_max_size             = IF({DBInstanceClassHugePagesDefault},
 {DBInstanceClassMemory*3/4}, 0)
use_large_pages          = {DBInstanceClassHugePagesDefault}
```

O grupo de parâmetros é usado por uma classe de instância de banco de dados db.r4 com menos de 100 GiB de memória. Com essas configurações de parâmetro e `use_large_pages` definido como `{DBInstanceClassHugePagesDefault}`, o recurso HugePages permanece ativado para a instância db.r4.

Considere outro exemplo com os valores de parâmetros a seguir definidos em um parameter group.

```
memory_target           = IF({DBInstanceClassHugePagesDefault}, 0,
 {DBInstanceClassMemory*3/4})
memory_max_target       = IF({DBInstanceClassHugePagesDefault}, 0,
 {DBInstanceClassMemory*3/4})
pga_aggregate_target    = IF({DBInstanceClassHugePagesDefault},
 {DBInstanceClassMemory*1/8}, 0)
sga_target              = IF({DBInstanceClassHugePagesDefault},
 {DBInstanceClassMemory*3/4}, 0)
sga_max_size            = IF({DBInstanceClassHugePagesDefault},
 {DBInstanceClassMemory*3/4}, 0)
use_large_pages         = FALSE
```

O grupo de parâmetros é usado por uma classe de instância de banco de dados db.r4 e por uma classe de instância de banco de dados db.r5, as duas com menos de 100 GiB de memória. Com essas configurações de parâmetro, o HugePages é desativado nas instâncias db.r4 e db.r5.

> **Note**
>
> Caso esse grupo de parâmetros seja usado por uma classe de instância de banco de dados db.r4 ou por uma classe de instância de banco de dados db.r5 com pelo menos 100 GiB de memória, a configuração de `FALSE` para `use_large_pages` será substituída e definida como `ONLY`. Nesse caso, uma notificação do cliente em relação à substituição é enviada.

Depois que o recurso HugePages estiver habilitado na instância de banco de dados, você poderá visualizar informações sobre o recurso HugePages habilitando o monitoramento avançado. Para obter mais informações, consulte Monitorar métricas do SO com o monitoramento avançado.

Ativar tipos de dados estendidos no RDS para Oracle

O Amazon RDS para Oracle é compatível com tipos de dados estendidos. Com tipos de dados estendidos, o tamanho máximo é 32.767 bytes para os tipos de dados `VARCHAR2`, `NVARCHAR2` e `RAW`. Para usar os tipos de dados estendidos, defina o parâmetro `MAX_STRING_SIZE` como `EXTENDED`. Para obter mais informações, consulte Tipos de dados estendidos na documentação da Oracle.

Se você não deseja usar os tipos de dados estendidos, mantenha o parâmetro MAX_STRING_SIZE definido como STANDARD (padrão). Nesse caso, os limites de tamanho são 4.000 bytes para os tipos de dados VARCHAR2 e NVARCHAR2 e 2.000 bytes para o tipo de dados RAW.

Você pode ativar os tipos de dados estendidos em uma instância de banco de dados nova ou existente. Para novas instâncias de banco de dados, o tempo de criação de instâncias de banco de dados normalmente é maior quando você ativa os tipos de dados estendidos. Para instâncias de banco de dados existentes, a instância fica indisponível durante o processo de conversão.

Considerações sobre tipos de dados estendidos

Considere o seguinte ao habilitar tipos de dados estendidos para sua instância de banco de dados:

- Ao ativar tipos de dados estendidos para uma instância de banco de dados nova ou existente, você deve reiniciar a instância para que a alteração entre em vigor.
- Depois de ativar tipos de dados estendidos, você não poderá voltar atrás e usar o tamanho padrão de tipos de dados para a instância. Se você definir o parâmetro MAX_STRING_SIZE novamente como STANDARD, ele assumirá o status incompatible-parameters.
- Quando você restaura uma instância de banco de dados que usa tipos de dados estendidos, é preciso especificar um parameter group com o parâmetro MAX_STRING_SIZE definido como EXTENDED. Durante a restauração, se você especificar um parameter group padrão ou qualquer outro parameter group com MAX_STRING_SIZE definido como STANDARD, o status incompatible-parameters será gerado.
- Quando o status da instância de banco e dados for incompatible-parameters por causa da configuração de MAX_STRING_SIZE, a instância permanecerá indisponível até você definir o parâmetro MAX_STRING_SIZE como EXTENDED e reinicializar a instância de banco de dados.

Ativar tipos de dados estendidos para uma nova instância de banco de dados

Quando você cria uma instância de banco de dados com MAX_STRING_SIZE definido como EXTENDED, a instância mostra MAX_STRING_SIZE definido como STANDARD padrão. Reinicialize a instância para ativar a alteração.

Para ativar tipos de dados estendidos para uma nova instância de banco de dados

1. Defina o parâmetro MAX_STRING_SIZE como EXTENDED em um parameter group.

Para definir o parâmetro, crie um novo parameter group ou modifique um parameter group existente.

Para obter mais informações, consulte Grupos de parâmetros para Amazon RDS.

2. Crie uma instância de banco de dados do RDS para Oracle.

   Para ter mais informações, consulte Criar uma instância de banco de dados do Amazon RDS.

3. Associe o grupo de parâmetros com `MAX_STRING_SIZE` definido como `EXTENDED` à instância de banco de dados.

   Para ter mais informações, consulte Criar uma instância de banco de dados do Amazon RDS.

4. Reinicialize a instância de banco de dados para que a alteração do parâmetro tenha efeito.

   Para ter mais informações, consulte Reinicializar uma instância de banco de dados .

Ativar tipos de dados estendidos para uma instância de banco de dados existente

Quando você modifica uma instância de banco de dados para ativar tipos de dados estendidos, o RDS converte os dados no banco de dados para usar os tamanhos estendidos. A conversão e o tempo de inatividade ocorrerão na próxima reinicialização do banco de dados depois da alteração do parâmetro. A instância de banco de dados fica indisponível durante a migração.

O tempo necessário para converter os dados depende da classe da instância de banco de dados, do tamanho do banco de dados e do horário do último snapshot de banco de dados. Para reduzir o tempo de inatividade, considere gerar um snapshot imediatamente antes da reinicialização. Isso reduz o tempo do backup que ocorre durante o fluxo de trabalho de conversão.

> **Note**
>
> Depois de ativar os tipos de dados estendidos, não é possível realizar uma recuperação a um ponto anterior no tempo durante o processo de conversão. Você pode restaurar para o momento imediatamente anterior à conversão ou após a conversão.

Como ativar tipos de dados estendidos para uma instância de banco de dados existente

1. Faça um snapshot do banco de dados.

Se houver objetos inválidos no banco de dados, o Amazon RDS tentará recompilá-los. A conversão para tipos de dados estendidos poderá falhar se o Amazon RDS não puder recompilar um objeto inválido. O snapshot permitirá que você restaure o banco de dados se houver um problema com a conversão. Sempre verifique se há objetos inválidos antes de iniciar a conversão e corrija ou descarte esses objetos inválidos. Para bancos de dados de produção, recomendamos primeiro testar o processo de conversão em uma cópia da instância de banco de dados.

Para obter mais informações, consulte Criar um snapshot de banco de dados para uma instância de banco de dados single-AZ para o Amazon RDS.

2. Defina o parâmetro `MAX_STRING_SIZE` como `EXTENDED` em um parameter group.

   Para definir o parâmetro, crie um novo parameter group ou modifique um parameter group existente.

   Para obter mais informações, consulte Grupos de parâmetros para Amazon RDS.

3. Modifique a instância de banco de dados para associá-la ao parameter group com `MAX_STRING_SIZE` definido como `EXTENDED`.

   Para obter mais informações, consulte Modificar uma instância de banco de dados do Amazon RDS.

4. Reinicialize a instância de banco de dados para que a alteração do parâmetro tenha efeito.

   Para obter mais informações, consulte Reinicializar uma instância de banco de dados .

Importar dados para o Oracle no Amazon RDS

A forma como você importa dados para uma instância de banco de dados do Amazon RDS para Oracle depende do seguinte:

- A quantidade de dados que você tem
- O número de objetos em seu banco de dados
- A variedade de objetos em seu banco de dados

Por exemplo, você pode usar as seguintes ferramentas, dependendo de seus requisitos:

- Oracle SQL Developer: importe um banco de dados simples de 20 MB.
- Oracle Data Pump: importar bancos de dados complexos ou bancos de dados com várias centenas de megabytes ou terabytes de tamanho. Por exemplo, você pode transportar espaços de tabela de um banco de dados on-premises para a instância de banco de dados do RDS para Oracle. Você pode usar o Amazon S3 ou o Amazon EFS para transferir os arquivos de dados e os metadados. Para mais informações, consulte Migrar usando espaços de tabela transportáveis da Oracle, Integração do Amazon EFS e Integração do Amazon S3.
- AWS Database Migration Service(AWS DMS): migre bancos de dados sem tempo de inatividade. Para obter mais informações sobre AWS DMS, consulte O que é o AWS Database Migration Service e a publicação de blog Migrar bancos de dados do Oracle com tempo de inatividade quase zero usando o AWS DMS.

> **Important**
>
> Antes de usar as técnicas de migração anteriores, recomendamos que você faça backup do banco de dados. Após importar os dados, você pode fazer backup de suas instâncias de banco de dados do RDS para Oracle criando snapshots. Mais tarde, você poderá restaurar os snapshots. Para obter mais informações, consulte Backup, restauração e exportação de dados.

Para vários mecanismos de banco de dados, a replicação contínua pode continuar até que você esteja pronto para migrar para o banco de dados de destino. Você pode usar AWS DMS a fim de migrar para o RDS para Oracle a partir do mesmo mecanismo de banco de dados ou um mecanismo de banco de dados diferente. Se você migrar de um mecanismo de banco de dados diferente,

poderá usar a AWS Schema Conversion Tool para migrar objetos de esquema que o AWS DMS não consegue migrar.

Tópicos

- Importar usando o Oracle SQL Developer
- Migrar usando espaços de tabela transportáveis da Oracle
- Importar usando o Oracle Data Pump
- Importar usando Oracle Export/Import
- Importar usando o Oracle SQL*Loader
- Migrar com visualizações materializadas do Oracle

Importar usando o Oracle SQL Developer

O Oracle SQL Developer é uma ferramenta gráfica Java distribuída sem custos pela Oracle. O SQL Developer fornece opções para migrar dados entre dois bancos de dados Oracle ou para migrar dados de outros bancos de dados, como o MySQL, para um banco de dados Oracle. Essa ferramenta é a mais adequada para migrar bancos de dados pequenos.

Você pode instalar essa ferramenta no seu computador desktop (Windows, Linux ou Mac) ou em um dos seus servidores. Depois de instalar o SQL Developer, você pode usá-lo para se conectar aos bancos de dados de origem e de destino. Use o comando Cópia do banco de dados no menu Ferramentas para copiar seus dados para a instância de banco de dados do RDS para Oracle.

Para baixar o SQL Developer, acesse http://www.oracle.com/technetwork/developer-tools/sql-developer.

Recomendamos que você leia a documentação do produto Oracle SQL Developer antes de começar a migrar seus dados. A Oracle também possui documentação sobre como migrar de outros bancos de dados, incluindo o MySQL e o SQL Server. Para obter mais informações, consulte http://www.oracle.com/technetwork/database/migration na documentação da Oracle.

Migrar usando espaços de tabela transportáveis da Oracle

Você pode usar o atributo de espaços de tabela transportáveis da Oracle para copiar um conjunto de espaços de tabela de um banco de dados Oracle on-premises para uma instância de banco de dados do RDS para Oracle. No nível físico, é necessário transferir os arquivos de dados de origem e os arquivos de metadados para a instância de banco de dados de destino usando o

Amazon EFS ou o Amazon S3. O recurso de espaços de tabela transportáveis usa o pacote `rdsadmin.rdsadmin_transport_util`. Consulte a sintaxe e a semântica desse pacote em Transportar espaços para tabela.

Para publicações de blog que explicam como transportar espaços de tabela, consulte Migrate Oracle Databases to AWS using transportable tablespace e Amazon RDS for Oracle Transportable Tablespaces using RMAN.

Tópicos

- Visão geral dos espaços de tabela transportáveis da Oracle
- Fase 1: Configurar o host de origem
- Fase 2: Preparar o backup completo dos espaços de tabela
- Fase 3: Criar e transferir backups incrementais
- Fase 4: Transportar os espaços de tabela
- Fase 5: Validar os espaços de tabela transportados
- Fase 6: Limpar os arquivos restantes

Visão geral dos espaços de tabela transportáveis da Oracle

Um conjunto de espaços de tabela transportáveis consiste em arquivos de dados para o conjunto de espaços de tabela que está sendo transportado e um arquivo de despejo de exportação que contém metadados do espaço de tabela. Em uma solução de migração física, como os espaços de tabela transportáveis, você transfere arquivos físicos: arquivos de dados, arquivos de configuração e arquivos de despejo do Data Pump.

Tópicos

- Vantagens e desvantagens dos espaços de tabela transportáveis
- Limitações dos espaços de tabela transportáveis
- Pré-requisitos para espaços de tabela transportáveis

Vantagens e desvantagens dos espaços de tabela transportáveis

Recomendamos que você use espaços de tabela transportáveis quando precisar migrar um ou mais espaços de tabela grandes para o RDS com o mínimo de tempo de inatividade. Os espaços de tabela transportáveis oferecem as seguintes vantagens em relação à migração lógica:

- O tempo de inatividade é menor do que a maioria das outras soluções de migração da Oracle.
- Como o atributo de espaço de tabela transportável copia somente arquivos físicos, ele evita os erros de integridade de dados e a corrupção lógica que podem ocorrer na migração lógica.
- Nenhuma licença adicional é necessária.
- Você pode migrar um conjunto de espaços de tabela entre diferentes plataformas e tipos de extremidade, por exemplo, de uma plataforma do Oracle Solaris para Linux. No entanto, não há suporte ao transporte de espaços de tabela de e para servidores Windows.

> **Note**
>
> O Linux foi totalmente testado e é totalmente compatível. Nem todas as variações do UNIX foram testadas.

Se você usar espaços de tabela transportáveis, poderá transportar dados usando o Amazon S3 ou o Amazon EFS:

- Ao usar o EFS, seus backups permanecem no sistema de arquivos do EFS durante a importação. Você poderá remover os arquivos posteriormente. Nessa técnica, não é necessário provisionar o armazenamento do EBS para sua instância de banco de dados. Por esse motivo, recomendamos usar o Amazon EFS em vez do S3. Para obter mais informações, consulte Integração do Amazon EFS.
- Ao usar o S3, baixe backups do RMAN para o armazenamento do EBS conectado à sua instância de banco de dados. Os arquivos permanecem no armazenamento do EBS durante a importação. Após a importação, você poderá liberar esse espaço, que permanecerá alocado para sua instância de banco de dados.

A principal desvantagem dos espaços de tabela transportáveis é a necessidade de um conhecimento relativamente avançado sobre o Oracle Database. Para obter mais informações, consulte Transporting Tablespaces Between Databases no Oracle Database Administrator's Guide.

Limitações dos espaços de tabela transportáveis

As limitações do Oracle Database para espaços de tabela transportáveis se aplicam quando você usa esse atributo no RDS para Oracle. Para obter mais informações, consulte Limitations on Transportable Tablespaces e General Limitations on Transporting Data no Oracle Database

Administrator's Guide. Observe as seguintes limitações adicionais para espaços de tabela transportáveis no RDS para Oracle:

- Nem o banco de dados de origem, nem o de destino podem usar a Standard Edition 2 (SE2). Somente a Enterprise Edition é compatível.
- Não é possível usar um banco de dados do Oracle Database 11g como fonte. O recurso de espaços de tabela transportáveis multiplataforma do RMAN depende do mecanismo de transporte RMAN, que o Oracle Database 11g não comporta.
- Não é possível migrar dados de uma instância de banco de dados do RDS para Oracle usando espaços de tabela transportáveis. Só é possível usar os espaços de tabela transportáveis para migrar dados para uma instância de banco de dados do RDS para Oracle.
- Não há suporte ao sistema operacional Windows.
- Não é possível transportar espaços de tabela para um banco de dados em um nível de versão inferior. O banco de dados de destino deve estar no mesmo nível de versão, ou posterior, do banco de dados de origem. Por exemplo, não é possível transportar espaços de tabela do Oracle Database 21c para o Oracle Database 19c.
- Não é possível transportar espaços de tabela administrativos, como `SYSTEM` e `SYSAUX`.
- Não é possível transportar objetos que não sejam de dados, como pacotes PL/SQL, classes Java, visualizações, gatilhos, sequências, usuários, perfis e tabelas temporárias. Para transportar objetos que não sejam de dados, crie-os manualmente ou use a exportação e importação de metadados do Data Pump. Para ter mais informações, consulte My Oracle Support Note 1454872.1.
- Não é possível transportar espaços de tabela que estão criptografados ou que usam colunas criptografadas.
- Ao transferir arquivos usando o Amazon S3, o tamanho máximo de arquivo compatível é de 5 TiB.
- Se o banco de dados de origem usar opções do Oracle como o Spatial, você não poderá transportar espaços de tabela, a menos que as mesmas opções estejam configuradas no banco de dados de destino.
- Não é possível transportar espaços de tabela para uma instância de banco de dados do RDS para Oracle em uma configuração de réplica do Oracle. Como solução alternativa, você pode excluir todas as réplicas, transportar os espaços de tabela e recriar as réplicas.

Pré-requisitos para espaços de tabela transportáveis

Antes de começar, conclua as seguintes tarefas:

- Analise os requisitos para espaços de tabela transportáveis descritos nos seguintes documentos em My Oracle Support:
  - Reduce Transportable Tablespace Downtime using Cross Platform Incremental Backup (ID: 2471245.1)
  - Transportable Tablespace (TTS) Restrictions and Limitations: Details, Reference, and Version Where Applicable (ID: 1454872.1)
  - Primary Note for Transportable Tablespaces (TTS) -- Common Questions and Issues (ID: 1166564.1)
- Planeje a conversão de extremidade. Se você especificar o ID da plataforma de origem, o RDS para Oracle converterá a extremidade automaticamente. Para saber como encontrar IDs de plataforma, consulte Data Guard Support for Heterogeneous Primary and Physical Standbys in Same Data Guard Configuration (Doc ID 413484.1).
- Verifique se o atributo de espaços de tabela transportáveis está habilitado em sua instância de banco de dados de destino. O atributo só estará habilitado se você não receber um erro `ORA-20304` ao executar a seguinte consulta:

  ```
  SELECT * FROM TABLE(rdsadmin.rdsadmin_transport_util.list_xtts_orphan_files);
  ```

  Se o atributo de espaços de tabela transportáveis não estiver habilitado, reinicialize a instância de banco de dados. Para obter mais informações, consulte Reinicializar uma instância de banco de dados .
- Verifique se o arquivo de fuso horário é o mesmo nos bancos de dados de origem e de destino.
- Certifique-se de que os conjuntos de caracteres do banco de dados nos bancos de dados de origem e de destino satisfaçam um dos seguintes requisitos:
  - Os conjuntos de caracteres são os mesmos.
  - Os conjuntos de caracteres são compatíveis. Para obter uma lista dos requisitos de compatibilidade, consulte Limitações gerais no transporte de dados na documentação do Oracle Database.
- Se você planeja transferir arquivos usando o Amazon S3, faça o seguinte:
  - Verifique se há um bucket do Amazon S3 disponível para transferências de arquivos e se o bucket do Amazon S3 está na mesma região da AWS que a sua instância de banco de dados. Para obter mais informações, consulte Criar um bucket no Guia de conceitos básicos do Amazon Simple Storage Service.

  - Prepare o bucket do Amazon S3 para a integração com o Amazon RDS seguindo as instruções em Configurar permissões do IAM para a integração do RDS para Oracle com o Amazon S3.
- Se você planeja transferir arquivos usando o Amazon EFS, verifique se configurou o EFS de acordo com as instruções em Integração do Amazon EFS.
- É altamente recomendável que você ative os backups automáticos na instância de banco de dados de destino. Como a etapa de importação de metadados pode falhar, é importante ser capaz de restaurar a instância de banco de dados para o estado anterior à importação, evitando assim a necessidade de fazer backup, transferir e importar os espaços de tabela novamente.

Fase 1: Configurar o host de origem

Nesta etapa, você copiará os scripts de transporte de espaços de tabela fornecidos em My Oracle Support e definirá os arquivos de configuração necessários. Nas etapas a seguir, o host de origem executará o banco de dados que contém os espaços de tabela que serão transportados para a instância de destino.

Como configurar o host de origem

1. Faça login no host de origem como proprietário do início do Oracle.
2. Certifique-se de que as variáveis de ambiente `ORACLE_HOME` e `ORACLE_SID` apontem para o banco de dados de origem.
3. Faça login em seu banco de dados como administrador e verifique se a versão do fuso horário, o conjunto de caracteres do banco de dados e o conjunto de caracteres nacionais são iguais aos do banco de dados de destino.

   ```
   SELECT * FROM V$TIMEZONE_FILE;
   SELECT * FROM NLS_DATABASE_PARAMETERS
     WHERE PARAMETER IN ('NLS_CHARACTERSET','NLS_NCHAR_CHARACTERSET');
   ```
4. Configure o utilitário de espaços de tabela transportáveis conforme descrito em Oracle Support note 2471245.1.

   A configuração inclui a edição do arquivo `xtt.properties` em seu host de origem. O exemplo de arquivo `xtt.properties` a seguir especifica backups de três espaços de tabela no diretório `/dsk1/backups`. Esses são os espaços de tabela que você pretende transportar para a instância de banco de dados de destino. Também especifica o ID da plataforma de origem para converter a extremidade automaticamente.

> **Note**
>
> Para saber como encontrar IDs de plataforma, consulte Data Guard Support for Heterogeneous Primary and Physical Standbys in Same Data Guard Configuration (Doc ID 413484.1).

```
#linux system
platformid=13
#list of tablespaces to transport
tablespaces=TBS1,TBS2,TBS3
#location where backup will be generated
src_scratch_location=/dsk1/backups
#RMAN command for performing backup
usermantransport=1
```

Fase 2: Preparar o backup completo dos espaços de tabela

Nesta fase, você fará backup dos espaços de tabela pela primeira vez, transferirá os backups para o host de destino e restaurará usando o procedimento `rdsadmin.rdsadmin_transport_util.import_xtts_tablespaces`. Quando essa fase estiver concluída, os backups iniciais dos espaços de tabela vão residir na instância de banco de dados de destino e poderão ser atualizados com backups incrementais.

Tópicos

- Etapa 1: Fazer backup dos espaços de tabela no host de origem
- Etapa 2: Transferir os arquivos de backup para a instância de banco de dados de destino
- Etapa 3: Importar os espaços de tabela na instância de banco de dados de destino

Etapa 1: Fazer backup dos espaços de tabela no host de origem

Nesta etapa, use o script `xttdriver.pl` para fazer um backup completo dos espaços de tabela. A saída de `xttdriver.pl` é armazenada na variável de ambiente `TMPDIR`.

Como fazer backup dos espaços de tabela

1. Se os espaços de tabela estiverem no modo somente leitura, faça login no banco de dados de origem como um usuário que tem o privilégio `ALTER TABLESPACE` e coloque os espaços de tabela no modo de leitura/gravação. Caso contrário, vá para a próxima etapa.

   O exemplo a seguir coloca `tbs1`, `tbs2` e `tbs3` no modo de leitura/gravação.

   ```
   ALTER TABLESPACE tbs1 READ WRITE;
   ALTER TABLESPACE tbs2 READ WRITE;
   ALTER TABLESPACE tbs3 READ WRITE;
   ```

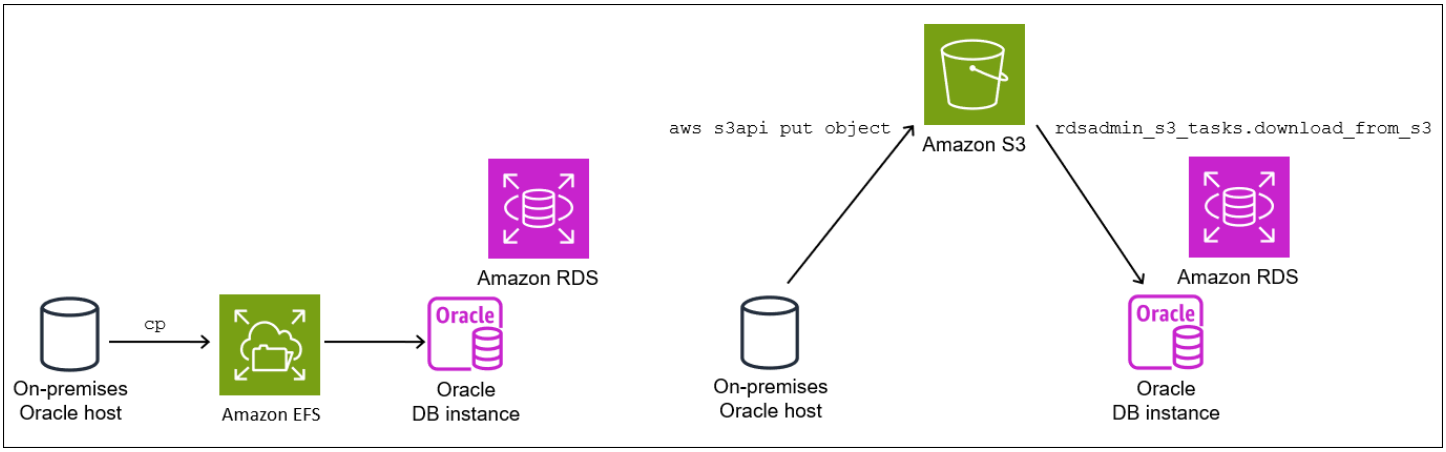
2. Faça backup dos espaços de tabela usando o script `xttdriver.pl`. Opcionalmente, você pode especificar `--debug` para executar o script no modo de depuração.

   ```
   export TMPDIR=location_of_log_files
   cd location_of_xttdriver.pl
   $ORACLE_HOME/perl/bin/perl xttdriver.pl --backup
   ```

Etapa 2: Transferir os arquivos de backup para a instância de banco de dados de destino

Nesta etapa, copie os arquivos de backup e configuração do local temporário para a instância de banco de dados de destino. Escolha uma das seguintes opções:

- Se os hosts de origem e de destino compartilharem um sistema de arquivos do Amazon EFS, use um utilitário de sistema operacional, como `cp`, para copiar os arquivos de backup e o arquivo `res.txt` do local temporário para um diretório compartilhado. Em seguida, vá para Etapa 3: Importar os espaços de tabela na instância de banco de dados de destino.
- Se você precisar preparar os backups em um bucket do Amazon S3, realize as etapas a seguir.



Etapa 2.2: Fazer upload dos backups no bucket do Amazon S3

Faça upload dos backups e do arquivo `res.txt` do diretório temporário para o bucket do Amazon S3. Para obter mais informações, consulte Fazer upload de objetos no Guia do usuário do Amazon Simple Storage Service.

Etapa 2.3: Fazer download dos backups do bucket do Amazon S3 para a instância de banco de dados de destino

Nesta etapa, use o procedimento `rdsadmin.rdsadmin_s3_tasks.download_from_s3` para fazer download dos backups para a instância de banco de dados do RDS para Oracle.

Como fazer download dos backups do bucket do Amazon S3

1. Inicie o SQL*Plus ou o Oracle SQL Developer e faça login em sua instância de banco de dados do RDS para Oracle.
2. Faça download dos backups do bucket do Amazon S3 para a instância de banco de dados de destino usando o procedimento `rdsadmin.rdsadmin_s3_tasks.download_from_s3` do Amazon RDS. O exemplo a seguir baixa todos os arquivos de um bucket do Amazon S3 chamado *`amzn-s3-demo-bucket`* no diretório *`DATA_PUMP_DIR`*.

```
EXEC UTL_FILE.FREMOVE ('DATA_PUMP_DIR', 'res.txt');
SELECT rdsadmin.rdsadmin_s3_tasks.download_from_s3(
  p_bucket_name    =>  'amzn-s3-demo-bucket',
  p_directory_name =>  'DATA_PUMP_DIR')
AS TASK_ID FROM DUAL;
```

A instrução SELECT retorna o ID da tarefa em um tipo de dados VARCHAR2. Para obter mais informações, consulte Baixar arquivos de um bucket do Amazon S3 para uma instância de banco de dados Oracle.

Etapa 3: Importar os espaços de tabela na instância de banco de dados de destino

Use o procedimento `rdsadmin.rdsadmin_transport_util.import_xtts_tablespaces` para restaurar os espaços de tabela na instância de banco de dados de destino. Esse procedimento converte automaticamente os arquivos de dados para o formato endian correto.

Se você importar de uma plataforma diferente do Linux, especifique a plataforma de origem usando o parâmetro `p_platform_id` ao chamar `import_xtts_tablespaces`. O ID da plataforma que você especificou deve corresponder ao que foi especificado no arquivo `xtt.properties` em Etapa 2: Exportar os metadados dos espaços de tabela no host de origem.

Importar os espaços de tabela na instância de banco de dados de destino

1. Inicie um cliente Oracle SQL e faça login como usuário principal na instância de banco de dados de destino do RDS para Oracle.
2. Execute o procedimento `rdsadmin.rdsadmin_transport_util.import_xtts_tablespaces`, especificando os espaços de tabela que serão importados e o diretório que contém os backups.

   O exemplo a seguir importa os espaços de tabela *TBS1*, *TBS2* e *TBS3* do diretório *DATA_PUMP_DIR*. A plataforma de origem é de sistemas baseados em AIX (64 bits), que tem o ID da plataforma de 6. É possível encontrar os IDs da plataforma consultando `V$TRANSPORTABLE_PLATFORM`.

```
VAR task_id CLOB

BEGIN
  :task_id:=rdsadmin.rdsadmin_transport_util.import_xtts_tablespaces(
        'TBS1,TBS2,TBS3',
        'DATA_PUMP_DIR',
        p_platform_id => 6);
END;
/
```

```
PRINT task_id
```

3. (Opcional) Monitore o progresso consultando a tabela `rdsadmin.rds_xtts_operation_info`. A coluna `xtts_operation_state` mostra o valor `EXECUTING`, `COMPLETED` ou `FAILED`.

```
SELECT * FROM rdsadmin.rds_xtts_operation_info;
```

> **Note**
>
> Para operações de longa duração, você também pode consultar `V$SESSION_LONGOPS`, `V$RMAN_STATUS` e `V$RMAN_OUTPUT`.

4. Veja o log da importação concluída usando o ID de tarefa da etapa anterior.

```
SELECT * FROM TABLE(rdsadmin.rds_file_util.read_text_file('BDUMP',
 'dbtask-'||'&task_id'||'.log'));
```

Verifique se importação foi bem-sucedida antes de prosseguir para a próxima etapa.

Fase 3: Criar e transferir backups incrementais

Nesta fase, você fará e transferirá backups incrementais periodicamente enquanto o banco de dados de origem estiver ativo. Essa técnica reduz o tamanho do backup final de espaços de tabela. Se você criar vários backups incrementais, deverá copiar o arquivo `res.txt` depois do último backup incremental antes de poder aplicá-lo na instância de destino.

As etapas são as mesmas de Fase 2: Preparar o backup completo dos espaços de tabela, exceto que a etapa de importação é opcional.

Fase 4: Transportar os espaços de tabela

Nesta fase, você fará backup dos espaços de tabela somente leitura e exportará os metadados do Data Pump, transferirá esses arquivos para o host de destino e importará os espaços de tabela e os metadados.

Tópicos

- Etapa 1: Fazer backup dos espaços de tabela somente leitura

- Etapa 2: Exportar os metadados dos espaços de tabela no host de origem
- Etapa 3: (somente Amazon S3) Transferir os arquivos de backup e exportação para a instância de banco de dados de destino
- Etapa 4: Importar os espaços de tabela na instância de banco de dados de destino
- Etapa 5: Importar os metadados de espaços de tabela na instância de banco de dados de destino

Etapa 1: Fazer backup dos espaços de tabela somente leitura

Esta etapa é idêntica a Etapa 1: Fazer backup dos espaços de tabela no host de origem, com uma única diferença importante: você coloca os espaços de tabela no modo somente leitura antes de fazer backup dos espaços de tabela pela última vez.

O exemplo a seguir coloca `tbs1`, `tbs2` e `tbs3` no modo somente leitura.

```
ALTER TABLESPACE tbs1 READ ONLY;
ALTER TABLESPACE tbs2 READ ONLY;
ALTER TABLESPACE tbs3 READ ONLY;
```

Etapa 2: Exportar os metadados dos espaços de tabela no host de origem

Exporte os metadados dos espaços de tabela executando o utilitário `expdb` no host de origem. O exemplo a seguir exporta os espaços de tabela *`TBS1`*, *`TBS2`* e *`TBS3`* para o arquivo de despejo *`xttdump.dmp`* no diretório *`DATA_PUMP_DIR`*.

```
expdp username/pwd \
dumpfile=xttdump.dmp \
directory=DATA_PUMP_DIR \
statistics=NONE \
transport_tablespaces=TBS1,TBS2,TBS3 \
transport_full_check=y \
logfile=tts_export.log
```

Se *`DATA_PUMP_DIR`* for um diretório compartilhado no Amazon EFS, vá para Etapa 4: Importar os espaços de tabela na instância de banco de dados de destino.

Etapa 3: (somente Amazon S3) Transferir os arquivos de backup e exportação para a instância de banco de dados de destino

Se você estiver usando o Amazon S3 para preparar os backups de espaços de tabela e o arquivo de exportação do Data Pump, conclua as etapas a seguir.

Etapa 3.1: Fazer upload dos backups e do arquivo de despejo do host de origem no bucket do Amazon S3

Faça upload dos arquivos de backup e despejo do host de origem no bucket do Amazon S3. Para obter mais informações, consulte Fazer upload de objetos no Guia do usuário do Amazon Simple Storage Service.

Etapa 3.2: Fazer download dos backups e do arquivo de despejo do bucket do Amazon S3 para a instância de banco de dados de destino

Nesta etapa, use o procedimento `rdsadmin.rdsadmin_s3_tasks.download_from_s3` para fazer download dos backups e do arquivo de despejo para a instância de banco de dados do RDS para Oracle. Siga as etapas em Etapa 2.3: Fazer download dos backups do bucket do Amazon S3 para a instância de banco de dados de destino.

Etapa 4: Importar os espaços de tabela na instância de banco de dados de destino

Use o procedimento `rdsadmin.rdsadmin_transport_util.import_xtts_tablespaces` para restaurar os espaços de tabela. Para conhecer a sintaxe e a semântica desse procedimento, consulte Importar espaços para tabela transportados para a instância de banco de dados.

> **Important**
>
> Depois de concluir a importação final de espaços de tabela, a próxima etapa será importar os metadados do Oracle Data Pump. Em caso de falha na importação, é importante restaurar o estado da instância de banco de dados anterior à falha. Portanto, recomendamos que você crie um snapshot de banco de dados da instância de banco de dados seguindo as instruções em Criar um snapshot de banco de dados para uma instância de banco de dados single-AZ para o Amazon RDS. O snapshot conterá todos os espaços de tabela importados, portanto, se a importação falhar, você não precisará repetir o processo de backup e importação.
> Se a instância de banco de dados de destino estiver com a opção de backups automáticos ativada e o Amazon RDS não detectar que um snapshot válido foi iniciado antes da importação dos metadados, o RDS tentará criar um snapshot. Dependendo da atividade da instância, esse snapshot pode ou não ser bem-sucedido. Se um snapshot válido não for detectado ou se não for possível iniciar o snapshot, a importação de metadados será encerrada com erros.

Importar os espaços de tabela na instância de banco de dados de destino

1. Inicie um cliente Oracle SQL e faça login como usuário principal na instância de banco de dados de destino do RDS para Oracle.
2. Execute o procedimento `rdsadmin.rdsadmin_transport_util.import_xtts_tablespaces`, especificando os espaços de tabela que serão importados e o diretório que contém os backups.

   O exemplo a seguir importa os espaços de tabela *TBS1*, *TBS2* e *TBS3* do diretório *DATA_PUMP_DIR*.

   ```
   BEGIN

    :task_id:=rdsadmin.rdsadmin_transport_util.import_xtts_tablespaces('TBS1,TBS2,TBS3','DATA_
   END;
   /
   PRINT task_id
   ```
3. (Opcional) Monitore o progresso consultando a tabela `rdsadmin.rds_xtts_operation_info`. A coluna `xtts_operation_state` mostra o valor `EXECUTING`, `COMPLETED` ou `FAILED`.

   ```
   SELECT * FROM rdsadmin.rds_xtts_operation_info;
   ```

   > **Note**
   >
   > Para operações de longa duração, você também pode consultar `V$SESSION_LONGOPS`, `V$RMAN_STATUS` e `V$RMAN_OUTPUT`.
4. Veja o log da importação concluída usando o ID de tarefa da etapa anterior.

   ```
   SELECT * FROM TABLE(rdsadmin.rds_file_util.read_text_file('BDUMP',
    'dbtask-'||'&task_id'||'.log'));
   ```

   Verifique se importação foi bem-sucedida antes de prosseguir para a próxima etapa.
5. Gere um snapshot manual do banco de dados seguindo as instruções em Criar um snapshot de banco de dados para uma instância de banco de dados single-AZ para o Amazon RDS.

Etapa 5: Importar os metadados de espaços de tabela na instância de banco de dados de destino

Nesta etapa, importe os metadados de espaços de tabela transportáveis para a instância de banco de dados do RDS para Oracle usando o procedimento `rdsadmin.rdsadmin_transport_util.import_xtts_metadata`. Para conhecer a sintaxe e a semântica desse procedimento, consulte Importar metadados de espaços para tabela transportáveis para a instância de banco de dados. Durante a operação, o status da importação é mostrado na tabela `rdsadmin.rds_xtts_operation_info`.

> **Important**
>
> Antes de importar metadados, é altamente recomendável que você confirme se um snapshot do banco de dados foi criado com êxito depois da importação dos espaços de tabela. Se a etapa de importação falhar, restaure a instância de banco de dados, resolva os erros de importação e tente importar novamente.

Importar os metadados do Data Pump para uma instância de banco de dados do RDS para Oracle

1. Inicie seu cliente Oracle SQL e faça login como usuário principal na instância de banco de dados de destino.
2. Crie os usuários proprietários de esquemas nos espaços de tabela transportados, caso esses usuários ainda não existam.

   ```
   CREATE USER tbs_owner IDENTIFIED BY password;
   ```

3. Importe os metadados, especificando o nome do arquivo de despejo e sua localização no diretório.

   ```
   BEGIN

    rdsadmin.rdsadmin_transport_util.import_xtts_metadata('xttdump.dmp','DATA_PUMP_DIR');
   END;
   /
   ```

4. (Opcional) Consulte a tabela de histórico de espaços de tabela transportáveis para ver o status da importação de metadados.

   ```
   SELECT * FROM rdsadmin.rds_xtts_operation_info;
   ```

Quando a operação for concluída, os espaços de tabela ficarão no modo somente leitura.

5. (Opcional) Veja o arquivo de log.

   O exemplo a seguir lista o conteúdo do diretório BDUMP, depois consulta o log de importação.

```
SELECT * FROM TABLE(rdsadmin.rds_file_util.listdir(p_directory => 'BDUMP'));

SELECT * FROM TABLE(rdsadmin.rds_file_util.read_text_file(
  p_directory => 'BDUMP',
  p_filename => 'rds-xtts-
import_xtts_metadata-2023-05-22.01-52-35.560858000.log'));
```

Fase 5: Validar os espaços de tabela transportados

Nesta etapa opcional, você validará os espaços de tabela transportados usando o procedimento `rdsadmin.rdsadmin_rman_util.validate_tablespace`, depois colocará os espaços de tabela no modo de leitura/gravação.

Como validar os dados transportados

1. Inicie o SQL*Plus ou o SQL Developer e faça login como usuário principal em sua instância de banco de dados de destino.
2. Valide os espaços de tabela usando o procedimento `rdsadmin.rdsadmin_rman_util.validate_tablespace`.

```
SET SERVEROUTPUT ON
BEGIN
    rdsadmin.rdsadmin_rman_util.validate_tablespace(
        p_tablespace_name     => 'TBS1',
        p_validation_type     => 'PHYSICAL+LOGICAL',
        p_rman_to_dbms_output => TRUE);
    rdsadmin.rdsadmin_rman_util.validate_tablespace(
        p_tablespace_name     => 'TBS2',
        p_validation_type     => 'PHYSICAL+LOGICAL',
        p_rman_to_dbms_output => TRUE);
    rdsadmin.rdsadmin_rman_util.validate_tablespace(
        p_tablespace_name     => 'TBS3',
        p_validation_type     => 'PHYSICAL+LOGICAL',
        p_rman_to_dbms_output => TRUE);
```

```
END;
/
```

3. Coloque os espaços de tabela no modo de leitura/gravação.

```
ALTER TABLESPACE TBS1 READ WRITE;
ALTER TABLESPACE TBS2 READ WRITE;
ALTER TABLESPACE TBS3 READ WRITE;
```

Fase 6: Limpar os arquivos restantes

Nesta etapa opcional, você removerá todos os arquivos desnecessários. Use o procedimento `rdsadmin.rdsadmin_transport_util.list_xtts_orphan_files` para listar arquivos de dados que ficaram órfãos após a importação de um espaço de tabela e, depois, use o procedimento `rdsadmin.rdsadmin_transport_util.list_xtts_orphan_files` para excluí-los. Para conhecer a sintaxe e a semântica desses procedimentos, consulte Listar arquivos órfãos após a importação de um espaço para tabela e Excluir arquivos de dados órfãos após a importação de espaços para tabela.

Como limpar os arquivos restantes

1. Remova os backups antigos em *DATA_PUMP_DIR* da seguinte forma:

   a. Liste os arquivos de backup executando `rdsadmin.rdsadmin_file_util.listdir`.

   ```
   SELECT * FROM TABLE(rdsadmin.rds_file_util.listdir(p_directory =>
    'DATA_PUMP_DIR'));
   ```

   b. Remova os backups um por um chamando `UTL_FILE.FREMOVE`.

   ```
   EXEC UTL_FILE.FREMOVE ('DATA_PUMP_DIR', 'backup_filename');
   ```

2. Se você importou espaços de tabela, mas não importou metadados para esses espaços de tabela, poderá excluir os arquivos de dados órfãos da seguinte forma:

   a. Liste os arquivos de dados órfãos que você precisa excluir. O exemplo a seguir executa o procedimento `rdsadmin.rdsadmin_transport_util.list_xtts_orphan_files`.

   ```
   SQL> SELECT * FROM
    TABLE(rdsadmin.rdsadmin_transport_util.list_xtts_orphan_files);
   ```

```
FILENAME       FILESIZE
-------------- ---------
datafile_7.dbf 104865792
datafile_8.dbf 104865792
```

b. Exclua os arquivos órfãos executando o procedimento `rdsadmin.rdsadmin_transport_util.cleanup_incomplete_xtts_import`.

```
BEGIN

 rdsadmin.rdsadmin_transport_util.cleanup_incomplete_xtts_import('DATA_PUMP_DIR');
END;
/
```

A operação de limpeza gera um arquivo de log que usa o formato de nome `rds-xtts-delete_xtts_orphaned_files-`*`YYYY-MM-DD.HH24-MI-SS.FF`*`.log` no diretório BDUMP.

c. Leia o arquivo de log gerado na etapa anterior. O exemplo a seguir lê o log `rds-xtts-delete_xtts_orphaned_files-`*`2023-06-01.09-33-11.868894000`*`.log`.

```
SELECT *
FROM TABLE(rdsadmin.rds_file_util.read_text_file(
       p_directory => 'BDUMP',
       p_filename  => 'rds-xtts-
delete_xtts_orphaned_files-2023-06-01.09-33-11.868894000.log'));

TEXT
--------------------------------------------------------------------------------
orphan transported datafile datafile_7.dbf deleted.
orphan transported datafile datafile_8.dbf deleted.
```

3. Se você importou espaços de tabela e importou metadados para esses espaços de tabela, mas encontrou erros de compatibilidade ou outros problemas do Oracle Data Pump, limpe os arquivos de dados parcialmente transportados da seguinte forma:

a. Liste os espaços de tabela que contêm arquivos de dados parcialmente transportados consultando `DBA_TABLESPACES`.

```
SQL> SELECT TABLESPACE_NAME FROM DBA_TABLESPACES WHERE PLUGGED_IN='YES';
```

```
TABLESPACE_NAME
------------------------------------------------------------------------------
TBS_3
```

b. Elimine os espaços de tabela e os arquivos de dados parcialmente transportados.

```
DROP TABLESPACE TBS_3 INCLUDING CONTENTS AND DATAFILES;
```

Importar usando o Oracle Data Pump

O Oracle Data Pump é um utilitário que permite exportar dados Oracle para um arquivo de despejo e importá-los para outro banco de dados Oracle. Ele é uma substituição a longo prazo para os utilitários de importação/exportação da Oracle. O Oracle Data Pump é a maneira preferida de mover grandes quantidades de dados de um banco de dados Oracle para uma instância de banco de dados do Amazon RDS.

Os exemplos desta seção mostram uma maneira de importar dados para um banco de dados Oracle, o Oracle Data Pump oferece suporte a outras técnicas. Para obter mais informações, consulte a documentação do Oracle Database.

Os exemplos desta seção usam o pacote `DBMS_DATAPUMP`. Você pode realizar as mesmas tarefas usando os utilitários da linha de comando do Oracle Data Pump `impdp` e `expdp`. Você pode instalar esses utilitários em um host remoto como parte de uma instalação do Oracle Client, incluindo o Oracle Instant Client. Para obter mais informações, consulte "Como usar o Oracle Instant Client para executar a importação ou exportação do Data Pump para a minha instância de banco de dados do Amazon RDS para Oracle?"

Tópicos

- Visão geral do Oracle Data Pump
- Importar dados com o Oracle Data Pump e um bucket do Amazon S3
- Importar dados com o Oracle Data Pump e um link de banco de dados

Visão geral do Oracle Data Pump

O Oracle Data Pump tem os seguintes componentes:

- Clientes de linha de comando `expdp` e `impdp`

- O pacote PL/SQL `DBMS_DATAPUMP`
- O pacote PL/SQL `DBMS_METADATA`

Você pode usar o Oracle Data Pump nos seguintes cenários:

- Importar dados de um banco de dados Oracle, on-premises ou em uma instância do Amazon EC2), para uma instância de banco de dados do RDS para Oracle.
- Importar dados de uma instância de banco de dados do RDS para Oracle para um banco de dados Oracle, on-premises ou em uma instância do Amazon EC2.
- Importar dados entre instâncias de banco de dados do RDS para Oracle, por exemplo, para migrar dados do EC2-Classic para a VPC.

Para baixar os utilitários do Oracle Data Pump, consulte "Oracle database software downloads" (Downloads do software Oracle Database) no site da Oracle Technology Network. Para considerações sobre compatibilidade ao migrar entre versões do Oracle Database, consulte a documentação do Oracle Database.

Fluxo de trabalho do Oracle Data Pump

Normalmente, você usa o Oracle Data Pump nos seguintes estágios:

1. Exportar seus dados em um arquivo de despejo no banco de dados de origem.
2. Carregar seu arquivo de despejo em sua instância de destino do RDS para Oracle. Você pode transferir usando um bucket do Amazon S3 ou usando um link de banco de dados entre os dois bancos de dados.
3. Importe os dados de seu arquivo de despejo em sua instância de banco de dados do RDS para Oracle.

Práticas recomendadas do Oracle Data Pump

Ao usar o Oracle Data Pump para importar dados para uma instância do RDS para Oracle, recomendamos as seguintes práticas:

- Faça importações no modo `schema` ou `table` para importar esquemas e objetos específicos.
- Limite os esquemas que você importa àqueles exigidos por sua aplicação.
- Não importe no modo `full` ou importar esquemas para componentes mantidos pelo sistema.

Como o RDS para Oracle não permite o acesso a usuários administrativos do SYS ou SYSDBA, essas ações podem danificar o dicionário de dados da Oracle e afetar a estabilidade de seu banco de dados.

- Ao carregar grandes quantidades de dados, faça o seguinte:
  1. Transfira o arquivo de despejo para a instância de banco de dados do RDS para Oracle de destino.
  2. Crie um snapshot do banco de dados da instância.
  3. Teste a importação para verificar se ela é bem-sucedida.

  Se os componentes do banco de dados forem invalidados, você poderá excluir a instância de banco de dados e recriá-la a partir do snapshot. A instância de banco de dados restaurada incluirá qualquer arquivo de despejo preparado na instância de banco de dados quando você criou o snapshot.
- Não importe arquivos de despejo que foram criados usando os parâmetros de exportação do Oracle Data Pump TRANSPORT_TABLESPACES, TRANSPORTABLE ou TRANSPORT_FULL_CHECK. As instâncias de banco de dados do RDS para Oracle não oferecem suporte à importação desses arquivos de despejo.
- Não importe arquivos de despejo que contenham objetos do Oracle Scheduler em SYS, SYSTEM, RDSADMIN, RDSSEC e RDS_DATAGUARD e pertençam às seguintes categorias:
  - Tarefas
  - Programas
  - Programações
  - Cadeias
  - Regras
  - Contextos de avaliação
  - Conjuntos de regras

  As instâncias de banco de dados do RDS para Oracle não oferecem suporte à importação desses arquivos de despejo.
- Para excluir objetos do Oracle Scheduler não aceitos, use diretivas adicionais durante a exportação do Data Pump. Se usar o DBMS_DATAPUMP, você poderá adicionar um METADATA_FILTER antes do DBMS_METADATA.START_JOB:

```
DBMS_DATAPUMP.METADATA_FILTER(
```

```
  v_hdnl,
  'EXCLUDE_NAME_EXPR',
  q'[IN (SELECT NAME FROM SYS.OBJ$
         WHERE TYPE# IN (66,67,74,79,59,62,46)
         AND OWNER# IN
           (SELECT USER# FROM SYS.USER$
            WHERE NAME IN ('RDSADMIN','SYS','SYSTEM','RDS_DATAGUARD','RDSSEC')
            )
        )
  ]',
  'PROCOBJ'
);
```

Se você usar o `expdp`, crie um arquivo de parâmetro que contenha a diretiva `exclude` mostrada no exemplo a seguir. Em seguida, use `PARFILE=`*`parameter_file`* com o comando `expdp`.

```
exclude=procobj:"IN
  (SELECT NAME FROM sys.OBJ$
   WHERE TYPE# IN (66,67,74,79,59,62,46)
   AND OWNER# IN
     (SELECT USER# FROM SYS.USER$
      WHERE NAME IN ('RDSADMIN','SYS','SYSTEM','RDS_DATAGUARD','RDSSEC')
     )
  )"
```

Importar dados com o Oracle Data Pump e um bucket do Amazon S3

O processo de importação a seguir usa o Oracle Data Pump e um bucket do Amazon S3. As etapas são:

1. Exporte dados no banco de dados de origem usando o pacoteDBMS_DATAPUMP da Oracle.
2. Coloque o arquivo de despejo em um bucket do Amazon S3.
3. Baixe o arquivo de despejo do bucket do Amazon S3 no diretório `DATA_PUMP_DIR` na instância de destino do banco de dados do RDS para Oracle.
4. Importe os dados do arquivo de despejo copiado para a instância de banco de dados do RDS para Oracle usando o pacote `DBMS_DATAPUMP`.

Tópicos

- Requisitos para importar dados com o Oracle Data Pump e um bucket do Amazon S3
- Etapa 1: Conceder privilégios ao usuário do banco de dados na instância de banco de dados de destino do RDS para Oracle
- Etapa 2: Exportar dados para um arquivo de despejo usando DBMS_DATAPUMP
- Etapa 3: Carregar o arquivo de despejo para o seu bucket do Amazon S3
- Etapa 4: Baixe o arquivo no bucket do Amazon S3 para a instância de banco de dados de destino
- Etapa 5: Importar seu arquivo de despejo para sua instância de banco de dados de destino usando DBMS_DATAPUMP
- Etapa 6: Limpeza

Requisitos para importar dados com o Oracle Data Pump e um bucket do Amazon S3

O processo tem estes requisitos:

- Verifique se há um bucket do Amazon S3 disponível para transferências de arquivos e se o bucket do Amazon S3 está na mesma Região da AWS que a instância de banco de dados. Para obter mais informações, consulte Criar um bucket no Guia de conceitos básicos do Amazon Simple Storage Service.
- O objeto carregado no bucket do Amazon S3 deve ter 5 TB ou menos. Para obter mais informações sobre como trabalhar com objetos no Amazon S3, consulte o Manual do usuário do Amazon Simple Storage Service.

  > **Note**
  >
  > Se o arquivo de despejo for superior a 5 TB, será possível executar a exportação do Oracle Data Pump com a opção paralela. Essa operação distribui os dados em vários arquivos de despejo para que você não exceda o limite de 5 TB para arquivos individuais.

- Você deve preparar o bucket do Amazon S3 para a integração com o Amazon RDS seguindo as instruções em Configurar permissões do IAM para a integração do RDS para Oracle com o Amazon S3.
- Você deve ter espaço de armazenamento suficiente para armazenar o arquivo de despejo na instância de origem e na instância de banco de dados de destino.

> **Note**
>
> Esse processo importa um arquivo de despejo no diretório `DATA_PUMP_DIR`, um diretório pré-configurado em todas as instâncias de banco de dados Oracle. Esse diretório está localizado no mesmo volume de armazenamento que seus arquivos de dados. Ao importar o arquivo de despejo, os arquivos de dados existes da Oracle usam mais espaço. Dessa forma, você deve verificar se a sua instância de banco de dados pode acomodar o uso de espaço adicional. O arquivo de despejo importado não é excluído ou limpo automaticamente do diretório `DATA_PUMP_DIR`. Para remover o arquivo de despejo importado, use UTL_FILE.FREMOVE, que pode ser encontrado no site da Oracle.

Etapa 1: Conceder privilégios ao usuário do banco de dados na instância de banco de dados de destino do RDS para Oracle

Nesta etapa, você cria os esquemas nos quais planeja importar dados e concede aos usuários os privilégios necessários.

Para criar usuários e conceder os privilégios necessários na instância do RDS para Oracle de destino

1. Use o SQL*Plus ou o Oracle SQL Developer para fazer login como usuário mestre na instância de banco de dados do RDS para Oracle para onde os dados serão importados. Para saber mais sobre como se conectar à sua instância de banco de dados, consulte Conectar-se a uma instância de banco de dados Oracle.
2. Crie os espaços de tabela necessários antes de importar os dados. Para obter mais informações, consulte Criação e uso de espaços de tabela.
3. Crie a conta de usuário e conceda as permissões e funções necessárias se a conta de usuário na qual os dados serão importados não existir. Se você planeja importar dados para vários esquemas de usuários, crie cada conta de usuário e conceda as funções e os privilégios necessários a ela.

   Por exemplo, as instruções SQL a seguir criam um novo usuário e concedem as permissões e funções necessárias para importar os dados para o esquema de propriedade do usuário. Substitua *schema_1* pelo nome de seu schema nesta e nas etapas a seguir.

   ```
   CREATE USER schema_1 IDENTIFIED BY my_password;
   GRANT CREATE SESSION, RESOURCE TO schema_1;
   ```

```
ALTER USER schema_1 QUOTA 100M ON users;
```

> **Note**
>
> Especifique uma senha diferente do prompt mostrado aqui como prática recomendada de segurança.

As instruções anteriores concedem ao novo usuário o privilégio `CREATE SESSION` e a função `RESOURCE`. Podem ser necessários privilégios e funções adicionais dependendo dos objetos de banco de dados que você importar.

Etapa 2: Exportar dados para um arquivo de despejo usando DBMS_DATAPUMP

Para criar um arquivo de despejo, use o pacote `DBMS_DATAPUMP`.

Como exportar dados Oracle para um arquivo de despejo

1. Use o SQL Plus ou o Oracle SQL Developer para se conectar à instância de banco de dados do RDS para Oracle de origem por meio de um usuário administrativo. Se o banco de dados de origem for uma instância de banco de dados do RDS para Oracle, conecte-se com o usuário mestre do Amazon RDS.
2. Exporte os dados chamando procedimentos do `DBMS_DATAPUMP`.

   O script a seguir exporta o esquema *`SCHEMA_1`* em um arquivo de despejo chamado `sample.dmp` no diretório `DATA_PUMP_DIR`. Substitua *`SCHEMA_1`* pelo nome do esquema que você quer exportar.

```
DECLARE
  v_hdnl NUMBER;
BEGIN
  v_hdnl := DBMS_DATAPUMP.OPEN(
    operation => 'EXPORT',
    job_mode  => 'SCHEMA',
    job_name  => null
  );
  DBMS_DATAPUMP.ADD_FILE(
    handle    => v_hdnl         ,
    filename  => 'sample.dmp'   ,
```

```
    directory => 'DATA_PUMP_DIR',
    filetype  => dbms_datapump.ku$_file_type_dump_file
  );
  DBMS_DATAPUMP.ADD_FILE(
    handle    => v_hdnl,
    filename  => 'sample_exp.log',
    directory => 'DATA_PUMP_DIR' ,
    filetype  => dbms_datapump.ku$_file_type_log_file
  );
  DBMS_DATAPUMP.METADATA_FILTER(v_hdnl,'SCHEMA_EXPR','IN (''SCHEMA_1'')');
  DBMS_DATAPUMP.METADATA_FILTER(
    v_hdnl,
    'EXCLUDE_NAME_EXPR',
    q'[IN (SELECT NAME FROM SYS.OBJ$
           WHERE TYPE# IN (66,67,74,79,59,62,46)
           AND OWNER# IN
             (SELECT USER# FROM SYS.USER$
              WHERE NAME IN ('RDSADMIN','SYS','SYSTEM','RDS_DATAGUARD','RDSSEC')
             )
          )
    ]',
    'PROCOBJ'
  );
  DBMS_DATAPUMP.START_JOB(v_hdnl);
END;
/
```

> **Note**
>
> O Data Pump inicia os trabalhos de forma assíncrona. Para obter informações sobre o monitoramento de um trabalho do Data Pump, consulte Monitoring Job Status (Monitorar status do trabalho) na documentação da Oracle.

3. (Opcional) Você pode visualizar o conteúdo do log de exportação chamando o procedimento `rdsadmin.rds_file_util.read_text_file`. Para obter mais informações, consulte Leitura de arquivos no diretório de uma instância de banco de dados.

Etapa 3: Carregar o arquivo de despejo para o seu bucket do Amazon S3

Use o procedimento `rdsadmin.rdsadmin_s3_tasks.upload_to_s3` do Amazon RDS para copiar o arquivo de despejo no bucket do Amazon S3. O exemplo a seguir faz upload todos os

arquivos do diretório DATA_PUMP_DIR para um bucket do Amazon S3 chamado *amzn-s3-demo-bucket*.

```
SELECT rdsadmin.rdsadmin_s3_tasks.upload_to_s3(
  p_bucket_name    =>  'amzn-s3-demo-bucket',
  p_directory_name =>  'DATA_PUMP_DIR')
AS TASK_ID FROM DUAL;
```

A instrução SELECT retorna o ID da tarefa em um tipo de dados VARCHAR2. Para obter mais informações, consulte Carregar arquivos de uma instância de banco de dados do RDS para Oracle para um bucket do Amazon S3.

Etapa 4: Baixe o arquivo no bucket do Amazon S3 para a instância de banco de dados de destino

Execute esta etapa usando o procedimento `rdsadmin.rdsadmin_s3_tasks.download_from_s3` do Amazon RDS. Quando você baixar um arquivo em um diretório, o procedimento `download_from_s3` ignorará o download se um arquivo com nome idêntico já existir no diretório. Para remover um arquivo do diretório de downloads, use UTL_FILE.FREMOVE, que pode ser encontrado no site da Oracle.

Como baixar seu arquivo de despejo

1. Inicie o SQL*Plus ou o Oracle SQL Developer e faça login como mestre em sua instância de banco de dados Oracle de destino do Amazon RDS
2. Baixe o arquivo de despejo usando o procedimento do Amazon RDS `rdsadmin.rdsadmin_s3_tasks.download_from_s3`.

   O exemplo a seguir baixa todos os arquivos de um bucket do Amazon S3 chamado *amzn-s3-demo-bucket* no diretório DATA_PUMP_DIR.

   ```
   SELECT rdsadmin.rdsadmin_s3_tasks.download_from_s3(
     p_bucket_name    =>  'amzn-s3-demo-bucket',
     p_directory_name =>  'DATA_PUMP_DIR')
   AS TASK_ID FROM DUAL;
   ```

   A instrução SELECT retorna o ID da tarefa em um tipo de dados VARCHAR2. Para obter mais informações, consulte Baixar arquivos de um bucket do Amazon S3 para uma instância de banco de dados Oracle.

Etapa 5: Importar seu arquivo de despejo para sua instância de banco de dados de destino usando DBMS_DATAPUMP

Use o `DBMS_DATAPUMP` para importar o esquema em sua instância de banco de dados do RDS para Oracle. Opções adicionais, como `METADATA_REMAP`, podem ser necessárias.

Para importar dados para sua instância de banco de dados de destino

1. Inicie o SQL*Plus ou o SQL Developer e faça login como usuário mestre em sua instância de banco de dados do RDS para Oracle.
2. Importe os dados chamando procedimentos `DBMS_DATAPUMP`.

   O exemplo a seguir importa os dados de *`SCHEMA_1`* de `sample_copied.dmp` em sua instância de banco de dados de destino.

```
DECLARE
  v_hdnl NUMBER;
BEGIN
  v_hdnl := DBMS_DATAPUMP.OPEN(
    operation => 'IMPORT',
    job_mode  => 'SCHEMA',
    job_name  => null);
  DBMS_DATAPUMP.ADD_FILE(
    handle    => v_hdnl,
    filename  => 'sample_copied.dmp',
    directory => 'DATA_PUMP_DIR',
    filetype  => dbms_datapump.ku$_file_type_dump_file);
  DBMS_DATAPUMP.ADD_FILE(
    handle    => v_hdnl,
    filename  => 'sample_imp.log',
    directory => 'DATA_PUMP_DIR',
    filetype  => dbms_datapump.ku$_file_type_log_file);
  DBMS_DATAPUMP.METADATA_FILTER(v_hdnl,'SCHEMA_EXPR','IN (''SCHEMA_1'')');
  DBMS_DATAPUMP.START_JOB(v_hdnl);
END;
/
```

> **Note**
>
> Os trabalhos do Data Pump são iniciados de forma assíncrona. Para obter informações sobre o monitoramento de um trabalho do Data Pump, consulte

Monitoring Job Status (Monitorar status do trabalho) na documentação da Oracle. Você pode visualizar o conteúdo do log de importação usando o procedimento `rdsadmin.rds_file_util.read_text_file`. Para obter mais informações, consulte Leitura de arquivos no diretório de uma instância de banco de dados.

3. Verifique a importação de dados listando as tabelas de esquema na sua instância de banco de dados de destino.

   Por exemplo, a consulta a seguir retorna o número de tabelas para o *SCHEMA_1*.

```
SELECT COUNT(*) FROM DBA_TABLES WHERE OWNER='SCHEMA_1';
```

Etapa 6: Limpeza

Depois que os dados foram importados, você pode excluir os arquivos que não deseja manter.

Como remover arquivos desnecessários

1. Inicie o SQL*Plus ou o SQL Developer e faça login como usuário mestre em sua instância de banco de dados do RDS para Oracle.
2. Liste os arquivos no `DATA_PUMP_DIR` usando o comando a seguir.

```
SELECT * FROM TABLE(rdsadmin.rds_file_util.listdir('DATA_PUMP_DIR')) ORDER BY
 MTIME;
```

3. Para excluir arquivos no `DATA_PUMP_DIR` que não são mais necessários, use o comando a seguir.

```
EXEC UTL_FILE.FREMOVE('DATA_PUMP_DIR','filename');
```

   Por exemplo, o seguinte comando exclui o arquivo chamado `sample_copied.dmp`.

```
EXEC UTL_FILE.FREMOVE('DATA_PUMP_DIR','sample_copied.dmp');
```

Importar dados com o Oracle Data Pump e um link de banco de dados

O processo a seguir usa o Oracle Data Pump e o pacote DBMS_FILE_TRANSFER do Oracle. As etapas são:

1. Conecte-se a um banco de dados Oracle de origem, que pode ser um banco de dados on-premises, uma instância do Amazon EC2 ou uma instância de banco de dados do RDS para Oracle.
2. Exporte dados usando o pacote DBMS_DATAPUMP.
3. Ele usa `DBMS_FILE_TRANSFER.PUT_FILE` para copiar o arquivo de despejo do banco de dados Oracle para o diretório `DATA_PUMP_DIR` na instância do banco de dados do RDS para Oracle de destino que está conectada por meio de um link de banco de dados.
4. Importe os dados do arquivo de despejo copiado para a instância de banco de dados do RDS para Oracle usando o pacote `DBMS_DATAPUMP`.

O processo de importação usando o Oracle Data Pump e o pacote `DBMS_FILE_TRANSFER` tem as etapas a seguir.

Tópicos

- Requisitos para importar dados com o Oracle Data Pump e um link de banco de dados
- Etapa 1: Conceder privilégios ao usuário na instância de destino do RDS para Oracle
- Etapa 2: Conceder privilégios ao usuário no banco de dados de origem
- Etapa 3: Criar um arquivo de despejo usando DBMS_DATAPUMP
- Etapa 4: Criar o vínculo de banco de dados com a instância de banco de dados de destino
- Etapa 5: Copiar o arquivo de despejo exportado para a instância de banco de dados de destino usando DBMS_FILE_TRANSFER
- Etapa 6: Importar o arquivo de dados para a instância de banco de dados de destino usando DBMS_DATAPUMP
- Etapa 7: Limpeza

Requisitos para importar dados com o Oracle Data Pump e um link de banco de dados

O processo tem estes requisitos:

- É necessário ter privilégios de execução nos pacotes `DBMS_FILE_TRANSFER` e `DBMS_DATAPUMP`.

- É necessário ter privilégios de gravação no diretório DATA_PUMP_DIR na instância de banco de dados de origem.
- Você deve ter espaço de armazenamento suficiente para armazenar o arquivo de despejo na instância de origem e na instância de banco de dados de destino.

> **Note**
>
> Esse processo importa um arquivo de despejo no diretório DATA_PUMP_DIR, um diretório pré-configurado em todas as instâncias de banco de dados Oracle. Esse diretório está localizado no mesmo volume de armazenamento que seus arquivos de dados. Ao importar o arquivo de despejo, os arquivos de dados existes da Oracle usam mais espaço. Dessa forma, você deve verificar se a sua instância de banco de dados pode acomodar o uso de espaço adicional. O arquivo de despejo importado não é excluído ou limpo automaticamente do diretório DATA_PUMP_DIR. Para remover o arquivo de despejo importado, use UTL_FILE.FREMOVE, que pode ser encontrado no site da Oracle.

Etapa 1: Conceder privilégios ao usuário na instância de destino do RDS para Oracle

Para conceder privilégios ao usuário na instância de banco de dados de destino do RDS para Oracle, sigas estas etapas:

1. Use o SQL Plus ou o Oracle SQL Developer para se conectar à instância de banco de dados do RDS para Oracle para onde você pretende importar os dados. Conecte-se como usuário mestre do Amazon RDS. Para obter informações sobre como se conectar à sua instância de banco de dados, consulte Conectar-se a uma instância de banco de dados Oracle.
2. Crie os espaços de tabela necessários antes de importar os dados. Para obter mais informações, consulte Criação e uso de espaços de tabela.
3. Se a conta de usuário na qual os dados serão importados não existir, crie-a e conceda as permissões e funções necessárias a ela. Se você planeja importar dados para vários esquemas de usuários, crie cada conta de usuário e conceda as funções e os privilégios necessários a ela.

   Por exemplo, os comandos a seguir criam um novo usuário chamado *schema_1* e concedem as permissões e funções necessárias para importar os dados para o esquema deste usuário.

```
CREATE USER schema_1 IDENTIFIED BY my-password;
GRANT CREATE SESSION, RESOURCE TO schema_1;
```

```
ALTER USER schema_1 QUOTA 100M ON users;
```

> **Note**
>
> Especifique uma senha diferente do prompt mostrado aqui como prática recomendada de segurança.

O exemplo anterior concede ao novo usuário o privilégio `CREATE SESSION` e a função `RESOURCE`. Podem ser necessários privilégios e funções adicionais dependendo dos objetos de banco de dados que você importa.

> **Note**
>
> Substitua *schema_1* pelo nome de seu schema nesta e nas etapas a seguir.

Etapa 2: Conceder privilégios ao usuário no banco de dados de origem

Use o SQL*Plus ou o Oracle SQL Developer para se conectar à instância de banco de dados do RDS para Oracle que contém os dados a serem importados. Se necessário, crie uma conta de usuário e conceda as permissões necessárias.

> **Note**
>
> Se o banco de dados de origem for uma instância do Amazon RDS, você poderá ignorar esta etapa. Use sua conta de usuário mestre do Amazon RDS para executar a exportação.

Os comandos a seguir criam um novo usuário e concedem as permissões necessárias.

```
CREATE USER export_user IDENTIFIED BY my-password;
GRANT CREATE SESSION, CREATE TABLE, CREATE DATABASE LINK TO export_user;
ALTER USER export_user QUOTA 100M ON users;
GRANT READ, WRITE ON DIRECTORY data_pump_dir TO export_user;
GRANT SELECT_CATALOG_ROLE TO export_user;
GRANT EXECUTE ON DBMS_DATAPUMP TO export_user;
GRANT EXECUTE ON DBMS_FILE_TRANSFER TO export_user;
```

> **Note**
>
> Especifique uma senha diferente do prompt mostrado aqui como prática recomendada de segurança.

Etapa 3: Criar um arquivo de despejo usando DBMS_DATAPUMP

Para criar o arquivo de despejo, faça o seguinte:

1. Use o SQL*Plus ou o Oracle SQL Developer para se conectar à instância de origem do Oracle por meio de um usuário administrativo ou o usuário que você criou na etapa 2. Se o banco de dados de origem for uma instância de banco de dados do Amazon RDS para Oracle, conecte-se com o usuário mestre do Amazon RDS.
2. Crie um arquivo de despejo usando o utilitário Oracle Data Pump.

   O script a seguir cria um arquivo de despejo chamado sample.dmp no diretório `DATA_PUMP_DIR`.

```
DECLARE
  v_hdnl NUMBER;
BEGIN
  v_hdnl := DBMS_DATAPUMP.OPEN(
    operation => 'EXPORT' ,
    job_mode  => 'SCHEMA' ,
    job_name  => null
  );
  DBMS_DATAPUMP.ADD_FILE(
    handle    => v_hdnl,
    filename  => 'sample.dmp'    ,
    directory => 'DATA_PUMP_DIR' ,
    filetype  => dbms_datapump.ku$_file_type_dump_file
  );
  DBMS_DATAPUMP.ADD_FILE(
    handle    => v_hdnl           ,
    filename  => 'sample_exp.log' ,
    directory => 'DATA_PUMP_DIR'  ,
    filetype  => dbms_datapump.ku$_file_type_log_file
  );
  DBMS_DATAPUMP.METADATA_FILTER(
    v_hdnl              ,
    'SCHEMA_EXPR'       ,
    'IN (''SCHEMA_1'')'
```

```
  );
  DBMS_DATAPUMP.METADATA_FILTER(
    v_hdnl,
    'EXCLUDE_NAME_EXPR',
    q'[IN (SELECT NAME FROM sys.OBJ$
           WHERE TYPE# IN (66,67,74,79,59,62,46)
           AND OWNER# IN
             (SELECT USER# FROM SYS.USER$
              WHERE NAME IN ('RDSADMIN','SYS','SYSTEM','RDS_DATAGUARD','RDSSEC')
             )
          )
    ]',
    'PROCOBJ'
  );
  DBMS_DATAPUMP.START_JOB(v_hdnl);
END;
/
```

> **Note**
>
> Os trabalhos do Data Pump são iniciados de forma assíncrona. Para obter informações sobre o monitoramento de um trabalho do Data Pump, consulte Monitoring Job Status (Monitorar status do trabalho) na documentação da Oracle.
> É possível visualizar o conteúdo do log de exportação usando o procedimento `rdsadmin.rds_file_util.read_text_file`. Para obter mais informações, consulte Leitura de arquivos no diretório de uma instância de banco de dados.

Etapa 4: Criar o vínculo de banco de dados com a instância de banco de dados de destino

Crie um link de banco de dados entre sua instância de banco de dados de origem e sua instância de banco de dados de destino. Observe que sua instância local do Oracle deve ter conectividade de rede com a instância de banco de dados para criar um link de banco de dados e transferir seu arquivo de despejo.

Realize esta etapa conectado com a mesma conta de usuário da etapa anterior.

Se você estiver criando um link de banco de dados entre duas instâncias de bancos de dados na mesma VPC ou VPCs emparelhadas, as duas instâncias de bancos de dados deverão ter uma rota válida entre elas. O grupo de segurança de cada instância de banco de dados deve permitir a entrada e a saída da outra instância de banco de dados. As regras de entrada e saída

do grupo de segurança podem se referir a grupos de segurança da mesma VPC ou de uma VPC emparelhada. Para obter mais informações, consulte Ajuste de vínculos de banco de dados para uso com instâncias de bancos de dados em uma VPC.

O comando a seguir cria um link de banco de dados chamado `to_rds` que se conecta ao usuário mestre do Amazon RDS na instância de banco de dados de destino.

```
CREATE DATABASE LINK to_rds
  CONNECT TO <master_user_account> IDENTIFIED BY <password>
  USING '(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=<dns or ip address of remote db>)
         (PORT=<listener port>))(CONNECT_DATA=(SID=<remote SID>)))';
```

Etapa 5: Copiar o arquivo de despejo exportado para a instância de banco de dados de destino usando DBMS_FILE_TRANSFER

Use `DBMS_FILE_TRANSFER` para copiar o arquivo de despejo da instância do banco de dados de origem para a instância de banco de dados de destino. O script a seguir copia um arquivo de despejo chamado sample.dmp da instância de origem para um link de banco de dados de destino chamado to_rds (criado na etapa anterior).

```
BEGIN
  DBMS_FILE_TRANSFER.PUT_FILE(
    source_directory_object       => 'DATA_PUMP_DIR',
    source_file_name              => 'sample.dmp',
    destination_directory_object  => 'DATA_PUMP_DIR',
    destination_file_name         => 'sample_copied.dmp',
    destination_database          => 'to_rds' );
END;
/
```

Etapa 6: Importar o arquivo de dados para a instância de banco de dados de destino usando DBMS_DATAPUMP

Use o Oracle Data Pump para importar o esquema na instância de banco de dados. Opções adicionais, como METADATA_REMAP, podem ser necessárias.

Conecte-se à instância de banco de dados com a conta de usuário mestre do Amazon RDS para executar a importação.

```
DECLARE
```

```
  v_hdnl NUMBER;
BEGIN
  v_hdnl := DBMS_DATAPUMP.OPEN(
    operation => 'IMPORT',
    job_mode  => 'SCHEMA',
    job_name  => null);
  DBMS_DATAPUMP.ADD_FILE(
    handle    => v_hdnl,
    filename  => 'sample_copied.dmp',
    directory => 'DATA_PUMP_DIR',
    filetype  => dbms_datapump.ku$_file_type_dump_file );
  DBMS_DATAPUMP.ADD_FILE(
    handle    => v_hdnl,
    filename  => 'sample_imp.log',
    directory => 'DATA_PUMP_DIR',
    filetype  => dbms_datapump.ku$_file_type_log_file);
  DBMS_DATAPUMP.METADATA_FILTER(v_hdnl,'SCHEMA_EXPR','IN (''SCHEMA_1'')');
  DBMS_DATAPUMP.START_JOB(v_hdnl);
END;
/
```

> **Note**
>
> Os trabalhos do Data Pump são iniciados de forma assíncrona. Para obter informações sobre o monitoramento de um trabalho do Data Pump, consulte Monitoring Job Status (Monitorar status do trabalho) na documentação da Oracle. Você pode visualizar o conteúdo do log de importação usando o procedimento `rdsadmin.rds_file_util.read_text_file`. Para obter mais informações, consulte Leitura de arquivos no diretório de uma instância de banco de dados.

Você pode verificar a importação de dados visualizando a tabela do usuário na instância de banco de dados. Por exemplo, a consulta a seguir retorna o número de tabelas para o *schema_1*.

```
SELECT COUNT(*) FROM DBA_TABLES WHERE OWNER='SCHEMA_1';
```

Etapa 7: Limpeza

Depois que os dados foram importados, você pode excluir os arquivos que não deseja manter. É possível listar os arquivos DATA_PUMP_DIR usando o seguinte comando.

```
SELECT * FROM TABLE(rdsadmin.rds_file_util.listdir('DATA_PUMP_DIR')) ORDER BY MTIME;
```

Para excluir arquivos no DATA_PUMP_DIR que não são mais necessários, use o comando a seguir.

```
EXEC UTL_FILE.FREMOVE('DATA_PUMP_DIR','<file name>');
```

Por exemplo, o seguinte comando exclui o arquivo chamado "sample_copied.dmp".

```
EXEC UTL_FILE.FREMOVE('DATA_PUMP_DIR','sample_copied.dmp');
```

Importar usando Oracle Export/Import

Pode ser útil considerar os utilitários Oracle Export/Import para migrações nas seguintes condições:

- O volume dos seus dados é pequeno.
- Tipos de dados como flutuante binário e duplo não são necessários.

O processo de importação cria os objetos de esquema necessários. Assim, você não precisa executar um script para criar os objetos antecipadamente.

A maneira mais fácil de instalar o Oracle e os utilitários de exportação e importação, é instalar o Oracle Instant Client. Para baixar o software, acesse https://www.oracle.com/database/technologies/instant-client.html. Para obter a documentação, consulte Instant Client for SQL*Loader, exportação e importação no manual Utilitários do banco de dados Oracle.

Como exportar e importar tabelas

1. Exporte as tabelas do banco de dados de origem usando o comando exp.

   O comando a seguir exporta as tabelas chamadas tab1, tab2 e tab3. O arquivo de despejo é exp_file.dmp.

   ```
   exp cust_dba@ORCL FILE=exp_file.dmp TABLES=(tab1,tab2,tab3) LOG=exp_file.log
   ```

   A exportação cria um arquivo de despejo binário que contém o esquema e os dados das tabelas especificadas.
2. Importe o esquema e os dados em um banco de dados de destino usando o comando imp.

O comando a seguir importa as tabelas `tab1`, `tab2` e `tab3` do arquivo de despejo `exp_file.dmp`.

```
imp cust_dba@targetdb FROMUSER=cust_schema TOUSER=cust_schema \
TABLES=(tab1,tab2,tab3) FILE=exp_file.dmp LOG=imp_file.log
```

A exportação e a importação têm outras variações que podem ser mais adequadas às suas necessidades. Consulte a documentação do Oracle Database para obter detalhes completos.

Importar usando o Oracle SQL*Loader

Você pode usar o Oracle SQL*Loader para bancos de dados grandes que contêm um número limitado de objetos. Como o processo de exportação de um banco de dados de origem e carregamento em um banco de dados de destino é específico para o esquema, o exemplo a seguir cria os objetos de esquema de amostra, exporta de uma origem, depois carrega os dados em um banco de dados de destino.

A maneira mais fácil de instalar o Oracle SQL*Loader é instalar o Oracle Instant Client. Para baixar o software, acesse https://www.oracle.com/database/technologies/instant-client.html. Para obter a documentação, consulte Instant Client for SQL*Loader, exportação e importação no manual Utilitários do banco de dados Oracle.

Para importar dados usando o Oracle SQL*Loader

1. Crie uma tabela de origem de exemplo usando a seguinte instrução SQL.

```
CREATE TABLE customer_0 TABLESPACE users
   AS (SELECT ROWNUM id, o.*
       FROM   ALL_OBJECTS o, ALL_OBJECTS x
       WHERE  ROWNUM <= 1000000);
```

2. Na instância de banco de dados de destino do RDS para Oracle, crie uma tabela de destino para carregar os dados. A cláusula `WHERE 1=2` garante que você copie a estrutura do `ALL_OBJECTS`, mas não copie nenhuma linha.

```
CREATE TABLE customer_1 TABLESPACE users
  AS (SELECT 0 AS ID, OWNER, OBJECT_NAME, CREATED
      FROM   ALL_OBJECTS
```

```
      WHERE  1=2);
```

3. Exporte os dados do banco de dados de origem para um arquivo de texto. O exemplo a seguir usa SQL*Plus. Para os seus dados, você provavelmente precisará gerar um script que faça a exportação para todos os objetos no banco de dados.

```
ALTER SESSION SET NLS_DATE_FORMAT = 'YYYY/MM/DD HH24:MI:SS'

SET LINESIZE 800 HEADING OFF FEEDBACK OFF ARRAY 5000 PAGESIZE 0
SPOOL customer_0.out
SET MARKUP HTML PREFORMAT ON
SET COLSEP ','

SELECT id, owner, object_name, created
FROM   customer_0;

SPOOL OFF
```

4. Crie um arquivo de controle para descrever os dados. Talvez seja necessário escrever um script para realizar essa etapa.

```
cat << EOF > sqlldr_1.ctl
load data
infile customer_0.out
into table customer_1
APPEND
fields terminated by "," optionally enclosed by '"'
(
  id           POSITION(01:10)    INTEGER EXTERNAL,
  owner        POSITION(12:41)    CHAR,
  object_name  POSITION(43:72)    CHAR,
  created      POSITION(74:92)    date "YYYY/MM/DD HH24:MI:SS"
)
```

   Se necessário, copie os arquivos gerados pelo código anterior para uma área de teste, como uma instância do Amazon EC2.

5. Importe os dados usando o SQL*Loader com o nome de usuário e a senha apropriados para o banco de dados de destino.

```
sqlldr cust_dba@targetdb CONTROL=sqlldr_1.ctl BINDSIZE=10485760 READSIZE=10485760
 ROWS=1000
```

Migrar com visualizações materializadas do Oracle

Para migrar conjuntos de dados grandes de forma eficiente, use a replicação de visualizações materializadas do Oracle. Com a replicação, você pode manter as tabelas de destino sincronizadas com as tabelas de origem. Assim, você pode mudar para o Amazon RDS posteriormente, se necessário.

Antes de poder migrar usando visualizações materializadas, cumpra os seguintes requisitos:

- Configure o acesso do banco de dados de destino ao banco de dados de origem. No exemplo a seguir, as regras de acesso foram habilitadas no banco de dados de origem para permitir que o banco de dados de destino do RDS para Oracle se conecte à origem via SQL*Net.
- Crie um link de banco de dados entre a instância de banco de dados do RDS para Oracle e o banco de dados de origem.

Como migrar dados usando visualizações materializadas

1. Crie uma conta de usuário na instância de origem e na instância de destino do RDS para Oracle que podem se autenticar com a mesma senha. O exemplo a seguir cria um usuário denominado `dblink_user`.

```
CREATE USER dblink_user IDENTIFIED BY my-password
  DEFAULT TABLESPACE users
  TEMPORARY TABLESPACE temp;

GRANT CREATE SESSION TO dblink_user;

GRANT SELECT ANY TABLE TO dblink_user;

GRANT SELECT ANY DICTIONARY TO dblink_user;
```

> **Note**
>
> Especifique uma senha diferente do prompt mostrado aqui como prática recomendada de segurança.

2. Crie um link de banco de dados da instância de destino do RDS para Oracle com a instância de origem usando o usuário recém-criado.

```
CREATE DATABASE LINK remote_site
  CONNECT TO dblink_user IDENTIFIED BY my-password
  USING '(description=(address=(protocol=tcp) (host=my-host)
    (port=my-listener-port)) (connect_data=(sid=my-source-db-sid)))';
```

> **Note**
>
> Especifique uma senha diferente do prompt mostrado aqui como prática recomendada de segurança.

3. Teste o link:

```
SELECT * FROM V$INSTANCE@remote_site;
```

4. Crie uma tabela de exemplo com uma chave primária e um log de visualização materializada na instância de origem.

```
CREATE TABLE customer_0 TABLESPACE users
  AS (SELECT ROWNUM id, o.*
      FROM   ALL_OBJECTS o, ALL_OBJECTS x
      WHERE  ROWNUM <= 1000000);

ALTER TABLE customer_0 ADD CONSTRAINT pk_customer_0 PRIMARY KEY (id) USING INDEX;

CREATE MATERIALIZED VIEW LOG ON customer_0;
```

5. Na instância de banco de dados de destino do RDS para Oracle, crie uma visualização materializada.

```
CREATE MATERIALIZED VIEW customer_0
  BUILD IMMEDIATE REFRESH FAST
  AS (SELECT *
      FROM   cust_dba.customer_0@remote_site);
```

6. Na instância de banco de dados de destino do RDS para Oracle, atualize a visualização materializada.

```
EXEC DBMS_MVIEW.REFRESH('CUSTOMER_0', 'f');
```

7. Elimine a visualização materializada e inclua a cláusula `PRESERVE TABLE` para reter a tabela de contêineres de visualização materializada e seus conteúdos.

```
DROP MATERIALIZED VIEW customer_0 PRESERVE TABLE;
```

   A tabela retida tem o mesmo nome que a visualização materializada que foi eliminada.

Trabalhar com réplicas de leitura do Amazon RDS para Oracle

Para configurar a replicação entre instâncias de banco de dados Oracle, você pode criar bancos de dados de réplica. Para obter uma visão geral das réplicas de leitura do Amazon RDS, consulte Visão geral das réplicas de leitura do Amazon RDS. Para ver um resumo das diferenças entre réplicas do Oracle e outros mecanismos de banco de dados, consulte Diferenças entre réplicas de leitura para mecanismos de banco de dados.

Tópicos

- Visão geral das réplicas do RDS para Oracle
- Requisitos e considerações sobre réplicas do RDS para Oracle
- Preparar-se para criar uma réplica do Oracle
- Criar uma réplica do RDS para Oracle no modo montado
- Modificar o modo da réplica do RDS para Oracle
- Como trabalhar com backups de réplicas do RDS para Oracle
- Executar uma alternância do Oracle Data Guard
- Solução de problemas de réplicas do RDS para Oracle

Visão geral das réplicas do RDS para Oracle

Um banco de dados de réplica do Oracle é uma cópia física do seu banco de dados primário. Uma réplica do Oracle no modo somente leitura é chamada de réplica de leitura. Uma réplica do Oracle no modo montado é chamada de réplica montada. O Oracle Database não permite gravações em uma réplica, mas é possível promover uma réplica para torná-la gravável. A réplica de leitura promovida terá os dados replicados para o ponto quando a solicitação foi feita para promovê-la.

O vídeo a seguir apresenta uma visão geral útil da recuperação de desastres do RDS para Oracle.

Para obter mais informações, consulte os artigos do blog Managed disaster recovery with Amazon RDS para Oracle cross-Region automated backups - Part 1 (Recuperação de desastres gerenciada com backups automatizados do Amazon RDS para Oracle entre regiões, Parte 1) e Managed disaster recovery with Amazon RDS para Oracle cross-Region automated backups - Part 2 (Recuperação de desastres gerenciada com backups automatizados do Amazon RDS para Oracle entre regiões, Parte 2).

Tópicos

- Réplicas somente leitura e montadas
- Leia réplicas de CDBs
- Retenção de logs redo arquivados
- Interrupções durante a replicação do Oracle

Réplicas somente leitura e montadas

Ao criar ou modificar uma réplica do Oracle, você pode colocá-la em um dos seguintes modos:

Somente leitura

Esse é o padrão. O Active Data Guard transmite e aplica alterações do banco de dados de origem a todos os bancos de dados da réplica de leitura.

Você pode criar até cinco réplicas de leitura de uma instância de banco de dados de origem. Para obter informações gerais sobre réplicas de leitura que se aplicam a todos os mecanismos de banco de dados, consulte Trabalhar com réplicas de leitura de instância de banco de dados. Para obter informações sobre o Oracle Data Guard, consulte Oracle Data Guard Concepts and Administration (Conceitos e administração do Oracle Data Guard) na documentação da Oracle.

Montada

Nesse caso, a replicação usa o Oracle Data Guard, mas o banco de dados de réplica não aceita conexões de usuário. O principal uso para réplicas montadas é a recuperação de desastres entre regiões.

Uma réplica montada não pode atender a uma workload somente leitura. A réplica montada exclui arquivos de log redo arquivados depois de aplicá-los, independentemente da política de retenção de logs arquivados.

É possível criar uma combinação de réplicas de banco de dados montadas e somente leitura para a mesma instância de banco de dados de origem. Você pode alterar uma réplica somente leitura para o modo montado ou alterar uma réplica montada para o modo somente leitura. Em ambos os casos, o banco de dados Oracle preserva a configuração de retenção de logs arquivados.

Leia réplicas de CDBs

O RDS para Oracle é compatível com réplicas de leitura do Data Guard para CDBs do Oracle Database 19c e 21c. Você pode criar, gerenciar e promover réplicas de leitura em um CDB, assim

como em um que não seja CDB. Também há suporte para réplicas montadas. Você obtém os seguintes benefícios:

- Recuperação de desastres gerenciada, alta disponibilidade e acesso somente leitura às suas réplicas
- A capacidade de criar réplicas de leitura em uma Região da AWS diferente.
- Integração com as APIs de réplica de leitura existentes do RDS: CreateDBInstanceReadReplica, PromoteReadReplica e SwitchoverReadReplica

Para usar esse atributo, você precisa de uma licença do Active Data Guard e de uma licença do Oracle Database Enterprise Edition para as instâncias de banco de dados primárias e de réplica. Não há custos adicionais relacionados ao uso da arquitetura CDB. Você paga somente por suas instâncias de banco de dados.

Para obter mais informações sobre as configurações de inquilino único e multilocatário da arquitetura CDB, consulte. Visão geral dos CDBs RDS para Oracle

Retenção de logs redo arquivados

Se uma instância de banco de dados primária não tiver réplicas de leitura entre regiões, o Amazon RDS para Oracle manterá um mínimo de duas horas de redo logs arquivados na instância de banco de dados de origem. Isso é válido independentemente da configuração para `archivelog retention hours` em `rdsadmin.rdsadmin_util.set_configuration`.

O RDS limpa logs da instância de banco de dados de origem após duas horas ou após o tempo definido na configuração de horas de retenção do log de arquivamento, o que for maior. O RDS limpará logs da réplica de leitura depois que após o tempo da configuração de horas de retenção de logs arquivados somente se esses logs tiverem sido aplicados com êxito ao banco de dados.

Em alguns casos, uma instância de banco de dados primária pode ter uma ou mais réplicas de leitura entre regiões. Nesse caso, o Amazon RDS for Oracle mantém os logs de transações na instância do banco de dados de origem até que eles tenham sido transmitidos e aplicados a todas as réplicas de leitura entre regiões. Para obter informações sobre `rdsadmin.rdsadmin_util.set_configuration`, consulte o tópico sobre Retenção de logs redo arquivados.

Interrupções durante a replicação do Oracle

Ao criar uma réplica de leitura, o Amazon RDS faz um snapshot do banco de dados da sua instância de banco de dados de origem e inicia a replicação. A instância de banco de dados de origem sofre uma suspensão de E/S muito breve quando a operação de snapshot do banco de dados se inicia. A suspensão de E/S geralmente dura um segundo. Você pode evitar a suspensão de E/S se a instância de banco de dados de origem for uma implantação multi-AZ. Nesse caso, o snapshot é retirado da instância de banco de dados secundária.

O snapshot de banco de dados torna-se a réplica do Oracle. O Amazon RDS define as permissões e os parâmetros necessários para o banco de dados de origem e para a réplica sem interrupção do serviço. Da mesma forma, se você excluir uma réplica, nenhuma interrupção ocorrerá.

Requisitos e considerações sobre réplicas do RDS para Oracle

Antes de criar uma réplica do Oracle, familiarize-se com os seguintes requisitos e considerações.

Tópicos

- Requisitos de versão e licenciamento para réplicas do RDS para Oracle
- Limitações do grupo de opções de réplicas do RDS para Oracle
- Considerações sobre backup e restauração de réplicas do RDS para Oracle
- Requisitos e limitações do Oracle Data Guard para as réplicas do RDS para Oracle
- Considerações diversas sobre réplicas do RDS para Oracle

Requisitos de versão e licenciamento para réplicas do RDS para Oracle

Antes de criar uma réplica do RDS para Oracle, considere o seguinte:

- Se a réplica estiver no modo somente leitura, verifique se você tem uma licença do Active Data Guard. Se você colocar a réplica no modo montado, não precisará de uma licença do Active Data Guard. Somente o mecanismo de banco de dados Oracle oferece suporte a réplicas montadas.
- As réplicas do Oracle são compatíveis apenas com o Oracle Enterprise Edition (EE).
- As réplicas não CDB do Oracle são compatíveis somente com instâncias de banco de dados criadas por meio de instâncias não CDB que executam o Oracle Database 19c.
- Réplicas do Oracle estão disponíveis somente para instâncias de banco de dados em execução em classes de instância de banco de dados com duas ou mais vCPUs. Uma instância de banco de dados de origem não pode usar a classe de instância db.t3.small.

- A versão do mecanismo de banco de dados Oracle da instância de banco de dados de origem e todas as réplicas devem ser iguais. O Amazon RDS atualiza as réplicas imediatamente após a atualização da instância de banco de dados de origem, desconsiderando a janela de manutenção de uma réplica. Para atualizações de versões principais de réplicas entre regiões, o Amazon RDS faz automaticamente o seguinte:
  - Gera um grupo de opções para a versão de destino.
  - Copia todas as opções e as configurações de opção do grupo de opções original para o novo grupo de opções.
  - Associa a réplica entre regiões atualizada ao novo grupo de opções.

  Para obter mais informações sobre como atualizar a versão do mecanismo de banco de dados, consulte Fazer upgrade do mecanismo de banco de dados RDS para Oracle.

Limitações do grupo de opções de réplicas do RDS para Oracle

Ao trabalhar com grupos de opções para sua réplica do RDS para Oracle, considere o seguinte:

- Não é possível usar um grupo de opções de réplica diferente do grupo de opções da instância de banco de dados de origem quando a origem e a réplica estão na mesma região da AWS.

  Modificações no grupo de opções de origem ou na associação do grupo de opções de origem são propagadas para as réplicas do Oracle. Essas alterações serão aplicadas às réplicas logo depois de serem aplicadas à instância de banco de dados de origem, independentemente da janela de manutenção da réplica. Para ter mais informações sobre grupos de opções, consulte Trabalhar com grupos de opções.
- Não é possível remover uma réplica entre regiões do RDS para Oracle do respectivo grupo de opções dedicado, que é criado automaticamente para a réplica.
- Não é possível adicionar o grupo de opções dedicado a uma réplica entre regiões do RDS para Oracle a uma instância de banco de dados diferente.
- Não é possível adicionar nem remover opções não replicadas de um grupo de opções dedicado para uma réplica entre regiões do RDS para Oracle, com exceção das seguintes opções:
  - `NATIVE_NETWORK_ENCRYPTION`
  - `OEM`
  - `OEM_AGENT`
  - `SSL`

Para adicionar outras opções a uma réplica entre regiões do RDS para Oracle, adicione-as ao grupo de opções da instância de banco de dados de origem. A opção também é instalada em todas as réplicas da instância do banco de dados de origem. Para opções licenciadas, verifique se existem licenças suficientes para as réplicas.

Quando você promove uma réplica entre regiões do RDS para Oracle, a réplica promovida se comporta da mesma maneira que outras instâncias de banco de dados do Oracle, inclusive com relação ao gerenciamento das opções. Você pode promover uma réplica, explícita ou implicitamente, excluindo a instância de banco de dados de origem.

Para obter mais informações sobre grupos de opções, consulte Trabalhar com grupos de opções.

- Não é possível adicionar a opção `EFS_INTEGRATION` às réplicas entre regiões do RDS para Oracle.

Considerações sobre backup e restauração de réplicas do RDS para Oracle

Antes de criar uma réplica do RDS para Oracle, considere o seguinte:

- Para criar snapshots de réplicas do RDS para Oracle ou ativar backups automáticos, defina o período de retenção de backup manualmente. Por padrão, os backups automáticos não estão ativados.
- Ao restaurar um backup de réplica, você restaura para a hora do banco de dados, não para a hora de criação do backup. A hora do banco de dados se refere ao horário de transação mais recente aplicado dos dados no backup. A diferença é significativa porque uma réplica pode apresentar atraso de minutos ou horas em relação à primária.

  Para descobrir a diferença, use o comando `describe-db-snapshots`. Compare `snapshotDatabaseTime`, que é a hora do banco de dados do backup da réplica, e o campo `OriginalSnapshotCreateTime`, que é a última transação aplicada no banco de dados primária.

Requisitos e limitações do Oracle Data Guard para as réplicas do RDS para Oracle

Antes de criar uma réplica do RDS para Oracle, observe os seguintes requisitos e limitações:

- Se a instância de banco de dados primária usa a configuração de locatário único da arquitetura multilocatário, pense no seguinte:

  - Você deve usar o Oracle Database 19c ou posterior com a Enterprise Edition.
  - Sua instância primária de CDB deve estar no ciclo de vida `ACTIVE`.
  - Você não pode converter uma instância primária que não seja CDB em uma instância de CDB e converter as respectivas réplicas na mesma operação. Em vez disso, exclua as réplicas que não são do CDB, converta a instância de banco de dados primária em um CDB e, depois, crie réplicas.
- Um acionador de login em uma instância primária de banco de dados deve permitir o acesso ao usuário `RDS_DATAGUARD` e a qualquer usuário cujo valor `AUTHENTICATED_IDENTITY` seja `RDS_DATAGUARD` ou `rdsdb`. Além disso, o gatilho não deve definir o esquema atual para o usuário `RDS_DATAGUARD`.
- Para evitar o bloqueio de conexões do processo do agente do Data Guard, não ative sessões restritas. Para obter mais informações sobre sessões restritas, consulte Habilitar e desabilitar sessões restritas.

Considerações diversas sobre réplicas do RDS para Oracle

Antes de criar uma réplica do RDS para Oracle, considere o seguinte:

- Se a instância de banco de dados for a origem de uma ou mais réplicas entre regiões, o banco de dados de origem reterá os arquivos de log redo arquivados até que sejam aplicados em todas as réplicas entre regiões. Os logs redo arquivados podem resultar em aumento no consumo de armazenamento.
- Para evitar interromper a automação do RDS, os gatilhos do sistema devem permitir que usuários específicos façam logon no banco de dados primário e de réplica. Os gatilhos do sistema incluem disparadores de função DDL, logon e banco de dados. Recomendamos que você adicione código aos seus gatilhos para excluir os usuários listados no seguinte código de exemplo:

```
-- Determine who the user is
SELECT SYS_CONTEXT('USERENV','AUTHENTICATED_IDENTITY') INTO CURRENT_USER FROM DUAL;
-- The following users should always be able to login to either the Primary or
 Replica
IF CURRENT_USER IN ('master_user', 'SYS', 'SYSTEM', 'RDS_DATAGUARD', 'rdsdb') THEN
RETURN;
END IF;
```

- O rastreamento de alterações de bloco é compatível com réplicas somente leitura, mas não com réplicas montadas. Você pode alterar uma réplica montada para uma réplica somente leitura e

habilitar o rastreamento de alterações de bloco. Para ter mais informações, consulte Habilitar e desabilitar o controle de alterações de bloco.

- Não é possível criar uma réplica de leitura do Oracle quando o banco de dados de origem gerencia as credenciais do usuário principal com o Secrets Manager.

Preparar-se para criar uma réplica do Oracle

Antes de começar a usar a réplica, execute as tarefas a seguir.

Tópicos

- Habilitar backups automáticos
- Habilitar o modo de registro em log forçado
- Alterar a configuração do registro em log
- Definir o parâmetro MAX_STRING_SIZE
- Planejar recursos de computação e de armazenamento

Habilitar backups automáticos

Antes que uma instância de banco de dados possa servir como uma instância de banco de dados de origem, habilite backups automáticos na instância de banco de dados de origem. Para saber como executar esse procedimento, consulte Ativar backups automáticos.

Habilitar o modo de registro em log forçado

Recomendamos que você habilite o modo de registro em log forçado. No modo de registro em log forçado, o banco de dados Oracle grava registros de redo mesmo quando `NOLOGGING` é usado com instruções da linguagem de definição de dados (DDL).

Como habilitar o modo de registro em log forçado

1. Faça login no banco de dados Oracle usando uma ferramenta cliente, como o SQL Developer.
2. Habilite o modo de registro em log forçado executando o procedimento a seguir.

```
exec rdsadmin.rdsadmin_util.force_logging(p_enable => true);
```

Para obter mais informações sobre esse procedimento, consulte Definição do registro em log forçado.

Alterar a configuração do registro em log

Para n redo logs on-line de tamanho m, o RDS cria automaticamente n +1 logs em espera de tamanho m na instância de banco de dados primária e em todas as réplicas. Sempre que você altera a configuração de registro em log na primária, as alterações se propagam automaticamente para as réplicas.

Se você alterar a configuração de registro em log, pense nestas diretrizes:

- Recomendamos concluir as alterações antes de transformar uma instância de banco de dados na fonte das réplicas. O RDS para Oracle também é compatível com a atualização da instância depois que ela se torna uma fonte.
- Antes de alterar a configuração de registro em log na instância de banco de dados primária, confira se cada réplica tem armazenamento suficiente para acomodar a nova configuração.

É possível modificar a configuração de registro em log de uma instância de banco de dados usando os procedimentos do Amazon RDS `rdsadmin.rdsadmin_util.add_logfile` e `rdsadmin.rdsadmin_util.drop_logfile`. Para obter mais informações, consulte Adição de logs redo online e Descarte de logs redo online.

Definir o parâmetro MAX_STRING_SIZE

Antes de criar uma réplica do Oracle, verifique se a configuração do parâmetro `MAX_STRING_SIZE` é a mesma na instância de banco de dados de origem e na réplica. Isso pode ser feito associando-os ao mesmo grupo de parâmetros. Se tiver diferentes grupos de parâmetros para a origem e a réplica, você poderá definir `MAX_STRING_SIZE` com o mesmo valor. Para obter mais informações sobre como configurar esse parâmetro, consulte Ativar tipos de dados estendidos para uma nova instância de banco de dados.

Planejar recursos de computação e de armazenamento

Verifique se a instância de banco de dados de origem e suas réplicas estão dimensionadas corretamente, em termos de computação e armazenamento, para atender às suas cargas operacionais. Se uma réplica atingir a capacidade de recursos de computação, rede ou armazenamento, a réplica deixará de receber ou aplicar alterações de sua origem. O Amazon RDS

for Oracle não intervém para atenuar atrasos altos de replicação entre uma instância de banco de dados de origem e suas réplicas. É possível modificar os recursos de armazenamento e de CPU de uma réplica independentemente de sua origem e de outras réplicas.

Criar uma réplica do RDS para Oracle no modo montado

Por padrão, as réplicas do Oracle são somente leitura. Para criar uma réplica no modo montado, use o console, a AWS CLI ou a API do RDS.

Console

Como criar uma réplica montada de uma instância de banco de dados Oracle de origem

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Databases (Bancos de dados).
3. Escolha a instância de banco de dados Oracle que você deseja usar como origem para uma réplica montada.
4. Em Actions (Ações), escolha Create replica (Criar réplica).
5. Em Replica mode (Modo de réplica), escolha Mounted (Montado).
6. Escolha as configurações que deseja usar. Em DB instance identifier (Identificador da instância de banco de dados), insira um nome para a réplica de leitura. Ajuste outras configurações, conforme necessário.
7. Em Regions (Regiões), escolha a região onde a réplica montada será iniciada.
8. Escolha o tamanho da instância e o tipo de armazenamento. Recomendamos que você use a classe de instância de banco de dados e o tipo de armazenamento iguais aos da instância de banco de dados de origem na réplica de leitura.
9. Em Multi-AZ deployment (Implantação Multi-AZ), escolha Create a standby instance (Criar uma instância de espera) para criar uma versão de espera de sua réplica em outra zona de disponibilidade a fim de oferecer suporte de failover para a réplica montada. Você pode criar a réplica montada como uma instância de banco de dados Multi-AZ independentemente de o banco de dados de origem ser ou não uma instância de banco de dados Multi-AZ.
10. Escolha as outras configurações que deseja usar.
11. Escolha Create replica (Criar réplica).

Na página Databases (Bancos de dados), a réplica montada tem a função Réplica.

AWS CLI

Para criar uma réplica do Oracle no modo montado, defina `--replica-mode` como `mounted` no comando create-db-instance-read-replica da AWS CLI.

Example

Para Linux, macOS ou Unix:

```
aws rds create-db-instance-read-replica \
    --db-instance-identifier myreadreplica \
    --source-db-instance-identifier mydbinstance \
    --replica-mode mounted
```

Para Windows:

```
aws rds create-db-instance-read-replica ^
    --db-instance-identifier myreadreplica ^
    --source-db-instance-identifier mydbinstance ^
    --replica-mode mounted
```

Para alterar uma réplica somente leitura para um estado montado, defina `--replica-mode` como `mounted` no comando modify-db-instance da AWS CLI. Para colocar uma réplica montada no modo somente leitura, defina `--replica-mode` como `open-read-only`.

API do RDS

Para criar uma réplica do Oracle no modo montado, especifique `ReplicaMode=mounted` na operação de API CreateDBInstanceReadReplica do RDS.

Modificar o modo da réplica do RDS para Oracle

Para alterar o modo de uma réplica existente, use o console, a AWS CLI ou a API do RDS. Quando você altera para o modo montado, a réplica desconecta todas as conexões ativas. Quando você muda para o modo somente leitura, o Amazon RDS inicializa o Active Data Guard.

A operação de alteração pode levar alguns minutos. Durante a operação, o status da instância de banco de dados muda para modifying (modificando). Para obter mais informações sobre alterações de status, consulte Visualizar o status de uma instância de banco de dados Amazon RDS em um .

Console

Como alterar o modo de uma réplica do Oracle de montada para somente leitura

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Databases (Bancos de dados).
3. Escolha o banco de dados de réplica montada.
4. Selecione Modify.
5. Em Replica mode (Modo de réplica), escolha Read-only (Somente leitura).
6. Escolha as outras configurações que deseja alterar.
7. Escolha Continue.
8. Em Scheduling of modifications (Programação de modificações), selecione Apply immediately (Aplicar imediatamente).
9. Selecione Modify DB instance (Modificar instância de banco de dados).

AWS CLI

Para alterar uma réplica de leitura para o modo montado, defina `--replica-mode` como `mounted` no comando modify-db-instance da AWS CLI. Para alterar uma réplica montada para o modo somente leitura, defina `--replica-mode` como `open-read-only`.

Example

Para Linux, macOS ou Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier myreadreplica \
    --replica-mode mode
```

Para Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier myreadreplica ^
    --replica-mode mode
```

API do RDS

Para alterar uma réplica somente leitura para o modo montado, defina `ReplicaMode=mounted` em ModifyDBInstance. Para alterar uma réplica montada para o modo somente leitura, defina `ReplicaMode=read-only`.

Como trabalhar com backups de réplicas do RDS para Oracle

Você pode criar e restaurar backups de uma réplica do RDS para Oracle. Tanto backups automáticos como snapshots manuais são compatíveis. Para obter mais informações, consulte Backup, restauração e exportação de dados. As seções a seguir descrevem as principais diferenças entre o gerenciamento de backups de uma primária e de uma réplica do RDS para Oracle.

Ativar backups de réplicas do RDS para Oracle

Uma réplica do Oracle não tem o recurso de backups automatizados ativado por padrão. Ative o recurso de backups automáticos configurando o período de retenção de backup para um valor positivo diferente de zero.

Console

Para habilitar imediatamente backups automáticos

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Bancos de dados e selecione a instância de banco de dados ou o cluster de banco de dados multi-AZ que você deseja modificar.
3. Escolha Modificar.
4. Em Backup Retention Period (Período de retenção de backup), escolha um valor positivo diferente de zero; por exemplo, três dias.
5. Escolha Continue.
6. Escolha Apply immediately (Aplicar imediatamente).
7. Escolha Modificar instância de banco de dados ou Modificar cluster para salvar as alterações e ativar os backups automatizados.

AWS CLI

Para habilitar backups automatizados, use o comando AWS CLI modify-db-instance ou modify-db-cluster.

Inclua os seguintes parâmetros:

- `--db-instance-identifier` (ou `--db-cluster-identifier` para um cluster de banco de dados multi-AZ)
- `--backup-retention-period`
- `--apply-immediately` ou `--no-apply-immediately`

No exemplo a seguir, habilitamos backups automatizados definindo o período de retenção de backup como três dias. As alterações são aplicadas imediatamente.

Example

Para Linux, macOS ou Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier my_db_instance  \
    --backup-retention-period 3 \
    --apply-immediately
```

Para Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier my_db_instance  ^
    --backup-retention-period 3 ^
    --apply-immediately
```

API do RDS

Para habilitar backups automatizados, use a operação ModifyDBInstance ou ModifyDBCluster da API do RDS com os seguintes parâmetros obrigatórios:

- `DBInstanceIdentifier` ou `DBClusterIdentifier`
- `BackupRetentionPeriod`

Restaurar um backup de réplica do RDS para Oracle

Você pode restaurar um backup de réplica do Oracle da mesma forma que pode restaurar um backup da instância primária. Para obter mais informações, consulte:

- Restaurar uma instância de banco de dados
- Restaurar uma instância de banco de dados para um momento especificado no Amazon RDS

A principal consideração ao restaurar um backup de réplica é determinar o momento para o qual está restaurando. A hora do banco de dados se refere ao horário de transação mais recente aplicado dos dados no backup. Ao restaurar um backup de réplica, você restaura para a hora do banco de dados, não para a hora em que o backup foi concluído. A diferença é significativa porque uma réplica do RDS para Oracle pode apresentar atraso de minutos ou horas em relação à primária. Portanto, a hora do banco de dados de um backup de réplica e, consequentemente, o momento para o qual vai restaurar podem ser muito antes do horário de criação do backup.

Para encontrar a diferença entre a hora do banco de dados e o horário de criação, use o comando `describe-db-snapshots`. Compare `SnapshotDatabaseTime`, que é a hora do banco de dados do backup da réplica, e o campo `OriginalSnapshotCreateTime`, que é a última transação aplicada no banco de dados primária. O exemplo a seguir mostra a diferença entre as duas horas:

```
aws rds describe-db-snapshots \
    --db-instance-identifier my-oracle-replica
    --db-snapshot-identifier my-replica-snapshot

{
    "DBSnapshots": [
        {
            "DBSnapshotIdentifier": "my-replica-snapshot",
            "DBInstanceIdentifier": "my-oracle-replica",
            "SnapshotDatabaseTime": "2022-07-26T17:49:44Z",
            ...
            "OriginalSnapshotCreateTime": "2021-07-26T19:49:44Z"
        }
    ]
}
```

Executar uma alternância do Oracle Data Guard

Uma alternância é uma inversão de funções entre um banco de dados primário e um banco de dados em espera. Durante uma alternância, o banco de dados primário original faz a transição para uma função em espera, enquanto o banco de dados auxiliar original faz a transição para a função principal.

Em um ambiente Oracle Data Guard, um banco de dados primário oferece suporte a um ou mais bancos de dados em espera. Você pode executar uma transição de função gerenciada e baseada em alternância de um banco de dados primário para um banco de dados em espera. Uma alternância é uma inversão de funções entre um banco de dados primário e um banco de dados em espera. Durante uma alternância, o banco de dados primário original faz a transição para uma função em espera, enquanto o banco de dados auxiliar original faz a transição para a função principal.

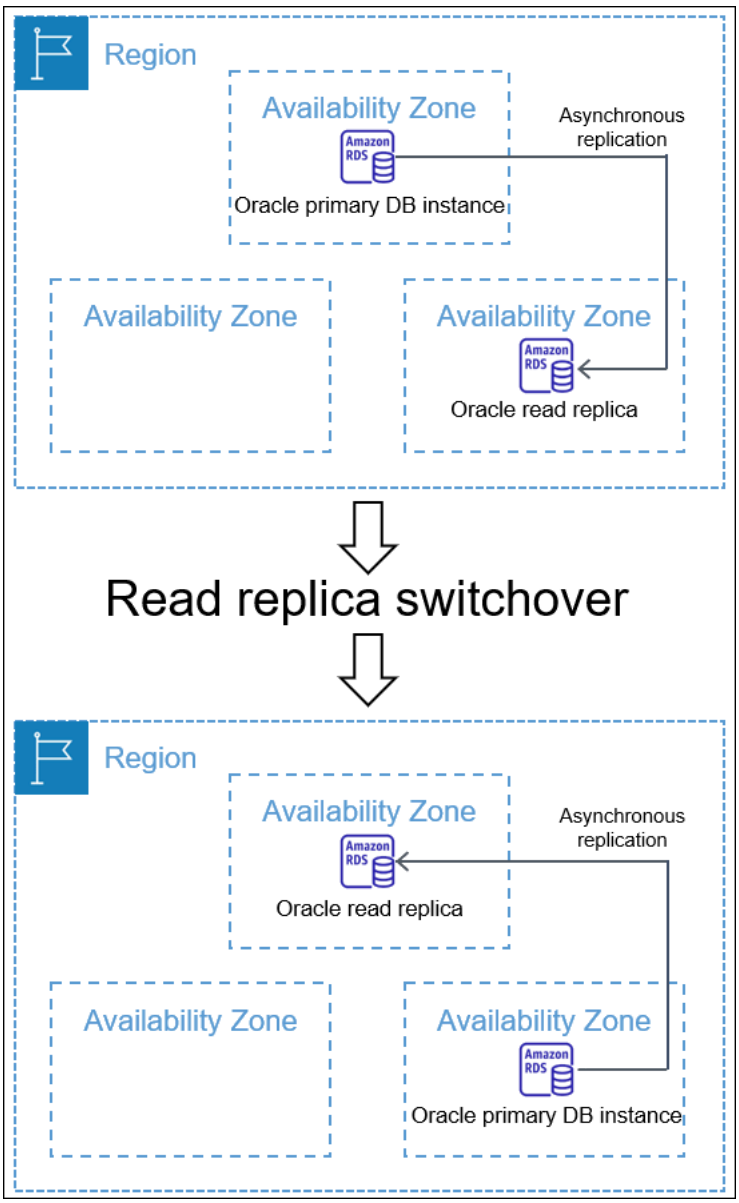
Tópicos

- Visão geral da alternância do Oracle Data Guard
- Requisitos para a transição do Oracle Data Guard
- Iniciar a alternância do Oracle Data Guard
- Monitorar a alternância do Oracle Data Guard

Visão geral da alternância do Oracle Data Guard

O Amazon RDS oferece suporte a uma transição de função totalmente gerenciada e baseada em alternância para réplicas do Oracle Database. Você pode iniciar uma alternância somente para um banco de dados em espera montado ou aberto somente para leitura.

As réplicas podem residir em Regiões da AWS separadas ou em diferentes zonas de disponibilidade (AZs) de uma única região. Todas as Regiões da AWS são aceitas.



A transição é diferente da promoção de réplicas de leitura. Em uma transição, as instâncias de banco de dados de origem e réplica mudam de função. Em uma promoção, uma réplica de leitura se torna uma instância de banco de dados de origem, mas a instância de banco de dados de origem não se torna uma réplica. Para ter mais informações, consulte Promoção de uma réplica de leitura a uma instância de banco de dados autônoma.

Tópicos

- Benefícios da alternância do Oracle Data Guard
- Versões compatíveis do Oracle Database
- Custo da alternância do Oracle Data Guard

- Como funciona a alternância do Oracle Data Guard

Benefícios da alternância do Oracle Data Guard

Assim como no caso das réplicas de leitura do RDS para Oracle, uma alternância gerenciada depende do Oracle Data Guard. A operação foi criada para não ter nenhuma perda de dados. O Amazon RDS automatiza os seguintes aspectos da alternância:

- Inverte as funções do banco de dados primária e do banco de dados em espera especificado, colocando o novo banco de dados em espera no mesmo estado (montado ou somente para leitura) do em espera original
- Garante a consistência de dados
- Mantém sua configuração de replicação após a transição
- Oferece suporte a reversões repetidas, permitindo que seu novo banco de dados em espera retorne à sua função primária original

Versões compatíveis do Oracle Database

O Oracle Database 19c e versões posteriores comportam a transição do Oracle Data Guard.

Custo da alternância do Oracle Data Guard

O recurso de alternância do Oracle Data Guard não está sujeito a custos adicionais. O Oracle Database Enterprise Edition inclui suporte a bancos de dados em espera no modo montado. Para abrir bancos de dados em espera no modo somente leitura, você precisa da opção Oracle Active Data Guard.

Como funciona a alternância do Oracle Data Guard

A alternância do Oracle Data Guard é uma operação totalmente gerenciada. Você pode iniciar a alternância para um banco de dados em espera chamando o comando da CLI `switchover-read-replica`. Em seguida, o Amazon RDS modifica as funções primárias e em espera em sua configuração de replicação.

A em espera original e a primário original são as funções que existem antes da alternância. A nova em espera e a novo primário são as funções que existem após a alternância. Uma réplica do espectador é um banco de dados de réplica que serve como banco de dados em espera no ambiente Oracle Data Guard, mas não está alternando de funções.

Tópicos

- Etapas da alternância do Oracle Data Guard
- Após a alternância do Oracle Data Guard

Etapas da alternância do Oracle Data Guard

Para realizar a alternância, o Amazon RDS deve seguir as seguintes etapas:

1. Bloqueie novas transações no banco de dados primário original. Durante a alternância, o Amazon RDS interrompe a replicação de todos os bancos de dados em sua configuração do Oracle Data Guard. Durante a alternância, o banco de dados primário original não pode processar solicitações de gravação.
2. Envie as transações não aplicadas para o banco de dados em espera original e aplique-as.
3. Reinicie o novo banco de dados em espera no modo somente leitura ou montado. O modo depende do estado aberto do banco de dados em espera original antes da alternância.
4. Abra o novo banco de dados primário no modo de leitura/gravação.

Após a alternância do Oracle Data Guard

O Amazon RDS alterna as funções do banco de dados primário e do banco de dados em espera. Você é responsável por reconectar sua aplicação e realizar qualquer outra configuração desejada.

Tópicos

- Critérios de sucesso
- Conexão com o novo banco de dados primário
- Configuração do novo banco de dados primário

Critérios de sucesso

A alternância do Oracle Data Guard é bem-sucedida quando o banco de dados em espera original faz o seguinte:

- Transições para sua função como novo banco de dados primário
- Conclui sua reconfiguração

Para limitar o tempo de inatividade, seu novo banco de dados primário se torna ativo o mais rápido possível. Como o Amazon RDS configura réplicas de espectador de forma assíncrona, essas réplicas podem se tornar ativas após o banco de dados primário original.

Conexão com o novo banco de dados primário

O Amazon RDS não propagará suas conexões de banco de dados atuais para o novo banco de dados primário após a alternância. Após a conclusão da alternância do Oracle Data Guard, reconecte sua aplicação ao novo banco de dados primário.

Configuração do novo banco de dados primário

Para realizar uma alternância para o novo banco de dados primário, o Amazon RDS altera o modo do banco de dados em espera original a ser aberto. A mudança na função é a única alteração no banco de dados. O Amazon RDS não configura recursos como a replicação Multi-AZ.

Se você realizar uma alternância para uma réplica entre regiões com opções diferentes, o novo banco de dados primário manterá suas próprias opções. O Amazon RDS não migrará as opções no banco de dados primário original. Se o banco de dados primário original tiver opções como SSL, NNE, OEM e OEM_AGENT, o Amazon RDS não as propagará para o novo banco de dados primário.

Requisitos para a transição do Oracle Data Guard

Antes de iniciar a alternância do Oracle Data Guard, verifique se o seu ambiente de replicação atende aos seguintes requisitos:

- O banco de dados em espera original é montado ou aberto somente para leitura.
- Os backups automáticos são habilitados no banco de dados em espera original.
- O banco de dados primário original e o banco de dados em espera original estão no estado `available`.
- O banco de dados primário original e o banco de dados em espera original não têm ações de manutenção pendentes em nenhum destes estados: `required`, `next window` ou `in progress`. As ações nesses estados bloqueiam a transição. Para saber como verificar o status de atualizações de manutenção pendentes, consulte Visualizar atualizações de manutenção pendentes.

  As ações de manutenção pendentes no estado `available` não bloqueiam a transição. O RDS para Oracle frequentemente lança atualizações do sistema operacional (SO) no estado `available`. Essas atualizações pendentes do sistema operacional não bloquearão uma

transição, a menos que você as agende para a próxima janela de manutenção, o que as coloca no estado `next window`.

> **Note**
>
> Se você quiser adiar uma ação de manutenção programada para poder executar uma transição, escolha Ações e, depois, Adiar atualização no console do RDS. Você também pode evitar que uma transição seja bloqueada aplicando uma ação de manutenção pendente ou movendo a janela de manutenção para um intervalo antes da transição. Consulte mais informações no artigo do re:Post How to remove RDS pending maintenance items.

- O banco de dados em espera original está no estado de replicação.
- Você não está tentando iniciar uma alternância quando o banco de dados primário ou o banco de dados em espera estão atualmente em um ciclo de vida de alternância. Se um banco de dados de réplica estiver sendo reconfigurado após uma alternância, o Amazon RDS impede que você inicie outra alternância.

> **Note**
>
> Uma réplica do espectador é uma réplica na configuração do Oracle Data Guard que não é o alvo da alternância. As réplicas do espectador podem estar em qualquer estado durante a alternância.

- O banco de dados em espera original tem uma configuração que é a mais próxima desejada do banco de dados primário original. Vamos supor que um cenário em que os bancos de dados primários originais e em espera originais tenham opções diferentes. Após a conclusão da alternância, o Amazon RDS não reconfigura automaticamente o novo banco de dados primário para ter as mesmas opções do banco de dados primário original.
- Você configura a implantação multi-AZ desejada antes de iniciar uma transição. O Amazon RDS não gerencia o multi-AZ como parte da transição. A implantação Multi-AZ permanece no estado em que se encontra.

  Suponha que db_maz seja o banco de dados principal em uma implantação multi-AZ e db_saz seja uma réplica Single-AZ. Você inicia uma transição de db_maz para db_saz. Posteriormente, db_maz é um banco de dados de réplica multi-AZ e db_saz é um banco de dados principal Single-AZ. O novo banco de dados principal já pode ser desprotegido por uma implantação multi-AZ.

- Em preparação para uma transição entre regiões, o banco de dados principal não usa o mesmo grupo de opções de uma instância de banco de dados fora da configuração de replicação. Para que uma transição entre regiões seja bem-sucedida, o banco de dados principal atual e suas réplicas de leitura devem ser as únicas instâncias de banco de dados a usar o grupo de opções do banco de dados principal atual. Caso contrário, o Amazon RDS impede a transição.

Iniciar a alternância do Oracle Data Guard

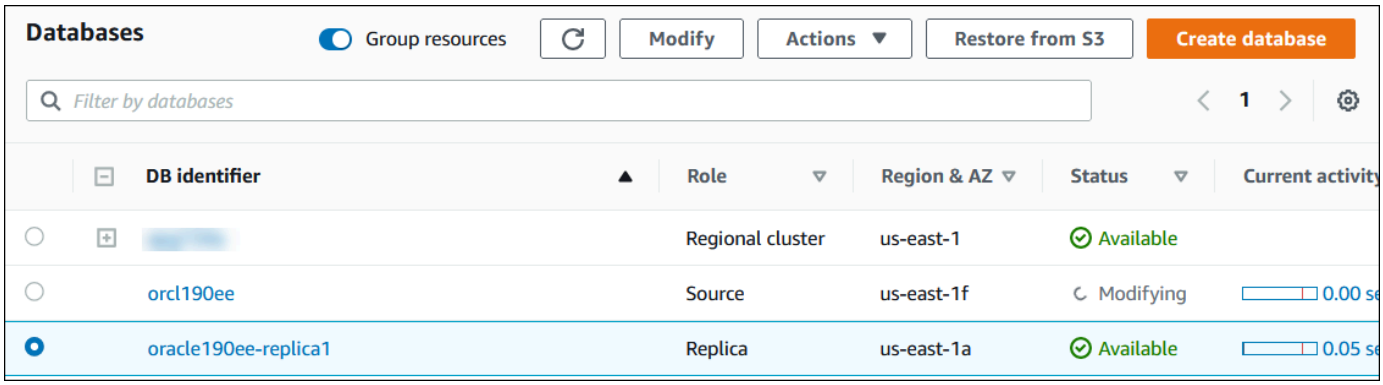
Você pode alternar uma réplica de leitura do RDS para Oracle para a função primária e a antiga instância de banco de dados primária para uma função de réplica.

Console

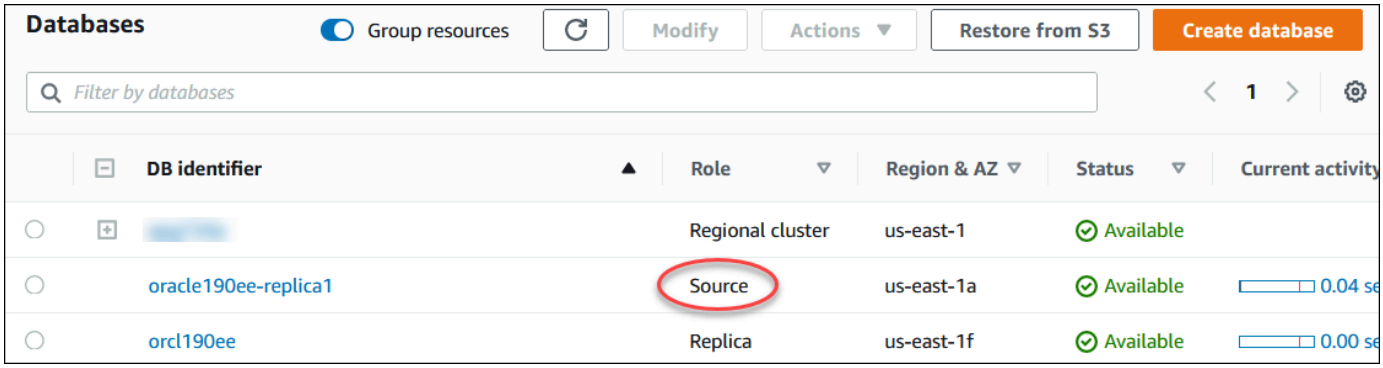
Como alternar uma réplica de leitura da Oracle para a função de banco de dados primária

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No console do Amazon RDS, escolha Databases (Bancos de dados).

   O painel Databases (Bancos de dados) é exibido. Todas as réplicas de leitura exibem Replica (Réplica) na coluna Role (Função).
3. Escolha a réplica de leitura que você deseja alternar para a função primária.
4. Para Actions (Ações), escolha Switch over replica (Alternar réplica).
5. Escolha I acknowledge (Eu aceito). Depois, escolha Switch over replica (Alternar réplica).
6. Na página Databases (Bancos de dados), monitore o progresso da transição.



   Quando a transição é concluída, a função do destino da alternância muda de Replica (Réplica) para Primary (Principal).



AWS CLI

Para mudar uma réplica do Oracle para a função de banco de dados primária, use o comando `switchover-read-replica` do AWS CLI. Os exemplos a seguir tornam a réplica do Oracle chamada *`replica-to-be-made-primary`* no novo banco de dados primário.

Example

Para Linux, macOS ou Unix:

```
aws rds switchover-read-replica \
    --db-instance-identifier replica-to-be-made-primary
```

Para Windows:

```
aws rds switchover-read-replica ^
    --db-instance-identifier replica-to-be-made-primary
```

API do RDS

Para alternar uma réplica do Oracle para a função de banco de dados primária, chame a operação da API `SwitchoverReadReplica` do Amazon RDS com o parâmetro necessário `DBInstanceIdentifier`. Esse parâmetro especifica o nome da réplica do Oracle que você deseja assumir como função de banco de dados primária.

Monitorar a alternância do Oracle Data Guard

Para verificar o status das suas instâncias usando o comando AWS da `describe-db-instances` CLI. O comando a seguir verifica o status da instância de banco de dados *`orcl2`*. Esse banco

de dados era um banco de dados em espera antes da alternância, mas é o novo banco de dados primário após a alternância.

```
aws rds describe-db-instances \
    --db-instance-identifier orcl2
```

Para verificar se a alternância foi concluída com êxito, consulte `V$DATABASE.OPEN_MODE`. Verifique se o valor do novo banco de dados primário é `READ WRITE`.

```
SELECT OPEN_MODE FROM V$DATABASE;
```

Para procurar eventos relacionados à alternância, use o comando `describe-events` da AWS.CLI. O exemplo a seguir procura eventos na instância *`orcl2`*.

```
aws rds describe-events \
    --source-identifier orcl2 \
    --source-type db-instance
```

Solução de problemas de réplicas do RDS para Oracle

Esta seção descreve possíveis problemas e soluções de replicação.

Tópicos

- Monitorar atraso da replicação Oracle
- Solução de problemas de replicação após a adição ou a modificação de acionadores

Monitorar atraso da replicação Oracle

Para monitorar o atraso de replicação no Amazon CloudWatch, visualize a métrica `ReplicaLag` do Amazon RDS. Para obter informações sobre o tempo de atraso de replicação, consulte Monitoramento da replicação de leitura e Métricas do Amazon CloudWatch para o Amazon RDS.

No caso de uma réplica de leitura, se o atraso for muito longo, consulte as seguintes visualizações:

- `V$ARCHIVED_LOG` – mostra quais confirmações foram aplicadas à réplica de leitura.
- `V$DATAGUARD_STATS` – mostra uma análise detalhada dos componentes que compõem a métrica `ReplicaLag`.
- `V$DATAGUARD_STATUS` – mostra a saída de log dos processos de replicação internos do Oracle.

No caso de uma réplica montada, se o tempo de atraso for muito longo, não será possível consultar as visualizações do `V$`. Em vez disso, faça o seguinte:

- Confira a métrica `ReplicaLag` no CloudWatch.
- Confira o arquivo de log de alertas para a réplica no console. Procure por erros nas mensagens de recuperação. As mensagens incluem o número da sequência de log, que você pode comparar com o número de sequência principal. Para obter mais informações, consulte Arquivos de log do banco de dados do Amazon RDS para Oracle.

Solução de problemas de replicação após a adição ou a modificação de acionadores

Se você adicionar ou modificar quaisquer triggers e a replicação falhar logo após, o problema pode estar nos triggers. Certifique-se de que o trigger exclua as seguintes contas de usuário, que são exigidas pelo RDS para replicação:

- Contas de usuário com privilégios de administrador
- `SYS`
- `SYSTEM`
- `RDS_DATAGUARD`
- `rdsdb`

Para obter mais informações, consulte Considerações diversas sobre réplicas do RDS para Oracle.

Adição de opções a instâncias de banco de dados Oracle

No Amazon RDS, uma opção é um recurso adicional. A seguir, você pode encontrar uma descrição das opções que você pode adicionar às instâncias do Amazon RDS que executam o mecanismo de banco de dados Oracle.

Tópicos

- Visão geral das opções do Oracle DB
- Integração do Amazon S3
- Oracle Application Express (APEX)
- Integração do Amazon EFS
- Oracle Java Virtual Machine
- Oracle Enterprise Manager
- Oracle Label Security
- Oracle Locator
- Oracle Native Network Encryption
- Oracle OLAP
- Oracle Secure Sockets Layer
- Oracle Spatial
- Oracle SQLT
- Oracle Statspack
- Fuso horário da Oracle
- Atualização automática do arquivo de fuso horário do Oracle
- Oracle Transparent Data Encryption
- Oracle UTL_MAIL
- Oracle XML DB

Visão geral das opções do Oracle DB

Para habilitar opções para seu banco de dados Oracle, adicione-as a um grupo de opções e, em seguida, associe o grupo de opções à sua instância de banco de dados. Para ter mais informações, consulte Trabalhar com grupos de opções.

Tópicos

- Resumo das opções do Oracle Database
- Opções compatíveis com diferentes edições
- Requisitos de memória para opções específicas

Resumo das opções do Oracle Database

Você pode adicionar as seguintes opções de instância de banco de dados Oracle.

Opção	ID da opção
Integração do Amazon S3	`S3_INTEGRATION`
Oracle Application Express (APEX)	`APEX` `APEX-DEV`
Oracle Enterprise Manager	`OEM` `OEM_AGENT`
Oracle Java Virtual Machine	`JVM`
Oracle Label Security	`OLS`
Oracle Locator	`LOCATOR`
Oracle Native Network Encryption	`NATIVE_NETWORK_ENCRYPTION`
Oracle OLAP	`OLAP`
Oracle Secure Sockets Layer	`SSL`
Oracle Spatial	`SPATIAL`
Oracle SQLT	`SQLT`
Oracle Statspack	`STATSPACK`

Opção	ID da opção
Fuso horário da Oracle	`Timezone`
Atualização automática do arquivo de fuso horário do Oracle	`TIMEZONE_FILE_AUTOUPGRADE`
Oracle Transparent Data Encryption	`TDE`
Oracle UTL_MAIL	`UTL_MAIL`
Oracle XML DB	`XMLDB`

Opções compatíveis com diferentes edições

O RDS for Oracle impede que você adicione opções a uma edição se elas não forem compatíveis. Para descobrir quais opções do RDS são aceitas em diferentes edições do Oracle Database, use o comando `aws rds describe-option-group-options`. O exemplo a seguir lista as opções compatíveis com o Oracle Database 19c Enterprise Edition.

```
aws rds describe-option-group-options \
    --engine-name oracle-ee \
    --major-engine-version 19
```

Para obter mais informações, consulte describe-option-group-groups na Referência de comandos da AWS CLI.

Requisitos de memória para opções específicas

Algumas opções exigem memória adicional para serem executadas na sua instância de banco de dados. Por exemplo, o Oracle Enterprise Manager Database Control usa cerca de 300 MB de RAM. Se você habilitar essa opção para uma instância de banco de dados pequena, poderá encontrar problemas de performance devido a restrições de memória. Você pode ajustar os parâmetros Oracle para que o banco de dados necessite de menos RAM. Como alternativa, você pode expandir a instância de banco de dados.

Integração do Amazon S3

É possível transferir arquivos entre uma instância de banco de dados do RDS para Oracle e um bucket do Amazon S3. É possível usar a integração do Amazon S3 com recursos do banco de dados do Oracle, como o Oracle Data Pump. Por exemplo, é possível baixar arquivos do Data Pump do Amazon S3 para a instância de banco de dados do RDS para Oracle. Para ter mais informações, consulteImportar dados para o Oracle no Amazon RDS

> **Note**
>
> A instância de banco de dados e o bucket do Amazon S3 devem estar na mesma Região da AWS.

Tópicos

- Configurar permissões do IAM para a integração do RDS para Oracle com o Amazon S3
- Adicionar a opção de integração do Amazon S3
- Transferir arquivos entre o Amazon RDS para Oracle e um bucket do Amazon S3
- Solução de problemas de integração do Amazon S3
- Remover a opção de integração do Amazon S3

Configurar permissões do IAM para a integração do RDS para Oracle com o Amazon S3

Para que o RDS para Oracle se integre com o Amazon S3, a instância de banco de dados deve ter acesso a um bucket do Amazon S3. A Amazon VPC usada pela instância de banco de dados não precisa conceder acesso aos endpoints do Amazon S3.

O RDS para Oracle é compatível com a transferência de arquivos entre uma instância de banco de dados em uma conta e um bucket do Amazon S3 em uma conta diferente. Quando etapas adicionais forem necessárias, elas serão indicadas nas seções a seguir.

Tópicos

- Etapa 1: Criar uma política do IAM para ser usada pelo perfil do Amazon RDS
- Etapa 2: (opcional) Criar uma política do IAM para o bucket do Amazon S3
- Etapa 3: Criar um perfil do IAM para a instância de banco de dados e anexar a política

- Etapa 4: Associar um perfil do IAM à sua instância de banco de dados do RDS para Oracle

Etapa 1: Criar uma política do IAM para ser usada pelo perfil do Amazon RDS

Nessa etapa, crie uma política do AWS Identity and Access Management (IAM) com as permissões necessárias para transferir arquivos entre o bucket do Amazon S3 e a instância de banco de dados do RDS. Esta etapa pressupõe que você já tenha criado um bucket do S3.

Antes de criar a política, observe as seguintes informações:

- Nome do recurso da Amazon (ARN) do bucket.
- O ARN da chave do AWS KMS, se o bucket usar criptografia SSE-KMS ou SSE-S3

> **Note**
>
> Uma instância de banco de dados do RDS para Oracle não pode acessar buckets do Amazon S3 criptografados com SSE-C.

Para ter mais informações, consulte Como proteger dados usando criptografia do lado do servidor no Guia do usuário do Amazon Simple Storage Service.

Console

Como criar uma política do IAM para permitir acesso do Amazon RDS a um bucket do Amazon S3

1. Abra o Console de Gerenciamento do IAM.
2. Em Access management (Gerenciamento de acesso), escolha Policies (Políticas).
3. Escolha Create Policy.
4. Na guia Editor visual, selecione Escolher um serviço e, em seguida, escolha S3.
5. Em Actions (Ações), escolha Expand all (Expandir tudo)e, em seguida, escolha as permissões de bucket e as permissões de objeto necessárias para transferir arquivos de um Amazon S3 bucket para Amazon RDS. Por exemplo, faça o seguinte:
   - Expanda List (Lista)e, em seguida, selecione ListBucket.
   - Expanda Read (Ler) e, em seguida, selecione GetObject.
   - Expanda Write (Gravar) e selecione PutObject e DeleteObject.

- Amplie Permissions management (Gerenciamento de permissões) e selecione PutObjectAcl. Essa permissão será necessária se você planejar o carregamento de arquivos em um bucket de propriedade de uma conta diferente, e essa conta precisar ter controle total do conteúdo do bucket.

Permissões de objetos são permissões para operações de objetos no Amazon S3. Você deve concedê-las para objetos em um bucket, não para o próprio bucket. Para ter mais informações, consulte Permissões para operações de objetos.

6. Selecione Recursos e, depois, faça o seguinte:

   a. Selecione Específico.

   b. Para bucket, selecione Adicionar ARN. Insira o ARN de seu bucket. O nome do bucket é preenchido automaticamente. Em seguida, escolha Adicionar.

   c. Se o recurso do objeto for exibido, selecione Adicionar ARN para adicionar recursos manualmente ou escolha Qualquer.

   > **Note**
   >
   > É possível definir o nome de recurso da Amazon (ARN) como um valor de ARN mais específico para permitir que o Amazon RDS acesse apenas arquivos ou pastas específicos em um bucket do Amazon S3. Para ter mais informações sobre como definir uma política de acesso para o Amazon S3, consulte Gerenciar permissões de acesso aos recursos do Amazon S3.

7. (Opcional) Escolha Adicionar permissões adicionais para adicionar recursos à política. Por exemplo, faça o seguinte:

   a. Se o bucket estiver criptografado com uma chave KMS personalizada, selecione KMS para o serviço.

   b. Em Ações manuais, selecione o seguinte:

      - Encrypt
      - Criptografar novamente de e Criptografar novamente para
      - Decrypt
      - DescribeKey
      - GenerateDataKey

c. Em Recursos, selecione Específico.

d. Para chave, selecione Adicionar ARN. Insira o ARN de sua chave personalizada como o recurso e, depois, selecione Adicionar.

   Para ter mais informações, consulte Como proteger os dados usando criptografia do lado do servidor com chaves do KMS armazenadas no AWS Key Management Service (SSE-KMS) do Guia do usuário do Amazon Simple Storage Service.

e. Se você Amazon RDS quiser acessar outros bucket, adicione os ARNs para esses buckets. Opcionalmente, você também pode conceder acesso a todos os buckets e objetos no Amazon S3.

8. Escolha Next: Tags (Próximo: tags) e Next: Review (Próximo: revisar).
9. Em Name (Nome), insira um nome para a sua política do IAM, por exemplo `rds-s3-integration-policy`. Você usará esse nome ao criar um perfil do IAM a ser associado à sua instância de banco de dados. Você também pode adicionar um valor opcional para Description (Descrição).
10. Escolha Create policy (Criar política).

AWS CLI

Crie uma política do AWS Identity and Access Management (IAM) que conceda ao Amazon RDS acesso a um bucket do Amazon S3. Depois de criar a política, anote o ARN da política. Você precisa do ARN para uma etapa subsequente.

Inclua as ações apropriadas na política com base no tipo de acesso necessário:

- `GetObject` – necessário para transferir arquivos de um bucket do Amazon S3 para o Amazon RDS.
- `ListBucket` – necessário para transferir arquivos de um bucket do Amazon S3 para o Amazon RDS.
- `PutObject` – Necessário para transferir arquivos do Amazon RDS para um bucket do Amazon S3.

O comando da AWS CLI a seguir cria uma política do IAM denominada *`rds-s3-integration-policy`* com essas opções. Ele concede acesso a um bucket denominado *`amzn-s3-demo-bucket`*.

Example

Para Linux, macOS ou Unix:

```
aws iam create-policy \
   --policy-name rds-s3-integration-policy \
   --policy-document '{
     "Version": "2012-10-17",
     "Statement": [
       {
         "Sid": "s3integration",
         "Action": [
           "s3:GetObject",
           "s3:ListBucket",
           "s3:PutObject"
         ],
         "Effect": "Allow",
         "Resource": [
           "arn:aws:s3:::amzn-s3-demo-bucket",
           "arn:aws:s3:::amzn-s3-demo-bucket/*"
         ]
       }
     ]
   }'
```

O exemplo a seguir inclui permissões para chaves KMS personalizadas.

```
aws iam create-policy \
   --policy-name rds-s3-integration-policy \
   --policy-document '{
     "Version": "2012-10-17",
     "Statement": [
       {
         "Sid": "s3integration",
         "Action": [
           "s3:GetObject",
           "s3:ListBucket",
           "s3:PutObject",
           "kms:Decrypt",
           "kms:Encrypt",
           "kms:ReEncrypt*",
           "kms:GenerateDataKey",
           "kms:DescribeKey",
```

```
        ],
        "Effect": "Allow",
        "Resource": [
          "arn:aws:s3:::amzn-s3-demo-bucket",
          "arn:aws:s3:::amzn-s3-demo-bucket/*",
          "arn:aws:kms:::your-kms-arn"
        ]
      }
    ]
  }'
```

Para Windows:

```
aws iam create-policy ^
   --policy-name rds-s3-integration-policy ^
   --policy-document '{
     "Version": "2012-10-17",
     "Statement": [
       {
         "Sid": "s3integration",
         "Action": [
           "s3:GetObject",
           "s3:ListBucket",
           "s3:PutObject"
         ],
         "Effect": "Allow",
         "Resource": [
           "arn:aws:s3:::amzn-s3-demo-bucket",
           "arn:aws:s3:::amzn-s3-demo-bucket/*"
         ]
       }
     ]
   }'
```

O exemplo a seguir inclui permissões para chaves KMS personalizadas.

```
aws iam create-policy ^
   --policy-name rds-s3-integration-policy ^
   --policy-document '{
     "Version": "2012-10-17",
     "Statement": [
       {
         "Sid": "s3integration",
```

```
        "Action": [
          "s3:GetObject",
          "s3:ListBucket",
          "s3:PutObject",
          "kms:Decrypt",
          "kms:Encrypt",
          "kms:ReEncrypt",
          "kms:GenerateDataKey",
          "kms:DescribeKey",
        ],
        "Effect": "Allow",
        "Resource": [
          "arn:aws:s3:::amzn-s3-demo-bucket",
          "arn:aws:s3:::amzn-s3-demo-bucket/*",
          "arn:aws:kms:::your-kms-arn"
        ]
      }
    ]
  }'
```

Etapa 2: (opcional) Criar uma política do IAM para o bucket do Amazon S3

Esta etapa é necessária somente nas seguintes condições:

- Você planeja carregar arquivos para um bucket do Amazon S3 de uma conta (conta A) e acessá-los em uma conta diferente (conta B).
- A conta B é a proprietária do bucket.
- A conta B precisa ter controle total dos objetos carregados no bucket.

Se as condições anteriores não se aplicarem a você, avance para Etapa 3: Criar um perfil do IAM para a instância de banco de dados e anexar a política.

Para criar a política de bucket, verifique se você tem o seguinte:

- O ID da conta A
- O nome do usuário da conta A
- O valor do ARN do bucket do Amazon S3 na conta B

Console

Para criar ou editar uma política de bucket

1. Faça login no AWS Management Console e abra o console do Amazon S3 em https://console.aws.amazon.com/s3/.
2. Na lista Buckets, escolha o nome do bucket para o qual você deseja criar uma política de bucket ou cuja política de bucket você deseja editar.
3. Escolha Permissions (Permissões).
4. Em Bucket policy (Política de bucket), escolha Edit (Editar). Isso abre a página Edit bucket policy (Editar política de bucket).
5. Na página Edit bucket policy (Editar política de buckets), explore Exemplos de políticas no Guia do usuário do Amazon S3, escolha Policy generator (Gerador de políticas) para gerar uma política automaticamente ou edite o JSON na seção Policy (Política).

   Se você escolher Policy generator (Gerador de políticas), o AWS Policy Generator será aberto em uma nova janela:

   a. Na página AWS Policy Generator, em Select Type of Policy (Selecionar tipo de política), S3 Bucket Policy (Política de bucket do S3).
   b. Adicione uma instrução inserindo as informações nos campos fornecidos e escolha Add Statement (Adicionar instrução). Repita para todas as instruções que deseja adicionar. Para ter mais informações sobre esses campos, consulte a Referência de elementos de política JSON do IAM no Guia do usuário do IAM.

      > **Note**
      >
      > Por conveniência, a página Edit bucket policy (Editar política do bucket) exibe o nome do recurso da Amazon (ARN) do bucket atual acima do campo de texto Policy (Política). Você pode copiar este ARN para usá-lo nas instruções da página do AWS Policy Generator.

   c. Depois de terminar de adicionar as instruções, escolha Generate Policy (Gerar política).
   d. Copie o texto da política gerada, escolha Close (Fechar) e retorne à página Edit bucket policy (Editar política de bucket) no console do Amazon S3.

6. Na caixa Policy (Política), edite a política existente ou cole a política de bucket do gerador de políticas. Solucione avisos de segurança, erros, avisos gerais e sugestões antes de salvar a política.

   JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Example permissions",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::account-A-ID:account-A-user"
      },
      "Action": [
        "s3:PutObject",
        "s3:PutObjectAcl"
      ],
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-destination-bucket",
        "arn:aws:s3:::amzn-s3-demo-destination-bucket/*"
      ]
    }
  ]
}
```

7. Escolha Save changes (Salvar alterações), o que o levará à página Bucket Permissions (Permissões do bucket).

Etapa 3: Criar um perfil do IAM para a instância de banco de dados e anexar a política

Esta etapa pressupõe que você tenha criado a política do IAM no Etapa 1: Criar uma política do IAM para ser usada pelo perfil do Amazon RDS. Nesta etapa, você cria um perfil para a instância de banco de dados do RDS para Oracle e anexa a política ao perfil.

Console

Como criar um perfil do IAM para permitir acesso do Amazon RDS a um bucket do Amazon S3

1. Abra o Console de Gerenciamento do IAM.

2. No painel de navegação, escolha Roles.
3. Escolha Criar Perfil.
4. Selecione Serviço da AWS.
5. Em Casos de uso de outros serviços da AWS:, selecione RDS e, depois, RDS: adicionar perfil ao banco de dados. Escolha Próximo.
6. Em Pesquisar, em Políticas de permissões, insira o nome da política do IAM que você criou em Etapa 1: Criar uma política do IAM para ser usada pelo perfil do Amazon RDS e selecione a política quando ela aparecer na lista. Escolha Próximo.
7. Em Nome do perfil, insira um nome para seu perfil do IAM, por exemplo, `rds-s3-integration-role`. Você também pode adicionar um valor opcional para Description (Descrição).
8. Selecione Criar perfil.

AWS CLI

Como criar um perfil e anexar uma política a ele

1. Crie um perfil do IAM que o Amazon RDS possa assumir em seu nome para acessar seus buckets do Amazon S3.

   Recomendamos usar as chaves de contexto de condição global `aws:SourceArn` e `aws:SourceAccount` em relações baseadas em recursos para limitar as permissões do serviço a um recurso específico. Essa é a maneira mais eficiente de se proteger contra o problema "confused deputy".

   Você pode usar as duas chaves de contexto de condição global e fazer com que o valor `aws:SourceArn` contenha o ID da conta. Nesses casos, verifique se o valor `aws:SourceAccount` e a conta no `aws:SourceArn` usa o mesmo ID de conta quando eles são usados na mesma instrução.

   - Use `aws:SourceArn` se quiser acesso entre serviços para um único recurso.
   - Use `aws:SourceAccount` se você quiser permitir que qualquer recurso nessa conta seja associado ao uso entre serviços.

Na relação de confiança, certifique-se de usar a chave de contexto de condição global `aws:SourceArn` com o nome do recurso da Amazon (ARN) completo dos recursos que acessam a função.

O seguinte comando da AWS CLI cria a função chamada *rds-s3-integration-role* para essa finalidade.

Example

Para Linux, macOS ou Unix:

```
aws iam create-role \
   --role-name rds-s3-integration-role \
   --assume-role-policy-document '{
     "Version": "2012-10-17",
     "Statement": [
       {
         "Effect": "Allow",
         "Principal": {
            "Service": "rds.amazonaws.com"
          },
         "Action": "sts:AssumeRole",
         "Condition": {
             "StringEquals": {
                 "aws:SourceAccount": "my_account_ID",
                 "aws:SourceArn": "arn:aws:rds:Region:my_account_ID:db:dbname"
             }
         }
       }
     ]
   }'
```

Para Windows:

```
aws iam create-role ^
   --role-name rds-s3-integration-role ^
   --assume-role-policy-document '{
     "Version": "2012-10-17",
     "Statement": [
       {
```

```
        "Effect": "Allow",
        "Principal": {
           "Service": "rds.amazonaws.com"
         },
        "Action": "sts:AssumeRole",
        "Condition": {
            "StringEquals": {
                "aws:SourceAccount": "my_account_ID",
                "aws:SourceArn": "arn:aws:rds:Region:my_account_ID:db:dbname"
            }
        }
      }
    ]
  }'
```

Para ter mais informações, consulte Criar uma função para delegar permissões a um usuário do IAM no Guia do usuário do IAM.

2. Depois que a função for criada, observe seu ARN. Você precisa do ARN para uma etapa subsequente.
3. Anexe a política criada à função criada.

   O seguinte comando da AWS CLI anexa a política à função denominada *rds-s3-integration-role*.

   Example

   Para Linux, macOS ou Unix:

   ```
   aws iam attach-role-policy \
      --policy-arn your-policy-arn \
      --role-name rds-s3-integration-role
   ```

   Para Windows:

   ```
   aws iam attach-role-policy ^
      --policy-arn your-policy-arn ^
      --role-name rds-s3-integration-role
   ```

   Substitua *your-policy-arn* pelo ARN da política que você anotou em uma etapa anterior.

Etapa 4: Associar um perfil do IAM à sua instância de banco de dados do RDS para Oracle

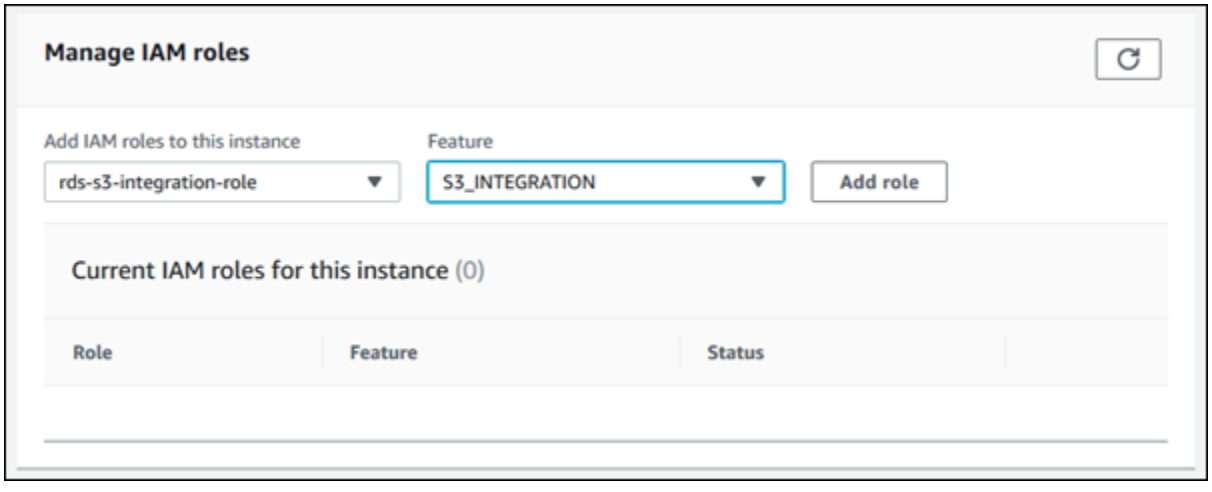
A última etapa na configuração de permissões para a integração com o Amazon S3 é associar seu perfil do IAM à sua instância de banco de dados. Observe os seguintes requisitos:

- Você precisa ter acesso a um perfil do IAM com a política de permissões do Amazon S3 necessária anexada a ele.
- Só é possível associar um perfil do IAM à instância de banco de dados do RDS para Oracle por vez.
- Sua instância de banco de dados deve estar no estado Disponível.

Console

Como associar um perfil do IAM à instância de banco de dados do RDS para Oracle

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. Escolha Databases (Bancos de dados) no painel de navegação.
3. Selecione o nome da instância de banco de dados Oracle para exibir os detalhes.
4. Na guia Connectivity & security (Conectividade e segurança), role para baixo até a seção Manage IAM roles (Gerenciar funções do IAM) na parte inferior da página.
5. Em Adicionar perfis do IAM a essa instância, selecione o perfil que você criou em Etapa 3: Criar um perfil do IAM para a instância de banco de dados e anexar a política.
6. Em Feature (Recurso), escolha S3_INTEGRATION.



7. Escolha Add role (adicionar função).

AWS CLI

O seguinte comando da AWS CLI adiciona a função a uma instância de banco de dados Oracle denominada *mydbinstance*.

Example

Para Linux, macOS ou Unix:

```
aws rds add-role-to-db-instance \
   --db-instance-identifier mydbinstance \
   --feature-name S3_INTEGRATION \
   --role-arn your-role-arn
```

Para Windows:

```
aws rds add-role-to-db-instance ^
   --db-instance-identifier mydbinstance ^
   --feature-name S3_INTEGRATION ^
   --role-arn your-role-arn
```

Substitua *your-role-arn* pelo ARN da função que você anotou em uma etapa anterior. `S3_INTEGRATION` deve ser especificado para a opção `--feature-name`.

Adicionar a opção de integração do Amazon S3

Para integrar o Amazon RDS para Oracle ao Amazon S3, sua instância de banco de dados deve estar associada a um grupo de opções que inclua a opção `S3_INTEGRATION`.

Console

Para configurar um grupo de opções para a integração com o Amazon S3

1. Crie um novo grupo de opções ou identifique um grupo de opções existente ao qual você pode adicionar a opção `S3_INTEGRATION`.

   Para obter informações sobre como criar um grupo de opções, consulte Criar um grupo de opções.
2. Adicione a opção `S3_INTEGRATION` ao grupo de opções.

   Para obter informações sobre como adicionar uma opção a um grupo de opções, consulte Adicionar uma opção a um grupo de opções.

3. Crie uma instância de banco de dados do RDS para Oracle e associe a ela o grupo de opções ou modifique uma instância de banco de dados do RDS para Oracle para associar o grupo de opções a ela.

   Para obter informações sobre como criar uma instância de banco de dados, consulte Criar uma instância de banco de dados do Amazon RDS.

   Para mais informações sobre a modificação de uma instância de banco de dados , consulte Modificar uma instância de banco de dados do Amazon RDS.

AWS CLI

Para configurar um grupo de opções para a integração com o Amazon S3

1. Crie um novo grupo de opções ou identifique um grupo de opções existente ao qual você pode adicionar a opção `S3_INTEGRATION`.

   Para obter informações sobre como criar um grupo de opções, consulte Criar um grupo de opções.

2. Adicione a opção `S3_INTEGRATION` ao grupo de opções.

   Por exemplo, o seguinte comando da AWS CLI adiciona a opção `S3_INTEGRATION` a um grupo de opções denominado **`myoptiongroup`**.

   Example

   Para Linux, macOS ou Unix:

```
aws rds add-option-to-option-group \
   --option-group-name myoptiongroup \
   --options OptionName=S3_INTEGRATION,OptionVersion=1.0
```

   Para Windows:

```
aws rds add-option-to-option-group ^
   --option-group-name myoptiongroup ^
   --options OptionName=S3_INTEGRATION,OptionVersion=1.0
```

3. Crie uma instância de banco de dados do RDS para Oracle e associe a ela o grupo de opções ou modifique uma instância de banco de dados do RDS para Oracle para associar o grupo de opções a ela.

   Para obter informações sobre como criar uma instância de banco de dados, consulte Criar uma instância de banco de dados do Amazon RDS.

   Para ter informações sobre como modificar uma instância de banco de dados do RDS para Oracle, consulte Modificar uma instância de banco de dados do Amazon RDS.

Transferir arquivos entre o Amazon RDS para Oracle e um bucket do Amazon S3

É possível usar um pacote `rdsadmin_s3_tasks` do Amazon RDS para transferir arquivos entre uma instância de banco de dados do RDS para Oracle e um bucket do Amazon S3. É possível compactar arquivos com o GZIP ao carregá-los e descompactá-los ao baixar.

Tópicos

- Requisitos e limitações das transferências de arquivos
- Carregar arquivos de uma instância de banco de dados do RDS para Oracle para um bucket do Amazon S3
- Baixar arquivos de um bucket do Amazon S3 para uma instância de banco de dados Oracle
- Monitorar o status de uma transferência de arquivo

Requisitos e limitações das transferências de arquivos

Antes de transferir arquivos entre a instância de banco de dados e um bucket do Amazon S3, observe o seguinte:

- O pacote `rdsadmin_s3_tasks` transfere arquivos localizados em um único diretório. Não é possível incluir subdiretórios em uma transferência.
- O tamanho máximo do objeto em um bucket do Amazon S3 é 5 TB.
- Tarefas criadas por `rdsadmin_s3_tasks` são executadas de forma assíncrona.
- É possível fazer upload de arquivos do diretório Data Pump, como `DATA_PUMP_DIR`, ou de qualquer diretório criado pelo usuário. Não é possível fazer upload de arquivos de um diretório usado pelos processos em segundo plano da Oracle, como os diretórios `adump`, `bdump` ou `trace`.

- O limite de download é de 2 mil arquivos por chamada de procedimento para `download_from_s3`. Se você precisar baixar mais de 2.000 arquivos do Amazon S3, divida o download em ações separadas, com no máximo 2.000 arquivos por chamado do procedimento.
- Se já existir um arquivo na pasta de download e você tentar baixar um arquivo com o mesmo nome, `download_from_s3` vai ignorar o download. Para remover um arquivo do diretório de downloads, use o procedimento PL/SQL UTL_FILE.FREMOVE.

Carregar arquivos de uma instância de banco de dados do RDS para Oracle para um bucket do Amazon S3

Para carregar arquivos da instância de banco de dados para um bucket do Amazon S3, use o procedimento `rdsadmin.rdsadmin_s3_tasks.upload_to_s3`. Por exemplo, você pode carregar arquivos de backup do Oracle Recovery Manager (RMAN) ou arquivos do Oracle Data Pump. Para ter mais informações sobre como trabalhar com objetos, consulte o Guia do usuário do Amazon Simple Storage Service. Para ter mais informações sobre como executar backups do RMAN, consulte Realização de tarefas comuns do RMAN para instâncias de banco de dados Oracle.

O procedimento `rdsadmin.rdsadmin_s3_tasks.upload_to_s3` tem os seguintes parâmetros.

Nome do parâmetro	Tipo de dados	Padrão	Obrigatório	Descrição
`p_bucket_name`	VARCHAR2	–	obrigatório	O nome do bucket do Amazon S3 no qual fazer o upload de arquivos.
`p_directory_name`	VARCHAR2	–	obrigatório	O nome do objeto de diretório Oracle do qual carregar arquivos. O diretório pode ser qualquer objeto de diretório criado pelo usuário ou o diretório Data Pump, como `DATA_PUMP_DIR` . Não é possível fazer upload de arquivos de

Nome do parâmetro	Tipo de dados	Padrão	Obrigatório	Descrição
				um diretório usado pelos processos em segundo plano, como `adump`, `bdump` e `trace`. **Note** Você só pode fazer o upload de arquivos do diretório especificado. Não é possível fazer upload de arquivos em subdiretórios no diretório especific ado.

Nome do parâmetro	Tipo de dados	Padrão	Obrigatório	Descrição
`p_s3_prefix`	VARCHAR2	–	obrigatório	Um prefixo de nome de arquivo do Amazon S3 no qual os arquivos são carregados. Um prefixo vazio faz upload de todos os arquivos para o nível superior no bucket do Amazon S3 especificado e não adiciona um prefixo aos nomes de arquivos. Por exemplo, se o prefixo for `folder_1/oradb` , os arquivos serão enviados em `folder_1`. Nesse caso, o prefixo `oradb` é adicionado a cada arquivo.
`p_prefix`	VARCHAR2	–	obrigatório	Um prefixo de nome de arquivo ao qual os nomes de arquivo devem corresponder para serem transferidos por upload. Um prefixo vazio faz upload de todos os arquivos no diretório especificado.

Nome do parâmetro	Tipo de dados	Padrão	Obrigatório	Descrição
`p_compression_level`	NUMBER	`0`	optional	O nível de compactação do GZIP. O intervalo de valores válidos varia de `0` a 9. • `0`: sem compactação • 1: compactação mais rápida • 9: maior compactação
`p_bucket_owner_full_control`	VARCHAR2	–	opcional	A configuração de controle de acesso do bucket. Os únicos valores válidos são nulo ou `FULL_CONTROL` . Essa configuração será necessária somente se você carregar arquivos de uma conta (conta A) em um bucket de propriedade de uma conta diferente (conta B), e a conta B precisar ter controle total dos arquivos.

O valor de retorno do procedimento `rdsadmin.rdsadmin_s3_tasks.upload_to_s3` é um ID de tarefa.

O exemplo a seguir faz upload de todos os arquivos do diretório *`DATA_PUMP_DIR`* no bucket do Amazon S3 chamado *`amzn-s3-demo-bucket`*. Os arquivos não são compactados.

```
SELECT rdsadmin.rdsadmin_s3_tasks.upload_to_s3(
      p_bucket_name    =>  'amzn-s3-demo-bucket',
```

```
      p_prefix         =>  '',
      p_s3_prefix      =>  '',
      p_directory_name =>  'DATA_PUMP_DIR')
   AS TASK_ID FROM DUAL;
```

O exemplo a seguir faz upload todos os arquivos com o prefixo *db* no diretório *DATA_PUMP_DIR* para o bucket do Amazon S3 denominado *amzn-s3-demo-bucket*. O Amazon RDS aplica o nível mais alto de compactação GZIP aos arquivos.

```
SELECT rdsadmin.rdsadmin_s3_tasks.upload_to_s3(
      p_bucket_name       =>  'amzn-s3-demo-bucket',
      p_prefix            =>  'db',
      p_s3_prefix         =>  '',
      p_directory_name    =>  'DATA_PUMP_DIR',
      p_compression_level =>  9)
   AS TASK_ID FROM DUAL;
```

O exemplo a seguir carrega todos os arquivos do diretório *DATA_PUMP_DIR* no bucket do Amazon S3 chamado *amzn-s3-demo-bucket*. Os arquivos são carregados em uma pasta `dbfiles`. Neste exemplo, o nível de compactação GZIP é *1*, que é o nível de compactação mais rápido.

```
SELECT rdsadmin.rdsadmin_s3_tasks.upload_to_s3(
      p_bucket_name       =>  'amzn-s3-demo-bucket',
      p_prefix            =>  '',
      p_s3_prefix         =>  'dbfiles/',
      p_directory_name    =>  'DATA_PUMP_DIR',
      p_compression_level =>  1)
   AS TASK_ID FROM DUAL;
```

O exemplo a seguir faz upload todos os arquivos no diretório *DATA_PUMP_DIR* para o bucket do Amazon S3 denominado *amzn-s3-demo-bucket*. Os arquivos são carregados em uma pasta `dbfiles` e `ora` é adicionado ao início de cada nome de arquivo. Nenhuma compactação é aplicada.

```
SELECT rdsadmin.rdsadmin_s3_tasks.upload_to_s3(
      p_bucket_name    =>  'amzn-s3-demo-bucket',
      p_prefix         =>  '',
      p_s3_prefix      =>  'dbfiles/ora',
      p_directory_name =>  'DATA_PUMP_DIR')
   AS TASK_ID FROM DUAL;
```

O exemplo a seguir pressupõe que o comando é executado na conta A, mas a conta B requer o controle total do conteúdo do bucket. O comando `rdsadmin_s3_tasks.upload_to_s3` transfere todos os arquivos no diretório *`DATA_PUMP_DIR`* para o bucket chamado*`s3bucketOwnedByAccountB`*. O controle de acesso está definido como `FULL_CONTROL` para que a conta B possa acessar os arquivos no bucket. O nível de compactação GZIP é *`6`*, o que equilibra a velocidade e o tamanho do arquivo.

```
SELECT rdsadmin.rdsadmin_s3_tasks.upload_to_s3(
      p_bucket_name               =>  's3bucketOwnedByAccountB',
      p_prefix                    =>  '',
      p_s3_prefix                 =>  '',
      p_directory_name            =>  'DATA_PUMP_DIR',
      p_bucket_owner_full_control =>  'FULL_CONTROL',
      p_compression_level         =>  6)
   AS TASK_ID FROM DUAL;
```

Em cada exemplo, a instrução `SELECT` retorna o ID da tarefa em um tipo de dados `VARCHAR2`.

Você pode visualizar o resultado exibindo o arquivo de saída da tarefa.

```
SELECT text FROM table(rdsadmin.rds_file_util.read_text_file('BDUMP','dbtask-task-id.log'));
```

Substitua *`task-id`* pelo ID da tarefa retornado pelo procedimento.

> **Note**
>
> As tarefas são executadas de forma assíncrona.

Baixar arquivos de um bucket do Amazon S3 para uma instância de banco de dados Oracle

Para baixar arquivos de um bucket do Amazon S3 para uma instância do RDS para Oracle, use o procedimento `rdsadmin.rdsadmin_s3_tasks.download_from_s3` do Amazon RDS.

O procedimento `download_from_s3` tem os seguintes parâmetros.

Nome do parâmetro	Tipo de dados	Padrão	Obrigatório	Descrição
p_bucket_name	VARCHAI	–	Obrigatório	O nome do bucket do Amazon S3 do qual baixar arquivos.
p_directory_name	VARCHAI	–	Obrigatório	O nome do objeto de diretório Oracle no qual baixar arquivos. O diretório pode ser qualquer objeto de diretório criado pelo usuário ou o diretório Data Pump, como DATA_PUMP_DIR .
p_error_on_zero_downloads	VARCHAI	FALSE	Opcional	Um sinalizador que determina se a tarefa gera um erro quando nenhum objeto no bucket do Amazon S3 corresponde ao prefixo. Se esse parâmetro não estiver definido ou estiver definido como FALSE (padrão), a tarefa imprimirá uma mensagem de que nenhum objeto foi encontrado, mas não vai gerar uma exceção nem uma falha. Se esse parâmetro for TRUE, a tarefa vai gerar uma exceção e ocorrerá uma falha. Exemplos de especificações de prefixo que podem falhar nos testes de correspondência são espaços em prefixos, como em ' import/test9.log' , e incompatibilidade de maiúsculas e minúsculas, como em test9.log e test9.LOG .
p_s3_prefix	VARCHAI	–	Obrigatório	Um prefixo de nome de arquivo ao qual os nomes de arquivo devem corresponder para serem baixados. Um prefixo vazio baixa todos os arquivos de nível

Nome do parâmetro	Tipo de dados	Padrão	Obrigatório	Descrição
				superior no bucket do Amazon S3 especificado, mas não os arquivos nas pastas do bucket. O procedimento baixa objetos Amazon S3 somente da pasta de primeiro nível que corresponde ao prefixo. Estruturas de diretório aninhadas que correspondem ao prefixo especificado não são baixados. Por exemplo, suponha que um bucket do Amazon S3 tenha a estrutura de pastas `folder_1/folder_2/folder_3` . Especifique o prefixo `'folder_1/folder_2/'` . Nesse caso, somente os arquivos em `folder_2` são baixados, e não os arquivos em `folder_1` ou `folder_3`. Em vez disso, se você especificar o prefixo `'folder_1/folder_2'` , todos os arquivos em `folder_1` que correspondem ao prefixo `'folder_2'` são baixados, e nenhum arquivo em `folder_2` é baixado.
`p_decompression_format`	VARCHAI	–	Opcional	O formato da descompactação. Os valores válidos são `NONE` para sem descompactação e `GZIP` para descompactação.

O valor de retorno do procedimento `rdsadmin.rdsadmin_s3_tasks.download_from_s3` é um ID de tarefa.

O exemplo a seguir baixa todos os arquivos no bucket do Amazon S3 denominado *amzn-s3-demo-bucket* para o diretório *DATA_PUMP_DIR*. Os arquivos não são compactados, portanto, nenhuma descompactação é aplicada.

```
SELECT rdsadmin.rdsadmin_s3_tasks.download_from_s3(
      p_bucket_name    =>  'amzn-s3-demo-bucket',
      p_directory_name =>  'DATA_PUMP_DIR')
   AS TASK_ID FROM DUAL;
```

O exemplo a seguir baixa todos os arquivos com o prefixo *db* no bucket do Amazon S3 denominado *amzn-s3-demo-bucket* para o diretório *DATA_PUMP_DIR*. Os arquivos são compactados com GZIP, portanto, nenhuma descompactação é aplicada. O parâmetro `p_error_on_zero_downloads` ativa a verificação de erros de prefixo, portanto, se o prefixo não corresponder a nenhum arquivo no bucket, a tarefa vai gerar uma exceção e ocorrerá uma falha.

```
SELECT rdsadmin.rdsadmin_s3_tasks.download_from_s3(
      p_bucket_name               =>  'amzn-s3-demo-bucket',
      p_s3_prefix                 =>  'db',
      p_directory_name            =>  'DATA_PUMP_DIR',
      p_decompression_format      =>  'GZIP',
      p_error_on_zero_downloads   =>  'TRUE')
   AS TASK_ID FROM DUAL;
```

O exemplo a seguir baixa todos os arquivos da pasta *myfolder/* no bucket do Amazon S3 denominado *amzn-s3-demo-bucket* para o diretório *DATA_PUMP_DIR*. Use o parâmetro `p_s3_prefix` para especificar a pasta Amazon S3. Os arquivos carregados são compactados com o GZIP, mas não são descompactados durante o download.

```
SELECT rdsadmin.rdsadmin_s3_tasks.download_from_s3(
      p_bucket_name          =>  'amzn-s3-demo-bucket',
      p_s3_prefix            =>  'myfolder/',
      p_directory_name       =>  'DATA_PUMP_DIR',
      p_decompression_format =>  'NONE')
   AS TASK_ID FROM DUAL;
```

O exemplo a seguir baixa o arquivo *mydumpfile.dmp* no bucket do Simple Storage Service (Amazon S3) denominado *amzn-s3-demo-bucket* para o diretório *DATA_PUMP_DIR*. Nenhuma descompactação é aplicada.

```
SELECT rdsadmin.rdsadmin_s3_tasks.download_from_s3(
```

```
      p_bucket_name    =>  'amzn-s3-demo-bucket',
      p_s3_prefix      =>  'mydumpfile.dmp',
      p_directory_name =>  'DATA_PUMP_DIR')
   AS TASK_ID FROM DUAL;
```

Em cada exemplo, a instrução `SELECT` retorna o ID da tarefa em um tipo de dados `VARCHAR2`.

Você pode visualizar o resultado exibindo o arquivo de saída da tarefa.

```
SELECT text FROM table(rdsadmin.rds_file_util.read_text_file('BDUMP','dbtask-task-id.log'));
```

Substitua *`task-id`* pelo ID da tarefa retornado pelo procedimento.

> **Note**
>
> As tarefas são executadas de forma assíncrona.
> Você pode usar o procedimento `UTL_FILE.FREMOVE` do Oracle para remover arquivos de um diretório. Para ter mais informações, consulte Procedimento FREMOVE na documentação da Oracle.

Monitorar o status de uma transferência de arquivo

Tarefas de transferência de arquivo de publicar eventos do Amazon RDS quando eles começam e quando são concluídos. A mensagem do evento contém o ID da tarefa para a transferência de arquivos. Para obter informações sobre como visualizar eventos, consulte Visualizar eventos do Amazon RDS.

Você pode visualizar o status de uma tarefa em andamento em um arquivo bdump. Os arquivos bdump estão localizados no diretório `/rdsdbdata/log/trace`. Cada nome de arquivo bdump está no seguinte formato.

```
dbtask-task-id.log
```

Substitua *`task-id`* pelo ID da tarefa que você deseja monitorar.

> **Note**
>
> As tarefas são executadas de forma assíncrona.

Você pode usar o procedimento armazenado `rdsadmin.rds_file_util.read_text_file` para ver o conteúdo dos arquivos bdump. Por exemplo, a consulta a seguir retorna o conteúdo do arquivo bdump *`dbtask-1234567890123-1234.log`*.

```
SELECT text FROM
 table(rdsadmin.rds_file_util.read_text_file('BDUMP','dbtask-1234567890123-1234.log'));
```

O exemplo a seguir mostra o arquivo de log de uma falha na transferência.

```
TASK_ID




                                                
```

Solução de problemas de integração do Amazon S3

Para obter dicas de solução de problemas, consulte o artigo do AWS re:Post Como solucionar problemas ao integrar o Amazon RDS para Oracle ao Amazon S3?.

Remover a opção de integração do Amazon S3

Você pode remover a opção de integração do Amazon S3 de uma instância de banco de dados.

Para remover a opção de integração do Amazon S3 de uma instância de banco de dados, faça o seguinte:

- Para remover a opção de integração do Amazon S3 de várias instâncias de bancos de dados, remova a opção `S3_INTEGRATION` do grupo de opções ao qual as instâncias de banco de dados pertencem. Essa alteração afeta todas as instâncias de bancos de dados que usam o grupo de opções. Para ter mais informações, consulteRemover uma opção de um grupo de opções
- Para remover a opção de integração do Amazon S3 de uma única instância de banco de dados, modifique a instância e especifique um grupo de opções diferente que não inclua a opção `S3_INTEGRATION`. Você pode especificar um grupo de opções padrão (vazio) ou criar um grupo de opções personalizado diferente. Para ter mais informações, consulteModificar uma instância de banco de dados do Amazon RDS

Oracle Application Express (APEX)

O Amazon RDS é compatível com o Oracle Application Express (APEX) com o uso das opções `APEX` e `APEX-DEV`. É possível implantar o Oracle APEX como um ambiente de tempo de execução ou como um ambiente de desenvolvimento completo para aplicações baseadas na web. Com o Oracle APEX, você pode criar aplicações inteiramente no navegador da web. Para ter mais informações, consulte Oracle Application Express na documentação da Oracle.

Tópicos

- Componentes do Oracle APEX
- Requisitos e limitações
- Configurar o Oracle APEX e o Oracle Rest Data Services (ORDS)
- Configurar o Oracle Rest Data Services (ORDS)
- Atualizar e remover o Oracle APEX

Componentes do Oracle APEX

O Oracle APEX consiste nos seguintes componentes principais:

- Um repositório que armazena metadados para aplicações e componentes do APEX. O repositório consiste em tabelas, índices e outros objetos instalados na sua instância de banco de dados do Amazon RDS.
- Um ouvinte que gerencia as comunicações HTTP com clientes Oracle APEX. O receptor reside em um host separado como uma instância do Amazon EC2, um servidor on-premises na sua empresa ou no seu computador desktop. O ouvinte aceita as conexões de entrada que os navegadores da web enviam à instância de banco de dados do Amazon RDS para fins de processamento. Em seguida, ele envia os resultados do repositório de volta para os navegadores.

  O RDS for Oracle é compatível com os seguintes tipos de receptores:

  - Para o Oracle APEX versão 5.0 e posterior, use o Oracle REST Data Services (ORDS) versão 19.1 e posterior. Recomendamos que você use a versão mais recente compatível do Oracle APEX e do ORDS. Essa documentação descreve versões mais antigas apenas para compatibilidade com versões anteriores.
  - Para o Oracle APEX versão 4.1.1, você pode usar o Oracle APEX Listener versão 1.1.4.
  - Você pode usar o Oracle HTTP Server e listeners `mod_plsql`.

> **Note**
>
> O Amazon RDS não é compatível com o servidor HTTP do banco de dados Oracle XML com o gateway PL/SQL incorporado como um listener do Oracle APEX. Em geral, a Oracle não recomenda o uso do gateway PL/SQL embutido para aplicativos executados na internet.

Para ter mais informações sobre esses tipos de listener, consulte About Choosing a Web Listener (Sobre escolher um listener da Web) na documentação do Oracle.

Ao adicionar as opções `APEX` e `APEX-DEV` à instância de banco de dados do RDS para Oracle, o Amazon RDS instala apenas o repositório do Oracle APEX. Instale seu receptor em um host separado.

Requisitos e limitações

O tópico a seguir lista os requisitos e limitações para o Oracle APEX e o ORDS.

Requisitos de versão do Oracle APEX

A opção `APEX` usa armazenamento na classe de instância de banco de dados para sua instância de banco de dados. A seguir estão as versões compatíveis e os requisitos aproximados de armazenamento para o Oracle APEX:

Versão do Oracle APEX	Requisitos de armazenamento	Versões compatíveis do Oracle Database	Observações
Oracle APEX versão 24.2.v1	114 MiB	Todos	Esta versão inclui o patch 37885097: PACOTE DO PSE PARA APEX 24.2 (PSES SOBRE 24.2.0), PATCH_VERSION 4.
Oracle APEX versão 24.1.v1	112 MiB	Todos	Esta versão inclui o patch 36695709: PSE BUNDLE FOR APEX 24.1 (PSES ON TOP OF 24.1.0), PATCH_VERSION 3. Se você precisar instalar exatamente a mesma versão de imagens

Versão do Oracle APEX	Requisitos de armazenamento	Versões compatíveis do Oracle Database	Observações
			do APEX em sua instância do EC2, baixe o Patch 37544819: 24.1.3 PSE BUNDLE FOR APEX 24.1 (PSES ON TOP OF 24.1.0).
Oracle APEX versão 23.2.v1	110 MiB	Todos	Esta versão inclui o patch 35895964: PACOTE DE PSEs PARA APEX 23.2 (PSEs ACIMA DE 23.2.0), PATCH_VERSION 6. Se você precisar instalar exatamente a mesma versão de imagens do APEX em sua instância do EC2, baixe o Patch 37593125: 23.2.6 PSE BUNDLE FOR APEX 23.2 (PSES ON TOP OF 23.2.0).
Oracle APEX versão 23.1.v1	106 MiB	Todos	Esta versão inclui o patch 35283657: PSE BUNDLE FOR APEX 23.1 (PSES ACIMA DE 23.1.0), PATCH_VERSION 2.
Oracle APEX versão 22.2.v1	106 MiB	Todos	Esta versão inclui o patch 34628174: PACOTE DO PSE PARA APEX 22.2 (PSES SOBRE 22.2.0), PATCH_VERSION 4.
Oracle APEX versão 22.1.v1	124 MiB	Todos	Esta versão inclui o patch 34020981: PACOTE DO PSE PARA APEX 22.1 (PSES SOBRE 22.1.0), PATCH_VERSION 6.
Oracle APEX versão 21.2.v1	125 MiB	Todos	Esta versão inclui o patch 33420059: PACOTE DO PSE PARA APEX 21.2 (PSES SOBRE 21.2.0), PATCH_VERSION 8.
Oracle APEX versão 21.1.v1	125 MiB	Todos	Esta versão inclui o patch 32598392: PACOTE PSE PARA APEX 21.1, PATCH_VERSÃO 3.

Versão do Oracle APEX	Requisitos de armazenamento	Versões compatíveis do Oracle Database	Observações
Oracle APEX versão 20.2.v1	148 MiB	Todas, exceto Oracle Database 21c	Esta versão inclui o patch 32006852: PACOTE PSE PARA APEX 20.2, PATCH_VERSÃO 2020.11.12. Você pode ver o número e a data do patch executando a seguinte consulta: `SELECT PATCH_VERSION, PATCH_NUMBER FROM APEX_PATCHES;`
Oracle APEX versão 20.1.v1	173 MiB	Todas, exceto Oracle Database 21c	Esta versão inclui o patch 30990551: PACOTE PSE PARA APEX 20.1, PATCH_VERSÃO 2020.07.15.
Oracle APEX versão 19.2.v1	149 MiB	Todas, exceto Oracle Database 21c	
Oracle APEX versão 19.1.v1	148 MiB	Todas, exceto Oracle Database 21c	

Com relação a arquivos .zip do Oracle APEX que podem ser baixados, consulte Oracle APEX Prior Release Archives no site da Oracle.

Pré-requisitos para Oracle APEX e ORDS

Observe os seguintes pré-requisitos para usar o Oracle APEX e o ORDS:

- O sistema precisa usar o Ambiente de Execução Java (JRE).
- Uma instalação de cliente Oracle deve incluir o seguinte:
  - SQL*Plus ou SQL Developer para tarefas de administração
  - Oracle Net Services, para configurar conexões com a instância de banco de dados do RDS para Oracle

Limitações do Oracle APEX

Você não pode modificar a conta de usuário `APEX_version`, que é gerenciada pelo Amazon RDS. Portanto, você não pode aplicar perfis de banco de dados nem impor regras de senha a esse usuário. Os perfis e as configurações de senha para `APEX_version` são predefinidos pela Oracle e pela AWS e projetados para satisfazer os requisitos de segurança do Amazon RDS.

Configurar o Oracle APEX e o Oracle Rest Data Services (ORDS)

O tópico a seguir lista as etapas necessárias para configurar o Oracle APEX e o ORDS.

Tópicos

- Adicionar as opções APEX e APEX-DEV à sua instância de banco de dados
- Desbloquear a conta de usuário público em sua instância de banco de dados
- Configuração de serviços RESTful para o Oracle APEX
- Preparar-se para instalar o ORDS em um host separado
- Configurar o Oracle APEX Listener

Adicionar as opções APEX e APEX-DEV à sua instância de banco de dados

Para adicionar as opções `APEX` e `APEX-DEV` à instância de banco de dados do RDS para Oracle, faça o seguinte:

1. Crie um novo grupo de opções, ou copie ou modifique um existente.
2. Adicione as opções `APEX` e `APEX-DEV` ao grupo de opções.
3. Associe o grupo de opções à instância de banco de dados.

Quando você adiciona as opções `APEX` e `APEX-DEV`, ocorre uma breve interrupção enquanto a instância de banco de dados é reiniciada automaticamente.

> **Note**
>
> `APEX_MAIL` está disponível quando a opção `APEX` está instalada. O privilégio de execução referente ao pacote `APEX_MAIL` é concedido a `PUBLIC`. Portanto, você não precisa da conta administrativa do APEX para usá-lo.

Como adicionar as opções APEX e APEX-DEV à instância de banco de dados

1. Determine o grupo de opções que você deseja usar. Você pode criar um novo grupo de opções ou usar um existente. Se você quiser usar um grupo de opções existente, vá para a próxima etapa. Caso contrário, crie um grupo de opções de banco de dados personalizado com as seguintes configurações:

   a. Para Engine (Mecanismo), escolha a edição do Oracle que deseja usar. As opções `APEX` e `APEX-DEV` podem usadas em todas as edições.

   b. Para Major engine version (Versão do mecanismo principal), escolha a versão da sua instância de banco de dados.

   Para ter mais informações, consulte Criar um grupo de opções.

2. Adicione opções ao grupo de opções. Se você quiser implantar somente o ambiente de tempo de execução do Oracle APEX, adicione apenas a opção `APEX`. Para implantar o ambiente de desenvolvimento completo, adicione as opções `APEX` e `APEX-DEV`.

   Em Versão, escolha a versão do Oracle APEX que você deseja usar.

   > **Important**
   >
   > Se você adicionar as opções `APEX` e `APEX-DEV` a um grupo de opções existente que já esteja anexado a uma ou mais instâncias de banco de dados, ocorrerá uma breve interrupção. Durante essa interrupção, todas as instâncias de banco de dados são automaticamente reiniciadas.

   Para mais informações sobre a adição de opções, consulte Adicionar uma opção a um grupo de opções.

3. Aplique o grupo de opções a uma instância de banco de dados nova ou existente:

- Para uma nova instância de banco de dados, você aplica o grupo de opções ao executar a instância. Para ter mais informações, consulte Criar uma instância de banco de dados do Amazon RDS.
- Para uma instância de banco de dados existente, você aplica o grupo de opções modificando a instância e anexando o novo grupo de opções. Quando você adiciona as opções `APEX` e `APEX-DEV` a uma instância de banco de dados existente, ocorre uma breve interrupção enquanto a instância de banco de dados é reiniciada automaticamente. Para obter mais informações, consulte Modificar uma instância de banco de dados do Amazon RDS.

Desbloquear a conta de usuário público em sua instância de banco de dados

Depois que as opções `APEX` e `APEX-DEV` da instância de banco de dados forem instaladas, faça o seguinte:

1. Altere a senha para a conta `APEX_PUBLIC_USER`.
2. Desbloqueie a conta.

Você pode fazer isso usando o utilitário de linha de comando Oracle SQL*Plus. Conecte-se à sua instância de banco de dados como usuário principal e emita os comandos a seguir. Substitua `new_password` por uma senha da sua escolha.

```
ALTER USER APEX_PUBLIC_USER IDENTIFIED BY new_password;
ALTER USER APEX_PUBLIC_USER ACCOUNT UNLOCK;
```

Configuração de serviços RESTful para o Oracle APEX

Para configurar serviços RESTful no Oracle APEX (não necessário para o Oracle APEX 4.1.1.V1), use o SQL*Plus para se conectar à instância de banco de dados como usuário principal. Depois que você fizer isso, execute o procedimento armazenado `rdsadmin.rdsadmin_run_apex_rest_config`. Ao executar o procedimento armazenado, você fornece senhas para os seguintes usuários:

- `APEX_LISTENER`
- `APEX_REST_PUBLIC_USER`

O procedimento armazenado executa o script `apex_rest_config.sql` que cria novas contas de banco de dados para esses usuários.

> **Note**
>
> A configuração não é obrigatória para o Oracle APEX versão 4.1.1.v1. Apenas para esta versão Oracle APEX, você não precisa executar o procedimento armazenado.

O comando a seguir executa o procedimento armazenado.

```
EXEC rdsadmin.rdsadmin_run_apex_rest_config('apex_listener_password',
 'apex_rest_public_user_password');
```

Preparar-se para instalar o ORDS em um host separado

Instale o ORDS em um host separado, como uma instância do Amazon EC2 em um servidor on-premises na sua empresa ou no seu computador desktop. Nos exemplos desta seção, suponha que o host execute Linux e tenha o nome `myapexhost.example.com`.

Antes de instalar o ORDS, é necessário criar um usuário de SO não privilegiado e, depois, baixar e descompactar o arquivo de instalação do Oracle APEX.

Como se preparar para a instalação do ORDS

1. Faça login em `myapexhost.example.com` como `root`.
2. Crie um usuário de SO sem privilégios para ser o proprietário da instalação do listener. O comando a seguir cria um novo usuário chamado apexuser.

   ```
   useradd -d /home/apexuser apexuser
   ```

   O comando a seguir atribui uma senha ao novo usuário.

   ```
   passwd apexuser;
   ```

3. Faça login em `myapexhost.example.com` como `apexuser` e baixe o arquivo de instalação do Oracle APEX no Oracle no diretório `/home/apexuser`:

   - http://www.oracle.com/technetwork/developer-tools/apex/downloads/index.html

- Oracle application Express prior release archives

4. Descompacte o arquivo no diretório `/home/apexuser`.

```
unzip apex_version.zip
```

Depois que você descompactar o arquivo, haverá um diretório `apex` no diretório `/home/apexuser`.

5. Enquanto você ainda está conectado no `myapexhost.example.com` como `apexuser`, baixe o arquivo do Oracle REST Data Services no diretório `/home/apexuser`: http://www.oracle.com/technetwork/developer-tools/apex-listener/downloads/index.html.

Configurar o Oracle APEX Listener

> **Note**
>
> O Oracle APEX Listener está defasado.

O Amazon RDS para Oracle continua a oferecer suporte ao Oracle APEX versão 4.1.1 e ao Oracle APEX Listener versão 1.1.4. Recomendamos que você use as versões com suporte mais recentes do Oracle APEX e do ORDS.

Instale o Oracle APEX Listener em um host separado, como uma instância do Amazon EC2, um servidor local na sua empresa ou seu computador desktop. Supomos que o nome do host é `myapexhost.example.com` e o seu host está sendo executado em Linux.

Preparar-se para instalar o Oracle APEX Listener

Antes de instalar o Oracle APEX Listener, é necessário criar um usuário de SO não privilegiado e, depois, baixar e descompactar o arquivo de instalação do Oracle APEX.

Como se preparar para a instalação do Oracle APEX Listener

1. Faça login em `myapexhost.example.com` como `root`.
2. Crie um usuário de SO sem privilégios para ser o proprietário da instalação do listener. O comando a seguir cria um novo usuário chamado apexuser.

```
useradd -d /home/apexuser apexuser
```

O comando a seguir atribui uma senha ao novo usuário.

```
passwd apexuser;
```

3. Faça login em `myapexhost.example.com` como `apexuser` e baixe o arquivo de instalação do Oracle APEX no Oracle no diretório `/home/apexuser`:

   - http://www.oracle.com/technetwork/developer-tools/apex/downloads/index.html
   - Oracle application Express prior release archives

4. Descompacte o arquivo no diretório `/home/apexuser`.

   ```
   unzip apex_<version>.zip
   ```

   Depois que você descompactar o arquivo, haverá um diretório `apex` no diretório `/home/apexuser`.

5. Enquanto ainda estiver conectado ao `myapexhost.example.com` como `apexuser`, baixe o arquivo do Oracle APEX Listener no diretório `/home/apexuser`.

Instalar e configurar o Oracle APEX Listener

Antes de usar o Oracle APEX, é necessário baixar o arquivo `apex.war`, usar o Java para instalar o Oracle APEX Listener e iniciar o Listener.

Como instalar e configurar o Oracle APEX Listener

1. Crie um diretório baseado no Oracle APEX Listener e abra o arquivo do listener.

   Execute o seguinte código:

   ```
   mkdir /home/apexuser/apexlistener
   cd /home/apexuser/apexlistener
   unzip ../apex_listener.version.zip
   ```

2. Execute o código a seguir.

   ```
   java -Dapex.home=./apex -Dapex.images=/home/apexuser/apex/images -Dapex.erase -
   jar ./apex.war
   ```

3. Insira as seguintes informações para os prompts do programa:

- O nome de usuário do administrador do APEX Listener. O padrão é adminlistener.
- A senha para o administrador do APEX Listener.
- O nome de usuário do gerente do APEX Listener. O padrão é managerlistener.
- A senha para o administrador do APEX Listener.

O programa imprime um URL necessário para concluir a configuração, conforme mostrado a seguir:

```
INFO: Please complete configuration at: http://localhost:8080/apex/
listenerConfigure
Database is not yet configured
```

4. Deixe o Oracle APEX Listener em execução para que você possa usar o Oracle Application Express. Quando você concluir o procedimento de configuração, poderá executar o ouvinte em segundo plano.
5. No navegador, acesse o URL fornecido pelo programa Oracle APEX Listener. A janela de administração do Oracle Application Express Listener é exibida. Insira as seguintes informações:
   - Nome de usuário – `APEX_PUBLIC_USER`
   - Senha – a senha de APEX_PUBLIC_USER. Essa é a senha que você especificou anteriormente quando configurou o repositório do Oracle APEX. Para obter mais informações, consulte Desbloquear a conta de usuário público em sua instância de banco de dados.
   - Tipo de conexão – básica
   - Nome de host – o endpoint da instância de banco de dados do Amazon RDS, como `mydb.f9rbfa893tft.us-east-1.rds.amazonaws.com`.
   - Porta – 1521
   - SID – o nome do banco de dados na sua instância de banco de dados do Amazon RDS, como `mydb`.
6. Escolha Apply (Aplicar). A janela de administração do Oracle APEX é exibida.
7. Defina uma senha para o usuário `admin` do Oracle APEX. Para fazer isso, use o SQL*Plus para conectar à sua instância de banco de dados como o usuário principal e execute os comandos a seguir.

```
EXEC rdsadmin.rdsadmin_util.grant_apex_admin_role;
grant APEX_ADMINISTRATOR_ROLE to master;
```

```
@/home/apexuser/apex/apxchpwd.sql
```

Substitua *master* pelo seu nome de usuário principal. Quando o script `apxchpwd.sql` solicitar, insira uma nova senha `admin`.

8. Volte para a janela de administração do Oracle APEX no navegador e escolha Administração. Em seguida, escolha Application Express Internal Administration (Administração interna do Application Express). Quando forem solicitadas as credenciais, insira as seguintes informações:

   - Nome de usuário – `admin`
   - Senha – a senha configurada com o script `apxchpwd.sql`

   Escolha Login (Fazer login) e, em seguida, defina uma nova senha para o usuário `admin`.

Agora o listener está pronto para ser usado.

Configurar o Oracle Rest Data Services (ORDS)

O tópico a seguir lista as opções de configuração para o ORDS 21 e 22:

Tópicos

- Instalar e configurar o ORDS 21 e anterior
- Instalar e configurar o ORDS 22 e posterior

Instalar e configurar o ORDS 21 e anterior

Agora você está pronto para instalar e configurar o Oracle Rest Data Services (ORDS) para uso com o Oracle APEX. Em relação ao Oracle APEX versão 5.0 e posterior, use o ORDS versões 19.1 a 21. Para saber como instalar o ORDS 22 e posterior, consulte Instalar e configurar o ORDS 22 e posterior.

Instale o listener em um host separado como uma instância do Amazon EC2 em um servidor no local na sua empresa ou no seu computador desktop. Para os exemplos desta seção, vamos supor que o nome do host seja `myapexhost.example.com` e que o host esteja executando o Linux.

Como instalar e configurar o ORDS 21 e anterior para usar com o Oracle APEX

1. Acesse Oracle REST Data Services e examine o Readme. Verifique se você tem a versão necessária do Java instalada.

2. Crie um diretório para a instalação do ORDS.

```
mkdir /home/apexuser/ORDS
cd /home/apexuser/ORDS
```

3. Baixe o arquivo `ords.version.number.zip` de Serviços de dados Oracle REST.
4. Descompacte o arquivo no diretório `/home/apexuser/ORDS`.
5. Se você estiver instalando ORDS em um banco de dados multilocativo, adicione a seguinte linha ao arquivo `/home/apexuser/ORDS/params/ords_params.properties`:

```
pdb.disable.lockdown=false
```

6. Conceda ao usuário mestre os privilégios necessários para instalar o ORDS.

   Depois que as opções para o Oracle APEX forem instaladas, conceda ao usuário principal os privilégios necessários para instalar o esquema do ORDS. É possível fazer isso conectando-se ao banco de dados e executando os comandos a seguir. Substitua *MASTER_USER* pelo nome do seu usuário mestre em letras maiúsculas.

   > **Important**
   >
   > Ao inserir o nome de usuário, use letras maiúsculas, a menos que você tenha criado o usuário com um identificador que diferencia letras maiúsculas de minúsculas. Por exemplo, se você executar `CREATE USER myuser` ou `CREATE USER MYUSER`, o dicionário de dados armazenará `MYUSER`. No entanto, se você usar aspas duplas em `CREATE USER "MyUser"`, o dicionário de dados armazenará `MyUser`. Para ter mais informações, consulte Concessão de privilégios SELECT ou EXECUTE a objetos SYS.

```
exec rdsadmin.rdsadmin_util.grant_sys_object('DBA_OBJECTS', 'MASTER_USER',
 'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('DBA_ROLE_PRIVS', 'MASTER_USER',
 'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('DBA_TAB_COLUMNS', 'MASTER_USER',
 'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('USER_CONS_COLUMNS', 'MASTER_USER',
 'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('USER_CONSTRAINTS', 'MASTER_USER',
 'SELECT', true);
```

```
exec rdsadmin.rdsadmin_util.grant_sys_object('USER_OBJECTS', 'MASTER_USER',
 'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('USER_PROCEDURES', 'MASTER_USER',
 'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('USER_TAB_COLUMNS', 'MASTER_USER',
 'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('USER_TABLES', 'MASTER_USER',
 'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('USER_VIEWS', 'MASTER_USER', 'SELECT',
 true);
exec rdsadmin.rdsadmin_util.grant_sys_object('WPIUTL', 'MASTER_USER', 'EXECUTE',
 true);
exec rdsadmin.rdsadmin_util.grant_sys_object('DBMS_SESSION', 'MASTER_USER',
 'EXECUTE', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('DBMS_UTILITY', 'MASTER_USER',
 'EXECUTE', true);
```

> **Note**
>
> Esses comandos se aplicam ao ORDS versão 19.1 e posterior.

7. Instale o esquema do ORDS usando o arquivo ords.war baixado.

```
java -jar ords.war install advanced
```

O prompt do programa solicitará as informações a seguir. Os valores padrão estão entre colchetes. Para ter mais informações, consulte o tópico de Introdução ao Oracle REST Data Services, na documentação da Oracle.

- Insira o local para armazenar dados de configuração:

  Insira */home/apexuser/ORDS*. Este é o local dos arquivos de configuração do ORDS.

- Especifique o tipo de conexão do banco de dados a ser usado. Digite o número para [1] Basic [2] TNS [3] URL personalizado [1]:

  Escolha o tipo de conexão desejada.

- Insira o nome do servidor de banco de dados [localhost]: *DB_instance_endpoint*

  Escolha o padrão ou insira o valor correto.

- Insira a porta do listener do banco de dados [1521]: *db_instance_port*

Escolha o padrão ou insira o valor correto.

- Digite 1 para especificar o nome de serviço do banco de dados ou 2 para especificar o SID do banco de dados [1]:

  Escolha 2 para especificar o SID do banco de dados.

- SID do banco de dados [xe]

  Escolha o padrão ou insira o valor correto.

- Digite 1 se desejar verificar/instalar o esquema do Oracle REST Data Services ou 2 para pular esta etapa [1]:

  Selecione 1. Esta etapa cria o usuário proxy do Oracle REST Data Services chamado ORDS_PUBLIC_USER.

- Digite a senha do banco de dados para ORDS_PUBLIC_USER:

  Digite a senha e confirme-a.

- Requer login com privilégios de administrador para verificar o esquema do Oracle REST Data Services.

  Insira o nome do usuário administrador: *master_user*

  Insira a senha do banco de dados para *master_user*: *master_user_password*

  Confirme a senha: *master_user_password*

  > **Note**
  >
  > Especifique uma senha diferente do prompt mostrado aqui como prática recomendada de segurança.

- Insira o espaço de tabela padrão para ORDS_METADATA [SYSAUX].

  Insira o espaço de tabela temporário para ORDS_METADATA [TEMP].

  Insira o espaço de tabela padrão para ORDS_PUBLIC_USER [USERS].

  Insira o espaço de tabela temporário para ORDS_PUBLIC_USER [TEMP].

- Insira 1 se quiser usar o gateway PL/SQL ou 2 para ignorar essa etapa. Se você está usando o Oracle Application Express ou migrando do mod_plsql, deve inserir 1 [1].

  Escolha o padrão.

- Insira o nome de usuário de banco de dados do gateway PL/SQL [APEX_PUBLIC_USER]

  Escolha o padrão.

- Digite a senha do banco de dados para APEX_PUBLIC_USER:

  Digite a senha e confirme-a.

- Digite 1 para especificar senhas para usuários de banco de dados de serviços RESTful do Application Express (APEX_LISTENER, APEX_REST_PUBLIC_USER) ou 2 para pular esta etapa [1]:

  Escolha 2 para APEX 4.1.1.V1; escolha 1 para todas as outras versões de APEX.

- [Não necessário para o APEX 4.1.1.v1] Senha de banco de dados para APEX_LISTENER

  Digite a senha (se necessário) e confirme-a.

- [Não necessário para o APEX 4.1.1.v1] Senha de banco de dados para APEX_REST_PUBLIC_USER

  Digite a senha (se necessário) e confirme-a.

- Digite um número para selecionar um recurso a ser ativado:

  Digite 1 para habilitar todos os recursos: SQL Developer Web, SQL habilitado para REST e API de banco de dados.

- Digite 1 se desejar iniciar no modo independente ou 2 para sair [1]:

  Digite 1.

- Insira o local dos recursos estáticos do APEX:

  Se você descompactou os arquivos de instalação do APEX em `/home/apexuser`, insira `/home/apexuser/apex/images`. Caso contrário, digite *`unzip_path`*`/apex/images`, onde *`unzip_path`* é o diretório onde você descompactou o arquivo.

- Digite 1 se estiver usando HTTP ou 2 se estiver usando HTTPS [1]:

Se você digitar 1, especifique a porta HTTP. Se você digitar 2, especifique a porta HTTPS e o nome do host SSL. A opção HTTPS solicita que você especifique como fornecerá o certificado:

- Digite 1 para usar o certificado autoassinado.
- Digite 2 para fornecer seu próprio certificado. Se você inserir 2, especifique o caminho para o certificado SSL e o caminho para a chave privada do certificado SSL.

8. Defina uma senha para o usuário `admin` do APEX. Para fazer isso, use o SQL*Plus para conectar à sua instância de banco de dados como o usuário principal e execute os comandos a seguir.

```
EXEC rdsadmin.rdsadmin_util.grant_apex_admin_role;
grant APEX_ADMINISTRATOR_ROLE to master;
@/home/apexuser/apex/apxchpwd.sql
```

Substitua *`master`* pelo seu nome de usuário principal. Quando o script `apxchpwd.sql` solicitar, insira uma nova senha `admin`.

9. Inicie o listener do ORDS. Execute o código a seguir.

```
java -jar ords.war
```

Na primeira vez que iniciar o ORDS, você será solicitado a fornecer a localização dos recursos estáticos do APEX. Essa pasta de imagens está localizada no diretório `/apex/images` do diretório de instalação do APEX.

10. Volte para a janela de administração do Oracle APEX no navegador e escolha Administração. Em seguida, escolha Application Express Internal Administration (Administração interna do Application Express). Quando forem solicitadas as credenciais, insira as seguintes informações:

- Nome de usuário – `admin`
- Senha – a senha configurada com o script `apxchpwd.sql`

Escolha Login (Fazer login) e, em seguida, defina uma nova senha para o usuário `admin`.

Agora o listener está pronto para ser usado.

Instalar e configurar o ORDS 22 e posterior

Agora você está pronto para instalar e configurar o Oracle Rest Data Services (ORDS) para uso com o Oracle APEX. Para os exemplos desta seção, vamos supor que o nome do host separado seja `myapexhost.example.com` e que o host esteja executando o Linux. As instruções do ORDS 22 são diferentes das instruções das versões anteriores.

Como instalar e configurar o ORDS 22 e posterior para usar com o Oracle APEX

1. Acesse Oracle REST Data Services e examine o Readme da versão do ORDS que você planeja baixar. Verifique se você tem a versão necessária do Java instalada.
2. Crie um diretório para a instalação do ORDS.

   ```
   mkdir /home/apexuser/ORDS
   cd /home/apexuser/ORDS
   ```

3. Baixe o arquivo `ords.version.number.zip` ou `ords-latest.zip` em Oracle REST Data Services.
4. Descompacte o arquivo no diretório `/home/apexuser/ORDS`.
5. Conceda ao usuário mestre os privilégios necessários para instalar o ORDS.

   Depois que a opção APEX for instalada, conceda ao usuário principal os privilégios necessários para instalar o esquema do ORDS. É possível fazer isso fazendo login no banco de dados e executando os comandos a seguir. Substitua *MASTER_USER* pelo nome do seu usuário mestre em letras maiúsculas.

   > **Important**
   >
   > Ao inserir o nome de usuário, use letras maiúsculas, a menos que você tenha criado o usuário com um identificador que diferencia letras maiúsculas de minúsculas. Por exemplo, se você executar `CREATE USER myuser` ou `CREATE USER MYUSER`, o dicionário de dados armazenará `MYUSER`. No entanto, se você usar aspas duplas em `CREATE USER "MyUser"`, o dicionário de dados armazenará `MyUser`. Para obter mais informações, consulte Concessão de privilégios SELECT ou EXECUTE a objetos SYS.

   ```
   exec rdsadmin.rdsadmin_util.grant_sys_object('DBA_OBJECTS', 'MASTER_USER',
    'SELECT', true);
   ```

```
exec rdsadmin.rdsadmin_util.grant_sys_object('DBA_ROLE_PRIVS', 'MASTER_USER',
 'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('DBA_TAB_COLUMNS', 'MASTER_USER',
 'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('USER_CONS_COLUMNS', 'MASTER_USER',
 'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('USER_CONSTRAINTS', 'MASTER_USER',
 'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('USER_OBJECTS', 'MASTER_USER',
 'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('USER_PROCEDURES', 'MASTER_USER',
 'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('USER_TAB_COLUMNS', 'MASTER_USER',
 'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('USER_TABLES', 'MASTER_USER',
 'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('USER_VIEWS', 'MASTER_USER', 'SELECT',
 true);
exec rdsadmin.rdsadmin_util.grant_sys_object('WPIUTL', 'MASTER_USER', 'EXECUTE',
 true);
exec rdsadmin.rdsadmin_util.grant_sys_object('DBMS_SESSION', 'MASTER_USER',
 'EXECUTE', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('DBMS_UTILITY', 'MASTER_USER',
 'EXECUTE', true);

exec rdsadmin.rdsadmin_util.grant_sys_object('DBMS_LOB', 'MASTER_USER', 'EXECUTE',
 true);
exec rdsadmin.rdsadmin_util.grant_sys_object('DBMS_ASSERT', 'MASTER_USER',
 'EXECUTE', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('DBMS_OUTPUT', 'MASTER_USER',
 'EXECUTE', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('DBMS_SCHEDULER', 'MASTER_USER',
 'EXECUTE', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('HTP', 'MASTER_USER', 'EXECUTE',
 true);
exec rdsadmin.rdsadmin_util.grant_sys_object('OWA', 'MASTER_USER', 'EXECUTE',
 true);
exec rdsadmin.rdsadmin_util.grant_sys_object('WPG_DOCLOAD', 'MASTER_USER',
 'EXECUTE', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('DBMS_CRYPTO', 'MASTER_USER',
 'EXECUTE', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('DBMS_METADATA', 'MASTER_USER',
 'EXECUTE', true);
```

```
exec rdsadmin.rdsadmin_util.grant_sys_object('DBMS_SQL', 'MASTER_USER', 'EXECUTE',
 true);
exec rdsadmin.rdsadmin_util.grant_sys_object('UTL_SMTP', 'MASTER_USER', 'EXECUTE',
 true);
exec rdsadmin.rdsadmin_util.grant_sys_object('DBMS_NETWORK_ACL_ADMIN',
 'MASTER_USER', 'EXECUTE', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('SESSION_PRIVS', 'MASTER_USER',
 'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('DBA_USERS', 'MASTER_USER', 'SELECT',
 true);
exec rdsadmin.rdsadmin_util.grant_sys_object('DBA_NETWORK_ACL_PRIVILEGES',
 'MASTER_USER', 'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('DBA_NETWORK_ACLS', 'MASTER_USER',
 'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('DBA_REGISTRY', 'MASTER_USER',
 'SELECT', true);
```

> **Note**
>
> Os comandos anteriores se aplicam ao ORDS 22 e posterior.

6. Instale o esquema do ORDS usando o script `ords` baixado. Especifique os diretórios para conter arquivos de configuração e arquivos de log. A Oracle Corporation recomenda não colocar esses diretórios dentro do diretório que contém o software do produto ORDS.

```
mkdir -p /home/apexuser/ords_config /home/apexuser/ords_logs

/home/apexuser/ORDS/bin/ords \
  --config /home/apexuser/ords_config \
  install --interactive --log-folder /home/apexuser/ords_logs
```

Para instâncias de banco de dados que executam a arquitetura de banco de dados de contêiner (CDB), use o ORDS 23.2 e posterior e transmita o argumento `--pdb-skip-disable-lockdown` ao instalar o ORDS.

```
/home/apexuser/ORDS/bin/ords \
  --config /home/apexuser/ords_config \
  install --interactive --log-folder /home/apexuser/ords_logs --pdb-skip-disable-
lockdown
```

O prompt do programa solicitará as informações a seguir. Os valores padrão estão entre colchetes. Para ter mais informações, consulte o tópico de Introdução ao Oracle REST Data Services, na documentação da Oracle.

- `Choose the type of installation:`

  Escolha **2** instalar esquemas do ORDS no banco de dados e criar um grupo de conexões de banco de dados nos arquivos de configuração do ORDS local.

- `Specify the database connection type to use. Enter number for [1] Basic [2] TNS [3] Custom URL:`

  Escolha o tipo de conexão desejada. Este exemplo pressupõe que você escolha **1**.

- `Enter the name of the database server [localhost]:` ***`DB_instance_endpoint`***

  Escolha o padrão ou insira o valor correto.

- `Enter the database listener port [1521]:` ***`DB_instance_port`***

  Escolha o padrão **1521** ou insira o valor correto.

- `Enter the database service name [orcl]:`

  Insira o nome do banco de dados usado pela instância de banco de dados do RDS para Oracle.

- `Provide database user name with administrator privileges`

  Digite o nome do usuário mestre da instância de banco de dados do RDS para Oracle.

- `Enter the database password for [username]:`

  Digite a senha do usuário mestre da instância de banco de dados do RDS para Oracle.

- `Enter the default tablespace for ORDS_METADATA and ORDS_PUBLIC_USER [SYSAUX]:`
- `Enter the temporary tablespace for ORDS_METADATA [TEMP]. Enter the default tablespace for ORDS_PUBLIC_USER [USERS]. Enter the temporary tablespace for ORDS_PUBLIC_USER [TEMP].`
- `Enter a number to select additional feature(s) to enable [1]:`
- `Enter a number to configure and start ORDS in standalone mode [1]:`

Escolha **2** para ignorar a inicialização do ORDS imediatamente no modo autônomo.

- `Enter a number to select the protocol [1] HTTP`
- `Enter the HTTP port [8080]:`
- `Enter the APEX static resources location:`

  Insira o caminho para os arquivos de instalação do Oracle APEX (`/home/apexuser/apex/images`).

7. Defina uma senha para o usuário `admin` do Oracle APEX. Para fazer isso, use o SQL*Plus para conectar à sua instância de banco de dados como o usuário principal e execute os comandos a seguir.

```
EXEC rdsadmin.rdsadmin_util.grant_apex_admin_role;
grant APEX_ADMINISTRATOR_ROLE to master;
@/home/apexuser/apex/apxchpwd.sql
```

   Substitua *`master`* pelo seu nome de usuário principal. Quando o script `apxchpwd.sql` solicitar, insira uma nova senha `admin`.

8. Execute o ORDS no modo autônomo usando o script `ords` com o comando `serve`. Em relação a implantações de produção, pense em usar servidores de aplicações Java EE compatíveis, como Apache Tomcat ou Oracle WebLogic Server. Para ter mais informações, consulte Deploying and Monitoring Oracle REST Data Services, na documentação do Oracle Database.

```
/home/apexuser/ORDS/bin/ords \
  --config /home/apexuser/ords_config serve \
  --port 8193 \
  --apex-images /home/apexuser/apex/images
```

   Se o ORDS estiver em execução e não conseguir acessar a instalação do Oracle APEX, você poderá ver o erro a seguir, principalmente em instâncias que não sejam de CDB.

```
The procedure named apex_admin could not be accessed, it may not be declared,
 or the user executing this request may not have been granted execute privilege
 on the procedure, or a function specified by security.requestValidationFunction
 configuration property has prevented access.
```

Para corrigir esse erro, altere a função de validação de solicitação usada pelo ORDS executando o script `ords` com o comando `config`. Por padrão, o ORDS usa o procedimento `ords_util.authorize_plsql_gateway`, que só é compatível com instâncias do CDB. Em relação a instâncias que não sejam de CDB, é possível alterar esse procedimento para o pacote `wwv_flow_epg_include_modules.authorize`. Consulte a documentação do Oracle Database e do Oracle Support para conhecer as práticas recomendadas sobre como configurar a função de validação de solicitações adequada para seu caso de uso.

9. Volte para a janela de administração do Oracle APEX no navegador e escolha Administração. Em seguida, escolha Application Express Internal Administration (Administração interna do Application Express). Quando forem solicitadas as credenciais, insira as seguintes informações:

   - Nome de usuário – `admin`
   - Senha – a senha configurada com o script `apxchpwd.sql`

   Escolha Login (Fazer login) e, em seguida, defina uma nova senha para o usuário `admin`.

Agora o listener está pronto para ser usado.

Atualizar e remover o Oracle APEX

Para atualizar ou remover o Oracle APEX, siga as instruções neste tópico:

Tópicos

- Atualizar a versão do Oracle APEX
- Remover as opções APEX e APEX-DEV

Atualizar a versão do Oracle APEX

> **Important**
>
> Faça backup da instância de banco de dados antes de atualizar o Oracle APEX. Para obter mais informações, consulte Criar um snapshot de banco de dados para uma instância de banco de dados single-AZ para o Amazon RDS e Testar uma atualização do Oracle DB.

Para atualizar o Oracle APEX com sua instância de banco de dados, faça o seguinte:

- Crie um novo grupo de opções para a versão atualizada da instância de banco de dados.
- Adicione as versões atualizadas do `APEX` e `APEX-DEV` ao novo grupo de opções. Inclua todas as outras opções usadas pela instância de banco de dados. Para ter mais informações, consulte Considerações de grupos de opções.
- Quando você atualizar a instância de banco de dados, especifique o novo grupo de opções para a instância de banco de dados atualizada.

Depois que você atualizar a versão do Oracle APEX, o esquema do Oracle APEX da versão anterior ainda poderá estar no banco de dados. Se não precisar mais dele, após a atualização você poderá ignorar o esquema do Oracle APEX anterior de seu banco de dados.

Se você atualizar a versão do Oracle APEX e os serviços RESTful não tiverem sido configurados na versão anterior do Oracle APEX, recomendamos que configure os serviços RESTful. Para obter mais informações, consulte Configuração de serviços RESTful para o Oracle APEX.

Em alguns casos, quando estiver pensando em fazer uma atualização de versão principal da instância de banco de dados, é possível que descubra que está usando uma versão do Oracle APEX que não é compatível com a versão do banco de dados de destino. Nesses casos, você pode atualizar sua versão do Oracle APEX antes de atualizar a instância de banco de dados. Atualizar o Oracle APEX primeiro pode reduzir o tempo necessário para atualizar a instância de banco de dados.

> **Note**
>
> Depois de atualizar o Oracle APEX, instale e configure um receptor para ser usado com a versão atualizada. Para instruções, consulte Configurar o Oracle APEX Listener.

Remover as opções APEX e APEX-DEV

Você pode remover as opções `APEX` e `APEX-DEV` de uma instância de banco de dados. Para remover as opções de uma instância de banco de dados, faça o seguinte:

- Para remover as opções `APEX` e `APEX-DEV` de várias instâncias de bancos de dados, remova as opções do grupo de opções ao qual elas pertencem. Essa alteração afeta todas as instâncias de bancos de dados que usam o grupo de opções. Quando você remove as opções de um grupo de opções existente anexado a várias instâncias de banco de dados, ocorre uma breve interrupção enquanto todas as instâncias do banco de dados são reiniciadas.

Para obter mais informações, consulte Remover uma opção de um grupo de opções.

- Para remover as opções `APEX` e `APEX-DEV` de uma única instância de banco de dados, modifique a instância e especifique um grupo de opções diferente que não inclua essas opções. Você pode especificar um grupo de opções padrão (vazio) ou criar um grupo de opções personalizado diferente. Quando você remove as opções, ocorre uma breve interrupção enquanto a instância de banco de dados é reiniciada automaticamente.

  Para obter mais informações, consulte Modificar uma instância de banco de dados do Amazon RDS.

Quando você remove as opções `APEX` e `APEX-DEV` de uma instância de banco de dados, o esquema do APEX é removido do banco de dados.

Integração do Amazon EFS

O Amazon Elastic File System (Amazon EFS) fornece armazenamento de arquivos sem servidor e totalmente elástico para que você possa compartilhar dados de arquivos sem provisionar ou gerenciar a capacidade e a performance do armazenamento. Com o Amazon EFS, você pode criar um sistema de arquivos e montá-lo em sua VPC por meio do protocolo NFS versões 4.0 e 4.1 (NFSv4). Depois, você pode usar o sistema de arquivos do EFS como qualquer outro sistema de arquivos compatível com POSIX. Para obter informações gerais, consulte O que é o Amazon Elastic File System? e o blog da AWS Integrar o Amazon RDS para Oracle com o Amazon EFS.

Tópicos

- Visão geral da integração com o Amazon EFS
- Configurar permissões de rede para a integração do RDS para Oracle com o Amazon EFS
- Configurar permissões do IAM para a integração do RDS para Oracle com o Amazon EFS
- Adicionar a opção EFS_INTEGRATION
- Configurar permissões do sistema de arquivos Amazon EFS
- Transferir arquivos entre o RDS para Oracle e um sistema de arquivos Amazon EFS
- Remover a opção EFS_INTEGRATION
- Solução de problemas de integração do Amazon EFS

Visão geral da integração com o Amazon EFS

Com o Amazon EFS, é possível transferir arquivos entre uma instância de banco de dados do RDS para Oracle e um sistema de arquivos do EFS. Por exemplo, você pode usar o EFS para oferecer suporte aos seguintes casos de uso:

- Compartilhar um sistema de arquivos entre aplicações e vários servidores de banco de dados.
- Criar um diretório compartilhado para arquivos relacionados à migração, incluindo arquivos de dados de espaços de tabela transportáveis. Para ter mais informações, consulte Migrar usando espaços de tabela transportáveis da Oracle.
- Armazenar e compartilhar arquivos de log de restauração arquivados sem alocar espaço de armazenamento adicional no servidor.
- Usar utilitários do Oracle Database, como o `UTL_FILE`, para ler e gravar arquivos.

Vantagens da integração com o Amazon EFS

Ao escolher um sistema de arquivos do EFS em vez de soluções alternativas de transferência de dados, você recebe os seguintes benefícios:

- É possível transferir arquivos do Oracle Data Pump entre o Amazon EFS e uma instância de banco de dados do RDS para Oracle. Você não precisa copiar esses arquivos localmente porque o Data Pump importa diretamente do sistema de arquivos EFS. Para ter mais informações, consulte Importar dados para o Oracle no Amazon RDS.
- A migração de dados é mais rápida do que ao usar um link de banco de dados.
- Você evita alocar espaço de armazenamento em sua instância de banco de dados do RDS para Oracle a fim de manter os arquivos.
- Um sistema de arquivos do EFS pode ajustar automaticamente a escala do armazenamento sem exigir que você o provisione.
- A integração com o Amazon EFS não tem taxas mínimas nem custos de configuração. Você paga somente pelo que usar.
- A integração do Amazon EFS comporta duas formas de criptografia: a criptografia de dados em trânsito e a criptografia em repouso. A criptografia de dados em trânsito é habilitada por padrão usando o TLS versão 1.2. É possível ativar a criptografia de dados em repouso ao criar um sistema de arquivos do Amazon EFS. Para obter mais informações, consulte Encrypting data at rest (Criptografia de dados em repouso) no Guia do usuário do Amazon Elastic File System.

Requisitos para a integração com o Amazon EFS

Certifique-se de cumprir os seguintes requisitos:

- Seu banco de dados deve executar o banco de dados versão 19.0.0.0.ru-2022-07.rur-2022-07.r1 ou posterior.
- A instância de banco de dados e o sistema de arquivos do EFS devem estar na mesma Região da AWS, VPC e Conta da AWS. O RDS para Oracle não permite o acesso entre contas e regiões do EFS.
- A VPC deve ter a Resolução DNS e os Nomes de host DNS habilitados. Para ter mais informações, consulte Atributos de DNS para sua VPC no Guia do usuário da Amazon Virtual Private Cloud.
- O sistema de arquivos EFS deve usar a classe de armazenamento Standard ou Standard-IA.

- Se você usar um nome do DNS no comando `mount`, a VPC deverá estar configurada para usar o servidor DNS fornecido pela Amazon. Os servidores DNS personalizados não são compatíveis.
- Você deve usar soluções que não são do RDS para fazer backup do sistema de arquivos do EFS. O RDS para Oracle não é compatível com backups automatizados nem snapshots manuais de banco de dados de um sistema de arquivos EFS. Para ter mais informações, consulte Backing up your Amazon EFS file systems (Fazer backup de seus sistemas de arquivos Amazon EFS).

Configurar permissões de rede para a integração do RDS para Oracle com o Amazon EFS

Para que o RDS para Oracle se integre ao Amazon EFS, sua instância de banco de dados deve ter acesso de rede a um sistema de arquivos EFS. Para ter mais informações, consulte Controlling network access to Amazon EFS file systems for NFS clients (Controlar o acesso à rede aos sistemas de arquivos Amazon EFS para clientes do NFS) no Guia do usuário do Amazon Elastic File System.

Tópicos

- Controlar acesso à rede com grupos de segurança
- Controlar o acesso à rede com políticas de sistema de arquivos

Controlar acesso à rede com grupos de segurança

Você pode controlar o acesso de sua instância de banco de dados aos sistemas de arquivos EFS usando mecanismos de segurança da camada de rede, como grupos de segurança da VPC. Para permitir o acesso a um sistema de arquivos EFS para sua instância de banco de dados, seu sistema de arquivos EFS deve atender aos seguintes requisitos:

- Existe um destino de montagem do EFS em cada zona de disponibilidade usada por uma instância de banco de dados do RDS para Oracle.

  Um destino de montagem do EFS fornece um endereço IP para um endpoint do NFSv4 no qual é possível montar um sistema de arquivos do EFS. Monte o sistema de arquivos usando o respectivo nome DNS, que é resolvido para o endereço IP do destino de montagem do EFS usado pela zona de disponibilidade da sua instância de banco de dados.

  Você pode configurar instâncias de banco de dados em diferentes AZs para usar o mesmo sistema de arquivos EFS. Para multi-AZ, você precisa de um ponto de montagem para cada AZ em sua implantação. Talvez você precise mover uma instância de banco de dados para uma AZ diferente.

Por esses motivos, recomendamos que você crie um ponto de montagem EFS em cada AZ em sua VPC. Por padrão, quando você cria um sistema de arquivos EFS usando o console, o RDS cria destinos de montagem para todas as AZs.

- Um grupo de segurança é anexado ao destino de montagem.
- O grupo de segurança tem uma regra de entrada para permitir a sub-rede de rede ou o grupo de segurança da instância de banco de dados do RDS para Oracle em TCP/2049 (tipo NFS).

Para ter mais informações, consulte Creating Amazon EFS file systems (Criar sistemas de arquivos Amazon EFS) e Creating and managing EFS mount targets and security groups (Criar e gerenciar destinos de montagem e grupos de segurança do EFS) no Guia do usuário do Amazon Elastic File System.

Controlar o acesso à rede com políticas de sistema de arquivos

A integração do Amazon EFS com o RDS para Oracle funciona com a política padrão (vazia) do sistema de arquivos EFS. A política padrão não usa o IAM para se autenticar. Em vez disso, ele concede acesso total a qualquer cliente anônimo que possa se conectar ao sistema de arquivos usando um destino de montagem. A política padrão estará em vigor sempre que uma política de sistema de arquivos configurada pelo usuário não estiver em vigor, inclusive na criação do sistema de arquivos. Para ter mais informações, consulte Default EFS file system policy (Política padrão do sistema de arquivos EFS) no Guia do usuário do Amazon Elastic File System.

Para fortalecer o acesso ao seu sistema de arquivos EFS para todos os clientes, inclusive o RDS para Oracle, você pode configurar as permissões do IAM. Nessa abordagem, você cria uma política do sistema de arquivos. Para ter mais informações, consulte Creating file system policies (Criar políticas do sistema de arquivos) no Guia do usuário do Amazon Elastic File System.

Configurar permissões do IAM para a integração do RDS para Oracle com o Amazon EFS

Por padrão, o recurso de integração do Amazon EFS não usa um perfil do IAM: a configuração da opção `USE_IAM_ROLE` é `FALSE`. Para integrar o RDS para Oracle ao Amazon EFS e um perfil do IAM, a instância de banco de dados deve ter permissões do IAM para acessar um sistema de arquivos Amazon EFS.

Tópicos

- Etapa 1: Criar um perfil do IAM para a instância de banco de dados e anexar a política
- Etapa 2: Criar uma política para seu sistema de arquivos Amazon EFS

- Etapa 3: Associar um perfil do IAM à sua instância de banco de dados do RDS para Oracle

Etapa 1: Criar um perfil do IAM para a instância de banco de dados e anexar a política

Nesta etapa, você criará uma função para a instância de banco de dados do RDS para Oracle para permitir que o Amazon RDS acesse o sistema de arquivos EFS.

Console

Como criar um perfil do IAM a fim de permitir que o Amazon RDS acesse um sistema de arquivos EFS

1. Abra o Console de Gerenciamento do IAM.
2. No painel de navegação, escolha Roles.
3. Escolha Criar Perfil.
4. Para Serviço da AWS, escolha RDS.
5. Em Select your use case (Selecionar o caso de uso), escolha RDS – Add Role to Database (Adicionar função ao banco de dados).
6. Escolha Próximo.
7. Não adicione nenhuma política de permissões. Escolha Próximo.
8. Defina Role name (Nome da função) como um nome para a sua função do IAM, por exemplo, `rds-efs-integration-role`. Você também pode adicionar um valor opcional para Description (Descrição).
9. Selecione Criar perfil.

AWS CLI

Para limitar as permissões do serviço a um recurso específico, recomendamos usar as chaves de contexto de condição global `aws:SourceArn` e `aws:SourceAccount` em relações de confiança baseadas em recursos. Essa é a maneira mais eficiente de se proteger contra o problema "confused deputy".

Você pode usar as duas chaves de contexto de condição global e fazer com que o valor `aws:SourceArn` contenha o ID da conta. Nesses casos, verifique se o valor `aws:SourceAccount` e a conta no `aws:SourceArn` usa o mesmo ID de conta quando eles são usados na mesma instrução.

- Use `aws:SourceArn` se quiser acesso entre serviços para um único recurso.
- Use `aws:SourceAccount` se você quiser permitir que qualquer recurso nessa conta seja associado ao uso entre serviços.

Na relação de confiança, certifique-se de usar a chave de contexto de condição global `aws:SourceArn` com o nome do recurso da Amazon (ARN) completo dos recursos que acessam a função.

O seguinte comando da AWS CLI cria a função chamada *`rds-efs-integration-role`* para essa finalidade.

Example

Para Linux, macOS ou Unix:

```
aws iam create-role \
   --role-name rds-efs-integration-role \
   --assume-role-policy-document '{
     "Version": "2012-10-17",
     "Statement": [
       {
         "Effect": "Allow",
         "Principal": {
            "Service": "rds.amazonaws.com"
          },
         "Action": "sts:AssumeRole",
         "Condition": {
             "StringEquals": {
                 "aws:SourceAccount": my_account_ID,
                 "aws:SourceArn": "arn:aws:rds:Region:my_account_ID:db:dbname"
             }
         }
       }
     ]
   }'
```

Para Windows:

```
aws iam create-role ^
   --role-name rds-efs-integration-role ^
   --assume-role-policy-document '{
```

```
    "Version": "2012-10-17",
    "Statement": [
      {
        "Effect": "Allow",
        "Principal": {
           "Service": "rds.amazonaws.com"
         },
        "Action": "sts:AssumeRole",
        "Condition": {
            "StringEquals": {
                "aws:SourceAccount": my_account_ID,
                "aws:SourceArn": "arn:aws:rds:Region:my_account_ID:db:dbname"
            }
        }
      }
    ]
  }'
```

Para ter mais informações, consulte Criar uma função para delegar permissões a um usuário do IAM no Guia do usuário do IAM.

Etapa 2: Criar uma política para seu sistema de arquivos Amazon EFS

Nesta etapa, crie uma política para seu sistema de arquivos EFS.

Como criar ou editar uma política de sistema de arquivos EFS

1. Abra o Console de Gerenciamento do EFS.
2. Escolha File Systems (Sistemas de arquivos).
3. Na página File systems (Sistemas de arquivos), selecione o sistema para o qual deseja editar ou criar uma política de sistema de arquivos. A página de detalhes desse sistema de arquivos é exibida.
4. Selecione a guia File system policy (Política do sistema de arquivos).

   Se a política estiver em branco, a política padrão do sistema de arquivos EFS estará em uso. Para ter mais informações, consulte Default EFS file system policy (Política padrão do sistema de arquivos EFS) no Guia do usuário do Amazon Elastic File System.
5. Escolha Editar. A página File system policy (Política de sistema de arquivos) é exibida.
6. No Policy editor (Editor de políticas), insira uma política como a seguinte e selecione Save (Salvar).

JSON

```
{
    "Version": "2012-10-17",
    "Id": "ExamplePolicy01",
    "Statement": [
        {
            "Sid": "ExampleStatement01",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::123456789012:role/rds-efs-integration-
role"
            },
            "Action": [
                "elasticfilesystem:ClientMount",
                "elasticfilesystem:ClientWrite",
                "elasticfilesystem:ClientRootAccess"
            ],
            "Resource": "arn:aws:elasticfilesystem:us-
east-1:123456789012:file-system/fs-1234567890abcdef0"
        }
    ]
}
```

Etapa 3: Associar um perfil do IAM à sua instância de banco de dados do RDS para Oracle

Nesta etapa, associe um perfil do IAM à instância de banco de dados. Esteja ciente dos seguintes requisitos:

- Você precisa ter acesso a um perfil do IAM com a política de permissões do Amazon EFS necessária anexada a ele.
- Só é possível associar um perfil do IAM à instância de banco de dados do RDS para Oracle por vez.
- O status de sua instância deve ser Available (Disponível).

Para ter mais informações, consulte Identity and access management for Amazon EFS (Identity and access management para Amazon EFS) no Guia do usuário do Amazon Elastic File System.

Console

Como associar um perfil do IAM à instância de banco de dados do RDS para Oracle

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. Escolha Databases (Bancos de dados).
3. Se a instância do banco de dados não estiver disponível, escolha Actions (Ações) e, em seguida, Start (Iniciar). Quando o status da instância for exibido Started (Iniciado), vá para a próxima etapa.
4. Escolha o nome da instância de banco de dados Oracle para exibir os detalhes.
5. Na guia Connectivity & security (Conectividade e segurança), role para baixo até a seção Manage IAM roles (Gerenciar funções do IAM) na parte inferior da página.
6. Escolha a função a ser adicionada na seção Add IAM roles to this instance (Adicionar funções do IAM a essa instância).
7. Em Feature (Recurso), selecione EFS_INTEGRATION.
8. Escolha Add role (adicionar função).

AWS CLI

O seguinte comando da AWS CLI adiciona a função a uma instância de banco de dados Oracle denominada *mydbinstance*.

Example

Para Linux, macOS ou Unix:

```
aws rds add-role-to-db-instance \
   --db-instance-identifier mydbinstance \
   --feature-name EFS_INTEGRATION \
   --role-arn your-role-arn
```

Para Windows:

```
aws rds add-role-to-db-instance ^
   --db-instance-identifier mydbinstance ^
   --feature-name EFS_INTEGRATION ^
```

```
    --role-arn your-role-arn
```

Substitua *your-role-arn* pelo ARN da função que você anotou em uma etapa anterior. `EFS_INTEGRATION` deve ser especificado para a opção `--feature-name`.

Adicionar a opção EFS_INTEGRATION

Para usar o Amazon RDS para Oracle com Amazon EFS, sua instância de banco de dados deve estar associada a um grupo de opções que inclua a opção `EFS_INTEGRATION`.

Várias instâncias de banco de dados Oracle que pertencem ao mesmo grupo de opções compartilham o mesmo sistema de arquivos EFS. Instâncias de banco de dados diferentes podem acessar os mesmos dados, mas o acesso pode ser dividido com o uso de diferentes diretórios Oracle. Para ter mais informações, consulte Transferir arquivos entre o RDS para Oracle e um sistema de arquivos Amazon EFS.

Console

Como configurar um grupo de opções para a integração do Amazon EFS

1. Crie um novo grupo de opções ou identifique um grupo de opções existente ao qual você pode adicionar a opção `EFS_INTEGRATION`.

   Para obter informações sobre como criar um grupo de opções, consulte Criar um grupo de opções.
2. Adicione a opção `EFS_INTEGRATION` ao grupo de opções. Você precisa especificar o ID do sistema de arquivos `EFS_ID` e definir o sinalizador `USE_IAM_ROLE`.

   Para ter mais informações, consulte Adicionar uma opção a um grupo de opções.
3. Associe o grupo de opções à instância de banco de dados de uma das seguintes formas:

   - Crie uma instância de banco de dados Oracle e associe o grupo de opções a ela. Para obter informações sobre como criar uma instância de banco de dados, consulte Criar uma instância de banco de dados do Amazon RDS.
   - Modifique uma instância de banco de dados do Oracle para associar o grupo de opções a ela. Para obter informações sobre como modificar uma instância de banco de dados Oracle, consulte Modificar uma instância de banco de dados do Amazon RDS.

AWS CLI

Como configurar um grupo de opções para a integração do EFS

1. Crie um novo grupo de opções ou identifique um grupo de opções existente ao qual você pode adicionar a opção `EFS_INTEGRATION`.

   Para obter informações sobre como criar um grupo de opções, consulte Criar um grupo de opções.

2. Adicione a opção `EFS_INTEGRATION` ao grupo de opções.

   Por exemplo, o seguinte comando da AWS CLI adiciona a opção `EFS_INTEGRATION` a um grupo de opções denominado **`myoptiongroup`**.

   Example

   Para Linux, macOS ou Unix:

   ```
   aws rds add-option-to-option-group \
      --option-group-name myoptiongroup \
      --options "OptionName=EFS_INTEGRATION,OptionSettings=\
      [{Name=EFS_ID,Value=fs-1234567890abcdef0},{Name=USE_IAM_ROLE,Value=TRUE}]"
   ```

   Para Windows:

   ```
   aws rds add-option-to-option-group ^
      --option-group-name myoptiongroup ^
      --options "OptionName=EFS_INTEGRATION,OptionSettings=^
      [{Name=EFS_ID,Value=fs-1234567890abcdef0},{Name=USE_IAM_ROLE,Value=TRUE}]"
   ```

3. Associe o grupo de opções à instância de banco de dados de uma das seguintes formas:

   - Crie uma instância de banco de dados Oracle e associe o grupo de opções a ela. Para obter informações sobre como criar uma instância de banco de dados, consulte Criar uma instância de banco de dados do Amazon RDS.
   - Modifique uma instância de banco de dados do Oracle para associar o grupo de opções a ela. Para obter informações sobre como modificar uma instância de banco de dados Oracle, consulte Modificar uma instância de banco de dados do Amazon RDS.

Configurar permissões do sistema de arquivos Amazon EFS

Por padrão, somente o usuário raiz (UID `0`) tem permissões de leitura, gravação e execução para um sistema de arquivos EFS recém-criado. Para os outros usuários modificarem o sistema de arquivos, o usuário raiz deve conceder explicitamente acesso a eles. O usuário da instância de banco de dados do RDS para Oracle está na categoria `others`. Para ter mais informações, consulte Working with users, groups, and permissions at the Network File System (NFS) Level (Trabalhar com usuários, grupos e permissões no nível do Network File System (NFS)) no Guia do usuário do Amazon Elastic File System.

Para permitir que a instância de banco de dados do RDS para Oracle leia e grave arquivos em um sistema de arquivos EFS, faça o seguinte:

- Monte um sistema de arquivos EFS localmente em sua instância do Amazon EC2 ou on-premises.
- Configure as permissões de granulação fina.

Por exemplo, para conceder a `other` usuários permissões para gravar na raiz do sistema de arquivos EFS, execute `chmod 777` nesse diretório. Para ter mais informações, consulte Example Amazon EFS file system use cases and permissions (Exemplos de casos de uso e permissões do sistema de arquivos Amazon EFS) no Guia do usuário do Amazon Elastic File System.

Transferir arquivos entre o RDS para Oracle e um sistema de arquivos Amazon EFS

Para transferir arquivos entre uma instância do RDS para Oracle e um sistema de arquivos do Amazon EFS, crie pelo menos um diretório Oracle e configure as permissões do sistema de arquivos EFS para controlar o acesso à instância de banco de dados.

Tópicos

- Criar um diretório Oracle
- Transferir dados de e para um sistema de arquivos EFS: exemplos

Criar um diretório Oracle

Para criar um diretório da Oracle, use o procedimento `rdsadmin.rdsadmin_util.create_directory_efs`. O procedimento tem os seguintes parâmetros.

Nome do parâmetro	Tipo de dados	Padrão	Obrigatório	Descrição
`p_directory_name`	VARCHAR	–	Sim	O nome do diretório da Oracle.
`p_path_on_efs`	VARCHAR	–	Sim	O caminho no sistema de arquivos EFS. O prefixo do nome do caminho usa o `/rdsefs-`*`fsid`*`/` padrão, em que *`fsid`* é um espaço reservado para o ID do sistema de arquivos EFS. Por exemplo, se seu sistema de arquivos EFS for denominado `fs-1234567890abcdef0` e você criar um subdiretório nesse sistema de arquivos chamado `mydir`, você poderá especificar o seguinte valor: `/rdsefs-fs-1234567890abcdef0/mydir`

Suponha que você crie um subdiretório denominado `/datapump1` no sistema de arquivos EFS `fs-1234567890abcdef0`. O exemplo a seguir cria um diretório Oracle `DATA_PUMP_DIR_EFS` que aponta para o diretório `/datapump1` no sistema de arquivos EFS. O valor do caminho do sistema de arquivos para o parâmetro `p_path_on_efs` é prefixado com a string `/rdsefs-`.

```
BEGIN
  rdsadmin.rdsadmin_util.create_directory_efs(
    p_directory_name => 'DATA_PUMP_DIR_EFS',
    p_path_on_efs    => '/rdsefs-fs-1234567890abcdef0/datapump1');
END;
/
```

Transferir dados de e para um sistema de arquivos EFS: exemplos

O exemplo a seguir usa o Oracle Data Pump para exportar a tabela denominada `MY_TABLE` para o arquivo `datapump.dmp`. Esse arquivo reside em um sistema de arquivos EFS.

```
DECLARE
```

```
  v_hdnl NUMBER;
BEGIN
  v_hdnl := DBMS_DATAPUMP.OPEN(operation => 'EXPORT', job_mode => 'TABLE',
 job_name=>null);
  DBMS_DATAPUMP.ADD_FILE(
    handle    => v_hdnl,
    filename  => 'datapump.dmp',
    directory => 'DATA_PUMP_DIR_EFS',
    filetype  => dbms_datapump.ku$_file_type_dump_file);
  DBMS_DATAPUMP.ADD_FILE(
    handle    => v_hdnl,
    filename  => 'datapump-exp.log',
    directory => 'DATA_PUMP_DIR_EFS',
    filetype  => dbms_datapump.ku$_file_type_log_file);
  DBMS_DATAPUMP.METADATA_FILTER(v_hdnl,'NAME_EXPR','IN (''MY_TABLE'')');
  DBMS_DATAPUMP.START_JOB(v_hdnl);
END;
/
```

O exemplo a seguir usa o Oracle Data Pump para importar a tabela denominada `MY_TABLE` do arquivo `datapump.dmp`. Esse arquivo reside em um sistema de arquivos EFS.

```
DECLARE
  v_hdnl NUMBER;
BEGIN
  v_hdnl := DBMS_DATAPUMP.OPEN(
    operation => 'IMPORT',
    job_mode  => 'TABLE',
    job_name  => null);
  DBMS_DATAPUMP.ADD_FILE(
    handle    => v_hdnl,
    filename  => 'datapump.dmp',
    directory => 'DATA_PUMP_DIR_EFS',
    filetype  => dbms_datapump.ku$_file_type_dump_file );
  DBMS_DATAPUMP.ADD_FILE(
    handle    => v_hdnl,
    filename  => 'datapump-imp.log',
    directory => 'DATA_PUMP_DIR_EFS',
    filetype  => dbms_datapump.ku$_file_type_log_file);
  DBMS_DATAPUMP.METADATA_FILTER(v_hdnl,'NAME_EXPR','IN (''MY_TABLE'')');
  DBMS_DATAPUMP.START_JOB(v_hdnl);
END;
/
```

Para ter mais informações, consulte Importar dados para o Oracle no Amazon RDS.

Remover a opção EFS_INTEGRATION

As etapas para remover a opção `EFS_INTEGRATION` dependem de você estar removendo a opção de várias instâncias de banco de dados ou de uma única instância.

Número de instâncias de banco de dado	Ação	Informações relacionadas
Vários	Remova a opção `EFS_INTEGRATION` do grupo de opções ao qual as instâncias de banco de dados pertencem. Essa alteração afeta todas as instâncias que usam o grupo de opções.	Remover uma opção de um grupo de opções
Único	Modifique a instância de banco de dados e especifique um grupo de opções diferente que não inclua a opção `EFS_INTEGRATION`. Você pode especificar um grupo de opções padrão (vazio) ou criar um grupo de opções personalizado diferente.	Modificar uma instância de banco de dados do Amazon RDS

Depois de remover a opção `EFS_INTEGRATION`, se desejar, você poderá excluir o sistema de arquivos do EFS que estava conectado às instâncias de banco de dados.

Solução de problemas de integração do Amazon EFS

A instância de banco de dados do RDS para Oracle monitora a conectividade com um sistema de arquivos Amazon EFS. Quando o monitoramento detecta um problema, ele pode tentar corrigi-lo e publicar um evento no console do RDS. Para ter mais informações, consulte Visualizar eventos do Amazon RDS.

Utilize as informações desta seção para ajudar a diagnosticar e corrigir problemas comuns ao trabalhar com a integração do Amazon EFS.

Notificação	Descrição	Ação
`The EFS for RDS Oracle instance` *instance_name* `isn't available on the primary host. NFS port 2049 of your EFS isn't reachable.`	A instância de banco de dados não pode se comunicar com o sistema de arquivos EFS.	Verifique o seguinte: • O sistema de arquivos EFS existe. • O grupo de segurança anexado ao destino de montagem do EFS tem uma regra de entrada para permitir o grupo de segurança ou a sub-rede de rede da instância de banco de dados do RDS para Oracle em TCP/2049 (tipo NFS).
`The EFS isn't reachable.`	Ocorreu um erro durante a instalação da opção `EFS_INTEGRATION` .	Verifique o seguinte: • O sistema de arquivos EFS existe. • O grupo de segurança anexado ao destino de montagem do EFS tem uma regra de entrada para permitir o grupo de segurança ou a sub-rede de rede da instância de banco de dados do RDS para Oracle em TCP/2049 (tipo NFS). • O atributo `enableDnsSupport` está ativado para sua VPC. • Você está usando o servidor DNS fornecido pela Amazon

Notificação	Descrição	Ação
		em sua VPC. A integraçã o do Amazon EFS não funciona com um DNS DHCP personalizado.
`The associated role with your DB instance wasn't found.`	Ocorreu um erro durante a instalação da opção `EFS_INTEGRATION` .	Associe um perfil do IAM à instância de banco de dados do RDS para Oracle.
`The associated role with your DB instance wasn't found.`	Ocorreu um erro durante a instalação da opção `EFS_INTEGRATION` . O RDS para Oracle foi restaurad o a partir de um snapshot de banco de dados com a configuração da opção `USE_IAM_ROLE` de `TRUE`.	Associe um perfil do IAM à instância de banco de dados do RDS para Oracle.
`The associated role with your DB instance wasn't found.`	Ocorreu um erro durante a instalação da opção `EFS_INTEGRATION` . O RDS para Oracle foi criado por meio de um modelo completo do CloudFormation com a configuração da opção `USE_IAM_ROLE` de `TRUE`.	Como solução alternativa, realize as seguintes etapas: 1. Crie uma instância de banco de dados com o perfil do IAM e o grupo de opções padrão. 2. Em uma atualização de pilha subsequente, adicione o grupo de opções personalizadas com a opção `EFS_INTEG RATION` .

Notificação	Descrição	Ação
`PLS-00302: component 'CREATE_DIRECTORY_ EFS' must be declared`	Esse erro pode ocorrer quando você está usando uma versão do RDS para Oracle não compatível com o Amazon EFS.	Você deve estar usando a instância de banco de dados do RDS para Oracle versão 19.0.0.0.ru-2022-07.rur-202 2-07.r1 ou posteriores.
`Read access of your EFS is denied. Check your file system policy.`	Sua instância de banco de dados não pode ler o sistema de arquivos EFS.	Seu sistema de arquivos EFS deve permitir acesso de leitura por meio do perfil do IAM ou no nível do sistema de arquivos EFS.
N/D	Sua instância de banco de dados não pode gravar no sistema de arquivos EFS.	Siga as seguintes etapas: 1. O sistema de arquivos EFS deve estar montado em uma instância do Amazon EC2. 2. Dê ao `others` grupo acesso de gravação ao seu usuário do RDS. A técnica mais simples é executar o comando `chmod 777` no diretório superior do sistema de arquivos EFS.

Oracle Java Virtual Machine

O Amazon RDS oferece suporte ao Oracle Java Virtual Machine (JVM) por meio da opção `JVM`. O Oracle Java oferece um esquema SQL e funções que facilitam os recursos do Oracle Java em um banco de dados Oracle. Para obter mais informações, consulte Introdução ao Java no banco de dados Oracle na documentação da Oracle. É possível usar o Oracle JVM com todas as versões do Oracle Database 21c (21.0.0) e do Oracle Database 19c (19.0.0).

Considerações sobre o Oracle JVM

A implementação do Java no Amazon RDS tem um conjunto limitado de permissões. O usuário mestre recebe a função `RDS_JAVA_ADMIN`, que concede um subconjunto dos privilégios concedidos pela função `JAVA_ADMIN`. Para listar os privilégios concedidos à função `RDS_JAVA_ADMIN`, execute a seguinte consulta na sua instância de banco de dados:

```
SELECT * FROM dba_java_policy
   WHERE grantee IN ('RDS_JAVA_ADMIN', 'PUBLIC')
   AND enabled = 'ENABLED'
   ORDER BY type_name, name, grantee;
```

Pré-requisitos para o Oracle JVM

Veja a seguir os pré-requisitos para usar o Oracle Java:

- A instância de banco de dados deve pertencer a uma classe grande o suficiente. O Oracle Java não é compatível com as classes de instância de banco de dados db.t3.small. Para ter mais informações, consulte Classes de instâncias de banco de dados .
- A instância de banco de dados deve ter Auto Minor Version Upgrade (Atualização automática de versão secundária) habilitada. Essa opção permite que a instância de banco de dados receba atualizações de versões secundárias do mecanismo automaticamente quando estiverem disponíveis. O Amazon RDS usa essa opção para atualizar a instância de banco de dados para a Oracle Patch Set Update (PSU) ou Release Update (RU) mais recente. Para obter mais informações, consulte Modificar uma instância de banco de dados do Amazon RDS.

Práticas recomendadas do Oracle JVM

Estas são as práticas recomendadas para usar o Oracle Java:

- Para obter segurança máxima, use a opção `JVM` com Secure Sockets Layer (SSL). Para obter mais informações, consulte Oracle Secure Sockets Layer.
- Configure a instância de banco de dados para restringir o acesso à rede. Para obter mais informações, consulte Cenários para acessar uma instância de banco de dados em uma VPC e Trabalhar com uma instância de banco de dados em uma VPC.
- Atualize a configuração de seus endpoints HTTPS para oferecer suporte ao TLSv1.2 se você atender às seguintes condições:
  - Você usa o Oracle Java Virtual Machine (JVM) para conectar um endpoint HTTPS por meio de protocolos TLSv1 ou TLSv1.1.
  - Seu endpoint não é compatível com o protocolo TLSv1.2.
  - Você não aplicou a atualização da versão de abril de 2021 ao Oracle DB.

  Ao atualizar a configuração do endpoint, você garante que a conectividade da JVM com o endpoint HTTPS continuará funcionando. Para obter mais informações sobre alterações TLS no Oracle JRE e no JDK, consulte Oracle JRE e JDK Cryptographic Roadmap.

Adicionar a opção Oracle JVM

Este é o processo geral para adicionar a opção `JVM` a uma instância de banco de dados:

1. Crie um novo grupo de opções, ou copie ou modifique um existente.
2. Adicione a opção ao grupo de opções.
3. Associe o grupo de opções à instância de banco de dados.

Há uma breve interrupção enquanto a opção `JVM` é adicionada. Depois de adicionar a opção , você não precisará reiniciar sua instância de banco de dados. Assim que o grupo de opções estiver ativo, o Oracle Java estará disponível.

> **Note**
>
> Durante essa interrupção, as funções de verificação de senha são desativadas brevemente. Você também pode esperar ver eventos relacionados às funções de verificação de senha durante a interrupção. As funções de verificação de senha são ativadas novamente antes que a instância de banco de dados Oracle esteja disponível.

Para adicionar a opção JVM a uma instância de banco de dados

1. Determine o grupo de opções que você deseja usar. Você pode criar um novo grupo de opções ou usar um existente. Se você quiser usar um grupo de opções existente, vá para a próxima etapa. Caso contrário, crie um grupo de opções de banco de dados personalizado com as seguintes configurações:

   - Em Engine (Mecanismo), selecione o mecanismo de banco de dados usado pela instância de banco de dados (oracle-ee, oracle-se, oracle-se1 ou oracle-se2).
   - Para Major engine version (Versão do mecanismo principal), escolha a versão da sua instância de banco de dados.

   Para obter mais informações, consulte Criar um grupo de opções.
2. Adicione a opção JVM ao grupo de opções. Para mais informações sobre a adição de opções, consulte Adicionar uma opção a um grupo de opções.
3. Aplique o grupo de opções a uma instância de banco de dados nova ou existente:

   - Para uma nova instância de banco de dados, aplique o grupo de opções ao executar a instância. Para obter mais informações, consulte Criar uma instância de banco de dados do Amazon RDS.
   - Para uma instância de banco de dados existente, aplique o grupo de opções modificando a instância e anexando o novo grupo de opções. Para obter mais informações, consulte Modificar uma instância de banco de dados do Amazon RDS.
4. Conceda as permissões necessárias aos usuários.

   Por padrão, o usuário mestre do Amazon RDS tem as permissões para usar a opção `JVM`. Se outros usuários precisarem dessas permissões, conecte-se à instância de banco de dados como o usuário mestre em um cliente SQL e conceda as permissões aos usuários.

   O exemplo a seguir concede as permissões para usar a opção `JVM` ao usuário `test_proc`.

```
create user test_proc identified by password;
CALL dbms_java.grant_permission('TEST_PROC',
 'oracle.aurora.security.JServerPermission', 'LoadClassInPackage.*', '');
```

> **Note**
>
> Especifique uma senha diferente do prompt mostrado aqui como prática recomendada de segurança.

Depois que o usuário receber as permissões, a seguinte consulta deverá retornar a saída.

```
select * from dba_java_policy where grantee='TEST_PROC';
```

> **Note**
>
> O nome do usuário do Oracle diferencia maiúsculas de minúsculas e, normalmente, usa somente caracteres maiúsculos.

Remover a opção Oracle JVM

Você pode remover a opção `JVM` de uma instância de banco de dados. Há uma breve interrupção enquanto a opção é removida. Depois de remover a opção `JVM`, você não precisará reiniciar a instância de banco de dados.

> **Warning**
>
> Remover a opção `JVM` poderá resultar em perda de dados se a instância de banco de dados estiver usando tipos de dados que foram habilitados como parte da opção. Faça backup dos dados antes de continuar. Para obter mais informações, consulte Backup, restauração e exportação de dados.

Para remover a opção `JVM` de uma instância de banco de dados, faça o seguinte:

- Remova a opção `JVM` do grupo de opções ao qual ela pertence. Essa alteração afeta todas as instâncias de bancos de dados que usam o grupo de opções. Para obter mais informações, consulte Remover uma opção de um grupo de opções.
- Modifique a instância de banco de dados e especifique um grupo de opções diferente que não inclua a opção `JVM`. Essa alteração afeta uma única instância de banco de dados. Você pode

especificar um grupo de opções padrão (vazio) ou criar um grupo de opções personalizado diferente. Para ter mais informações, consulte Modificar uma instância de banco de dados do Amazon RDS.

Oracle Enterprise Manager

O Amazon RDS oferece suporte para o Oracle Enterprise Manager (OEM). OEM é a linha de produtos da Oracle para o gerenciamento integrado de tecnologia da informação empresarial.

O Amazon RDS é compatível com o OEM em CDBs ou não CDBs do Oracle Database 19c. A tabela a seguir descreve as opções de OEM compatíveis.

Opção	ID da opção	Versões do OEM compatíveis
OEM Database Express	`OEM`	OEM Database Express 19c
OEM Management Agent	`OEM_AGENT`	OEM Cloud Control para 13c

> **Note**
>
> É possível usar o OEM Database ou o OEM Management Agent, mas não ambos.

Oracle Enterprise Manager Database Express

O Amazon RDS comporta o Oracle Enterprise Manager Database Express (EM Express) com o uso da opção OEM. O Amazon RDS é compatível com o EM Express para Oracle Database 19c usando a arquitetura CDB ou não.

O EM Express é uma ferramenta de gerenciamento de banco de dados baseada na web incluída em seu banco de dados e só está disponível quando está aberta. Ele comporta as principais funções de gerenciamento de performance e de administração básica de banco de dados. Para ter mais informações, consulte Introduction to Oracle Enterprise Manager Database Express, na documentação do Oracle Database.

> **Note**
>
> O EM Express não é compatível com a classe de instância de banco de dados db.t3.small. Para ter mais informações sobre classes de instância de banco de dados, consulte Classes de instância de banco de dados do RDS para Oracle.

Configurações da opção OEM

O Amazon RDS oferece suporte para as seguintes configurações da opção OEM.

Configuração da opção	Valores válidos	Descrição
Porta	Um valor inteiro	A porta na instância de banco de dados do RDS para Oracle que escuta o EM Express. O padrão é 5500.
Grupos de segurança	—	Um grupo de segurança tem acesso à Port (Porta).

Etapa 1: Adicionar a opção OEM

O processo geral para adicionar as opções do OEM a uma instância de banco de dados é o seguinte:

1. Crie um novo grupo de opções, ou copie ou modifique um existente.

2. Adicione a opção ao grupo de opções.
3. Associe o grupo de opções à instância de banco de dados.

Quando você adiciona a opção OEM, uma breve interrupção ocorre enquanto a instância de banco de dados é reiniciada automaticamente.

Para adicionar a opção OEM a uma instância de banco de dados

1. Determine o grupo de opções que você deseja usar. Você pode criar um novo grupo de opções ou usar um existente. Se você quiser usar um grupo de opções existente, vá para a próxima etapa. Caso contrário, crie um grupo de opções de banco de dados personalizado com as seguintes configurações:

   a. Em Engine (Mecanismo), escolha a edição Oracle para sua instância de banco de dados.

   b. Para Major engine version (Versão do mecanismo principal), escolha a versão da sua instância de banco de dados.

   Para obter mais informações, consulte Criar um grupo de opções.

2. Adicione a opção OEM ao grupo de opções e defina as configurações da opção. Para mais informações sobre a adição de opções, consulte Adicionar uma opção a um grupo de opções. Para ter mais informações sobre cada configuração, consulte Configurações da opção OEM.

   > **Note**
   >
   > Se você adicionar a opção OEM a um grupo de opções existente que já está anexado a uma ou mais instâncias de banco de dados, uma breve interrupção ocorrerá enquanto todas as instâncias de banco de dados forem reiniciadas automaticamente.

3. Aplique o grupo de opções a uma instância de banco de dados nova ou existente:

   - Para uma nova instância de banco de dados, aplique o grupo de opções ao executar a instância. Para obter mais informações, consulte Criar uma instância de banco de dados do Amazon RDS.
   - Para uma instância de banco de dados existente, aplique o grupo de opções modificando a instância e anexando o novo grupo de opções. Quando você adiciona a opção OEM, uma breve interrupção ocorre enquanto a instância de banco de dados é reiniciada

automaticamente. Para obter mais informações, consulte Modificar uma instância de banco de dados do Amazon RDS.

> **Note**
>
> Também é possível usar a AWS CLI para adicionar a opção OEM. Para ver exemplos, consulte Adicionar uma opção a um grupo de opções.

Etapa 2: (Somente CDB) Desbloquear a conta de usuário DBSNMP

Se sua instância de banco de dados usa a arquitetura CDB, você deve fazer login no EM Express como `DBSNMP`, em um CDB, `DBSNMP` é um usuário comum. Por padrão, essa conta está bloqueada. Se sua instância de banco de dados não usa a arquitetura CDB, ignore essa etapa.

Como desbloquear a conta de usuário DBSNMP em uma instância de CDB

1. No SQL*Plus ou em outra aplicação Oracle SQL, faça login como usuário principal em sua instância de banco de dados.
2. Execute o seguinte procedimento armazenado para desbloquear a conta `DBSNMP`:

   ```
   EXEC rdsadmin.rdsadmin_util.reset_oem_agent_password('new_password');
   ```

   Se você receber um erro informando que o procedimento não existe, reinicie sua instância do CDB para instalá-la automaticamente. Para obter mais informações, consulte Reinicializar uma instância de banco de dados .

Etapa 3: Acessar o EM Express por meio do navegador

Ao acessar o EM Express pelo navegador da web, é exibida uma janela de login que solicita um nome de usuário e uma senha.

Para acessar o EM Express por meio do navegador

1. Identifique o endpoint e a porta do EM Express para sua instância de banco de dados do Amazon RDS. Para obter informações sobre como localizar o endpoint de sua instância de banco de dados do Amazon RDS, consulte Encontrar o endpoint da instância de banco de dados do RDS para Oracle.

2. Insira um URL na barra de localização do navegador usando o formato a seguir.

```
https://endpoint.rds.amazonaws.com:port/em
```

Por exemplo, se o endpoint da sua instância de banco de dados do Amazon RDS for `mydb.a1bcde234fgh.us-east-1.rds.amazonaws.com`, e a porta do EM Express for `1158`, use o URL a seguir para acessar o EM Express.

```
https://mydb.f9rbfa893tft.us-east-1.rds.amazonaws.com:1158/em
```

3. Quando os detalhes de login forem solicitados, realize uma das seguintes ações, dependendo da arquitetura do banco de dados:

O banco de dados não é CDB.

Digite o nome de usuário mestre e a senha mestre da sua instância de banco de dados.

O banco de dados é CDB.

Digite `DBSNMP` para o usuário e a senha do `DBSNMP`. Deixe o campo `Container` vazio.

Modificar configurações do banco de dados OEM

Depois de habilitar o banco de dados OEM, você pode modificar a configuração Security Groups para essa opção.

Você não pode modificar o número da porta OEM depois de ter associado o grupo de opções uma instância de banco de dados. Para alterar o número da porta OEM para uma instância de banco de dados, faça o seguinte:

1. Crie um novo grupo de opções.
2. Adicione a opção OEM com o novo número de porta ao novo grupo de opções.
3. Remova o grupo de opções existente da instância de banco de dados.
4. Adicione o novo grupo de opções à instância de banco de dados.

Para mais informações sobre como modificar as configurações da opção, consulte Modificar uma configuração de opção. Para ter mais informações sobre cada configuração, consulte Configurações da opção OEM.

Executar tarefas do OEM Database Express

É possível usar os procedimentos do Amazon RDS para executar determinadas tarefas do OEM Database Express. Executando esses procedimentos, você pode executar as tarefas listadas a seguir.

> **Note**
>
> As tarefas do OEM Database Express são executadas de forma assíncrona.

Tarefas

- Mudar o front-end do site do OEM Database Express para Adobe Flash
- Mudar o front-end do site do OEM Database Express para Oracle JET

Mudar o front-end do site do OEM Database Express para Adobe Flash

> **Note**
>
> Essa tarefa está disponível somente para não CDBs do Oracle Database 19c.

A partir do Oracle Database 19c, a Oracle descontinuou a interface de usuário anterior do OEM Database Express, que é baseada no Adobe Flash. Em vez disso, o OEM Database Express agora usa uma interface criada com o Oracle JET. Se você tiver dificuldades com a nova interface, é possível mudar para a interface defasada baseada em Flash. As dificuldades que você pode ter com a nova interface incluem ficar preso em uma tela `Loading` após fazer o login no OEM Database Express. Você também pode sentir falta de alguns recursos que estavam presentes na versão baseada em Flash do OEM Database Express.

Para mudar o front-end do site do OEM Database Express para o Adobe Flash, execute o procedimento `rdsadmin.rdsadmin_oem_tasks.em_express_frontend_to_flash` do Amazon RDS. Este procedimento é equivalente ao comando SQL `execemx emx`.

As práticas recomendadas não recomendam o uso do Adobe Flash. Embora você possa reverter para o OEM Database Express baseado em Flash, recomendamos o uso de sites do OEM Database Express baseados em JET, se possível. Se você reverter para o uso do Adobe Flash e quiser mudar novamente para o Oracle JET, use o procedimento

`rdsadmin.rdsadmin_oem_tasks.em_express_frontend_to_jet`. Após uma atualização do banco de dados Oracle, uma versão mais recente do Oracle JET pode resolver problemas relacionados ao JET no OEM Database Express. Para obter mais informações sobre como mudar para o Oracle JET, consulte Mudar o front-end do site do OEM Database Express para Oracle JET.

> **Note**
>
> Executar essa tarefa na instância de banco de dados de origem para uma réplica de leitura também faz com que a réplica de leitura mude os front-ends do site do OEM Database Express para o Adobe Flash.

A chamada do procedimento a seguir cria uma tarefa para mudar o site do OEM Database Express para Adobe Flash e retorna o ID da tarefa.

```
SELECT rdsadmin.rdsadmin_oem_tasks.em_express_frontend_to_flash() as TASK_ID from DUAL;
```

Você pode visualizar o resultado exibindo o arquivo de saída da tarefa.

```
SELECT text FROM table(rdsadmin.rds_file_util.read_text_file('BDUMP','dbtask-task-id.log'));
```

Substitua *`task-id`* pelo ID da tarefa retornado pelo procedimento. Para obter mais informações sobre o procedimento `rdsadmin.rds_file_util.read_text_file` do Amazon RDS, consulte Leitura de arquivos no diretório de uma instância de banco de dados

Também é possível visualizar o conteúdo do arquivo de saída da tarefa no AWS Management Console ao pesquisar o `task-id` nas entradas do log na seção Logs & events (Logs e eventos).

Mudar o front-end do site do OEM Database Express para Oracle JET

> **Note**
>
> Essa tarefa está disponível somente para não CDBs do Oracle Database 19c.

Para mudar o front-end do site do OEM Database Express para o Oracle JET, execute o Amazon RDS procedimento `rdsadmin.rdsadmin_oem_tasks.em_express_frontend_to_jet`. Este procedimento é equivalente ao comando SQL `execemx omx`.

Por padrão, os sites do OEM Database Express para instâncias de banco de dados Oracle que executam a versão 19c ou posterior usam o Oracle JET. Caso tenha usado o procedimento `rdsadmin.rdsadmin_oem_tasks.em_express_frontend_to_flash` para mudar o front-end do site do OEM Database Express para o Adobe Flash, você pode revertê-lo para o Oracle JET. Para fazer isso, use o procedimento `rdsadmin.rdsadmin_oem_tasks.em_express_frontend_to_jet`. Para obter mais informações sobre como mudar para o Adobe Flash, consulte Mudar o front-end do site do OEM Database Express para Adobe Flash.

> **Note**
>
> Executar esta tarefa a partir da instância de banco de dados de origem para uma réplica de leitura também faz com que a réplica de leitura mude seus front-ends do site do OEM Database Express para o Oracle JET.

A chamada do procedimento a seguir cria uma tarefa para mudar o site do OEM Database Express para Oracle JET e retorna o ID da tarefa.

```
SELECT rdsadmin.rdsadmin_oem_tasks.em_express_frontend_to_jet() as TASK_ID from DUAL;
```

Você pode visualizar o resultado exibindo o arquivo de saída da tarefa.

```
SELECT text FROM table(rdsadmin.rds_file_util.read_text_file('BDUMP','dbtask-task-id.log'));
```

Substitua *`task-id`* pelo ID da tarefa retornado pelo procedimento. Para obter mais informações sobre o procedimento `rdsadmin.rds_file_util.read_text_file` do Amazon RDS, consulte Leitura de arquivos no diretório de uma instância de banco de dados

Também é possível visualizar o conteúdo do arquivo de saída da tarefa no AWS Management Console ao pesquisar o `task-id` nas entradas do log na seção Logs & events (Logs e eventos).

Remover a opção de banco de dados OEM

Você pode remover a opção OEM de uma instância de banco de dados. Quando você remove a opção OEM, ocorre uma breve interrupção enquanto a instância de banco de dados é reiniciada automaticamente. Assim, depois de remover a opção OEM, você não precisará reiniciar sua instância de banco de dados.

Para remover a opção OEM de uma instância de banco de dados, siga um destes procedimentos:

- Remova a opção OEM do grupo de opções ao qual ela pertence. Essa alteração afeta todas as instâncias de bancos de dados que usam o grupo de opções. Para obter mais informações, consulte Remover uma opção de um grupo de opções.
- Modifique a instância de banco de dados e especifique um grupo de opções diferente que não inclua a opção OEM. Essa alteração afeta uma única instância de banco de dados. Você pode especificar um grupo de opções padrão (vazio) ou criar um grupo de opções personalizado diferente. Para ter mais informações, consulte Modificar uma instância de banco de dados do Amazon RDS.

Oracle Management Agent para Enterprise Manager Cloud Control

O Management Agent Oracle Enterprise Manager (OEM) é um componente de software que monitora destinos em execução em hosts e que comunica essas informações ao Oracle Management Service (OMS) na camada intermediária. O Amazon RDS suporta o Management Agent pelo uso da opção `OEM_AGENT`.

Para obter mais informações, consulte Visão geral do Oracle Enterprise Manager Cloud Control 12c e Visão geral do Oracle Enterprise Manager Cloud Control 13c na documentação da Oracle.

Tópicos

- Requisitos para o Management Agent
- Pré-requisitos de comunicação do host OMS
- Limitações para o Management Agent
- Configurações de opção do Management Agent
- Habilitar a opção Management Agent para uma instância de banco de dados
- Remover a opção Management Agent
- Como executar tarefas de banco de dados com o Management Agent

Requisitos para o Management Agent

Veja a seguir os requisitos gerais para usar o Management Agent:

- A instância de banco de dados deve executar o Oracle Database 19c (19.0.0.0). É possível usar a arquitetura CDB ou não CDB.
- É necessário usar um Oracle Management Service (OMS) configurado para estabelecer conexão com a instância de banco de dados. Observe os seguintes requisitos de OMS:
  - As versões 13.5.0.0.v2 e 13.5.0.0.v3 do Management Agent requerem o OMS versão 13.5.0.23 ou 24.1.
  - A versão 13.5.0.0.v1 do Management Agent requer o OMS versão 13.5.0.0 ou 24.1.
  - As versões 13.4.0.9.v1 e 13.4.0.9.v2 do Management Agent requerem a versão 13.4.0.9 ou posterior do OMS e o patch 32198287.
- Na maioria dos casos, você precisa configurar sua VPC para permitir conexões do OMS com a sua instância de banco de dados. Se não tiver familiaridade com a Amazon Virtual Private Cloud (Amazon VPC), recomendamos que você conclua as etapas em Tutorial: Criar uma VPC para usar com uma instância de banco de dados (somente IPv4) antes de continuar.

- É possível usar o Management Agent com o Oracle Enterprise Manager Cloud Control para 12c e 13c. Certifique-se de que você tenha espaço de armazenamento suficiente para sua versão do OEM:
  - Pelo menos 8,5 GiB para OEM 13c versão 5
  - Pelo menos 8,5 GiB para OEM 13c versão 4
  - Pelo menos 8,5 GiB para OEM 13c versão 3
  - Pelo menos 5,5 GiB para OEM 13c versão 2
  - Pelo menos 4,5 GiB para o OEM 13c versão 1
  - Pelo menos 2,5 GiB para OEM 12c
- Se você estiver usando as versões do Management Agent `OEM_AGENT 13.2.0.0.v3` e `13.3.0.0.v2` e deseja usar a conectividade TCPS, siga as instruções no tópico Configuring third party CA certificates for communication with target databases na documentação da Oracle. Além disso, atualize o JDK em seu OMS seguindo as instruções no documento da Oracle com o Oracle Doc ID 2241358.1. Isso garante que o OMS ofereça suporte para todos os pacotes de criptografia compatíveis com o banco de dados.

> **Note**
>
> A conectividade TCPS entre o Management Agent e a instância de banco de dados é compatível com o Management Agent `OEM_AGENT 13.2.0.0.v3`, `13.3.0.0.v2`, `13.4.0.9.v1` e versões posteriores.

Pré-requisitos de comunicação do host OMS

Certifique-se de que seu host OMS e sua instância de banco de dados do Amazon RDS podem se comunicar. Faça o seguinte:

- Para se conectar do Management Agent com o host do OMS, se o host do OMS estiver atrás de um firewall, adicione os endereços IP das suas instâncias de banco de dados ao firewall. Certifique-se de que o firewall do OMS permita o seguinte tráfego de rede:

  Do host do OMS à sua instância de banco de dados

  Configure uma regra de firewall unidirecional que permita tráfego do host do OMS para a porta do receptor do banco de dados (padrão 1521) e para a porta do OEM Agent (padrão 3872).

Da sua instância de banco de dados para o host do OMS

Configure uma regra de firewall unidirecional que permita tráfego do da instância de banco de dados para a porta HTTP do OMS (padrão 4903).

- Para se conectar do OMS ao Management Agent, se o OMS tiver um nome de host que possa ser publicamente resolvido, você deverá adicionar o endereço do OMS a um grupo de segurança. O grupo de segurança deve ter regras de entrada que permitam o acesso à porta do listener de banco de dados e à porta do Management Agent. Para obter um exemplo de como criar regras de segurança e adicionar regras de entrada, consulte Tutorial: Criar uma VPC para usar com uma instância de banco de dados (somente IPv4).
- Para se conectar do OMS ao Management Agent, se o OMS não tiver um nome de host que possa ser publicamente resolvido, use um dos seguintes:
  - Se o OMS estiver hospedado em uma instância do Amazon Elastic Compute Cloud (Amazon EC2) em uma VPC privada, você poderá configurar o emparelhamento de VPCs para se conectar do OMS ao Management Agent. Para obter mais informações, consulte Uma instância de banco de dados em uma VPC acessada por uma instância do EC2 em uma VPC diferente.
  - Se o OMS estiver hospedado no local, você poderá configurar uma conexão VPN para permitir o acesso do OMS ao Management Agent. Para obter mais informações, consulte Uma instância de banco de dados em uma VPC acessada por uma aplicação cliente pela internet ou Conexões VPN.
- Para conectar o OEM Management Agent versão 13.5.0.0 (v1-v3) a um host do OMS 24.1, defina `MINIMUM_TLS_VERSION` para usar o protocolo `TLSv1.2` do TLS 1.2 em suas opções de configuração.

Limitações para o Management Agent

Veja a seguir algumas limitações do uso do Management Agent:

- Não é possível fornecer imagens personalizadas do Oracle Management Agent.
- Tarefas administrativas, como a execução do trabalhos e aplicação de patch de banco de dados, que exigem credenciais de host, não são compatíveis.
- Não há garantias de que as métricas de host e a lista de processos reflitam o estado do sistema atual. Por isso, você não deve usar o OEM para monitorar o sistema de arquivos raiz ou o sistema de arquivos do ponto de montagem. Para obter mais informações sobre como monitorar o sistema operacional, consulte Monitorar métricas do SO com o monitoramento avançado.

- A descoberta automática não é aceita. Você deve adicionar destinos de banco de dados manualmente.
- A disponibilidade do módulo OMS depende de sua edição do banco de dados. Por exemplo, o módulo de ajuste e diagnóstico de performance do banco de dados só está disponível para o Oracle Database Enterprise Edition.
- O Management Agent consome memória adicional e recursos de computação. Se você tiver problemas de desempenho depois de habilitar a opção `OEM_AGENT`, recomendamos que você dimensione para uma classe de instância de banco de dados maior. Para obter mais informações, consulte Classes de instâncias de banco de dados e Modificar uma instância de banco de dados do Amazon RDS.
- O usuário que executa o `OEM_AGENT` no host do Amazon RDS não tem acesso ao log de alertas pelo sistema operacional. Assim, você não pode coletar métricas para `DB Alert Log` e `DB Alert Log Error Status` no OEM.

Configurações de opção do Management Agent

O Amazon RDS oferece suporte para as seguintes configurações da opção Management Agent.

Configuração da opção	Obrigatório	Valores válidos	Descrição

Version (Versão (`AGENT_VERSION`)	Sim	`13.5.0.0.v3`
		`13.5.0.0.v2`
		`13.5.0.0.v1`
		`13.4.0.9.v2`
		`13.4.0.9.v1`
		`13.3.0.0.v2`
			A versão do software Management Agent. A versão mínima compatível é `13.1.0.0.v1` .
			O nome da opção da AWS CLI é `OptionVersion` .
			Note
			Nas regiões AWS GovCloud (US), as versões 13.1 não estão disponíveis.

Configuração da opção	Obrigatório	Valores válidos	Descrição

		`13.3.0.0.v1`
		`13.2.0.0.v3`
		`13.2.0.0.v2`
		`13.2.0.0.v1`
		`13.1.0.0.v1`	

Port (Porta (`AGENT_PORT`)	Sim	Um valor inteiro	A porta na instância de banco de dados que escuta o host do OMS. O padrão é 3872. Seu host OMS deve pertencer a um grupo de segurança que tenha acesso a essa porta. O nome da opção da AWS CLI é `Port`.
Grupos de segurança	Sim	Grupos de segurança existentes	Um grupo de segurança tem acesso à Port (Porta). Seu host OMS deve pertencer a esse grupo de segurança. O nome da opção da AWS CLI é `VpcSecurityGroupMemberships` ou `DBSecurityGroupMemberships`.

Configuração da opção	Obrigatório	Valores válidos	Descrição
OMS_HOST	Sim	Um valor de string, por exemplo, *my.example.oms*	O nome do host ou endereço IP publicamente acessível do OMS. O nome da opção da AWS CLI é `OMS_HOST`.
OMS_PORT	Sim	Um valor inteiro	A porta de upload HTTPS no host do OMS que escuta o Management Agent. Para determinar a porta de upload HTTPS, conecte-se ao host do OMS e execute o seguinte comando (que requer a senha `SYSMAN`): `emctl status oms -details` O nome da opção da AWS CLI é `OMS_PORT`.
AGENT_REGISTRATION_PASSWORD	Sim	Um valor de string	A senha usada pelo Management Agent para autenticar-se no OMS. Recomendamos criar uma senha persistente no OMS antes de habilitar a opção `OEM_AGENT` . Com uma senha persistente, você pode compartilhar um único grupo de opções do Management Agent entre vários bancos de dados do Amazon RDS. O nome da opção da AWS CLI é `AGENT_REGISTRATION_PASSWORD` .

Configuração da opção	Obrigatório	Valores válidos	Descrição
ALLOW_TLS_ONLY	Não	`true`, `false` (padrão)	Um valor que configura o OEM Agent para oferecer suporte apenas ao protocolo `TLSv1` enquanto o agente escuta como um servidor. Essa configuração não é mais compatível. Por padrão, as versões do Management Agent 13.1.0.0.v1 e posterior são compatíveis com Transport Layer Security (TLS).
MINIMUM_TLS_VERSION	Não	`TLSv1` (padrão), `TLSv1.2`	Um valor que especifica a versão mínima do TLS compatível com o OEM Agent enquanto o agente escuta como um servidor. Versões sem suporte do agente são compatíveis apenas com a definição de `TLSv1`. Para conectar a versão 13.5.0.0 (v1-v3) a um host do OMS 24.1, defina essa configuração como `TLSv1.2`.
TLS_CIPHER_SUITE	Não	Consulte Configurações de opção do Management Agent.	Um valor que especifica o conjunto de cifras TLS usado pelo OEM Agent enquanto o agente escuta como um servidor.

A tabela a seguir lista os conjuntos de cifras TLS compatíveis com a opção Management Agent.

Pacote de cifras	Versão do Agent compatível	Conformidade com FedRAMP
TLS_RSA_WITH_AES_128_CBC_SHA	Todos	Não
TLS_RSA_WITH_AES_128_CBC_SHA256	13.1.0.0.v1 e posterior	Não
TLS_RSA_WITH_AES_256_CBC_SHA	13.2.0.0.v3 e posterior	Não
TLS_RSA_WITH_AES_256_CBC_SHA256	13.2.0.0.v3 e posterior	Não
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	13.2.0.0.v3 e posterior	Sim
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	13.2.0.0.v3 e posterior	Sim
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256	13.2.0.0.v3 e posterior	Sim
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384	13.2.0.0.v3 e posterior	Sim

Habilitar a opção Management Agent para uma instância de banco de dados

Para habilitar a opção Management Agent, use as seguintes etapas:

Tópicos

- Etapa 1: adicionar a opção Management Agent a uma instância de banco de dados
- Etapa 2: Desbloquear a conta de usuário DBSNMP
- Etapa 3: Adicionar seus destinos ao console do Agente de gerenciamento

Etapa 1: adicionar a opção Management Agent a uma instância de banco de dados

Para adicionar a opção Agente de gerenciamento à instância de banco de dados, faça o seguinte:

1. Crie um novo grupo de opções, ou copie ou modifique um existente.
2. Adicione a opção ao grupo de opções.
3. Associe o grupo de opções à instância de banco de dados.

Se você encontrar erros, poderá verificar os documentos no My Oracle Support para obter informações sobre como resolver problemas específicos.

Depois de adicionar a opção Management Agent, você não precisará reiniciar sua instância de banco de dados. Assim que o grupo de opções estiver ativo, o OEM Agent estará ativo.

Se o host OMS estiver usando um certificado de terceiros não confiáveis, o Amazon RDS retorna o erro a seguir.

```
You successfully installed the OEM_AGENT option. Your OMS host is using an untrusted
 third party certificate.
Configure your OMS host with the trusted certificates from your third party.
```

Se o erro for retornado, a opção do Management Agent não estará habilitada até que o problema seja corrigido. Para obter informações sobre como corrigir o problema, consulte o documento My Oracle Support 2202569.1.

Console

Como adicionar a opção Agente de gerenciamento a uma instância de banco de dados

1. Determine o grupo de opções que você deseja usar. Você pode criar um novo grupo de opções ou usar um existente. Se você quiser usar um grupo de opções existente, vá para a próxima etapa. Caso contrário, crie um grupo de opções de banco de dados personalizado com as seguintes configurações:

   a. Em Engine (Mecanismo), escolha a edição Oracle para sua instância de banco de dados.

   b. Para Major engine version (Versão do mecanismo principal), escolha a versão da sua instância de banco de dados.

   Para obter mais informações, consulte Criar um grupo de opções.
2. Adicione a opção OEM_AGENT ao grupo de opções e defina as configurações da opção. Para obter mais informações sobre a adição de opções, consulte Adicionar uma opção a um grupo de

opções. Para ter mais informações sobre cada configuração, consulte Configurações de opção do Management Agent.

3. Aplique o grupo de opções a uma instância de banco de dados nova ou existente:

   - Para uma nova instância de banco de dados, você aplica o grupo de opções ao executar a instância. Para ter mais informações, consulte Criar uma instância de banco de dados do Amazon RDS.
   - Para uma instância de banco de dados existente, você aplica o grupo de opções modificando a instância e anexando o novo grupo de opções. Para obter mais informações, consulte Modificar uma instância de banco de dados do Amazon RDS.

AWS CLI

O exemplo a seguir usa o comando da AWS CLI add-option-to-option-group para adicionar a opção `OEM_AGENT` a um grupo de opções chamado `myoptiongroup`.

Para Linux, macOS ou Unix:

```
aws rds add-option-to-option-group \
    --option-group-name "myoptiongroup" \
    --options
 OptionName=OEM_AGENT,OptionVersion=13.1.0.0.v1,Port=3872,VpcSecurityGroupMemberships=sg-123456
{Name=OMS_PORT,Value=4903},{Name=AGENT_REGISTRATION_PASSWORD,Value=password}] \
    --apply-immediately
```

Para Windows:

```
aws rds add-option-to-option-group ^
    --option-group-name "myoptiongroup" ^
    --options
 OptionName=OEM_AGENT,OptionVersion=13.1.0.0.v1,Port=3872,VpcSecurityGroupMemberships=sg-123456
{Name=OMS_PORT,Value=4903},{Name=AGENT_REGISTRATION_PASSWORD,Value=password}] ^
    --apply-immediately
```

Etapa 2: Desbloquear a conta de usuário DBSNMP

O Agente de Gerenciamento usa a conta do usuário `DBSNMP` para se conectar ao banco de dados e relatar problemas ao Oracle Enterprise Manager. Em um CDB, `DBSNMP` é um usuário comum. Essa conta de usuário é necessária tanto para o Agente de gerenciamento quanto para o OEM Database

Express. Por padrão, essa conta está bloqueada. O procedimento para desbloquear essa conta difere dependendo se seu banco de dados usa a arquitetura não CDB ou CDB.

Como desbloquear a conta de usuário DBSNMP

1. No SQL*Plus ou em outra aplicação Oracle SQL, faça login como usuário principal em sua instância de banco de dados.
2. Realize uma das seguintes ações, dependendo da arquitetura do banco de dados:

   O banco de dados não é CDB.

   Execute a seguinte declaração SQL:

   ```
   ALTER USER dbsnmp IDENTIFIED BY new_password ACCOUNT UNLOCK;
   ```

   O banco de dados é CDB.

   Execute o seguinte procedimento armazenado para desbloquear a conta `DBSNMP`:

   ```
   EXEC rdsadmin.rdsadmin_util.reset_oem_agent_password('new_password');
   ```

   Se você receber um erro informando que o procedimento não existe, reinicie sua instância do CDB para instalá-la automaticamente. Para obter mais informações, consulte Reinicializar uma instância de banco de dados .

Etapa 3: Adicionar seus destinos ao console do Agente de gerenciamento

Para adicionar uma instância de banco de dados como destino, você deve conhecer o endpoint e a porta. Para obter informações sobre como localizar o endpoint de sua instância de banco de dados do Amazon RDS, consulte Encontrar o endpoint da instância de banco de dados do RDS para Oracle. Se o banco de dados usa a arquitetura CDB, adicione o contêiner `CDB$ROOT` separadamente como destino.

Como adicionar destinos ao console do Agente de gerenciamento

1. No seu console do OMS, escolha Setup (Configurar), Add Target (Adicionar destino), Add Targets Manually (Adicionar destinos manualmente).
2. Escolha Add Targets Declaratively by Specifying Target Monitoring Properties (Adicionar destinos de forma declarativa especificando as propriedades de monitoramento do destino).

3. Para Target Type (Tipo de destino), escolha Database Instance (Instância de banco de dados).
4. Em Monitoring Agent (Agente de monitoramento), escolha o agente com o identificador igual ao identificador de instância de banco de dados do RDS.
5. Escolha Add Manually (Adicionar manualmente).
6. Insira o endpoint da instância de banco de dados do Amazon RDS ou escolha-o na lista de nomes de host. Verifique se o nome de host especificado corresponde ao endpoint da instância de banco de dados do Amazon RDS.
7. Especifique as seguintes propriedades do banco de dados:
   - Em Target Name (Nome do destino), insira um nome.
   - Em Database system name (Nome do sistema de banco de dados), insira um nome.
   - Em Monitor username (Nome de usuário do monitor), insira **`dbsnmp`**.
   - Em Senha do monitor, insira a senha de Etapa 2: Desbloquear a conta de usuário DBSNMP.
   - Em Role (Função), insira normal.
   - Em Oracle home path (Caminho de início Oracle), insira **`/oracle`**.
   - Para Listener Machine name (Nome da máquina ouvinte), o identificador do agente já é exibido.
   - Em Port (Porta), insira a porta do banco de dados. A porta padrão do RDS é 1521.
   - Em Database name (Nome do banco de dados), insira o nome do banco de dados. Se o banco de dados for um CDB, esse nome será `RDSCDB`.
8. Escolha Test Connection (Testar conexão).
9. Escolha Next (Próximo). O banco de dados de destino aparece na sua lista de recursos monitorados.

Remover a opção Management Agent

Você pode remover o OEM Agent de uma instância de banco de dados. Depois de remover a opção OEM Agent, você não precisará reiniciar sua instância de banco de dados.

Para remover o OEM Agent de uma instância de banco de dados, siga um destes procedimentos:

- Remova a opção OEM Agent do grupo de opções ao qual ela pertence. Essa alteração afeta todas as instâncias de bancos de dados que usam o grupo de opções. Para obter mais informações, consulte Remover uma opção de um grupo de opções.

- Modifique a instância de banco de dados e especifique um grupo de opções diferente que não inclua a opção OEM Agent. Essa alteração afeta uma única instância de banco de dados. Você pode especificar um grupo de opções padrão (vazio) ou criar um grupo de opções personalizado diferente. Para obter mais informações, consulte Modificar uma instância de banco de dados do Amazon RDS.

Como executar tarefas de banco de dados com o Management Agent

Você pode usar os procedimentos do Amazon RDS para executar determinados comandos EMCTL no Management Agent. Executando esses procedimentos, você pode executar as tarefas listadas a seguir.

> **Note**
>
> As tarefas são executadas de forma assíncrona.

Tarefas

- Proteger o Management Agent
- Obter o status do Management Agent
- Como reiniciar o Management Agent
- Como listar os destinos monitorados pelo Management Agent
- Listar os threads de coleta monitorados pelo Management Agent
- Como limpar o estado do Management Agent
- Como fazer o Management Agent carregar o OMS
- Como executar ping no OMS
- Como visualizar o status de uma tarefa em andamento

Proteger o Management Agent

Para proteger o Management Agent, execute o procedimento `rdsadmin.rdsadmin_oem_agent_tasks.secure_oem_agent` do Amazon RDS. Este procedimento é equivalente à execução do comando `emctl secure agent`.

O seguinte procedimento cria uma tarefa para proteger o Management Agent e exibe o ID da tarefa:

```
SELECT rdsadmin.rdsadmin_oem_agent_tasks.secure_oem_agent as TASK_ID from DUAL;
```

Para exibir a saída referente à tarefa e visualizar o resultado, consulte Como visualizar o status de uma tarefa em andamento.

Obter o status do Management Agent

Para obter o status do Management Agent, execute o procedimento do Amazon RDS `rdsadmin.rdsadmin_oem_agent_tasks.get_status_oem_agent`. Este procedimento é equivalente ao comando `emctl status agent`.

O procedimento a seguir cria uma tarefa para obter o status do Management Agent e retorna o ID da tarefa.

```
SELECT rdsadmin.rdsadmin_oem_agent_tasks.get_status_oem_agent() as TASK_ID from DUAL;
```

Para exibir a saída referente à tarefa e visualizar o resultado, consulte Como visualizar o status de uma tarefa em andamento.

Como reiniciar o Management Agent

Para reiniciar o Management Agent, execute o procedimento `rdsadmin.rdsadmin_oem_agent_tasks.restart_oem_agent` do Amazon RDS. Este procedimento é equivalente à execução dos comandos `emctl stop agent` e `emctl start agent`.

O procedimento a seguir cria uma tarefa para reiniciar o Management Agent e retorna o ID da tarefa.

```
SELECT rdsadmin.rdsadmin_oem_agent_tasks.restart_oem_agent as TASK_ID from DUAL;
```

Para exibir a saída referente à tarefa e visualizar o resultado, consulte Como visualizar o status de uma tarefa em andamento.

Como listar os destinos monitorados pelo Management Agent

Para listar os destinos monitorados pelo Management Agent, execute o procedimento `rdsadmin.rdsadmin_oem_agent_tasks.list_targets_oem_agent` do Amazon RDS. Este procedimento é equivalente à execução do comando `emctl config agent listtargets`.

O procedimento a seguir cria uma tarefa para listar os destinos monitorados pelo Management Agent e retorna o ID da tarefa.

```
SELECT rdsadmin.rdsadmin_oem_agent_tasks.list_targets_oem_agent as TASK_ID from DUAL;
```

Para exibir a saída referente à tarefa e visualizar o resultado, consulte Como visualizar o status de uma tarefa em andamento.

Listar os threads de coleta monitorados pelo Management Agent

Para listar todos os threads de coleta em execução, prontos e programados que são monitorados pelo Management Agent, execute o procedimento `rdsadmin.rdsadmin_oem_agent_tasks.list_clxn_threads_oem_agent` do Amazon RDS. Este procedimento é equivalente ao comando `emctl status agent scheduler`.

O procedimento a seguir cria uma tarefa para listar os threads de coleta e retorna o ID da tarefa.

```
SELECT rdsadmin.rdsadmin_oem_agent_tasks.list_clxn_threads_oem_agent() as TASK_ID from
 DUAL;
```

Para exibir a saída referente à tarefa e visualizar o resultado, consulte Como visualizar o status de uma tarefa em andamento.

Como limpar o estado do Management Agent

Para limpar o estado do Management Agent, execute o procedimento `rdsadmin.rdsadmin_oem_agent_tasks.clearstate_oem_agent` do Amazon RDS. Este procedimento é equivalente à execução do comando `emctl clearstate agent`.

O procedimento a seguir cria uma tarefa que limpa o estado do Management Agent e retorna o ID da tarefa.

```
SELECT rdsadmin.rdsadmin_oem_agent_tasks.clearstate_oem_agent() as TASK_ID from DUAL;
```

Para exibir a saída referente à tarefa e visualizar o resultado, consulte Como visualizar o status de uma tarefa em andamento.

Como fazer o Management Agent carregar o OMS

Para que o Management Agent carregue o Oracle Management Server (OMS) associado a ele, execute o procedimento `rdsadmin.rdsadmin_oem_agent_tasks.upload_oem_agent` do Amazon RDS. Este procedimento é equivalente à execução do comando `emclt upload agent`.

O procedimento a seguir cria uma tarefa que faz com que o Management Agent faça upload do OMS associado e retorne o ID da tarefa.

```
SELECT rdsadmin.rdsadmin_oem_agent_tasks.upload_oem_agent() as TASK_ID from DUAL;
```

Para exibir a saída referente à tarefa e visualizar o resultado, consulte Como visualizar o status de uma tarefa em andamento.

Como executar ping no OMS

Para executar ping no OMS do Management Agent, execute o procedimento `rdsadmin.rdsadmin_oem_agent_tasks.ping_oms_oem_agent` do Amazon RDS. Este procedimento é equivalente à execução do comando `emctl pingOMS`.

O procedimento a seguir cria uma tarefa que executa ping no OMS do Management Agent e retorna o ID da tarefa.

```
SELECT rdsadmin.rdsadmin_oem_agent_tasks.ping_oms_oem_agent() as TASK_ID from DUAL;
```

Para exibir a saída referente à tarefa e visualizar o resultado, consulte Como visualizar o status de uma tarefa em andamento.

Como visualizar o status de uma tarefa em andamento

Você pode visualizar o status de uma tarefa em andamento em um arquivo bdump. Os arquivos bdump estão localizados no diretório `/rdsdbdata/log/trace`. Cada nome de arquivo bdump está no seguinte formato.

```
dbtask-task-id.log
```

Quando quiser monitorar uma tarefa, substitua *`task-id`* pelo ID da tarefa que deseja monitorar.

Para visualizar o conteúdo dos arquivos bdump, execute o procedimento `rdsadmin.rds_file_util.read_text_file` do Amazon RDS. A consulta a seguir retorna o conteúdo do arquivo bdump `dbtask-1546988886389-2444.log`.

```
SELECT text FROM
 table(rdsadmin.rds_file_util.read_text_file('BDUMP','dbtask-1546988886389-2444.log'));
```

Para obter mais informações sobre o procedimento `rdsadmin.rds_file_util.read_text_file` do Amazon RDS, consulte Leitura de arquivos no diretório de uma instância de banco de dados.

Oracle Label Security

O Amazon RDS é compatível com o Oracle Label Security para a Enterprise Edition do Oracle Database, com o uso da opção OLS.

A maioria dos controles de segurança do banco de dados acessa o nível do objeto. O Oracle Label Security fornece um controle de acesso granular a linhas de tabela individuais. Por exemplo, você pode usar o Label Security para impor a conformidade regulamentar com um modelo de administração baseado em políticas. Você pode usar as políticas do Label Security para controlar o acesso a dados confidenciais e permitir o acesso somente para os usuários com o nível de autorização apropriado. Para obter mais informações, consulte o tópico de introdução ao Oracle Label Security, na documentação da Oracle.

Tópicos

- Requisitos para o Oracle Label Security
- Considerações sobre o uso do Oracle Label Security
- Adicionar a opção Oracle Label Security
- Solução de problemas

Requisitos para o Oracle Label Security

Familiarize-se com os seguintes requisitos para o Oracle Label Security:

- Sua instância de banco de dados deve usar o modelo "Traga sua própria licença". Para obter mais informações, consulte Opções de licenciamento do RDS para Oracle.
- Você deve ter uma licença válida para o Oracle Enterprise Edition com Suporte e licença para atualização de software.
- Sua licença Oracle deve incluir a opção Label Security.

Considerações sobre o uso do Oracle Label Security

Para usar o Oracle Label Security, você cria políticas que controlam o acesso a linhas específicas nas suas tabelas. Para obter mais informações, consulte o tópico sobre como Criar uma política do Oracle Label Security, na documentação da Oracle.

Considere o seguinte:

- O Oracle Label Security é uma opção permanente e persistente. Como a opção é permanente, não é possível removê-la de um grupo de opções. Se você adicionar o Oracle Label Security a um grupo de opções e associá-lo à sua instância de banco de dados, poderá associar posteriormente um grupo de opções diferente à sua instância de banco de dados, mas esse grupo também deverá conter a opção Oracle Label Security.
- Ao trabalhar com o Label Security, você realiza todas as ações como o perfil `LBAC_DBA`. O usuário mestre da sua instância de banco de dados recebe o perfil `LBAC_DBA`. É possível conceder o perfil `LBAC_DBA` a outros usuários, para que eles possam administrar políticas do Label Security.
- Conceda acesso ao pacote `OLS_ENFORCEMENT` a novos usuários que precisem de acesso ao Oracle Label Security. Para conceder acesso ao pacote `OLS_ENFORCEMENT`, conecte-se à instância de banco de dados como o usuário mestre e execute a seguinte instrução SQL:

```
GRANT ALL ON LBACSYS.OLS_ENFORCEMENT TO username;
```

- É possível configurar o Label Security por meio do Oracle Enterprise Manager (OEM) Cloud Control. O Amazon RDS é compatível com o OEM Cloud Control pelo uso da opção Agente de gerenciamento. Para obter mais informações, consulte Oracle Management Agent para Enterprise Manager Cloud Control.

Adicionar a opção Oracle Label Security

O processo geral para adicionar a opção Oracle Label Security a uma instância de banco de dados é o seguinte:

1. Crie um novo grupo de opções, ou copie ou modifique um existente.
2. Adicione a opção ao grupo de opções.

> **Important**
>
> O Oracle Label Security é uma opção permanente e persistente.

3. Associe o grupo de opções à instância de banco de dados.

Depois de adicionar a opção Label Security, assim que o grupo de opções estiver ativo, o Label Security ficará ativo.

Como adicionar a opção Label Security a uma instância de banco de dados

1. Determine o grupo de opções que você deseja usar. Você pode criar um novo grupo de opções ou usar um existente. Se você quiser usar um grupo de opções existente, vá para a próxima etapa. Caso contrário, crie um grupo de opções de banco de dados personalizado com as seguintes configurações:

   a. Para Engine (Mecanismo), escolha oracle-ee.

   b. Para Major engine version (Versão do mecanismo principal), escolha a versão da sua instância de banco de dados.

   Para obter mais informações, consulte Criar um grupo de opções.

2. Adicione a opção OLS ao grupo de opções. Para mais informações sobre a adição de opções, consulte Adicionar uma opção a um grupo de opções.

   > **Important**
   >
   > Se você adicionar Label Security a um grupo de opções existente que já esteja conectado a uma ou mais instâncias de banco de dados, todas estas serão reiniciadas.

3. Aplique o grupo de opções a uma instância de banco de dados nova ou existente:

   - Para uma nova instância de banco de dados, você aplica o grupo de opções ao executar a instância. Para ter mais informações, consulte Criar uma instância de banco de dados do Amazon RDS.

   - Para uma instância de banco de dados existente, você aplica o grupo de opções modificando a instância e anexando o novo grupo de opções. Quando você adiciona a opção Label Security a uma instância de banco de dados existente, ocorre uma breve interrupção enquanto sua instância de banco de dados é reiniciada automaticamente. Para ter mais informações, consulte Modificar uma instância de banco de dados do Amazon RDS.

Solução de problemas

Os problemas a seguir podem ocorrer ao usar o Oracle Label Security.

Problema	Sugestões de solução de problemas
Ao tentar criar uma política, você vê uma mensagem de erro semelhante à seguinte: `insufficient authorization for the SYSDBA package`.	Um problema conhecido com o recurso Label Security da Oracle impede que os usuários com nomes de usuários de 16 ou 24 caracteres executem comandos do Label Security. Você pode criar um novo usuário com um número diferente de caracteres, conceder LBAC_DBA ao novo usuário, fazer login como esse novo usuário e executar os comandos do OLS como esse novo usuário. Se precisar de mais informações, entre em contato com o suporte da Oracle.

Oracle Locator

O Amazon RDS dá suporte ao Oracle Locator por meio do uso da opção `LOCATOR`. O Oracle Locator oferece recursos normalmente obrigatórios para dar suporte à Internet e a aplicativos baseados em serviço sem fio além de soluções GIS baseadas no parceiro. O Oracle Locator é um subconjunto limitado do Oracle Spatial. Para obter mais informações, consulte Oracle Locator na documentação da Oracle.

> **Important**
>
> Se você usar o Oracle Locator, o Amazon RDS atualizará automaticamente a instância de banco de dados para o Oracle PSU mais recente se houver vulnerabilidades de segurança com uma pontuação Common Vulnerability Scoring System (CVSS – Sistema de pontuação de vulnerabilidades comuns) superior a 9 ou outras vulnerabilidades de segurança anunciadas.

Versões de banco de dados compatíveis com o Oracle Locator

O RDS para Oracle é compatível com o Oracle Locator para o Oracle Database 19c. O Oracle Locator não é compatível com o Oracle Database 21c, mas sua funcionalidade está disponível na opção Oracle Spatial. Anteriormente, a opção Spatial exigia licenças adicionais. O Oracle Locator representava um subconjunto de recursos do Oracle Spatial e não exigia licenças adicionais. Em 2019, a Oracle anunciou que todos os recursos do Oracle Spatial haviam sido incluídos nas licenças Enterprise Edition e Standard Edition 2 sem custo adicional. Consequentemente, a opção Oracle Spatial não exigia mais licenciamento adicional. Para obter mais informações, consulte Machine Learning, Spatial and Graph - No License Required! no blog do Oracle Database Insider.

Pré-requisitos do Oracle Locator

Estes são os pré-requisitos para usar o Oracle Locator:

- A instância de banco de dados deve ser de classe suficiente. O Oracle Locator não é compatível com as classes de instância de banco de dados db.t3.small DB. Para ter mais informações, consulte Classes de instância de banco de dados do RDS para Oracle.
- A instância de banco de dados deve ter Auto Minor Version Upgrade (Atualização automática de versão secundária) habilitada. Essa opção permite que a instância de banco de dados receba atualizações de versões secundárias do mecanismo de banco de dados automaticamente,

quando disponibilizadas e é necessária para qualquer opção que instala o Oracle Java Virtual Machine (JVM). O Amazon RDS usa essa opção para atualizar a instância de banco de dados para a Oracle Patch Set Update (PSU) ou Release Update (RU) mais recente. Para obter mais informações, consulte Modificar uma instância de banco de dados do Amazon RDS.

Práticas recomendadas do Oracle Locator

Estas são as práticas recomendadas para usar o Oracle Locator:

- Para obter segurança máxima, use a opção `LOCATOR` com Secure Sockets Layer (SSL). Para obter mais informações, consulte Oracle Secure Sockets Layer.
- Configure a instância de banco de dados para restringir o acesso à instância de banco de dados. Para obter mais informações, consulte Cenários para acessar uma instância de banco de dados em uma VPC e Trabalhar com uma instância de banco de dados em uma VPC.

Adicionar a opção Oracle Locator

Este é o processo geral para adicionar a opção `LOCATOR` a uma instância de banco de dados:

1. Crie um novo grupo de opções, ou copie ou modifique um existente.
2. Adicione a opção ao grupo de opções.
3. Associe o grupo de opções à instância de banco de dados.

Se o Oracle Java Virtual Machine (JVM) não estiver instalado na instância de banco de dados, haverá uma breve interrupção enquanto a opção `LOCATOR` é adicionada. Não haverá interrupção se o Oracle Java Virtual Machine (JVM) já estiver instalado na instância de banco de dados. Depois de adicionar a opção , você não precisará reiniciar sua instância de banco de dados. Assim que o grupo de opções estiver ativo, o Oracle Locator estará disponível.

> **Note**
>
> Durante essa interrupção, as funções de verificação de senha são desativadas brevemente. Você também pode esperar ver eventos relacionados às funções de verificação de senha durante a interrupção. As funções de verificação de senha são ativadas novamente antes que a instância de banco de dados Oracle esteja disponível.

Para adicionar a opção **LOCATOR** a uma instância de banco de dados

1. Determine o grupo de opções que você deseja usar. Você pode criar um novo grupo de opções ou usar um existente. Se você quiser usar um grupo de opções existente, vá para a próxima etapa. Caso contrário, crie um grupo de opções de banco de dados personalizado com as seguintes configurações:

   a. Em Engine (Mecanismo), escolha a edição Oracle para sua instância de banco de dados.

   b. Para Major engine version (Versão do mecanismo principal), escolha a versão da sua instância de banco de dados.

   Para obter mais informações, consulte Criar um grupo de opções.

2. Adicione a opção LOCATOR ao grupo de opções. Para mais informações sobre a adição de opções, consulte Adicionar uma opção a um grupo de opções.

3. Aplique o grupo de opções a uma instância de banco de dados nova ou existente:

   - Para uma nova instância de banco de dados, você aplica o grupo de opções ao executar a instância. Para ter mais informações, consulte Criar uma instância de banco de dados do Amazon RDS.
   - Para uma instância de banco de dados existente, você aplica o grupo de opções modificando a instância e anexando o novo grupo de opções. Para obter mais informações, consulte Modificar uma instância de banco de dados do Amazon RDS.

Usar o Oracle Locator

Depois de habilitar a opção Oracle Locator, você poderá começar a usá-lo. Você deve usar somente recursos do Oracle Locator. Não use recursos do Oracle Spatial, a menos que você tenha uma licença para o Oracle Spatial.

Para obter uma lista de recursos compatíveis com o Oracle Locator, consulte Recursos incluídos com o Locator na documentação da Oracle.

Para obter uma lista de recursos não compatíveis com o Oracle Locator, consulte Recursos não incluídos com o Locator na documentação da Oracle.

Remover a opção Oracle Locator

Depois de descartar todos os objetos que usam tipos de dados fornecidos pela opção LOCATOR, você pode removê-los de uma instância de banco de dados. Se o Oracle Java Virtual Machine (JVM) não estiver instalado na instância de banco de dados, haverá uma breve interrupção enquanto a opção LOCATOR é removida. Não haverá interrupção se o Oracle Java Virtual Machine (JVM) já estiver instalado na instância de banco de dados. Depois de remover a opção LOCATOR, você não precisará reiniciar a instância de banco de dados.

Como descartar a opção **LOCATOR**

1. Faça backup de seus dados.

   > **Warning**
   >
   > Se a instância usar tipos de dados que foram ativados como parte da opção e se você remover a opção LOCATOR, você poderá perder dados. Para obter mais informações, consulte Backup, restauração e exportação de dados.

2. Verifique se os objetos existentes fazem referência a tipos de dados ou recursos da opção LOCATOR.

   Se existirem opções LOCATOR, a instância poderá ficar presa ao aplicar o novo grupo de opções que não tem a opção LOCATOR. Você pode identificar os objetos usando as seguintes consultas:

```
SELECT OWNER, SEGMENT_NAME, TABLESPACE_NAME, BYTES/1024/1024 mbytes
FROM   DBA_SEGMENTS
WHERE  SEGMENT_TYPE LIKE '%TABLE%'
AND    (OWNER, SEGMENT_NAME) IN
       (SELECT DISTINCT OWNER, TABLE_NAME
        FROM   DBA_TAB_COLUMNS
        WHERE  DATA_TYPE='SDO_GEOMETRY'
        AND    OWNER <> 'MDSYS')
ORDER BY 1,2,3,4;

SELECT OWNER, TABLE_NAME, COLUMN_NAME
FROM   DBA_TAB_COLUMNS
WHERE  DATA_TYPE = 'SDO_GEOMETRY'
AND    OWNER <> 'MDSYS'
ORDER BY 1,2,3;
```

3. Solte todos os objetos que fazem referência a tipos de dados ou recursos da opção `LOCATOR`.
4. Execute um destes procedimentos:

   - Remova a opção `LOCATOR` do grupo de opções ao qual ela pertence. Essa alteração afeta todas as instâncias de bancos de dados que usam o grupo de opções. Para obter mais informações, consulte Remover uma opção de um grupo de opções.
   - Modifique a instância de banco de dados e especifique um grupo de opções diferente que não inclua a opção `LOCATOR`. Essa alteração afeta uma única instância de banco de dados. Você pode especificar um grupo de opções padrão (vazio) ou criar um grupo de opções personalizado diferente. Para ter mais informações, consulte Modificar uma instância de banco de dados do Amazon RDS.

Oracle Native Network Encryption

O Amazon RDS oferece suporte para a Oracle Native Network Encryption (NNE). Com a opção `NATIVE_NETWORK_ENCRYPTION`, é possível criptografar dados à medida que eles entram e saem de uma instância de banco de dados. O Amazon RDS é compatível com a NNE para todas as edições do Oracle Database.

Uma discussão detalhada sobre a Oracle Native Network Encryption está além do escopo deste guia, mas você deve compreender os pontos fortes e fracos de cada algoritmo e chave antes de decidir sobre uma solução para a sua implantação. Para obter informações sobre os algoritmos e as chaves que estão disponíveis pela Oracle Native Network Encryption, consulte Configurar a criptografia de dados de rede, na documentação da Oracle. Para ter mais informações sobre a segurança da AWS, consulte a Central de Segurança da AWS.

> **Note**
>
> Você pode usar a Native Network Encryption ou a Secure Sockets Layer, mas não ambas. Para ter mais informações, consulte Oracle Secure Sockets Layer.

Tópicos

- Configurações da opção NATIVE_NETWORK_ENCRYPTION
- Adicionar a opção NATIVE_NETWORK_ENCRYPTION
- Como definir valores de NNE no sqlnet.ora
- Modificar a configurações da opção NATIVE_NETWORK_ENCRYPTION
- Remover a opção NATIVE_NETWORK_ENCRYPTION

Configurações da opção NATIVE_NETWORK_ENCRYPTION

Você pode especificar requisitos de criptografia no servidor e no cliente. A instância de banco de dados pode atuar como um cliente quando, por exemplo, usa um link de banco de dados para se conectar a outro banco de dados. Você pode preferir evitar forçar a criptografia no lado do servidor. Por exemplo, talvez você não queira forçar todas as comunicações do cliente a usar criptografia porque o servidor exige isso. Nesse caso, você pode forçar a criptografia no lado do cliente usando as opções do `SQLNET.*CLIENT`.

O Amazon RDS é compatível com as configurações a seguir da opção `NATIVE_NETWORK_ENCRYPTION`.

> **Note**
>
> Ao usar vírgulas para separar valores para uma configuração de opção, não coloque um espaço após a vírgula.

Configuração da opção	Valores válidos	Valores padrão	Descrição
`SQLNET.ALLOW_WEAK_CRYPTO_CLIENTS`	`TRUE`, `FALSE`	`TRUE`	O comportamento do servidor quando um cliente que usa uma cifra não segura tenta se conectar ao banco de dados. Se for `TRUE`, os clientes poderão se conectar mesmo que não sejam corrigidos com a PSU de julho de 2021. Se a configuração for `FALSE`, os clientes poderão se conectar ao banco de dados somente quando forem corrigidos com a PSU de julho de 2021. Antes de definir `SQLNET.ALLOW_WEAK_CRYPTO_CLIENTS` para `FALSE`, certifique-se de que as seguintes condições sejam atendidas: • `SQLNET.ENCRYPTION_TYPES_SERVER` e `SQLNET.ENCRYPTION_TYPES_CLIENT` têm um método de criptografia correspondente que não é `DES`, `3DES` ou `RC4` (todos os comprimentos de chaves). • `SQLNET.CHECKSUM_TYPES_SERVER` e `SQLNET.CHECKSUM_TYPES_CLIENT` têm

Configuração da opção	Valores válidos	Valores padrão	Descrição
			um método de soma de verificação segura correspondente que não é `MD5`. • O cliente é corrigido com a PSU de julho de 2021. Se o cliente não for corrigido, o cliente perderá a conexão e receberá o erro `ORA-12269` .

Configuração da opção	Valores válidos	Valores padrão	Descrição
SQLNET.ALLOW_WEAK_CRYPTO	TRUE, FALSE	TRUE	O comportamento do servidor quando um cliente que usa uma cifra não segura tenta se conectar ao banco de dados. As seguintes cifras são consideradas não seguras: • Método de criptografia DES (todos os comprimentos de chave) • Método de criptografia 3DES (todos os comprimentos de chave) • Método de criptografia RC4 (todos os comprimentos de chave) • Método de soma de verificação MD5 Se a configuração for TRUE, os clientes poderão se conectar quando usarem as cifras não seguras anteriores. Se a configuração for FALSE, o banco de dados impedirá os clientes e se conectarem quando usarem as cifras não seguras anteriores. Antes de definir SQLNET.ALLOW_WEAK_CRYPTO para FALSE, certifique-se de que as seguintes condições sejam atendidas: • SQLNET.ENCRYPTION_TYPES_SERVER e SQLNET.ENCRYPTION_TYPES_CLIENT têm um método de criptografia correspondente que não é DES, 3DES ou RC4 (todos os comprimentos de chaves).

Configuração da opção	Valores válidos	Valores padrão	Descrição
			• SQLNET.CHECKSUM_TYPES_SERVER e SQLNET.CHECKSUM_TYPES_CLIENT têm um método de soma de verificação segura correspondente que não é MD5. • O cliente é corrigido com a PSU de julho de 2021. Se o cliente não for corrigido, o cliente perderá a conexão e receberá o erro ORA-12269 .
SQLNET.CRYPTO_CHECKSUM_CLIENT	Accepted Rejected Requested , Required	Requested	O comportamento de integridade de dados quando uma instância de banco de dados se conecta ao cliente ou um servidor atuando como um cliente. Quando uma instância de banco de dados usa um link de banco de dados, ela atua como um cliente. Requested indica que o cliente não exige que a instância de banco de dados realize uma soma de verificação.
SQLNET.CRYPTO_CHECKSUM_SERVER	Accepted Rejected Requested , Required	Requested	O comportamento de integridade de dados quando um cliente, ou um servidor atuando como um cliente, conecta-se à instância de banco de dados. Quando uma instância de banco de dados usa um link de banco de dados, ela atua como um cliente. Requested indica que a instância de banco de dados não exige que o cliente faça uma soma de verificação.

Configuração da opção	Valores válidos	Valores padrão	Descrição
SQLNET.CRYPTO_CHECKSUM_TYPES_CLIENT	SHA256, SHA384, SHA512, SHA1, MD5	SHA256, SHA384, SHA512	Uma lista de algoritmos de soma de verificação. É possível especificar um valor ou uma lista de valores separados por vírgulas. Se você usar uma vírgula, não insira um espaço após a vírgula; caso contrário, você receberá um erro InvalidParameterValue . Este parâmetro e o SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER devem ter uma cifra comum.
SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER	SHA256, SHA384, SHA512, SHA1, MD5	SHA256, SHA384, SHA512, SHA1, MD5	Uma lista de algoritmos de soma de verificação. É possível especificar um valor ou uma lista de valores separados por vírgulas. Se você usar uma vírgula, não insira um espaço após a vírgula; caso contrário, você receberá um erro InvalidParameterValue . Este parâmetro e o SQLNET.CRYPTO_CHECKSUM_TYPES_CLIENT devem ter uma cifra comum.

Configuração da opção	Valores válidos	Valores padrão	Descrição
SQLNET.ENCRYPTION_CLIENT	Accepted Rejected Requested , Required	Requested	O comportamento de criptografia do cliente quando um cliente, ou um servidor atuando como cliente, se conecta à instância de banco de dados. Quando uma instância de banco de dados usa um link de banco de dados, ela atua como um cliente. Requested indica que o cliente não exige que o tráfego do servidor seja criptografado.
SQLNET.ENCRYPTION_SERVER	Accepted Rejected Requested , Required	Requested	O comportamento de criptografia do servidor quando um cliente, ou um servidor atuando como cliente, se conecta à instância de banco de dados. Quando uma instância de banco de dados usa um link de banco de dados, ela atua como um cliente. Requested indica que a instância de banco de dados não requer que o tráfego a partir do cliente seja criptografado.

Configuração da opção	Valores válidos	Valores padrão	Descrição
SQLNET.ENCRYPTION_TYPES_CLIENT	RC4_256, AES256, AES192, 3DES168, RC4_128, AES128, 3DES112, RC4_56, DES, RC4_40, DES40	RC4_256, AES256, AES192, 3DES168, RC4_128, AES128, 3DES112, RC4_56, DES, RC4_40, DES40	Uma lista de algoritmos de criptografia usados pelo cliente. Para descriptografar a entrada do servidor, o cliente experimenta cada algoritmo, em ordem, até que um deles seja bem-sucedido ou até atingir o final da lista. O Amazon RDS usa a seguinte lista padrão do Oracle. O RDS inicia com RC4_256 e segue a sequência na lista. Você pode alterar a ordem ou limitar os algoritmos que a instância de banco de dados aceitará. 1. RC4_256: RSA RC4 (tamanho da chave de 256 bits) 2. AES256: AES (tamanho da chave de 256 bits) 3. AES192: AES (tamanho da chave de 192 bits) 4. 3DES168: DES triplo de 3 chaves (tamanho da chave efetivo de 112 bits) 5. RC4_128: RSA RC4 (tamanho da chave de 128 bits) 6. AES128: AES (tamanho da chave de 128 bits) 7. 3DES112: DES triplo de 2 chaves (tamanho da chave efetivo de 80 bits) 8. RC4_56: RSA RC4 (tamanho da chave de 56 bits)

Configuração da opção	Valores válidos	Valores padrão	Descrição
			9. `DES`: DES padrão (tamanho da chave de 56 bits) 10 `RC4_40`: RSA RC4 (tamanho da chave de 40 bits) 11 `DES40`: DES40 (tamanho da chave de 40 bits) É possível especificar um valor ou uma lista de valores separados por vírgulas. Se você usar uma vírgula, não insira um espaço após a vírgula; caso contrário, você receberá um erro `InvalidParameterValue` . Este parâmetro e o `SQLNET.SQLNET.ENCRYPTION_TYPES_SERVER` devem ter uma cifra comum.

Configuração da opção	Valores válidos	Valores padrão	Descrição
SQLNET.ENCRYPTION_TYPES_SERVER	RC4_256, AES256, AES192, 3DES168, RC4_128, AES128, 3DES112, RC4_56, DES, RC4_40, DES40	RC4_256, AES256, AES192, 3DES168, RC4_128, AES128, 3DES112, RC4_56, DES, RC4_40, DES40	Uma lista de algoritmos de criptografia usados pela instância de banco de dados. A instância de banco de dados usa cada algoritmo, em ordem, para tentar descriptografar a entrada do cliente até que um algoritmo seja bem-sucedido ou até atingir o final da lista. O Amazon RDS usa a seguinte lista padrão do Oracle. Você pode alterar a ordem ou restringir os algoritmos que o cliente aceitará. 1. RC4_256: RSA RC4 (tamanho da chave de 256 bits) 2. AES256: AES (tamanho da chave de 256 bits) 3. AES192: AES (tamanho da chave de 192 bits) 4. 3DES168: DES triplo de 3 chaves (tamanho da chave efetivo de 112 bits) 5. RC4_128: RSA RC4 (tamanho da chave de 128 bits) 6. AES128: AES (tamanho da chave de 128 bits) 7. 3DES112: DES triplo de 2 chaves (tamanho da chave efetivo de 80 bits) 8. RC4_56: RSA RC4 (tamanho da chave de 56 bits)

Configuração da opção	Valores válidos	Valores padrão	Descrição
			9. `DES`: DES padrão (tamanho da chave de 56 bits) 10 `RC4_40`: RSA RC4 (tamanho da chave de 40 bits) 11 `DES40`: DES40 (tamanho da chave de 40 bits) É possível especificar um valor ou uma lista de valores separados por vírgulas. Se você usar uma vírgula, não insira um espaço após a vírgula; caso contrário, você receberá um erro `InvalidParameterValue` . Este parâmetro e o `SQLNET.SQLNET.ENCRYPTION_TYPES_SERVER` devem ter uma cifra comum.

Adicionar a opção NATIVE_NETWORK_ENCRYPTION

Este é o processo geral para adicionar a opção `NATIVE_NETWORK_ENCRYPTION` a uma instância de banco de dados:

1. Crie um novo grupo de opções, ou copie ou modifique um existente.
2. Adicione a opção ao grupo de opções.
3. Associe o grupo de opções à instância de banco de dados.

Quando o grupo de opções estiver ativo, a NNE ficará ativa.

Como adicionar a opção NATIVE_NETWORK_ENCRYPTION a uma instância de banco de dados usando o AWS Management Console

1. Para Engine (Mecanismo), escolha a edição do Oracle que deseja usar. NNE é compatível em todas as edições.
2. Para Major engine version (Versão do mecanismo principal), escolha a versão da sua instância de banco de dados.

   Para ter mais informações, consulte Criar um grupo de opções.
3. Adicione a opção NATIVE_NETWORK_ENCRYPTION ao grupo de opções. Para ter mais informações sobre a adição de opções, consulte Adicionar uma opção a um grupo de opções.

   > **Note**
   >
   > Depois de adicionar a opção NATIVE_NETWORK_ENCRYPTION, você não precisará reiniciar as instâncias de banco de dados. Assim que o grupo de opções estiver ativo, o NNE ficará ativo.
4. Aplique o grupo de opções a uma instância de banco de dados nova ou existente:

   - Para uma nova instância de banco de dados, você aplica o grupo de opções ao executar a instância. Para ter mais informações, consulte Criar uma instância de banco de dados do Amazon RDS.
   - Para uma instância de banco de dados existente, você aplica o grupo de opções modificando a instância e anexando o novo grupo de opções. Depois de adicionar a opção NATIVE_NETWORK_ENCRYPTION, você não precisará reiniciar a instância de banco de dados. Assim que o grupo de opções estiver ativo, o NNE ficará ativo. Para ter mais informações, consulte Modificar uma instância de banco de dados do Amazon RDS.

Como definir valores de NNE no sqlnet.ora

Com a Oracle Native Network Encryption, você pode configurar a criptografia de rede no lado do servidor e no lado do cliente. O cliente é o computador usado para realizar a conexão com a instância de banco de dados. Você pode especificar as seguintes configurações de cliente no slqnet.ora:

- `SQLNET.ALLOW_WEAK_CRYPTO`

- SQLNET.ALLOW_WEAK_CRYPTO_CLIENTS
- SQLNET.CRYPTO_CHECKSUM_CLIENT
- SQLNET.CRYPTO_CHECKSUM_TYPES_CLIENT
- SQLNET.ENCRYPTION_CLIENT
- SQLNET.ENCRYPTION_TYPES_CLIENT

Para obter informações, consulte Configurar a criptografia de dados da rede e a integridade para clientes e servidores Oracle na documentação da Oracle.

Às vezes, a instância de banco de dados rejeita uma solicitação de conexão de uma aplicação. Por exemplo, uma rejeição pode ocorrer quando os algoritmos de criptografia no cliente e no servidor não correspondem. Para testar a Oracle Native Network Encryption, adicione as seguintes linhas ao arquivo sqlnet.ora no cliente:

```
DIAG_ADR_ENABLED=off
TRACE_DIRECTORY_CLIENT=/tmp
TRACE_FILE_CLIENT=nettrace
TRACE_LEVEL_CLIENT=16
```

Quando ocorre uma tentativa de conexão, as linhas anteriores geram um arquivo de rastreamento no cliente chamado `/tmp/nettrace*`. O arquivo de rastreamento contém informações sobre a conexão. Para ter mais informações sobre problemas de conexão quando você estiver usando a Oracle Native Network Encryption, consulte About negotiating encryption and integrity (Sobre a negociação da criptografia e a integridade) na documentação do Oracle Database.

Modificar a configurações da opção NATIVE_NETWORK_ENCRYPTION

Depois de habilitar a opção `NATIVE_NETWORK_ENCRYPTION`, é possível modificar as configurações. Atualmente, é possível modificar as configurações da opção `NATIVE_NETWORK_ENCRYPTION` somente com a AWS CLI ou a API do RDS. Não é possível usar o console. O exemplo a seguir modifica duas configurações na opção.

```
aws rds add-option-to-option-group \
    --option-group-name my-option-group \
    --options
 "OptionName=NATIVE_NETWORK_ENCRYPTION,OptionSettings=[{Name=SQLNET.CRYPTO_CHECKSUM_TYPES_SERVE
{Name=SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER,Value=SHA256}]" \
```

```
    --apply-immediately
```

Para saber como modificar as configurações da opção usando a CLI, consulte AWS CLI. Para ter mais informações sobre cada configuração, consulte Configurações da opção NATIVE_NETWORK_ENCRYPTION.

Tópicos

- Modificar valores de CRYPTO_CHECKSUM_*
- Modificar as configurações de ALLOW_WEAK_CRYPTO*

Modificar valores de CRYPTO_CHECKSUM_*

Se você modificar as configurações de opções de NATIVE_NETWORK_ENCRYPTION, as seguintes configurações de opções devem ter pelo menos uma cifra comum:

- `SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER`
- `SQLNET.CRYPTO_CHECKSUM_TYPES_CLIENT`

O exemplo a seguir mostra um cenário no qual você modifica `SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER`. A configuração é válida porque `CRYPTO_CHECKSUM_TYPES_CLIENT` e `CRYPTO_CHECKSUM_TYPES_SERVER` usam `SHA256`.

Configuração da opção	Valores antes da modificação	Valores após modificação
`SQLNET.CRYPTO_CHECKSUM_TYPES_CLIENT`	**`SHA256`**, `SHA384`, `SHA512`	Sem alterações
`SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER`	**`SHA256`**, `SHA384`, `SHA512`, `SHA1`, `MD5`	`SHA1,MD5,` **`SHA256`**

Como outro exemplo, suponha que você queira modificar `SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER` de sua configuração padrão para `SHA1,MD5`. Nesse caso, certifique-se de que tenha definido `SQLNET.CRYPTO_CHECKSUM_TYPES_CLIENT` como `SHA1` ou `MD5`. Esses algoritmos não são incluídos nos valores padrão para `SQLNET.CRYPTO_CHECKSUM_TYPES_CLIENT`.

Modificar as configurações de ALLOW_WEAK_CRYPTO*

Para definir as opções SQLNET.ALLOW_WEAK_CRYPTO* do valor padrão para FALSE, certifique-se de que as seguintes condições sejam atendidas:

- SQLNET.ENCRYPTION_TYPES_SERVER e SQLNET.ENCRYPTION_TYPES_CLIENT têm um método de criptografia segura correspondente. Um método é considerado seguro se não for DES, 3DES ou RC4 (todos os comprimentos de chaves).
- SQLNET.CHECKSUM_TYPES_SERVER e SQLNET.CHECKSUM_TYPES_CLIENT têm um método de soma de verificação segura correspondente. Um método é considerado seguro se não for MD5.
- O cliente é corrigido com a PSU de julho de 2021. Se o cliente não for corrigido, o cliente perderá a conexão e receberá o erro ORA-12269.

O exemplo a seguir mostra as configurações de NNE de exemplo. Suponha que você queira definir SQLNET.ENCRYPTION_TYPES_SERVER e SQLNET.ENCRYPTION_TYPES_CLIENT para FALSE, bloqueando conexões não seguras. As configurações da opção de soma de verificação atendem aos pré-requisitos porque ambas têm SHA256. No entanto, SQLNET.ENCRYPTION_TYPES_CLIENT e SQLNET.ENCRYPTION_TYPES_SERVER usam os métodos de criptografia DES, 3DES e RC4, que não são seguros. Portanto, para definir as opções de SQLNET.ALLOW_WEAK_CRYPTO* para FALSE, primeiro defina SQLNET.ENCRYPTION_TYPES_SERVER e SQLNET.ENCRYPTION_TYPES_CLIENT para um método de criptografia seguro, como AES256.

Configuração da opção	Valores
SQLNET.CRYPTO_CHECKSUM_TYPES_CLIENT	SHA256, SHA384, SHA512
SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER	SHA1,MD5,SHA256
SQLNET.ENCRYPTION_TYPES_CLIENT	RC4_256, 3DES168, DES40
SQLNET.ENCRYPTION_TYPES_SERVER	RC4_256, 3DES168, DES40

Remover a opção NATIVE_NETWORK_ENCRYPTION

Você pode remover o NNE de uma instância de banco de dados.

Para remover a opção `NATIVE_NETWORK_ENCRYPTION` de uma instância de banco de dados, faça o seguinte:

- Para remover a opção de várias instâncias de bancos de dados, remova a opção `NATIVE_NETWORK_ENCRYPTION` do grupo de opções ao qual elas pertencem. Essa alteração afeta todas as instâncias de bancos de dados que usam o grupo de opções. Depois de remover a opção `NATIVE_NETWORK_ENCRYPTION`, você não precisará reiniciar as instâncias de banco de dados. Para ter mais informações, consulte Remover uma opção de um grupo de opções.
- Para remover a opção de uma única instância de banco de dados, modifique essa instância e especifique um grupo de opções diferente que não inclua a opção `NATIVE_NETWORK_ENCRYPTION`. Você pode especificar um grupo de opções padrão (vazio) ou criar um grupo de opções personalizado diferente. Depois de remover a opção `NATIVE_NETWORK_ENCRYPTION`, você não precisará reiniciar a instância de banco de dados. Para ter mais informações, consulte Modificar uma instância de banco de dados do Amazon RDS.

Oracle OLAP

O Amazon RDS oferece suporte ao Oracle OLAP por meio do uso da opção OLAP. Essa opção fornece OLAP (On-line Analytical Processing, Processamento analítico online) para instâncias de banco de dados Oracle. É possível usar o Oracle OLAP para analisar grandes quantidades de dados criando cubos e objetos dimensionais de acordo com o padrão OLAP. Para obter mais informações, consulte a documentação da Oracle.

> **Important**
>
> Se você usar o Oracle OLAP, o Amazon RDS atualizará automaticamente a instância de banco de dados para o Oracle PSU mais recente se houver vulnerabilidades de segurança com uma pontuação Common Vulnerability Scoring System (CVSS – Sistema de pontuação de vulnerabilidades comuns) superior a 9 ou outras vulnerabilidades de segurança anunciadas.

O Amazon RDS é compatível com o Oracle OLAP para a Enterprise Edition do Oracle Database 19c e posterior.

Pré-requisitos para o Oracle OLAP

Veja a seguir os pré-requisitos para usar o Oracle OLAP:

- É necessário ter uma licença do Oracle OLAP da Oracle. Para obter mais informações, consulte Licensing Information na documentação da Oracle.
- A instância de banco de dados deve ser de uma classe de instância suficiente. O Oracle OLAP não é compatível com as classes de instância de banco de dados db.t3.small. Para ter mais informações, consulte Classes de instância de banco de dados do RDS para Oracle.
- A instância de banco de dados deve ter Auto Minor Version Upgrade (Atualização automática de versão secundária) habilitada. Essa opção permite que a instância de banco de dados receba atualizações de versões secundárias do mecanismo de banco de dados automaticamente, quando disponibilizadas e é necessária para qualquer opção que instala o Oracle Java Virtual Machine (JVM). O Amazon RDS usa essa opção para atualizar a instância de banco de dados para a Oracle Patch Set Update (PSU) ou Release Update (RU) mais recente. Para obter mais informações, consulte Modificar uma instância de banco de dados do Amazon RDS.
- A instância de banco de dados não deve ter um usuário chamado OLAPSYS. Se isso acontecer, a instalação da opção OLAP falhará.

Práticas recomendadas do Oracle OLAP

Estas são as práticas recomendadas para usar o Oracle OLAP:

- Para obter segurança máxima, use a opção `OLAP` com Secure Sockets Layer (SSL). Para obter mais informações, consulte Oracle Secure Sockets Layer.
- Configure a instância de banco de dados para restringir o acesso à instância de banco de dados. Para obter mais informações, consulte Cenários para acessar uma instância de banco de dados em uma VPC e Trabalhar com uma instância de banco de dados em uma VPC.

Como adicionar a opção do Oracle OLAP

Este é o processo geral para adicionar a opção `OLAP` a uma instância de banco de dados:

1. Crie um novo grupo de opções, ou copie ou modifique um existente.
2. Adicione a opção ao grupo de opções.
3. Associe o grupo de opções à instância de banco de dados.

Se o Oracle Java Virtual Machine (JVM) não estiver instalado na instância de banco de dados, haverá uma breve interrupção enquanto a opção `OLAP` é adicionada. Não haverá interrupção se o Oracle Java Virtual Machine (JVM) já estiver instalado na instância de banco de dados. Depois de adicionar a opção , você não precisará reiniciar sua instância de banco de dados. Assim que o grupo de opções estiver ativo, o Oracle OLAP estará disponível.

Como adicionar a opção do OLAP a uma instância de banco de dados

1. Determine o grupo de opções que você deseja usar. Você pode criar um novo grupo de opções ou usar um existente. Se você quiser usar um grupo de opções existente, vá para a próxima etapa. Caso contrário, crie um grupo de opções de banco de dados personalizado com as seguintes configurações:

   - Em Engine (Mecanismo), escolha a edição Oracle para sua instância de banco de dados.
   - Para Major engine version (Versão do mecanismo principal), escolha a versão da sua instância de banco de dados.

   Para obter mais informações, consulte Criar um grupo de opções.

2. Adicione a opção OLAP ao grupo de opções. Para mais informações sobre a adição de opções, consulte Adicionar uma opção a um grupo de opções.
3. Aplique o grupo de opções a uma instância de banco de dados nova ou existente:
   - Para uma nova instância de banco de dados, aplique o grupo de opções ao executar a instância. Para obter mais informações, consulte Criar uma instância de banco de dados do Amazon RDS.
   - Para uma instância de banco de dados existente, aplique o grupo de opções modificando a instância e anexando o novo grupo de opções. Para obter mais informações, consulte Modificar uma instância de banco de dados do Amazon RDS.

Como usar o Oracle OLAP

Depois de habilitar a opção Oracle OLAP, será possível começar a usá-lo. Para obter uma lista de recursos compatíveis com o Oracle OLAP, consulte a documentação da Oracle.

Como remover a opção do Oracle OLAP

Depois de descartar todos os objetos que usam tipos de dados fornecidos pela opção `OLAP`, você pode removê-los de uma instância de banco de dados. Se o Oracle Java Virtual Machine (JVM) não estiver instalado na instância de banco de dados, haverá uma breve interrupção enquanto a opção `OLAP` é removida. Não haverá interrupção se o Oracle Java Virtual Machine (JVM) já estiver instalado na instância de banco de dados. Depois de remover a opção `OLAP`, você não precisará reiniciar a instância de banco de dados.

Como descartar a opção **`OLAP`**

1. Faça backup de seus dados.

   > **Warning**
   >
   > Se a instância usar tipos de dados que foram ativados como parte da opção e se você remover a opção `OLAP`, você poderá perder dados. Para obter mais informações, consulte Backup, restauração e exportação de dados.

2. Verifique se os objetos existentes fazem referência a tipos de dados ou recursos da opção `OLAP`.
3. Solte todos os objetos que fazem referência a tipos de dados ou recursos da opção `OLAP`.

4. Execute um destes procedimentos:

   - Remova a opção 0LAP do grupo de opções ao qual ela pertence. Essa alteração afeta todas as instâncias de bancos de dados que usam o grupo de opções. Para obter mais informações, consulte Remover uma opção de um grupo de opções.
   - Modifique a instância de banco de dados e especifique um grupo de opções diferente que não inclua a opção 0LAP. Essa alteração afeta uma única instância de banco de dados. Você pode especificar um grupo de opções padrão (vazio) ou criar um grupo de opções personalizado diferente. Para ter mais informações, consulte Modificar uma instância de banco de dados do Amazon RDS.

Oracle Secure Sockets Layer

Para habilitar a criptografia SSL para uma instância de banco de dados do RDS para Oracle, adicione a opção Oracle SSL ao grupo de opções associado à instância de banco de dados. O Amazon RDS usa uma segunda porta, conforme exigido pela Oracle, para conexões SSL. Essa abordagem permite que comunicações de texto sem formatação e de texto com criptografia SSL ocorram ao mesmo tempo entre uma instância de banco de dados e o SQL*Plus. Por exemplo, você pode usar a porta com comunicação de texto simples para se comunicar com outros recursos dentro de uma VPC enquanto usa a porta com comunicação criptografada em SSL para se comunicar com recursos fora da VPC.

> **Note**
>
> É possível usar SSL ou a Native Network Encryption (NNE) na mesma instância de banco de dados do RDS para Oracle, mas não ambas. Se você usar a criptografia SSL, desative qualquer outra criptografia de conexão. Para ter mais informações, consulte Oracle Native Network Encryption.

SSL/TLS e NNE não fazem mais parte do Oracle Advanced Security. No RDS para Oracle, é possível usar a criptografia SSL com todas as edições licenciadas das seguintes versões do banco de dados:

- Oracle Database 21c (21.0.0)
- Oracle Database 19c (19.0.0)

Tópicos

- Versões do TLS para a opção Oracle SSL
- Pacotes de criptografia para a opção Oracle SSL
- Suporte ao FIPS
- Adicionar a opção do SSL
- Configurar o SQL*Plus para usar SSL com uma instância de banco de dados do RDS para Oracle
- Conectar-se a uma instância de banco de dados do RDS para Oracle usando SSL
- Configurar uma conexão SSL via JDBC
- Forçar uma correspondência de DN com uma conexão SSL

- Solução de problemas de conexões SSL

Versões do TLS para a opção Oracle SSL

O Amazon RDS for Oracle oferece suporte ao Transport Layer Security (TLS) versões 1.0 e 1.2. Quando você adiciona uma nova opção Oracle SSL, defina `SQLNET.SSL_VERSION` explicitamente como um valor válido. Os seguintes valores são permitidos para essa configuração de opção:

- `"1.0"`: os clientes só podem se conectar à instância de banco de dados usando o TLS versão 1.0. Para opções Oracle SSL existentes, a `SQLNET.SSL_VERSION` é definida como `"1.0"` automaticamente. Você pode alterar a configuração se necessário.
- `"1.2"` – os clientes podem se conectar à instância de banco de dados usando somente o TLS 1.2.
- `"1.2 or 1.0"` – os clientes podem se conectar à instância de banco de dados usando o TLS 1.2 ou 1.0.

Pacotes de criptografia para a opção Oracle SSL

O Amazon RDS for Oracle oferece suporte para vários pacotes de criptografia SSL.
Por padrão, a opção Oracle SSL está configurada para usar o pacote de criptografia `SSL_RSA_WITH_AES_256_CBC_SHA`. Para especificar um pacote de criptografia diferente para usar em conexões SSL, use a configuração da opção `SQLNET.CIPHER_SUITE`.

É possível especificar vários valores para `SQLNET.CIPHER_SUITE`. Essa técnica é útil caso você tenha links de banco de dados entre suas instâncias de banco de dados e decida atualizar seus pacotes de criptografia.

A tabela a seguir resume o suporte a SSL no RDS para Oracle em todas as edições do Oracle Database 19c e 21c.

Suíte de cifras (SQLNET.CIPHER_SUITE)	Versão do TLS compatível (SQLNET.SSL_VERSION)	Suporte ao FIPS	Conformidade com FedRAMP
SSL_RSA_WITH_AES_256_CBC_SHA (padrão)	1.0 e 1.2	Sim	Não

Suíte de cifras (SQLNET.CIPHER_SUITE)	Versão do TLS compatível (SQLNET.SSL_VERSION)	Suporte ao FIPS	Conformidade com FedRAMP
SSL_RSA_WITH_AES_256_CBC_SHA256	1.2	Sim	Não
SSL_RSA_WITH_AES_256_GCM_SHA384	1.2	Sim	Não
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	1.2	Sim	Sim
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	1.2	Sim	Sim
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384	1.2	Sim	Sim
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256	1.2	Sim	Sim
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	1.2	Sim	Sim
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	1.2	Sim	Sim

Suporte ao FIPS

O RDS para Oracle permite usar o padrão FIPS (Federal Information Processing Standard) para 140-2. O FIPS 140-2 é um padrão do governo dos Estados Unidos que define os requisitos de segurança de módulos criptográficos. Ative o padrão FIPS definindo `FIPS.SSLFIPS_140` como `TRUE` para a opção do Oracle SSL. Quando o FIPS 140-2 é configurado para SSL, as bibliotecas criptográficas criptografam dados entre o cliente e a instância de banco de dados do RDS para Oracle.

Os clientes devem usar o pacote de criptografia compatível com o FIPS. Ao estabelecer uma conexão, o cliente e a instância de banco de dados do RDS para Oracle negociam qual pacote de criptografia deverá ser usado ao transmitir mensagens nas duas direções. A tabela em Pacotes de

criptografia para a opção Oracle SSL mostra os pacotes de criptografia SSL compatíveis com o FIPS para cada versão do TLS. Para receber mais informações, consulte Oracle Database FIPS 140-2 Settings na documentação da Oracle.

Adicionar a opção do SSL

Para usar SSL, sua instância de banco de dados do RDS para Oracle deve estar associada a um grupo de opções que inclua a opção `SSL`.

Console

Para adicionar a opção SSL a um grupo de opções

1. Crie um novo grupo de opções ou identifique um grupo de opções existente ao qual você pode adicionar a opção `SSL`.

   Para obter informações sobre como criar um grupo de opções, consulte Criar um grupo de opções.
2. Adicione a opção `SSL` ao grupo de opções.

   Se você quiser usar somente pacotes de criptografia verificados para o FIPS para conexões SSL, defina a opção `FIPS.SSLFIPS_140` como `TRUE`. Para obter informações sobre o padrão FIPS, consulte Suporte ao FIPS.

   Para obter informações sobre como adicionar uma opção a um grupo de opções, consulte Adicionar uma opção a um grupo de opções.
3. Crie uma instância de banco de dados do RDS para Oracle e associe a ela o grupo de opções ou modifique uma instância de banco de dados do RDS para Oracle para associar o grupo de opções a ela.

   Para obter informações sobre como criar uma instância de banco de dados, consulte Criar uma instância de banco de dados do Amazon RDS.

   Para ter mais informações sobre a modificação de uma instância de banco de dados, consulte Modificar uma instância de banco de dados do Amazon RDS.

AWS CLI

Para adicionar a opção SSL a um grupo de opções

1. Crie um novo grupo de opções ou identifique um grupo de opções existente ao qual você pode adicionar a opção `SSL`.

   Para obter informações sobre como criar um grupo de opções, consulte Criar um grupo de opções.

2. Adicione a opção `SSL` ao grupo de opções.

   Especifique as seguintes configurações de opção:

   - `Port` – O número da porta SSL
   - `VpcSecurityGroupMemberships` – O grupo de segurança da VPC para o qual a opção está ativada
   - `SQLNET.SSL_VERSION` – A versão do TLS que o cliente pode usar para se conectar à instância de banco de dados

   Por exemplo, o seguinte comando da AWS CLI adiciona a opção `SSL` a um grupo de opções denominado `ora-option-group`.

   Example

   Para Linux, macOS ou Unix:

```
aws rds add-option-to-option-group --option-group-name ora-option-group \
  --options
 'OptionName=SSL,Port=2484,VpcSecurityGroupMemberships="sg-68184619",OptionSettings=[{Name=
```

   Para Windows:

```
aws rds add-option-to-option-group --option-group-name ora-option-group ^
  --options
 'OptionName=SSL,Port=2484,VpcSecurityGroupMemberships="sg-68184619",OptionSettings=[{Name=
```

3. Crie uma instância de banco de dados do RDS para Oracle e associe a ela o grupo de opções ou modifique uma instância de banco de dados do RDS para Oracle para associar o grupo de opções a ela.

   Para obter informações sobre como criar uma instância de banco de dados, consulte Criar uma instância de banco de dados do Amazon RDS.

   Para ter mais informações sobre a modificação de uma instância de banco de dados, consulte Modificar uma instância de banco de dados do Amazon RDS.

Configurar o SQL*Plus para usar SSL com uma instância de banco de dados do RDS para Oracle

É necessário configurar o SQL*Plus para se conectar a uma instância de banco de dados do RDS para Oracle que use a opção Oracle SSL.

> **Note**
>
> Para permitir o acesso à instância de banco de dados a partir dos clientes apropriados, verifique se os grupos de segurança estão configurados corretamente. Para ter mais informações, consulte Controlar acesso com grupos de segurança. Além disso, essas instruções são para o SQL*Plus e outros clientes que usam diretamente um Oracle Home. Para conexões JDBC, consulte Configurar uma conexão SSL via JDBC.

Para configurar o SQL*Plus para usar SSL para se conectar a uma instância de banco de dados do RDS para Oracle

1. Defina a variável de ambiente `ORACLE_HOME` como o local do diretório inicial do Oracle.

   O caminho para o diretório inicial do Oracle depende da instalação. O exemplo a seguir define a variável de ambiente `ORACLE_HOME`.

   ```
   prompt>export ORACLE_HOME=/home/user/app/user/product/19.0.0/dbhome_1
   ```

   Para obter informações sobre como definir variáveis de ambiente do Oracle, consulte Variáveis de ambiente do SQL*Plus na documentação da Oracle e consulte também o guia de instalação da Oracle para o sistema operacional.

2. Adicione a variável de ambiente `$ORACLE_HOME/lib` ao `LD_LIBRARY_PATH`.

   O exemplo a seguir define a variável de ambiente LD_LIBRARY_PATH.

   ```
   prompt>export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ORACLE_HOME/lib
   ```

3. Crie um diretório para o Oracle Wallet em `$ORACLE_HOME/ssl_wallet`.

   Este é um exemplo que cria o diretório do Oracle Wallet.

   ```
   prompt>mkdir $ORACLE_HOME/ssl_wallet
   ```

4. Baixe o pacote do certificado .pem que funciona para todas as Regiões da AWS e coloque o arquivo no diretório ssl_wallet. Para ter mais informações, consulte Usar SSL/TLS para criptografar uma conexão com uma instância ou um cluster de banco de dados.

5. No diretório `$ORACLE_HOME/network/admin`, modifique ou crie o arquivo `tnsnames.ora` e inclua a entrada a seguir.

   ```
   net_service_name =
     (DESCRIPTION =
       (ADDRESS_LIST =
         (ADDRESS =
           (PROTOCOL = TCPS)
           (HOST = endpoint)
           (PORT = ssl_port_number)
         )
       )
       (CONNECT_DATA =
         (SID = database_name)
       )
       (SECURITY =
         (SSL_SERVER_CERT_DN =
    "C=US,ST=Washington,L=Seattle,O=Amazon.com,OU=RDS,CN=endpoint")
       )
     )
   ```

6. No mesmo diretório, modifique ou crie o arquivo sqlnet.ora e inclua os seguintes parâmetros.

> **Note**
>
> Para se comunicar com entidades por meio de uma conexão protegida por TLS, a Oracle requer uma carteira com os certificados necessários para autenticação. Você pode usar o utilitário ORAPKI da Oracle para criar e manter carteiras Oracle, conforme mostrado na etapa 7. Para ter mais informações, consulte Configurar o Oracle Wallet usando ORAPKI na documentação da Oracle.

```
WALLET_LOCATION = (SOURCE = (METHOD = FILE) (METHOD_DATA = (DIRECTORY =
 $ORACLE_HOME/ssl_wallet)))
SSL_CLIENT_AUTHENTICATION = FALSE
SSL_VERSION = 1.0
SSL_CIPHER_SUITES = (SSL_RSA_WITH_AES_256_CBC_SHA)
SSL_SERVER_DN_MATCH = ON
```

> **Note**
>
> Você poderá definir `SSL_VERSION` como um valor mais alto se sua instância de banco de dados for compatível.

7. Execute os comandos a seguir para criar o Oracle Wallet.

```
prompt>orapki wallet create -wallet $ORACLE_HOME/ssl_wallet -auto_login_only
```

8. Extraia cada certificado no arquivo do pacote .pem em um arquivo .pem separado usando um utilitário do sistema operacional.
9. Adicione cada certificado à carteira usando comandos `orapki` separados, substituindo *`certificate-pem-file`* pelo nome absoluto do arquivo .pem.

```
prompt>orapki wallet add -wallet $ORACLE_HOME/ssl_wallet -trusted_cert -cert
      certificate-pem-file -auto_login_only
```

Para ter mais informações, consulte Alternar o certificado SSL/TLS.

Conectar-se a uma instância de banco de dados do RDS para Oracle usando SSL

Depois de configurar o SQL*Plus para usar SSL, conforme descrito anteriormente, você pode se conectar à instância de banco de dados do RDS para Oracle com a opção SSL. Opcionalmente, primeiro é possível exportar o valor de `TNS_ADMIN` que aponta para o diretório que contém os arquivos tnsnames.ora e sqlnet.ora. Isso garante que o SQL*Plus possa encontrar esses arquivos de forma consistente. O exemplo a seguir exporta o valor de `TNS_ADMIN`.

```
export TNS_ADMIN = ${ORACLE_HOME}/network/admin
```

Conecte-se à instância de banco de dados. Por exemplo, você pode se conectar usando o SQL*Plus e um *<net_service_name>* em um arquivo tnsnames.ora.

```
sqlplus mydbuser@net_service_name
```

Você também pode se conectar à instância de banco de dados usando o SQL*Plus sem utilizar um arquivo tnsnames.ora, por meio do seguinte comando.

```
sqlplus 'mydbuser@(DESCRIPTION = (ADDRESS = (PROTOCOL = TCPS)(HOST = endpoint) (PORT
 = ssl_port_number))(CONNECT_DATA = (SID = database_name)))'
```

Você também pode se conectar à instância de banco de dados do RDS para Oracle sem usar a SSL. Por exemplo, o seguinte comando se conecta à instância de banco de dados por meio da porta de texto sem formatação sem criptografia SSL.

```
sqlplus 'mydbuser@(DESCRIPTION = (ADDRESS = (PROTOCOL = TCP)(HOST = endpoint) (PORT
 = port_number))(CONNECT_DATA = (SID = database_name)))'
```

Se você deseja fechar o acesso à porta do protocolo TCP, crie um security group sem entradas de endereço IP e adicione-o à instância. Essa inclusão encerra as conexões através da porta TCP, embora ainda permita conexões sobre a porta SSL especificadas a partir de endereços IP dentro do intervalo permitido pelo security group da opção SSL.

Configurar uma conexão SSL via JDBC

Para usar uma conexão SSL via JDBC, é necessário criar um repositório de chaves, confiar no certificado da CA raiz do Amazon RDS e usar o trecho de código especificado a seguir.

Para criar o repositório de chaves no formato JKS, é possível usar o comando a seguir. Para ter mais informações sobre como criar o repositório de chaves, consulte Creating a keystore na documentação da Oracle. Para ter informações de referência, consulte keytool na Java Platform, Standard Edition Tools Reference.

```
keytool -genkey -alias client -validity 365 -keyalg RSA -keystore clientkeystore
```

Siga estas etapas para confiar no certificado da CA raiz do Amazon RDS.

Para confiar no certificado da CA raiz do Amazon RDS

1. Baixe o pacote do certificado .pem que funciona para todas as Regiões da AWS e coloque o arquivo no diretório ssl_wallet.

   Para obter informações sobre como baixar certificados, consulte Usar SSL/TLS para criptografar uma conexão com uma instância ou um cluster de banco de dados.
2. Extraia cada certificado no arquivo .pem em um arquivo separado usando um utilitário do sistema operacional.
3. Converta cada certificado no formato .der usando um comando `openssl` separado, substituindo *`certificate-pem-file`* pelo nome do arquivo .pem do certificado (sem a extensão .pem).

   ```
   openssl x509 -outform der -in certificate-pem-file.pem -out certificate-pem-file.der
   ```
4. Importe cada certificado para o repositório de chaves usando o comando a seguir.

   ```
   keytool -import -alias rds-root -keystore clientkeystore.jks -file certificate-pem-file.der
   ```

   Para ter mais informações, consulte Alternar o certificado SSL/TLS.
5. Confirme se o armazenamento de chaves foi criado com êxito.

   ```
   keytool -list -v -keystore clientkeystore.jks
   ```

   Insira a senha do repositório de chaves quando solicitado.

O exemplo de código a seguir mostra como configurar a conexão SSL usando o JDBC.

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.util.Properties;

public class OracleSslConnectionTest {
    private static final String DB_SERVER_NAME = "dns-name-provided-by-amazon-rds";
    private static final Integer SSL_PORT = "ssl-option-port-configured-in-option-
group";
    private static final String DB_SID = "oracle-sid";
    private static final String DB_USER = "user-name";
    private static final String DB_PASSWORD = "password";
    // This key store has only the prod root ca.
    private static final String KEY_STORE_FILE_PATH = "file-path-to-keystore";
    private static final String KEY_STORE_PASS = "keystore-password";

    public static void main(String[] args) throws SQLException {
        final Properties properties = new Properties();
        final String connectionString = String.format(
                "jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCPS)(HOST=%s)(PORT=
%d))(CONNECT_DATA=(SID=%s)))",
                DB_SERVER_NAME, SSL_PORT, DB_SID);
        properties.put("user", DB_USER);
        properties.put("password", DB_PASSWORD);
        properties.put("oracle.jdbc.J2EE13Compliant", "true");
        properties.put("javax.net.ssl.trustStore", KEY_STORE_FILE_PATH);
        properties.put("javax.net.ssl.trustStoreType", "JKS");
        properties.put("javax.net.ssl.trustStorePassword", KEY_STORE_PASS);
        final Connection connection = DriverManager.getConnection(connectionString,
 properties);
        // If no exception, that means handshake has passed, and an SSL connection can
 be opened
    }
}
```

> **Note**
>
> Especifique uma senha diferente do prompt mostrado aqui como prática recomendada de segurança.

Forçar uma correspondência de DN com uma conexão SSL

Você pode usar o parâmetro Oracle `SSL_SERVER_DN_MATCH` para forçar que o nome distinto (DN) do servidor de banco de dados corresponda ao nome do serviço. Se você forçar as verificações de correspondência, a SSL assegurará que o certificado seja proveniente do servidor. Se você não forçar a verificação de correspondência, a SSL realizará a verificação, mas permitirá a conexão, independentemente de haver ou não uma correspondência. Se você não forçar a correspondência, permitirá que o servidor falsifique a identidade.

Para forçar a correspondência de DN, adicione a propriedade de correspondência de DN e use a string de conexão especificada abaixo.

Adicione a propriedade à conexão de cliente para reforçar a correspondência de DN.

```
properties.put("oracle.net.ssl_server_dn_match", "TRUE");
```

Use a seguinte string de conexão para forçar a correspondência de DN ao usar a SSL.

```
final String connectionString = String.format(
    "jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCPS)(HOST=%s)(PORT=%d))" +
    "(CONNECT_DATA=(SID=%s))" +
    "(SECURITY = (SSL_SERVER_CERT_DN =
\"C=US,ST=Washington,L=Seattle,O=Amazon.com,OU=RDS,CN=%s\")))",
    DB_SERVER_NAME, SSL_PORT, DB_SID, DB_SERVER_NAME);
```

Solução de problemas de conexões SSL

Você pode consultar seu banco de dados e receber o erro `ORA-28860`.

```
ORA-28860: Fatal SSL error
28860. 00000 - "Fatal SSL error"
*Cause: An error occurred during the SSL connection to the peer. It is likely that this
 side sent data which the peer rejected.
*Action: Enable tracing to determine the exact cause of this error.
```

Esse erro ocorre quando o cliente tenta se conectar usando uma versão do TLS não compatível com o servidor. Para evitar esse erro, edite o sqlnet.ora e defina `SSL_VERSION` como a versão correta do TLS. Para ter mais informações, consulte o Documento de suporte da Oracle 2748438.1 em My Oracle Support.

Oracle Spatial

O Amazon RDS dá suporte ao Oracle Spatial por meio do uso da opção `SPATIAL`. O Oracle Spatial fornece um esquema SQL e funções que facilitam o armazenamento, a recuperação, a atualização e a consulta de coleções de dados espaciais em um banco de dados Oracle. Para obter mais informações, consulte Conceitos do Spatial na documentação da Oracle. O Amazon RDS é compatível com o Oracle Spatial em todas as edições de todas as versões compatíveis.

Como funcionam os pacotes de patches espaciais (SPBs)

A cada trimestre, o RDS para Oracle lança novas versões secundárias de mecanismos para cada mecanismo principal compatível. Uma atualização de versão (RU) do mecanismo incorpora correções de bugs da Oracle incluindo os patches de RU referentes ao trimestre especificado. Uma versão do mecanismo do pacote de patches espaciais (SPB) contém patches de RU e patches específicos para o Oracle Spatial. Por exemplo, 19.0.0.0.ru-2025-01.spb-1.r1 é uma versão secundária do mecanismo que contém os patches de RU na versão do mecanismo 19.0.0.0.ru-2025-01.rur-2025-01.r1 e patches do Spatial. Os SPBs são compatíveis somente com o Oracle Database 19c.

Eles funcionam da mesma forma que as RUs, embora tenham nomes diferentes. Uma RU usa o formato de nomenclatura 19.0.0.0.ru-2025-01.rur-2025-01.r1. Um nome de SPB inclui o texto “spb”, como em 19.0.0.0.ru-2025-01.spb-1.r1. Normalmente, um SPB é lançado duas a três semanas após a RU trimestral correspondente. Por exemplo, 19.0.0.0.ru-2025-01.spb-1.r1 é lançada após a 19.0.0.0.ru-2025-01.rur-2025-01.r1.

O RDS para Oracle tem caminhos separados para atualizações automáticas de versões secundárias de RUs e SPBs. Se sua instância de banco de dados usa uma RU, o RDS a atualiza automaticamente para uma RU. Se sua instância de banco de dados usa um SPB, o RDS o atualiza automaticamente para um SPB. Quando você adiciona a opção Oracle Spatial à sua instância de banco de dados e habilita atualizações automáticas, o RDS coloca sua instância no caminho de atualização para SPBs.

Para ter mais informações sobre RUs e SPBs, consulte Atualizações de versões secundárias Oracle. Para examinar uma lista de RUs e SPBs compatíveis com o Oracle Database 19c, consulte Amazon RDS for Oracle Database 19c (19.0.0.0) nas Notas de lançamento do Amazon RDS para Oracle.

Pré-requisitos do Oracle Spatial

Estes são os pré-requisitos para usar o Oracle Spatial:

- A instância de banco de dados deve ser de uma classe de instância suficiente. O Oracle Spatial não é compatível com as classes de instância de banco de dados db.t3.small. Para ter mais informações, consulte Classes de instância de banco de dados do RDS para Oracle.
- O Upgrade automático de versões secundárias deve estar habilitado na instância de banco de dados. Essa opção permite que a instância de banco de dados receba atualizações de versões secundárias do mecanismo de banco de dados automaticamente, quando disponibilizadas e é necessária para qualquer opção que instala o Oracle Java Virtual Machine (JVM). O Amazon RDS usa essa opção para atualizar a instância de banco de dados para a Oracle Patch Set Update (PSU) ou Release Update (RU) mais recente. Para obter mais informações, consulte Modificar uma instância de banco de dados do Amazon RDS.

Práticas recomendadas do Oracle Spatial

Estas são as práticas recomendadas para usar o Oracle Spatial:

- Para obter segurança máxima, use a opção `SPATIAL` com Secure Sockets Layer (SSL). Para obter mais informações, consulte Oracle Secure Sockets Layer.
- Configure a instância de banco de dados para restringir o acesso à instância de banco de dados. Para obter mais informações, consulte Cenários para acessar uma instância de banco de dados em uma VPC e Trabalhar com uma instância de banco de dados em uma VPC.

Adicionar a opção Oracle Spatial

Este é o processo geral para adicionar a opção `SPATIAL` a uma instância de banco de dados:

1. Crie um novo grupo de opções, ou copie ou modifique um existente.
2. Adicione a opção ao grupo de opções.
3. Associe o grupo de opções à instância de banco de dados.

Se o Oracle Java Virtual Machine (JVM) não estiver instalado na instância de banco de dados, haverá uma breve interrupção enquanto a opção `SPATIAL` é adicionada. Não haverá interrupção se o Oracle Java Virtual Machine (JVM) já estiver instalado na instância de banco de dados. Depois de adicionar a opção , você não precisará reiniciar sua instância de banco de dados. Assim que o grupo de opções estiver ativo, o Oracle Spatial estará disponível.

> **Note**
>
> Durante essa interrupção, as funções de verificação de senha são desativadas brevemente. Você também pode esperar ver eventos relacionados às funções de verificação de senha durante a interrupção. As funções de verificação de senha são ativadas novamente antes que a instância de banco de dados Oracle esteja disponível.

Para adicionar a opção **`SPATIAL`** a uma instância de banco de dados

1. Determine o grupo de opções que você deseja usar. Você pode criar um novo grupo de opções ou usar um existente. Se você quiser usar um grupo de opções existente, vá para a próxima etapa. Caso contrário, crie um grupo de opções de banco de dados personalizado com as seguintes configurações:

   a. Em Engine (Mecanismo), escolha a edição Oracle para sua instância de banco de dados.

   b. Para Major engine version (Versão do mecanismo principal), escolha a versão da sua instância de banco de dados.

   Para obter mais informações, consulte Criar um grupo de opções.
2. Adicione a opção SPATIAL ao grupo de opções. Para mais informações sobre a adição de opções, consulte Adicionar uma opção a um grupo de opções.
3. Aplique o grupo de opções a uma instância de banco de dados nova ou existente:

   - Para uma nova instância de banco de dados, você aplica o grupo de opções ao executar a instância. Para ter mais informações, consulte Criar uma instância de banco de dados do Amazon RDS.
   - Para uma instância de banco de dados existente, você aplica o grupo de opções modificando a instância e anexando o novo grupo de opções. Para obter mais informações, consulte Modificar uma instância de banco de dados do Amazon RDS.

Remover a opção Oracle Spatial

Depois de descartar todos os objetos que usam tipos de dados fornecidos pela opção `SPATIAL`, você pode descartá-los de uma instância de banco de dados. Se o Oracle Java Virtual Machine (JVM) não estiver instalado na instância de banco de dados, haverá uma breve interrupção enquanto

a opção SPATIAL é removida. Não haverá interrupção se o Oracle Java Virtual Machine (JVM) já estiver instalado na instância de banco de dados. Depois de remover a opção SPATIAL, você não precisará reiniciar a instância de banco de dados.

Como descartar a opção **SPATIAL**

1. Faça backup de seus dados.

   > **Warning**
   >
   > Se a instância usar tipos de dados que foram ativados como parte da opção e se você remover a opção SPATIAL, você poderá perder dados. Para obter mais informações, consulte Backup, restauração e exportação de dados.

2. Verifique se os objetos existentes fazem referência a tipos de dados ou recursos da opção SPATIAL.

   Se existirem opções SPATIAL, a instância poderá ficar presa ao aplicar o novo grupo de opções que não tem a opção SPATIAL. Você pode identificar os objetos usando as seguintes consultas:

   ```
   SELECT OWNER, SEGMENT_NAME, TABLESPACE_NAME, BYTES/1024/1024 mbytes
   FROM   DBA_SEGMENTS
   WHERE  SEGMENT_TYPE LIKE '%TABLE%'
   AND    (OWNER, SEGMENT_NAME) IN
          (SELECT DISTINCT OWNER, TABLE_NAME
           FROM   DBA_TAB_COLUMNS
           WHERE  DATA_TYPE='SDO_GEOMETRY'
           AND    OWNER <> 'MDSYS')
   ORDER BY 1,2,3,4;

   SELECT OWNER, TABLE_NAME, COLUMN_NAME
   FROM   DBA_TAB_COLUMNS
   WHERE  DATA_TYPE = 'SDO_GEOMETRY'
   AND    OWNER <> 'MDSYS'
   ORDER BY 1,2,3;
   ```

3. Solte todos os objetos que fazem referência a tipos de dados ou recursos da opção SPATIAL.
4. Execute um destes procedimentos:

- Remova a opção `SPATIAL` do grupo de opções ao qual ela pertence. Essa alteração afeta todas as instâncias de bancos de dados que usam o grupo de opções. Para obter mais informações, consulte Remover uma opção de um grupo de opções.
- Modifique a instância de banco de dados e especifique um grupo de opções diferente que não inclua a opção `SPATIAL`. Essa alteração afeta uma única instância de banco de dados. Você pode especificar um grupo de opções padrão (vazio) ou criar um grupo de opções personalizado diferente. Para ter mais informações, consulte Modificar uma instância de banco de dados do Amazon RDS.

Oracle SQLT

O Amazon RDS dá suporte ao Oracle SQLTXPLAIN (SQLT) por meio do uso da opção SQLT. É possível usar o SQLT com qualquer edição do Oracle Database 19c e posterior.

A instrução `EXPLAIN PLAN` do Oracle pode determinar o plano de execução de uma instrução SQL. Ela pode verificar se o otimizador Oracle escolha um determinado plano de execução, como uma junção aninhada em loop. Ela também ajuda a compreender as decisões do otimizador, como por que escolheu uma junção aninhada em loop em uma junção de hash. Assim, `EXPLAIN PLAN` ajuda a compreender a performance da instrução.

SQLT é um utilitário Oracle que produz um relatório. O relatório inclui estatísticas de objeto, metadados de objeto, parâmetros de inicialização relacionados ao otimizador e outras informações que um administrador de banco de dados pode usar para ajustar uma instrução SQL para performance ideal. SQLT produz um relatório HTML com hiperlinks para todas as seções no relatório.

Diferentemente de relatórios Automatic Workload Repository ou Statspack, o SQLT funciona em instruções SQL individuais. O SQLT é uma coleção de arquivos SQL, PL/SQL e SQL*Plus que coletam, armazenam e exibem dados de performance.

A seguir estão as versões com suporte do Oracle para cada versão do SQLT.

Versão do SQLT	Oracle Database 21c	Oracle Database 19c
2018-07-25.v1	Compatível	Compatível
2018-03-31.v1	Sem compatibilidade	Sem compatibilidade
2016-04-29.v1	Sem compatibilidade	Sem compatibilidade

Para baixar o SQLT e as instruções de acesso para usá-lo:

- Faça login na conta My Oracle Support e abra os documentos a seguir:
- Para baixar o SQLT: Documento 215187.1
- Para obter instruções de uso do SQLT: Documento 1614107.1
- Para ver as perguntas frequentes sobre o SQLT, consulte Documento 1454160.1.

- Para obter informações sobre como ler a saída do SQLT: Documento 1456176.1
- Para a interpretação do relatório principal: Documento 1922234.1

O Amazon RDS não é compatível com os seguintes métodos do SQLT:

- `XPLORE`
- `XHUME`

Pré-requisitos para o SQLT

Veja a seguir os pré-requisitos para usar o SQLT:

- Será necessário remover usuários e funções solicitados pelo SQLT, se eles existirem.

  A opção do SQLT cria os seguintes usuários e as funções em uma instância de banco de dados:

  - `SQLTXPLAIN`Usuário
  - `SQLTXADMIN`Usuário
  - `SQLT_USER_ROLE`Função

  Se a instância de banco de dados tiver algum desses usuários ou funções, faça login na instância de banco de dados usando um cliente SQL e os ignore usando as seguintes instruções:

```
DROP USER SQLTXPLAIN CASCADE;
DROP USER SQLTXADMIN CASCADE;
DROP ROLE SQLT_USER_ROLE CASCADE;
```

- Será necessário remover espaços de tabela solicitados pelo SQLT, se eles existirem.

  A opção do SQLT cria os seguintes espaços de tabela em uma instância de banco de dados:

  - `RDS_SQLT_TS`
  - `RDS_TEMP_SQLT_TS`

  Se a instância de banco de dados tiver esses espaços de tabela, faça login na instância de banco de dados usando um cliente SQL e os ignore.

Configurações de opção SQLT

O SQLT pode funcionar com recursos licenciados fornecidos pelo Oracle Tuning Pack e pelo Oracle Diagnostics Pack. O Oracle Tuning Pack inclui o SQL Tuning Advisor, e o Oracle Diagnostics Pack inclui o Automatic Workload Repository. As configurações do SQLT habilitam ou desabilitam acesso a esses recursos do SQLT.

O Amazon RDS oferece suporte para as seguintes configurações da opção SQLT.

Configuração da opção	Valores válidos	Valor padrão	Descrição
`LICENSE_PACK`	`T`, `D`, `N`	`N`	Os Oracle Management Packs que você deseja acessar com o SQLT. Insira um dos seguintes valores: • `T` indica que você tem uma licença do Oracle Tuning Pack e do Oracle Diagnostics Pack e deseja acessar o SQL Tuning Advisor e o Automatic Workload Repository no SQLT. • `D` indica que você tem uma licença do Oracle Diagnostics Pack e deseja acessar o Automatic Workload Repository no SQLT. • `N` Indica que você não tem uma licença do Oracle Tuning Pack e do Oracle Diagnostics Pack, ou que você tem uma licença para um ou ambos, mas não deseja que o SQLT os acesse. **Note** O Amazon RDS não fornece licenças para esses Oracle Management Packs. Se

Configuração da opção	Valores válidos	Valor padrão	Descrição
			indicar que deseja usar um pacote não incluído na instância de banco de dados, você poderá usar o SQLT com a instância de banco de dados. No entanto, o SQLT não consegue acessar o pacote, e o relatório SQLT não inclui os dados do pacote. Por exemplo, se você especific ar T, mas a instância de banco de dados não incluir o Oracle Tuning Pack, o SQLT funcionará na instância de banco de dados, mas o relatório gerado não conterá dados relacionados ao Oracle Tuning Pack.
VERSION	2016-04-29.v1 2018-03-31.v1 2018-07-25.v1	2016-04-29.v1	A versão do SQLT que você deseja instalar. **Note** Para o Oracle Database 19c e 21c, a única versão compatíve l é 2018-07-25.v1 . Para essas versões, essa é a padrão.

Adicionar a opção SQLT

Este é o processo geral para adicionar as opções do SQLT a uma instância de banco de dados:

1. Crie um novo grupo de opções, ou copie ou modifique um existente.
2. Adicione a opção SQLT ao grupo de opções.
3. Associe o grupo de opções à instância de banco de dados.

Depois de adicionar a opção SQLT, assim que o grupo de opções estiver ativo, o SQLT ficará ativo.

Para adicionar a opção SQLT a uma instância de banco de dados

1. Determine o grupo de opções que você deseja usar. Você pode criar um novo grupo de opções ou usar um existente. Se você quiser usar um grupo de opções existente, vá para a próxima etapa. Caso contrário, crie um grupo de opções de banco de dados personalizado com as seguintes configurações:

   a. Para Engine (Mecanismo), escolha a edição do Oracle que deseja usar. A opção do SQLT é compatível em todas as edições.

   b. Para Major engine version (Versão do mecanismo principal), escolha a versão da sua instância de banco de dados.

   Para obter mais informações, consulte Criar um grupo de opções.
2. Adicione a opção SQLT ao grupo de opções. Para mais informações sobre a adição de opções, consulte Adicionar uma opção a um grupo de opções.
3. Aplique o grupo de opções a uma instância de banco de dados nova ou existente:

   - Para uma nova instância de banco de dados, você aplica o grupo de opções ao executar a instância. Para ter mais informações, consulte Criar uma instância de banco de dados do Amazon RDS.

   - Para uma instância de banco de dados existente, você aplica o grupo de opções modificando a instância e anexando o novo grupo de opções. Para obter mais informações, consulte Modificar uma instância de banco de dados do Amazon RDS.
4. (Opcional) Verifique a instalação do SQLT em cada instância de banco de dados com a opção do SQLT.

   a. Use um cliente SQL para se conectar à instância de banco de dados como o usuário mestre.

Para obter informações sobre como se conectar a uma instância de banco de dados Oracle usando um cliente SQL, consulte Conectar-se a uma instância de banco de dados Oracle.

b. Execute a seguinte consulta:

```
SELECT sqltxplain.sqlt$a.get_param('tool_version') sqlt_version FROM DUAL;
```

A consulta retorna a versão atual da opção do SQLT no Amazon RDS. `12.1.160429` é um exemplo de uma versão do SQLT disponível no Amazon RDS.

5. Altere as senhas dos usuários criados pela opção do SQLT.

a. Use um cliente SQL para se conectar à instância de banco de dados como o usuário mestre.

b. Execute a seguinte instrução SQL para alterar a senha do usuário `SQLTXADMIN`:

```
ALTER USER SQLTXADMIN IDENTIFIED BY new_password ACCOUNT UNLOCK;
```

> **Note**
>
> Especifique uma senha diferente do prompt mostrado aqui como prática recomendada de segurança.

c. Execute a seguinte instrução SQL para alterar a senha do usuário `SQLTXPLAIN`:

```
ALTER USER SQLTXPLAIN IDENTIFIED BY new_password ACCOUNT UNLOCK;
```

> **Note**
>
> Especifique uma senha diferente do prompt mostrado aqui como prática recomendada de segurança.

> **Note**
>
> Atualizar o SQLT exige desinstalar uma versão mais antiga do SQLT e instalar a nova versão. Dessa forma, todos os metadados do SQLT podem ser perdidos quando você atualiza o SQLT. Uma atualização de versão principal de um banco de dados também desinstala e reinstala o SQLT. Um exemplo de uma atualização de versão principal é uma atualização do Oracle Database 19c para o Oracle Database 21c.

Uso de SQLT

O SQLT funciona com o utilitário Oracle SQL*Plus.

Para usar o SQLT

1. Baixe o arquivo .zip SQLT no Documento 215187.1 no site do My Oracle Support.

   > **Note**
   >
   > Não é possível baixar o SQLT 12.1.160429 do site do My Oracle Support. A Oracle desativou essa versão mais antiga.

2. Descompacte o arquivo .zip do SQLT.
3. Em um prompt de comando, altere o diretório `sqlt/run` no sistema de arquivos.
4. No prompt de comando, abra o SQL*Plus e se conecte à instância de banco de dados como o usuário mestre.

   Para obter mais informações sobre como se conectar a uma instância de banco de dados usando o SQL *Plus, consulte Conectar-se a uma instância de banco de dados Oracle.
5. Obtenha o ID do SQL de uma instrução SQL:

```
SELECT SQL_ID FROM V$SQL WHERE SQL_TEXT='sql_statement';
```

A saída é semelhante à seguinte:

```
SQL_ID
-------------
chvsmttqjzjkn
```

6. Analise uma instrução SQL com o SQLT:

```
START sqltxtract.sql sql_id sqltxplain_user_password
```

Por exemplo, para o ID SQL `chvsmttqjzjkn`, insira o seguinte:

```
START sqltxtract.sql chvsmttqjzjkn sqltxplain_user_password
```

O SQLT gera o relatório HTML e os recursos relacionados como um arquivo .zip no diretório no qual o comando do SQLT foi executado.

7. (Opcional) Para permitir que usuários do aplicativo façam diagnóstico de instruções SQL com o SQLT, conceda `SQLT_USER_ROLE` a cada usuário do aplicativo com a seguinte instrução:

```
GRANT SQLT_USER_ROLE TO application_user_name;
```

> **Note**
>
> O Oracle não recomenda executar o SQLT com o usuário `SYS` ou com os usuários que tenham a função `DBA`. Trata-se de uma melhor prática executar o diagnóstico do SQLT

usando a conta do usuário do aplicativo concedendo `SQLT_USER_ROLE` ao usuário do aplicativo.

Atualizar a opção SQLT

Com o Amazon RDS for Oracle, você pode atualizar a opção SQLT da sua versão existente para uma versão superior. Para atualizar a opção SQLT, conclua as etapas 1 – 3 em Uso de SQLT para a nova versão do SQLT. Além disso, se você tiver concedido privilégios na versão anterior do SQLT, na etapa 7 daquela seção, conceda os privilégios novamente na nova versão do SQLT.

A atualização da opção SQLT resulta na perda dos metadados da versão anterior do SQLT. O esquema e os objetos relacionados da versão anterior do SQLT são descartados e a nova versão do SQLT é instalada. Para obter mais informações sobre as alterações na versão mais recente do SQLT, consulte Documento 1614201.1 no site My Oracle Support.

> **Note**
>
> Downgrades de versões não são compatíveis.

Modificar as configurações do SQLT

Depois de habilitar o SQLT, você pode modificar as configurações de `LICENSE_PACK` e `VERSION` para essa opção.

Para mais informações sobre como modificar as configurações da opção, consulte Modificar uma configuração de opção. Para mais informações sobre cada configuração, consulte Configurações de opção SQLT.

Remover a opção SQLT

Você pode remover o SQLT de uma instância de banco de dados.

Para remover o SQLT de uma instância de banco de dados, siga um destes procedimentos:

- Para remover o SQLT de várias instâncias de bancos de dados, remova a opção SQLT do grupo de opções ao qual as instâncias de banco de dados pertencem. Essa alteração afeta todas as instâncias de bancos de dados que usam o grupo de opções. Para obter mais informações, consulte Remover uma opção de um grupo de opções.

- Para remover o SQLT de uma única instância de banco de dados, modifique essa instância e especifique um grupo de opções diferente que não inclua a opção SQLT. Você pode especificar um grupo de opções padrão (vazio) ou criar um grupo de opções personalizado diferente. Para ter mais informações, consulteModificar uma instância de banco de dados do Amazon RDS

Oracle Statspack

A opção Oracle Statspack instala e habilita o recurso de estatísticas de performance Oracle Statspack. O Oracle Statspack é uma coleção de scripts SQL, PL/SQL e SQL*Plus que coletam, armazenam e exibem dados de performance. Para obter informações sobre como usar o Oracle Statspack, consulte Oracle Statspack na documentação da Oracle.

> **Note**
>
> O Oracle Statspack não tem mais suporte pela Oracle e foi substituído pelo mais avançado AWR (Automatic Workload Repository). O AWR está disponível apenas para clientes do Oracle Enterprise Edition que compraram o Diagnostics Pack. Você pode usar o Oracle Statspack com qualquer mecanismo de banco de dados Oracle no Amazon RDS. Você não pode executar o Oracle Statspack em réplicas de leitura do Amazon RDS.

Configurar o Oracle Statspack

Para executar scripts do Statspack, é necessário adicionar a opção Statspack.

Como configurar o Oracle Statspack

1. Em um cliente SQL, faça login no banco de dados Oracle com uma conta administrativa.
2. Execute uma das seguintes ações, dependendo se o Statspack está instalado:
   - Se o Statspack estiver instalado e a conta `PERFSTAT` estiver associada ao Statspack, vá para a Etapa 4.
   - Se o Statspack não estiver instalado e a conta `PERFSTAT` existir, descarte a conta da seguinte maneira:

     ```
     DROP USER PERFSTAT CASCADE;
     ```

     Caso contrário, tentar adicionar a opção Statspack gerará um erro e `RDS-Event-0058`.
3. Adicione a opção Statspack a um grupo de opções. Consulte Adicionar uma opção a um grupo de opções.

   O Amazon RDS instala automaticamente os scripts do Statspack na instância de banco de dados e configura a conta `PERFSTAT`.

4. Redefina a senha usando a seguinte instrução SQL, substituindo pwd por sua nova senha:

```
ALTER USER PERFSTAT IDENTIFIED BY pwd ACCOUNT UNLOCK;
```

   Você pode fazer login usando a conta do usuário PERFSTAT e executar os scripts do Statspack.

5. Conceda o privilégio CREATE JOB à conta PERFSTAT usando a seguinte instrução:

```
GRANT CREATE JOB TO PERFSTAT;
```

6. Verifique se os eventos de espera ociosos na tabela PERFSTAT.STATS$IDLE_EVENT estão preenchidos.

   Devido ao erro 28523746 do Oracle, os eventos de espera ociosos em PERFSTAT.STATS$IDLE_EVENT podem não estar preenchidos. Para garantir que todos os eventos ociosos estejam disponíveis, execute a seguinte instrução:

```
INSERT INTO PERFSTAT.STATS$IDLE_EVENT (EVENT)
SELECT NAME FROM V$EVENT_NAME WHERE WAIT_CLASS='Idle'
MINUS
SELECT EVENT FROM PERFSTAT.STATS$IDLE_EVENT;
COMMIT;
```

Gerar relatórios do Statspack

Um relatório do Statspack compara dois snapshots.

Como gerar relatórios do Statspack

1. Em um cliente SQL, faça login no banco de dados Oracle com a conta PERFSTAT.
2. Crie um snapshot usando uma das seguintes técnicas:
   - Crie um snapshot do Statspack manualmente.
   - Crie um trabalho que faça um snapshot do Statspack após determinado intervalo de tempo. Por exemplo, o seguinte trabalho cria um snapshot do Statspack a cada hora:

```
VARIABLE jn NUMBER;
exec dbms_job.submit(:jn, 'statspack.snap;',SYSDATE,'TRUNC(SYSDATE
+1/24,''HH24'')');
```

```
COMMIT;
```

3. Visualize os snapshots usando a seguinte consulta:

```
SELECT SNAP_ID, SNAP_TIME FROM STATS$SNAPSHOT ORDER BY 1;
```

4. Execute o procedimento `rdsadmin.rds_run_spreport` do Amazon RDS, substituindo begin_snap e end_snap pelos IDs do snapshot:

```
exec rdsadmin.rds_run_spreport(begin_snap,end_snap);
```

Por exemplo, o seguinte comando cria um relatório baseado no intervalo entre os snapshots 1 e 2 do Statspack:

```
exec rdsadmin.rds_run_spreport(1,2);
```

O nome de arquivo do relatório do Statspack inclui o número dos dois snapshots. Por exemplo, um arquivo de relatório criado usando os snapshots 1 e 2 do Statspack se chamaria `ORCL_spreport_1_2.lst`.

5. Verifique se há erros na saída.

O Oracle Statspack realiza verificações antes de executar o relatório. Portanto, você também pode ver mensagens de erro na saída do comando. Por exemplo, você pode tentar gerar um relatório com base em um intervalo inválido, em que o valor inicial do snapshot do Statspack é maior do que o valor final. Nesse caso, a saída mostra a mensagem de erro, mas o mecanismo de banco de dados não gera um arquivo de erro.

```
exec rdsadmin.rds_run_spreport(2,1);
*
ERROR at line 1:
ORA-20000: Invalid snapshot IDs. Find valid ones in perfstat.stats$snapshot.
```

Se você usar um número inválido para um snapshot do Statspack, a saída mostrará um erro. Por exemplo, se você tentar gerar um relatório para os snapshots 1 e 50, mas o snapshot 50 não existir, a saída mostrará um erro.

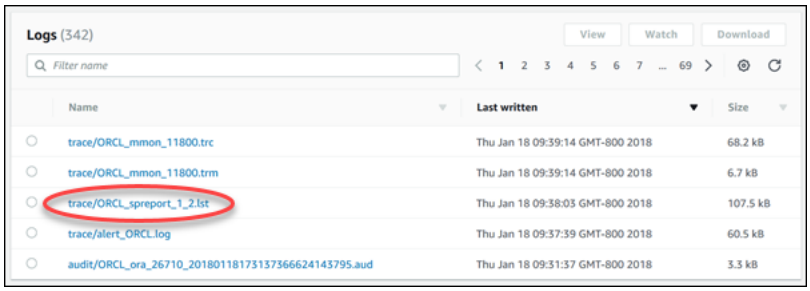
```
exec rdsadmin.rds_run_spreport(1,50);
*
ERROR at line 1:
```

```
ORA-20000: Could not find both snapshot IDs
```

6. (Optional)

   Para recuperar o relatório, chame os procedimentos de arquivo de rastreamento, conforme explicado em Trabalhar com arquivos de rastreamento da Oracle.

   Como alternativa, baixe o relatório do Statspack pelo console do RDS. Acesse a seção Log dos detalhes da instância de banco de dados e selecione Baixar. O exemplo a seguir mostra `trace/ORCL_spreport_1_2.lst`.



   Se ocorrer um erro ao gerar um relatório, o mecanismo de banco de dados usará as mesmas convenções de nomenclatura de um relatório, mas com uma extensão de `.err`. Por exemplo, se ocorresse um erro durante a criação de um relatório usando os snapshots 1 e 7 do Statspack, o arquivo de relatório se chamaria `ORCL_spreport_1_7.err`. É possível baixar o relatório de erros usando as mesmas técnicas de um relatório de snapshot padrão.

Remover snapshots do Statspack

Para remover vários snapshots do Oracle Statspack, use o seguinte comando:

```
exec statspack.purge(begin snap, end snap);
```

Fuso horário da Oracle

Para alterar o fuso horário do sistema usado pela sua instância de banco de dados Oracle, use a opção de fuso horário. Por exemplo, é possível alterar o fuso horário de uma instância de banco de dados para que ele seja compatível com um ambiente local ou um aplicativo herdado. A opção de fuso horário altera o fuso horário no nível do host. Alterar o fuso horário afeta todas as colunas e valores de data, incluindo `SYSDATE` e `SYSTIMESTAMP`.

A opção de fuso horário difere do comando `rdsadmin_util.alter_db_time_zone`. O comando `alter_db_time_zone` altera o fuso horário apenas para certos tipos de dados. A opção de fuso horário altera o fuso horário para todas as colunas e valores de data. Para ter mais informações sobre o `alter_db_time_zone`, consulte Definição do fuso horário do banco de dados. Para ter mais informações sobre as considerações de atualização, consulte Considerações sobre fuso horário.

Restrições para definir o fuso horário

A opção de fuso horário é uma opção permanente e persistente. Portanto, você não pode fazer o seguinte:

- Remover a opção de um grupo de opções depois de adicionar a opção de fuso horário.
- Remover o grupo de opções de uma instância de banco de dados depois de adicioná-lo.
- Modificar a configuração de fuso horário da opção para um fuso horário diferente.

Recomendações para definir o fuso horário

Antes de adicionar a opção de fuso horário ao seu banco de dados de produção, recomendamos que você faça o seguinte:

- Crie um snapshot de sua instância de banco de dados. Se você definir o fuso horário incorretamente por engano, será necessário recuperar a instância de banco de dados para sua configuração de fuso horário anterior. Para ter mais informações, consulte Criar um snapshot de banco de dados para uma instância de banco de dados single-AZ para o Amazon RDS.
- Para adicionar a opção de fuso horário a uma instância de banco de dados Adicionar a opção de fuso horário pode causar problemas com tabelas que usam a data do sistema para adicionar datas ou horários. Recomendamos que você analise dados e aplicações na instância de teste. Dessa forma, é possível avaliar o impacto da mudança do fuso horário na instância de produção.

Se a instância de banco de dados usa o grupo de opções padrão, siga estas etapas:

1. Crie um snapshot de sua instância de banco de dados.
2. Adicione a opção de fuso horário a uma instância de banco de dados.

Se a instância de banco de dados usa o grupo de opções não padrão, siga estas etapas:

1. Crie um snapshot de sua instância de banco de dados.
2. Crie um novo grupo de opções.
3. Adicione a opção de fuso horário à instância, além de todas as outras opções atualmente associadas ao grupo de opções existente.

   Isso evita que as opções existentes sejam desinstaladas ao habilitar a opção de fuso horário.
4. Adicione o grupo de opções à sua instância de banco de dados.

Configurações da opção de fuso horário

O Amazon RDS oferece suporte para as seguintes configurações da opção de fuso horário.

Configuração da opção	Valores válidos	Descrição
`TIME_ZONE`	Um dos fusos horários disponíveis. Para ver a lista completa, consulte Fusos horários disponíveis.	O novo fuso horário da sua instância de banco de dados.

Adicionar a opção de fuso horário

Conclua as seguintes etapas para adicionar a opção de fuso horário à instância de banco de dados:

1. (Recomendado) Crie um snapshot da instância de banco de dados.
2. Realize uma destas tarefas:
   - Crie um grupo de opções do zero. Para ter mais informações, consulte Criar um grupo de opções.
   - Copie um grupo de opções existente usando a AWS CLI ou a API. Para ter mais informações, consulte Copiar um grupo de opções.

- Reutilize um grupo de opções não padrão. Uma prática recomendada é usar um grupo de opções que não esteja atualmente associado a nenhuma instância de banco de dados ou snapshot.

3. Adicione a nova opção ao grupo de opções da etapa anterior.
4. Se o grupo de opções atualmente associado à instância de banco de dados tiver opções habilitadas, adicione essas opções ao novo grupo de opções. Essa estratégia evita que as opções existentes sejam desinstaladas ao habilitar a nova opção.
5. Adicione o novo grupo de opções à instância de banco de dados.

Quando você adiciona a opção de fuso horário, ocorre uma breve interrupção enquanto sua instância de banco de dados é reiniciada automaticamente.

Console

Como adicionar a opção de fuso horário a um grupo de opções e associá-la à instância de banco de dados

1. No console do RDS, escolha Grupos de opções.
2. Escolha o nome do grupo de opções ao qual você deseja adicionar a opção.
3. Escolha Add option (Adicionar opção).
4. Em Nome da opção, escolha Fuso horário e defina as configurações da opção.
5. Associe o grupo de opções a uma instância de banco de dados nova ou existente:

   - Para uma nova instância de banco de dados, aplique o grupo de opções ao executar a instância. Para obter mais informações, consulte Criar uma instância de banco de dados do Amazon RDS.
   - Para uma instância de banco de dados existente, aplique o grupo de opções modificando a instância e anexando o novo grupo de opções. Quando você adiciona a nova opção a uma instância de banco de dados existente, ocorre uma breve interrupção enquanto a instância de banco de dados é reiniciada automaticamente. Para ter mais informações, consulte Modificar uma instância de banco de dados do Amazon RDS.

AWS CLI

O exemplo a seguir usa o comando add-option-to-option-group da AWS CLI para adicionar a opção `Timezone` e a configuração da opção `TIME_ZONE` a um grupo de opções chamado de `myoptiongroup`. O fuso horário é definido como `Africa/Cairo`.

Para Linux, macOS ou Unix:

```
aws rds add-option-to-option-group \
    --option-group-name "myoptiongroup" \
    --options "OptionName=Timezone,OptionSettings=[{Name=TIME_ZONE,Value=Africa/
Cairo}]" \
    --apply-immediately
```

Para Windows:

```
aws rds add-option-to-option-group ^
    --option-group-name "myoptiongroup" ^
    --options "OptionName=Timezone,OptionSettings=[{Name=TIME_ZONE,Value=Africa/
Cairo}]" ^
    --apply-immediately
```

Modificar configurações de fuso horário

A opção de fuso horário é uma opção permanente e persistente. Você não poderá remover a opção de um grupo de opções depois de adicioná-la. Você não poderá remover o grupo de opções de uma instância de banco de dados depois de adicioná-lo. Não é possível modificar a configuração de fuso horário da opção para um fuso horário diferente. Se você definir o fuso horário incorretamente, restaure um snapshot da sua instância de banco de dados antes de adicionar a opção de fuso horário.

Remover opção de fuso horário

A opção de fuso horário é uma opção permanente e persistente. Você não poderá remover a opção de um grupo de opções depois de adicioná-la. Você não poderá remover o grupo de opções de uma instância de banco de dados depois de adicioná-lo. Pare remover a opção de fuso horário, restaure um snapshot da sua instância de banco de dados antes de adicionar a opção de fuso horário.

Fusos horários disponíveis

Os valores a seguir podem ser usados para a opção de fuso horário.

Zona	Time zone (Fuso horário)
África	África/Cairo, África/Casablanca, África/Harare, África/Lagos, África/Luanda, África/Monrovia, África/Nairobi, África/Tripoli, África/Windhoek
América	América/Araguaina, América/Argentina/Buenos_Aires, América/Asuncion, América/Bogota, América/Caracas, América/Chicago, América/Chihuahua, América/Cuiaba, América/Denver, América/Detroit, América/Fortaleza, América/Godthab, América/Guatemala, América/Halifax, América/Lima, América/Los_Angeles, América/Manaus, América/Matamoros, América/Mexico_City, América/Monterrey, América/Montevideo, América/New_York, América/Phoenix, América/Santiago, América/Sao_Paulo, América/Tijuana, América/Toronto
Ásia	Ásia/Amman, Ásia/Ashgabat, Ásia/Baghdad, Ásia/Baku, Ásia/Bangkok, Ásia/Beirut, Ásia/Calcutta, Ásia/Damascus, Ásia/Dhaka, Ásia/Hong_Kong, Ásia/Irkutsk, Ásia/Jakarta, Ásia/Jerusalem, Ásia/Kabul, Ásia/Karachi, Ásia/Kathmandu, Ásia/Kolkata, Ásia/Krasnoyarsk, Ásia/Magadan, Ásia/Manila, Ásia/Muscat, Ásia/Novosibirsk, Ásia/Rangoon, Ásia/Riyadh, Ásia/Seoul, Ásia/Shanghai, Ásia/Singapore, Ásia/Taipei, Ásia/Tehran, Ásia/Tokyo, Ásia/Ulaanbaatar, Ásia/Vladivostok, Ásia/Yakutsk, Ásia/Yerevan
Atlântico	Atlântico/Açores, Atlântico/Cape_Verde
Austrália	Austrália/Adelaide, Austrália/Brisbane, Austrália/Darwin, Austrália/Eucla, Austrália/Hobart, Austrália/Lord_Howe, Austrália/Perth, Austrália/Sydney
Brasil	Brasil/DeNoronha, Brasil/Leste
Canadá	Canadá/Newfoundland, Canadá/Saskatchewan
Etc	Etc/GMT-3
Europa	Europa/Amsterdam, Europa/Athens, Europa/Berlin, Europa/Dublin, Europa/Helsinki, Europa/Kaliningrad, Europa/London, Europa/Madrid, Europa/Moscow, Europa/Paris, Europa/Prague, Europa/Rome, Europa/Sarajevo

Zona	Time zone (Fuso horário)
Pacífico	Pacífico/Apia, Pacífico/Auckland, Pacífico/Chatham, Pacífico/Fiji, Pacífico/Guam, Pacífico/Honolulu, Pacífico/Kiritimati, Pacífico/Marquesas, Pacífico/Samoa, Pacífico/Tongatapu, Pacífico/Wake
EUA	EUA/Alasca, EUA/Central, EUA/East-Indiana, EUA/Eastern, EUA/Pacífico
UTC	UTC

Atualização automática do arquivo de fuso horário do Oracle

Com a opção `TIMEZONE_FILE_AUTOUPGRADE`, é possível atualizar o arquivo de fuso horário atual para a versão mais recente na instância de banco de dados do RDS para Oracle.

Tópicos

- Visão geral dos arquivos de fuso horário da Oracle
- Estratégias para atualizar seu arquivo de fuso horário
- Tempo de inatividade durante a atualização do arquivo de fuso horário
- Preparação para atualizar o arquivo de fuso horário
- Adição da opção de atualização automática do arquivo de fuso horário
- Verificação dos dados após a atualização do arquivo de fuso horário

Visão geral dos arquivos de fuso horário da Oracle

Um arquivo de fuso horário do Oracle Database armazena as seguintes informações:

- Distância em relação ao Tempo Universal Coordenado (UTC)
- Horários de transição para horário de verão (DST)
- Abreviações para horário padrão e DST

O Oracle Database fornece várias versões de arquivos de fuso horário. Ao criar um banco de dados Oracle em um ambiente local, você escolhe a versão do arquivo de fuso horário. Para ter mais informações, consulte Choosing a Time Zone File (Escolher um arquivo de fuso horário) no Oracle Database Globalization Support Guide (Guia de suporte à globalização de bancos de dados da Oracle).

Se as regras mudarem para DST, a Oracle publicará novos arquivos de fuso horário. A Oracle lança esses novos arquivos de fuso horário independentemente do cronograma de atualizações de versão (RUs) e revisões de atualização de versões (RURs) trimestrais. Os arquivos de fuso horário residem no host do banco de dados no diretório `$ORACLE_HOME/oracore/zoneinfo/`. Os nomes de arquivo de fuso horário usam o formato DStv*`versão`*, como em DSTv35.

Como o arquivo de fuso horário afeta a transferência de dados

No Oracle Database, o tipo de dados `TIMESTAMP WITH TIME ZONE` armazena carimbo de data/hora e dados de fuso horário. Dados com o tipo de dados `TIMESTAMP WITH TIME ZONE` usam as

regras na versão do arquivo de fuso horário associado. Assim, os dados `TIMESTAMP WITH TIME ZONE` existentes são afetados ao atualizar o arquivo de fuso horário.

Podem ocorrer problemas quando você transfere dados entre bancos de dados que usam versões diferentes do arquivo de fuso horário. Por exemplo, se você importar dados de um banco de dados de origem com uma versão de arquivo de fuso horário mais recente do que o do banco de dados de destino, o banco de dados emitirá o erro `ORA-39405`. Anteriormente, era necessário contornar esse erro usando uma das seguintes técnicas:

- Crie uma instância de banco de dados do RDS para Oracle com o arquivo de fuso horário desejado, exporte dados do banco de dados de origem e importe-os para o novo banco de dados.
- Use o AWS DMS ou replicação lógica para migrar seus dados.

Atualizações automáticas usando a opção TIMEZONE_FILE_AUTOUPGRADE

Quando o grupo de opções anexado à instância de banco de dados do RDS para Oracle inclui a opção `TIMEZONE_FILE_AUTOUPGRADE`, o RDS atualiza os arquivos de fuso horário automaticamente. Ao garantir que os bancos de dados Oracle usem a mesma versão de arquivo de fuso horário, você evita técnicas manuais demoradas ao mover dados entre ambientes diferentes. A opção `TIMEZONE_FILE_AUTOUPGRADE` é compatível com bancos de dados de contêiner (CDBs) e não CDBs.

Ao adicionar a opção `TIMEZONE_FILE_AUTOUPGRADE` ao seu grupo de opções, você pode escolher se deseja adicionar a opção imediatamente ou durante a janela de manutenção. Depois que a instância de banco de dados usa a nova opção, o RDS confere se pode instalar um arquivo DSTv*`version`* mais recente. A *`versão`* de destino do DSTv depende do seguinte:

- A versão de mecanismo secundária que sua instância de banco de dados está em execução atualmente
- A versão secundária do mecanismo para a qual você deseja atualizar sua instância de banco de dados

Por exemplo, suponhamos que a versão atual do arquivo de fuso horário seja DSTv33. Ao aplicar a atualização ao grupo de opções, o RDS poderá determinar se DSTv34 está disponível atualmente no sistema de arquivos da instância de banco de dados. O RDS então atualizará o arquivo de fuso horário para DSTv34 automaticamente.

Para encontrar as versões de DST disponíveis nas atualizações da versão compatível do RDS, confira os patches em Notas de versão do Amazon Relational Database Service (Amazon RDS) para Oracle. Por exemplo, a versão 19.0.0.0.ru-2022-10.rur-2022-10.r1 lista o patch 34533061: RDBMS – DSTV39 UPDATE – TZDATA2022C.

Estratégias para atualizar seu arquivo de fuso horário

Atualizar o mecanismo de banco de dados e adicionar a opção `TIMEZONE_FILE_AUTOUPGRADE` a um grupo de opções são operações separadas. Adicionar a opção `TIMEZONE_FILE_AUTOUPGRADE` iniciará a atualização do arquivo de fuso horário se um mais recente estiver disponível. É necessário executar os seguintes comandos (somente as opções relevantes são mostradas) imediatamente ou na próxima janela de manutenção:

- Atualize o mecanismo de banco de dados usando somente o seguinte comando da CLI do RDS:

```
modify-db-instance --engine-version name ...
```

- Adicione a opção `TIMEZONE_FILE_AUTOUPGRADE` somente usando o seguinte comando da CLI:

```
add-option-to-option-group --option-group-name name --options
 OptionName=TIMEZONE_FILE_AUTOUPGRADE ...
```

- Atualize o mecanismo de banco de dados e adicione um novo grupo de opções à instância usando o seguinte comando da CLI:

```
modify-db-instance --engine-version name --option-group-name name ...
```

A estratégia de atualização depende de atualizar o banco de dados e arquivo de fuso horário juntos ou realizar apenas uma dessas operações. Lembre-se de que, se você atualizar o grupo de opções e, depois, atualizar o mecanismo de banco de dados em operações de API separadas, é possível que uma atualização do arquivo de fuso horário esteja em andamento ao atualizar o mecanismo de banco de dados.

Para os exemplos nesta seção, suponha o seguinte:

- Você ainda não adicionou `TIMEZONE_FILE_AUTOUPGRADE` ao grupo de opções atualmente associado à instância de banco de dados.
- Sua instância de banco de dados usa a versão de banco de dados 19.0.0.0.ru-2019-07.rur-2019-07.r1 e o arquivo de fuso horário DSTv33.

- O sistema de arquivos da sua instância de banco de dados inclui o arquivo DStv34.
- A atualização de versão 19.0.0.0.ru-2022-10.rur-2022-10.r1 inclui DSTv35.

Para atualizar o arquivo de fuso horário, você pode usar as seguintes estratégias.

Tópicos

- Atualizar o arquivo de fuso horário sem atualizar o mecanismo
- Atualizar o arquivo de fuso horário e a versão do mecanismo de banco de dados
- Atualizar a versão do mecanismo de banco de dados sem atualizar o arquivo de fuso horário

Atualizar o arquivo de fuso horário sem atualizar o mecanismo

Nesse cenário, seu banco de dados está usando o DSTv33, mas o DSTv34 está disponível no sistema de arquivos de sua instância de banco de dados. Você deseja atualizar o arquivo de fuso horário usado por sua instância de banco de dados de DSTv33 para DSTv34, mas não deseja atualizar seu mecanismo para uma nova versão secundária, que inclua DSTv35.

Em um comando `add-option-to-option-group`, adicione `TIMEZONE_FILE_AUTOUPGRADE` ao grupo de opções usado pela instância de banco de dados. Especifique se deseja adicionar a opção imediatamente ou adiá-la para a janela de manutenção. Depois de aplicar a opção `TIMEZONE_FILE_AUTOUPGRADE`, o RDS faz o seguinte:

1. Confere se há uma nova versão do DST.
2. Determina se o DSTv34 está disponível no sistema de arquivos.
3. Atualiza o arquivo de fuso horário imediatamente.

Atualizar o arquivo de fuso horário e a versão do mecanismo de banco de dados

Nesse cenário, seu banco de dados está usando o DSTv33, mas o DSTv34 está disponível no sistema de arquivos de sua instância de banco de dados. Você deseja atualizar seu mecanismo de banco de dados para a versão secundária 19.0.0.0.ru-2022-10.rur-2022-10.r1, que inclui DSTv35 e atualizar o arquivo de fuso horário para DSTv35 durante a atualização do mecanismo. Portanto, seu objetivo é ignorar o DSTv34 e atualizar os arquivos de fuso horário diretamente para DSTv35.

Para atualizar o mecanismo e o arquivo de fuso horário juntos, execute `modify-db-instance` com as opções `--option-group-name` e `--engine-version`. É possível executar o comando

imediatamente ou adiá-lo para a janela de manutenção. In `--option-group-name`, especifique um grupo de opções que inclua a opção `TIMEZONE_FILE_AUTOUPGRADE`. Por exemplo:

```
aws rds modify-db-instance
    --db-instance-identifier my-instance \
    --engine-version new-version \
    ----option-group-name og-with-timezone-file-autoupgrade \
    --apply-immediately
```

O RDS começa a atualizar o mecanismo para 19.0.0.0.ru-2022-10.rur-2022-10.r1. Depois que a opção `TIMEZONE_FILE_AUTOUPGRADE` é aplicada, o RDS confere se há uma nova versão do DST, percebe que o DSTv35 está disponível em 19.0.0.0.ru-2022-10.rur-2022-10.r1 e inicia imediatamente a atualização para DSTv35.

Para atualizar o mecanismo imediatamente e, depois, atualizar o arquivo de fuso horário, execute as operações na sequência:

1. Atualize o mecanismo de banco de dados usando o seguinte comando da CLI:

   ```
   aws rds modify-db-instance \
       --db-instance-identifier my-instance \
       --engine-version new-version \
       --apply-immediately
   ```

2. Adicione a opção `TIMEZONE_FILE_AUTOUPGRADE` ao grupo de opções anexado à instância usando o seguinte comando da CLI:

   ```
   aws rds add-option-to-option-group \
       --option-group-name og-in-use-by-your-instance \
       --options OptionName=TIMEZONE_FILE_AUTOUPGRADE \
       --apply-immediately
   ```

Atualizar a versão do mecanismo de banco de dados sem atualizar o arquivo de fuso horário

Nesse cenário, seu banco de dados está usando o DSTv33, mas o DSTv34 está disponível no sistema de arquivos de sua instância de banco de dados. Você deseja atualizar seu mecanismo de banco de dados para a versão 19.0.0.0.ru-2022-10.rur-2022-10.r1, que inclui o DSTv35, mas reter o arquivo de fuso horário DSTv33. Você pode fazer isso pelas seguintes razões:

- Seus dados não usam o tipo de dados `TIMESTAMP WITH TIME ZONE`.

- Seus dados usam o tipo de dados `TIMESTAMP WITH TIME ZONE`, mas eles não são afetados pelas alterações de fuso horário.
- Você deseja adiar a atualização do arquivo de fuso horário porque não pode tolerar o tempo de inatividade extra.

Sua estratégia depende de qual das seguintes possibilidades é verdadeira:

- Sua instância de banco de dados não está associada a um grupo de opções que inclui `TIMEZONE_FILE_AUTOUPGRADE`. No comando `modify-db-instance`, não especifique um novo grupo de opções para que o RDS não atualize o arquivo de fuso horário.
- A instância de banco de dados está atualmente associada a um grupo de opções que inclui `TIMEZONE_FILE_AUTOUPGRADE`. Em m um único comando `modify-db-instance`, associe a instância de banco de dados a um grupo de opções que não inclua `TIMEZONE_FILE_AUTOUPGRADE` e, depois, atualize o mecanismo de banco de dados para 19.0.0.0.ru-2022-10.rur-2022-10.r1.

Tempo de inatividade durante a atualização do arquivo de fuso horário

Quando o RDS atualiza o arquivo de fuso horário, os dados existentes que utilizam `TIMESTAMP WITH TIME ZONE` podem mudar. Nesse caso, sua consideração principal é o tempo de inatividade.

> **Warning**
>
> Se você adicionar a opção `TIMEZONE_FILE_AUTOUPGRADE`, a atualização do mecanismo poderá exigir um tempo de inatividade prolongado. A atualização de dados de fuso horário para um banco de dados grande pode levar horas ou até dias.

A duração da atualização do arquivo de fuso horário depende de fatores como os seguintes:

- A quantidade de dados `TIMESTAMP WITH TIME ZONE` em seu banco de dados
- A configuração da instância de banco de dados
- Classe da instância de banco de dados
- A configuração do armazenamento
- A configuração do banco de dados:
- As configurações de parâmetros do banco de dados

Um período de inatividade adicional pode ocorrer quando você:

- Adiciona a opção ao grupo de opções quando a instância de banco de dados usa um arquivo de fuso horário desatualizado
- Atualiza o mecanismo de banco de dados Oracle quando a nova versão do mecanismo contém uma nova versão do arquivo de fuso horário

> **Note**
>
> Durante a atualização do arquivo de fuso horário, o RDS for Oracle chama `PURGE DBA_RECYCLEBIN`.

Preparação para atualizar o arquivo de fuso horário

Uma atualização de arquivo de fuso horário tem duas fases separadas: preparar e atualizar. Embora não seja necessário, recomendamos enfaticamente que você execute a etapa de preparação. Nessa etapa, você descobre quais dados serão afetados pela execução do procedimento `DBMS_DST.FIND_AFFECTED_TABLES` de PL/SQL. Para ter mais informações sobre a janela de preparação, consulte Upgrading the Time Zone File and Timestamp with Time Zone Data (Atualização do arquivo de fuso horário e o carimbo de data/hora com os dados de fuso horário) na documentação do Oracle Database.

Para preparar a atualização do arquivo de fuso horário

1. Conecte ao banco de dados Oracle usando um cliente SQL.
2. Determine a versão atual do arquivo de fuso horário usada.

   ```
   SELECT * FROM V$TIMEZONE_FILE;
   ```

3. Determine a versão mais recente do arquivo de fuso horário disponível na instância de banco de dados

   ```
   SELECT DBMS_DST.GET_LATEST_TIMEZONE_VERSION FROM DUAL;
   ```

4. Determine o tamanho total das tabelas que têm colunas do tipo `TIMESTAMP WITH LOCAL TIME ZONE` ou `TIMESTAMP WITH TIME ZONE`.

```
SELECT SUM(BYTES)/1024/1024/1024 "Total_size_w_TSTZ_columns_GB"
FROM   DBA_SEGMENTS
WHERE  SEGMENT_TYPE LIKE 'TABLE%'
AND    (OWNER, SEGMENT_NAME) IN
         (SELECT OWNER, TABLE_NAME
          FROM   DBA_TAB_COLUMNS
          WHERE  DATA_TYPE LIKE 'TIMESTAMP%TIME ZONE');
```

5. Determine os nomes e tamanhos dos segmentos que têm colunas do tipo `TIMESTAMP WITH LOCAL TIME ZONE` ou `TIMESTAMP WITH TIME ZONE`.

```
SELECT OWNER, SEGMENT_NAME, SUM(BYTES)/1024/1024/1024
 "SEGMENT_SIZE_W_TSTZ_COLUMNS_GB"
FROM   DBA_SEGMENTS
WHERE  SEGMENT_TYPE LIKE 'TABLE%'
AND    (OWNER, SEGMENT_NAME) IN
         (SELECT OWNER, TABLE_NAME
          FROM   DBA_TAB_COLUMNS
          WHERE  DATA_TYPE LIKE 'TIMESTAMP%TIME ZONE')
GROUP BY OWNER, SEGMENT_NAME;
```

6. Execute a etapa de preparação.

   - O procedimento `DBMS_DST.CREATE_AFFECTED_TABLE` cria uma tabela para armazenar quaisquer dados afetados. Passe o nome desta tabela ao procedimento `DBMS_DST.FIND_AFFECTED_TABLES`. Para ter mais informações, consulte CREATE_AFFECTED_TABLE Procedure (Procedimento CREATE_AFECTED_TABLE) na documentação do Oracle Database.
   - O procedimento `CREATE_ERROR_TABLE` cria uma tabela para registrar erros. Para ter mais informações, consulte CREATE_ERROR_TABLE Procedure (Procedimento ERROR_TABLE) na documentação do Oracle Database.

   O exemplo a seguir cria os dados afetados e as tabelas de erro e localiza todas as tabelas de afetados.

```
EXEC DBMS_DST.CREATE_ERROR_TABLE('my_error_table')
EXEC DBMS_DST.CREATE_AFFECTED_TABLE('my_affected_table')

EXEC DBMS_DST.BEGIN_PREPARE(new_version);
```

```
EXEC DBMS_DST.FIND_AFFECTED_TABLES('my_affected_table', TRUE, 'my_error_table');
EXEC DBMS_DST.END_PREPARE;

SELECT * FROM my_affected_table;
SELECT * FROM my_error_table;
```

7. Consulte as tabelas de afetados e de erro.

```
SELECT * FROM my_affected_table;
SELECT * FROM my_error_table;
```

Adição da opção de atualização automática do arquivo de fuso horário

Quando você adiciona a opção a um grupo de opções, o grupo de opções está em um dos seguintes estados:

- Um grupo de opções existente está atualmente anexado a pelo menos uma instância de banco de dados. Quando você adiciona a opção, todas as instâncias de banco de dados que usam esse grupo de opções são reiniciadas automaticamente. Isso causa uma breve interrupção.
- Um grupo de opções existente não está anexado a nenhuma instância de banco de dados. Você planeja adicionar a opção, depois associar o grupo de opções existente às instâncias de banco de dados existentes ou a uma nova instância de banco de dados.
- Você cria um grupo de opções e adiciona a opção. Você planeja adicionar o novo grupo de opções às instâncias de banco de dados existentes ou a uma nova instância de banco de dados.

Console

Para adicionar a opção de atualização automática do arquivo de fuso horário a uma instância de banco de dados

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Option groups (Grupos de opções).
3. Determine o grupo de opções que você deseja usar. Você pode criar um novo grupo de opções ou usar um existente. Se você quiser usar um grupo de opções existente, vá para a próxima etapa. Caso contrário, crie um grupo de opções de banco de dados personalizado com as seguintes configurações:

a. Em Engine (Mecanismo), escolha a edição do Oracle Database para sua instância de banco de dados.

b. Para Major engine version (Versão do mecanismo principal), escolha a versão da sua instância de banco de dados.

Para ter mais informações, consulte Criar um grupo de opções.

4. Selecione o grupo de opções que você deseja modificar e escolha Add Option (Adicionar opção).
5. Na janela Add option (Adicionar opção), faça o seguinte:

a. Escolha TIMEZONE_FILE_AUTOUPGRADE.

b. Para habilitar a opção em todas as instâncias de banco de dados associadas assim que você a adicionar, em Apply Immediately (Aplicar imediatamente), escolha Yes (Sim). Se você escolher No (Não) (o padrão), a opção será habilitada para cada instância de banco de dados associada durante sua próxima janela de manutenção.

6. Quando estiver satisfeito com as configurações, escolha Add Option (Adicionar opção).

AWS CLI

O exemplo a seguir usa o comando da AWS CLI add-option-to-option-group para adicionar a opção `TIMEZONE_FILE_AUTOUPGRADE` a um grupo de opções chamado `myoptiongroup`.

Para Linux, macOS ou Unix:

```
aws rds add-option-to-option-group \
    --option-group-name "myoptiongroup" \
    --options "OptionName=TIMEZONE_FILE_AUTOUPGRADE" \
    --apply-immediately
```

Para Windows:

```
aws rds add-option-to-option-group ^
    --option-group-name "myoptiongroup" ^
    --options "OptionName=TIMEZONE_FILE_AUTOUPGRADE" ^
    --apply-immediately
```

Verificação dos dados após a atualização do arquivo de fuso horário

Recomendamos verificar seus dados após atualizar o arquivo de fuso horário. Durante a etapa de preparação, o RDS for Oracle cria automaticamente as seguintes tabelas:

- `rdsadmin.rds_dst_affected_tables`: lista as tabelas que contêm dados afetados pela atualização
- `rdsadmin.rds_dst_error_table`: lista os erros gerados durante a atualização

Essas tabelas são independentes de todas as tabelas criadas na janela de preparação. Para ver os resultados da atualização, consulte as tabelas conforme mostrado a seguir.

```
SELECT * FROM rdsadmin.rds_dst_affected_tables;
SELECT * FROM rdsadmin.rds_dst_error_table;
```

Para ter mais informações sobre o esquema das tabelas de dados afetados e de erros, consulte FIND_AFFECTED_TABLES Procedure (Processo FIND_AFECTED_TABLES) na documentação do Oracle.

Oracle Transparent Data Encryption

O Amazon RDS oferece suporte para ao Oracle Transparent Data Encryption (TDE), um recurso da opção Oracle Advanced Security disponível no Oracle Enterprise Edition. Esse recurso criptografa os dados automaticamente antes de gravá-los no armazenamento e os descriptografa automaticamente quando os são lidos. A opção é compatível somente com o modelo traga a sua própria licença (BYOL).

A TDE é útil em situações nas quais é preciso criptografar dados confidenciais, caso os arquivos de dados e os backups sejam obtidos por um terceiro. Ela também é útil quando você precisa estar em conformidade com os regulamentos relacionados à segurança.

Uma explicação detalhada sobre a TDE no Oracle Database está além do escopo deste guia. Para ter informações, consulte os seguintes recursos do Oracle Database:

- Introduction to Transparent Data Encryption na documentação do banco de dados Oracle
- Oracle advanced security na documentação do Oracle Database.
- Oracle advanced security Transparent Data Encryption best practices, que é um whitepaper da Oracle.

Para ter mais informações sobre como usar o TDE com o RDS para Oracle, consulte os seguintes blogs:

- Oracle Database Encryption Options on Amazon RDS
- Migrate a cross-account TDE-enabled Amazon RDS for Oracle DB instance with reduced downtime using AWS DMS

Modos de criptografia do TDE

O Oracle Transparent Data Encryption oferece suporte para dois modos de criptografia: a criptografia de espaço de tabela do TDE e a criptografia de coluna do TDE. A criptografia de espaço de tabela do TDE é usada para criptografar todas as tabelas de aplicativos. A criptografia de coluna do TDE é usada para criptografar elementos de dados individuais que contêm dados confidenciais. Você também pode aplicar uma solução de criptografia híbrida que use tanto a criptografia de espaço de tabela quanto a criptografia de coluna do TDE.

> **Note**
>
> O Amazon RDS gerencia a chave mestra do Oracle Wallet e do TDE para a instância de banco de dados. Você não precisa definir a chave de criptografia usando o comando `ALTER SYSTEM set encryption key`.

Depois de habilitar a opção `TDE`, será possível conferir o status do Oracle Wallet usando o seguinte comando:

```
SELECT * FROM v$encryption_wallet;
```

Para criar um espaço de tabela criptografado, use o seguinte comando:

```
CREATE TABLESPACE encrypt_ts ENCRYPTION DEFAULT STORAGE (ENCRYPT);
```

Para especificar o algoritmo de criptografia, use o seguinte comando:

```
CREATE TABLESPACE encrypt_ts ENCRYPTION USING 'AES256' DEFAULT STORAGE (ENCRYPT);
```

As declarações anteriores para criptografar um espaço de tabela são as mesmas que você usaria em um banco de dados Oracle on-premises.

Restrições para a opção TDE

A opção TDE é permanente e persistente. Após a associação da instância de banco de dados a um grupo de opções com a opção TDE habilitada, não é possível realizar as seguintes ações:

- Desabilitar a opção `TDE` no grupo de opções atualmente associado.
- Associar a instância de banco de dados a um grupo de opções diferente que não inclua a opção `TDE`.
- Compartilhar um snapshot de banco de dados que usa a opção `TDE`. Para obter mais informações sobre como compartilhar snapshot de banco de dadoss, consulte Compartilhar um snapshot de banco de dados para o Amazon RDS.

Para ter mais informações sobre opções persistentes e permanentes, consulte Opções persistentes e permanentes.

Determinar se a instância de banco de dados está usando a TDE

Convém determinar se a instância de banco de dados está associada a um grupo de opções que tenha a opção TDE habilitada. Para visualizar o grupo de opções ao qual uma instância de banco de dados está associada, use o console do RDS, o comando describe-db-instance da AWS CLI ou a operação DescribeDBInstances da API.

Adicionar a opção TDE

Para adicionar a opção TDE à instância de banco de dados, siga estas etapas:

1. (Recomendado) Crie um snapshot da instância de banco de dados.
2. Realize uma destas tarefas:
   - Crie um grupo de opções do zero. Para ter mais informações, consulte Criar um grupo de opções.
   - Copie um grupo de opções existente usando a AWS CLI ou a API. Para ter mais informações, consulte Copiar um grupo de opções.
   - Reutilize um grupo de opções não padrão. Uma prática recomendada é usar um grupo de opções que não esteja atualmente associado a nenhuma instância de banco de dados ou snapshot.
3. Adicione a nova opção ao grupo de opções da etapa anterior.
4. Se o grupo de opções atualmente associado à instância de banco de dados tiver opções habilitadas, adicione essas opções ao novo grupo de opções. Essa estratégia evita que as opções existentes sejam desinstaladas ao habilitar a nova opção.
5. Adicione o novo grupo de opções à instância de banco de dados.

Console

Como adicionar a opção TDE a um grupo de opções e associá-la à instância de banco de dados

1. No console do RDS, escolha Grupos de opções.
2. Escolha o nome do grupo de opções ao qual você deseja adicionar a opção.
3. Escolha Add option (Adicionar opção).
4. Em Nome da opção, escolha TDE e defina as configurações da opção.
5. Escolha Add option (Adicionar opção).

> **Important**
>
> Se você adicionar a opção TDE a um grupo de opções que está anexado a uma ou mais instâncias de banco de dados, uma breve interrupção ocorrerá enquanto todas as instâncias de banco de dados são reiniciadas automaticamente.

Para ter mais informações sobre a adição de opções, consulte Adicionar uma opção a um grupo de opções.

6. Associe o grupo de opções a uma instância de banco de dados nova ou existente:

   - Para uma nova instância de banco de dados, aplique o grupo de opções ao executar a instância. Para obter mais informações, consulte Criar uma instância de banco de dados do Amazon RDS.
   - Para uma instância de banco de dados existente, aplique o grupo de opções modificando a instância e anexando o novo grupo de opções. A instância de banco de dados não é reiniciada como parte dessa operação. Para ter mais informações, consulte Modificar uma instância de banco de dados do Amazon RDS.

AWS CLI

No exemplo a seguir, você usa o comando da AWS CLI add-option-to-option-group para adicionar a opção TDE a um grupo de opções chamado `myoptiongroup`. Consulte mais informações em Getting started: Flink 1.13.2.

Para Linux, macOS ou Unix:

```
aws rds add-option-to-option-group \
    --option-group-name "myoptiongroup" \
    --options "OptionName=TDE" \
    --apply-immediately
```

Para Windows:

```
aws rds add-option-to-option-group ^
    --option-group-name "myoptiongroup" ^
    --options "OptionName=TDE" ^
```

```
    --apply-immediately
```

Copiar os dados para uma instância de banco de dados que não inclua a opção TDE

Não é possível remover a opção TDE de uma instância de banco de dados nem a associar a um grupo de opções sem a opção TDE. Para migrar os dados para uma instância que não inclua a opção TDE, faça o seguinte:

1. Descriptografe os dados na instância de banco de dados.
2. Copie os dados para uma nova instância de banco de dados que não esteja associada a um grupo de opções com a opção `TDE` habilitada.
3. Exclua a instância de banco de dados original.

É possível usar o mesmo nome da instância de banco de dados anterior para a nova instância.

Considerações ao usar o TDE com o Oracle Data Pump

É possível usar o Oracle Data Pump para importar ou exportar arquivos de despejo criptografados. O Amazon RDS é compatível com modo de criptografia de senha (`ENCRYPTION_MODE=PASSWORD`) para o Oracle Data Pump. O Amazon RDS não é compatível com o modo de criptografia transparente (`ENCRYPTION_MODE=TRANSPARENT`) para o Oracle Data Pump. Para ter mais informações, consulte Importar usando o Oracle Data Pump.

Oracle UTL_MAIL

O Amazon RDS oferece suporte para Oracle UTL_MAIL com o uso da opção UTL_MAIL e de servidores SMTP. É possível enviar e-mails diretamente de seu banco de dados usando o pacote UTL_MAIL. O Amazon RDS é compatível com o UTL_MAIL para as seguintes edições e versões do Oracle:

- Oracle Database 21c (21.0.0.0), todas as versões
- Oracle Database 19c (19.0.0.0), todas as versões

Veja a seguir algumas limitações do uso do UTL_MAIL:

- UTL_MAIL não oferece suporta para Transport Layer Security (TLS) e, portanto, os e-mails não são criptografados.

  Para se conectar com segurança a recursos SSL/TLS remotos criando e carregando carteiras Oracle personalizadas, siga as instruções em Configurar o acesso UTL_HTTP usando certificados e uma carteira Oracle.

  Os certificados específicos que são necessários para sua carteira variam por serviço. Para serviços da AWS, eles geralmente podem ser encontrados no Repositório de serviços de confiança da Amazon.
- UTL_MAIL não oferece suporte para autenticação com servidores SMTP.
- Você só pode enviar um anexo em um e-mail.
- Você não pode enviar anexos com mais de 32 K.
- Você só pode usar as codificações de caracteres ASCII e Extended Binary Coded Decimal Interchange Code (EBCDIC).
- A porta SMTP (25) é limitada com base nas políticas do proprietário da interface de rede elástica.

Quando você habilita o UTL_MAIL, somente o usuário mestre da sua instância de banco de dados recebe o privilégio de execução. Se necessário, o usuário mestre pode conceder o privilégio de execução a outros usuários para que eles possam usar o UTL_MAIL.

> **Important**
>
> Recomendamos que você habilite o recurso de auditoria incorporada da Oracle para rastrear o uso dos procedimentos UTL_MAIL.

Pré-requisitos para o Oracle UTL_MAIL

Veja a seguir os pré-requisitos para usar o Oracle UTL_MAIL:

- Um ou mais servidores SMTP, e os endereços IP correspondentes ou nomes de servidor de nomes de domínio (DNS) públicos ou particulares. Para obter mais informações sobre nomes DNS privados resolvidos através de um servidor DNS personalizado, consulte Configuração de um servidor DNS personalizado.

Adicionar a opção Oracle UTL_MAIL

O processo geral para adicionar a opção Oracle UTL_MAIL a uma instância de banco de dados é o seguinte:

1. Crie um novo grupo de opções, ou copie ou modifique um existente.
2. Adicione a opção ao grupo de opções.
3. Associe o grupo de opções à instância de banco de dados.

Depois de adicionar a opção UTL_MAIL, assim que o grupo de opções estiver ativo, o UTL_MAIL ficará ativo.

Para adicionar a opção UTL_MAIL a uma instância de banco de dados

1. Determine o grupo de opções que você deseja usar. Você pode criar um novo grupo de opções ou usar um existente. Se você quiser usar um grupo de opções existente, vá para a próxima etapa. Caso contrário, crie um grupo de opções de banco de dados personalizado com as seguintes configurações:
   a. Em Engine (Mecanismo), selecione a edição do Oracle que você deseja usar.
   b. Para Major engine version (Versão do mecanismo principal), escolha a versão da sua instância de banco de dados.

Para obter mais informações, consulte Criar um grupo de opções.

2. Adicione a opção UTL_MAIL ao grupo de opções. Para mais informações sobre a adição de opções, consulte Adicionar uma opção a um grupo de opções.
3. Aplique o grupo de opções a uma instância de banco de dados nova ou existente:

   - Para uma nova instância de banco de dados, você aplica o grupo de opções ao executar a instância. Para ter mais informações, consulte Criar uma instância de banco de dados do Amazon RDS.
   - Para uma instância de banco de dados existente, você aplica o grupo de opções modificando a instância e anexando o novo grupo de opções. Para obter mais informações, consulte Modificar uma instância de banco de dados do Amazon RDS.

Uso do Oracle UTL_MAIL

Depois de habilitar a opção UTL_MAIL, você deve configurar o servidor SMTP antes de começar a usá-lo.

Você configura o servidor SMTP definindo o parâmetro SMTP_OUT_SERVER como um endereço IP válido ou um nome DNS público. Para o parâmetro SMTP_OUT_SERVER, você pode especificar uma lista separada por vírgulas dos endereços de vários servidores. Se o primeiro servidor não estiver disponível, o UTL_MAIL tentará o próximo servidor, e assim por diante.

Você pode definir o SMTP_OUT_SERVER padrão para uma instância de banco de dados usando um parameter group de banco de dados. Você pode definir o parâmetro SMTP_OUT_SERVER para uma sessão executando o seguinte código no seu banco de dados na sua instância de banco de dados.

```
ALTER SESSION SET smtp_out_server = mailserver.domain.com:25;
```

Depois que a opção UTL_MAIL estiver ativada e o seu SMTP_OUT_SERVER estiver configurado, você poderá enviar o e-mail usando o procedimentoSEND. Para obter mais informações, consulte UTL_MAIL na documentação da Oracle.

Remover a opção Oracle UTL_MAIL

Você pode remover a opção Oracle UTL_MAIL de uma instância de banco de dados.

Para remover UTL_MAIL de uma instância de banco de dados, siga um destes procedimentos:

- Para remover UTL_MAIL de várias instâncias de bancos de dados, remova a opção UTL_MAIL do grupo de opções ao qual elas pertencem. Essa alteração afeta todas as instâncias de bancos de dados que usam o grupo de opções. Para obter mais informações, consulte Remover uma opção de um grupo de opções.
- Para remover UTL_MAIL de uma única instância de banco de dados, modifique essa instância e especifique um grupo de opções diferente que não inclua a opção UTL_MAIL. Você pode especificar um grupo de opções padrão (vazio) ou criar um grupo de opções personalizado diferente. Para obter mais informações, consulte Modificar uma instância de banco de dados do Amazon RDS.

Solução de problemas

Os problemas a seguir podem ocorrer ao usar o UTL_MAIL com o Amazon RDS.

- Limitação. A porta SMTP (25) é limitada com base nas políticas do proprietário da interface de rede elástica. Se você puder enviar e-mails com sucesso usando o UTL_MAIL e vir o erro `ORA-29278: SMTP transient error: 421 Service not available`, você provavelmente está sendo limitado. Se você perceber limitação com a entrega de e-mails, recomendamos implementar um algoritmo de recuo. Para obter mais informações sobre algoritmos de recuo, consulte Repetições de erro e recuo exponencial na AWS e Como lidar com um erro "Limitação taxa máxima de envio excedida".

  Você pode solicitar que essa limitação seja removida. Para obter mais informações, consulte Como faço para remover a limitação da porta 25 na minha instância do EC2?.

Oracle XML DB

O Oracle XML DB adiciona suporte para XML nativo à sua instância de banco de dados. Com o XML DB, é possível armazenar e recuperar XML estruturado ou não estruturado, além de dados relacionais. O servidor de protocolo XML DB não comporta o RDS para Oracle.

O XML DB está pré-instalado no Oracle Database 12c e posterior. Portanto, você não precisa usar um grupo de opções para instalar explicitamente o XML DB como um recurso adicional.

Para saber como configurar e usar o XML DB, consulte o Oracle XML DB Developer's Guide na documentação do Oracle Database.

Fazer upgrade do mecanismo de banco de dados RDS para Oracle

Quando o Amazon RDS oferece suporte a uma nova versão do Oracle Database, você pode fazer upgrade de suas instâncias de banco de dados para essa nova versão. Para obter informações sobre quais versões do Oracle estão disponíveis no Amazon RDS, consulte Notas de lançamento do Amazon RDS for Oracle.

> **Important**
>
> Os bancos de dados RDS para Oracle 11g, 12c e 18c não são mais compatíveis. Se você mantiver snapshots do Oracle Database 11g, 12c ou 18v, poderá atualizá-los para uma versão posterior. Para ter mais informações, consulte Como atualizar para um snapshot de banco de dados Oracle.

Tópicos

- Visão geral das atualizações do mecanismo RDS para Oracle
- Atualizações da versão principal do Oracle
- Atualizações de versões secundárias Oracle
- Considerações sobre atualizações de bancos de dados Oracle
- Testar uma atualização do Oracle DB
- Atualizar a versão de uma instância de banco de dados do RDS para Oracle
- Como atualizar para um snapshot de banco de dados Oracle

Visão geral das atualizações do mecanismo RDS para Oracle

Antes de atualizar sua instância de banco de dados do RDS para Oracle, familiarize-se com os seguintes conceitos.

Tópicos

- Upgrades de versões principais e secundárias
- Datas de suporte e atualizações obrigatórias referentes ao RDS para Oracle
- Oracle Engine Version Management
- Instantâneos automáticos durante atualizações do mecanismo

- Atualizações do Oracle em uma implantação Multi-AZ
- Atualizações do Oracle de réplicas de leitura

Upgrades de versões principais e secundárias

As versões principais são os principais lançamentos do Oracle Database que ocorrem a cada 1–2 anos. O Oracle Database 19c e o Oracle Database 21c são exemplos de versões principais.

A cada trimestre, o RDS para Oracle lança novas versões secundárias de mecanismos para cada mecanismo principal compatível. Uma atualização de versão (RU) do mecanismo incorpora correções de bugs da Oracle incluindo os patches de RU referentes ao trimestre especificado. Por exemplo, 21.0.0.0.ru-2024-10.rur-2024-10.r1 é uma versão secundária do Oracle Database 21c que incorpora a RU de outubro de 2024.

Uma versão do mecanismo do pacote de patches espaciais (SPB) contém patches de RU e patches específicos para o Oracle Spatial. Por exemplo, 19.0.0.0.ru-2025-01.spb-1.r1 é uma versão secundária do mecanismo que contém os patches de RU na versão do mecanismo 19.0.0.0.ru-2025-01.rur-2025-01.r1 e patches do Spatial. Normalmente, o RDS para Oracle lança SPBs duas a três semanas após a RU correspondente. Para obter uma explicação sobre as diferenças entre RUs e SPBs, consulte Atualizações de versão (RUs) e pacotes de patches espaciais (SPBs). Para ter mais informações, consulte Release notes for Amazon Relational Database Service (Amazon RDS) for Oracle.

O RDS para Oracle oferece suporte aos seguintes upgrades para uma instância de banco de dados.

Tipo de upgrade	Compatibilidade de aplicações	Métodos de upgrade	Exemplo de caminho de upgrade
Versão principal	Um upgrade de versão principal pode apresentar alterações não compatíveis com aplicações existentes.	Somente manual	Do Oracle Database 19c para o Oracle Database 21c
Versão secundária	Uma atualização de versão secundária inclui somente alterações compatíveis com versões anteriores dos aplicativos existentes.	Automático ou manual	De 21.0.0.0.ru-2023-07.rur-2022-07.r1 para 21.0.0.0.ru-2023-10.rur-2022-10.r1

> **Important**
>
> Quando você atualiza o mecanismo de banco de dados, ocorre uma interrupção. A duração da interrupção depende da versão do mecanismo e do tamanho da instância de banco de dados.
> Teste completamente qualquer atualização para verificar se as aplicações funcionam corretamente antes de aplicá-la aos bancos de dados de produção. Para obter mais informações, consulte Testar uma atualização do Oracle DB.

Datas de suporte e atualizações obrigatórias referentes ao RDS para Oracle

As versões de banco de dados do RDS para Oracle têm datas de suporte previstas. Quando uma versão principal ou secundária de um mecanismo de banco de dados do RDS para Oracle se aproxima da data de fim do suporte, o RDS inicia as atualizações obrigatórias, também conhecidas como atualizações forçadas. O RDS publica as seguintes informações:

- Uma recomendação para você começar a atualizar manualmente as instâncias em versões obsoletas para versões compatíveis.
- Uma data após a qual não é possível mais criar instâncias nas versões não compatíveis.
- Uma data em que o RDS começa a atualizar automaticamente as instâncias para versões compatíveis durante as janelas de manutenção.
- Uma data em que o RDS começa a atualizar automaticamente as instâncias para versões compatíveis fora das janelas de manutenção.

> **Important**
>
> As atualizações forçadas podem gerar consequências inesperadas para as pilhas do AWS CloudFormation. Caso dependa do RDS para atualizar suas instâncias de banco de dados automaticamente, você poderá encontrar problemas com o AWS CloudFormation.

Esta seção contém os seguintes tópicos:

Tópicos

- Datas de suporte para versões principais do RDS para Oracle

- Datas de suporte para versões secundárias do RDS para Oracle

Datas de suporte para versões principais do RDS para Oracle

As versões principais do RDS para Oracle continuam disponíveis pelo menos até a data de fim do suporte para a versão correspondente do Oracle Database. É possível usar as seguintes datas para planejar seus ciclos de teste e atualização. Essas datas representam a data mais antiga em que uma atualização para uma versão mais recente pode ser necessária. Se a Amazon estender o suporte para uma versão do RDS para Oracle por mais tempo do que o planejado originalmente, planejamos atualizar essa tabela para refletir a data posterior.

> **Note**
>
> Você pode visualizar as versões principais dos bancos de dados Oracle executando o comando describe-db-major-engine-versions AWS CLI ou usando a operação de API DescribeDBMajorEngineVersions do RDS.

Versão principal do Oracle Database	Data esperada para atualização para uma versão mais recente
Oracle Database 19c	31 de dezembro de 2029 com o BYOL Premier Support (isenção de taxas para suporte estendido) 31 de dezembro de 2032 com suporte estendido BYOL (custo adicional) ou um contrato de licença ilimitada 31 de dezembro de 2029 com Licença Incluída (LI)
Oracle Database 21c	31 de julho de 2027 (não disponível para Suporte estendido)

O RDS enviará uma notificação pelo menos 12 meses antes de você precisar atualizar para uma versão principal mais recente. A notificação descreve o processo de atualização, como o momento de marcos importantes, o impacto nas instâncias de banco de dados e as ações recomendadas. Recomendamos que você teste detalhadamente suas aplicações com as novas versões do RDS para Oracle antes de atualizar seu banco de dados para uma versão principal.

Depois desse período de notificação prévia, um upgrade automático para a versão principal subsequente poderá ser aplicado a qualquer instância de banco de dados que ainda estiver executando a versão mais antiga. Nesse caso, a atualização é iniciada durante as janelas de manutenção agendada.

Para ter mais informações, consulte Release Schedule of Current Database Releases no My Oracle Support.

Datas de suporte para versões secundárias do RDS para Oracle

As seguintes versões secundárias do Oracle Database 19c estão se aproximando da data de encerramento do suporte:

- 19.0.0.0.ru-2019-07.rur-2019-07.r1
- 19.0.0.0.ru-2019-10.rur-2019-10.r1
- 19.0.0.0.ru-2020-01.rur-2020-01.r1

As seguintes versões secundárias do Oracle Database 21c estão se aproximando da data de encerramento do suporte:

- 21.0.0.0.ru-2022-01.rur-2022-01.r1
- 21.0.0.0.ru-2022-04.rur-2022-04.r1
- 21.0.0.0.ru-2022-07.rur-2022-07.r1
- 21.0.0.0.ru-2022-10.rur-2022-10.r1
- 21.0.0.0.ru-2023-01.rur-2023-01.r1
- 21.0.0.0.ru-2023-01.rur-2023-01.r2

A tabela a seguir mostra um cronograma das atualizações obrigatórias de versão secundária referentes a versões anteriores descontinuadas.

Data	Ação referente a versões descontinuadas
2/7/2025	Não é possível mais criar instâncias de banco de dados executando versões descontinuadas. Atualize suas instâncias de banco de dados manualmente para a última atualização de versão (RU) ou espere que o RDS as atualize automaticamente.

Data	Ação referente a versões descontinuadas
1.º/8/25-15/9/25	O RDS força a atualização das instâncias de banco de dados na janela de manutenção programada para a RU mais recente. O RDS também atualiza todas as instâncias restauradas com base em snapshots das versões descontinuadas.
15/9/2025	O RDS força a atualização das instâncias de banco de dados para a RU mais recente independentemente da janela de manutenção programada. As atualizações forçadas continuam até que todas as instâncias estejam executando versões secundárias compatíveis.

Se você não quiser que o RDS atualize automaticamente suas instâncias que estão executando versões obsoletas, faça o seguinte:

- Atualize suas versões secundárias do Oracle Database 19c e 21c manualmente para a RU mais recente ou para qualquer RU que não esteja programada para atualizações obrigatórias. Por exemplo, é possível atualizar 21.0.0.0.ru-2023-01.rur-2023-01.r2 para 21.0.0.0.ru-2023-04.rur-2023-04.r2 ou atualizar 19.0.0.0.ru-2019-07.rur-2019-07.r1 para 19.0.0.0.ru-2025-01.rur-2025-01.r2.
- Atualize os snapshots de banco de dados que usam versões secundárias obsoletas e, em seguida, restaure os snapshots.

Para obter mais informações sobre versões secundárias, consulte Release notes for Amazon Relational Database Service (Amazon RDS) para Oracle.

Oracle Engine Version Management

Com o gerenciamento de versão do mecanismo de banco de dados, você controla quando e como o mecanismo é corrigido e atualizado. Com esse recurso, você obtém flexibilidade para manter a compatibilidade com versões de patch do mecanismo de banco de dados. Também é possível testar novas versões de patch do RDS para Oracle para garantir que elas funcionem de forma eficiente com a aplicação antes de implantá-las na produção. Além disso, você atualiza as versões em seus próprios prazos e cronogramas.

> **Note**
>
> O Amazon RDS agrega periodicamente os patches oficiais de bancos de dados Oracle usando uma versão do mecanismo de banco de dados específica do Amazon RDS. Para ver uma lista de quais patches do Oracle estão contidos em uma versão de mecanismo específica do Oracle do Amazon RDS, acesse Notas de lançamento do Amazon RDS for Oracle.

Instantâneos automáticos durante atualizações do mecanismo

Quando você atualiza uma instância de banco de dados Oracle, os snapshots oferecem proteção contra problemas de atualização. Se o período de retenção de backup da instância de banco de dados for maior que 0, o Amazon RDS os seguintes snapshots de banco de dados serão realizados durante a atualização:

1. Um snapshot da instância de banco de dados antes de qualquer alteração de atualização ter sido feita. Se o upgrade falhar, você poderá restaurar esse snapshot para criar uma instância de banco de dados executando a versão antiga.
2. Um snapshot da instância de banco de dados após a conclusão da atualização.

> **Note**
>
> Para alterar o período de retenção de backup, consulte Modificar uma instância de banco de dados do Amazon RDS.

Após uma atualização, você não pode reverter para a versão anterior do mecanismo. No entanto, você pode criar uma nova instância de banco de dados Oracle restaurando o snapshot pré-atualização.

Atualizações do Oracle em uma implantação Multi-AZ

Se a instância de banco de dados estiver em uma implantação Multi-AZ, o Amazon RDS atualizará as réplicas primária e em espera. Se nenhuma atualização do sistema operacional for necessária, as atualizações primária e em espera ocorrerão simultaneamente. As instâncias não estarão disponíveis até que a atualização seja concluída.

Se forem necessárias atualizações do sistema operacional em uma implantação multi-AZ, o Amazon RDS aplicará as atualizações quando você solicitar o upgrade do banco de dados. O Amazon RDS executa as seguintes etapas:

1. Atualiza o sistema operacional na instância de banco de dados atual em espera.
2. Executa failover na instância de banco de dados principal para a instância de banco de dados em espera.
3. Faz upgrade da versão do banco de dados na nova instância de banco de dados primária, que anteriormente era a instância em espera. O banco de dados primário fica indisponível durante a migração.
4. Atualiza o sistema operacional na nova instância de banco de dados em espera, que anteriormente era a instância de banco de dados principal.
5. Atualiza a versão do banco de dados na nova instância de banco de dados em espera.
6. Faz failover da nova instância de banco de dados primária de volta para a instância de banco de dados primária original e a nova instância de banco de dados em espera volta para a instância de banco de dados em espera original. Assim, o Amazon RDS retorna a configuração de replicação ao seu estado original.

Atualizações do Oracle de réplicas de leitura

A versão do mecanismo de banco de dados Oracle da instância de banco de dados de origem e todas as réplicas de leitura devem ser iguais. O Amazon RDS executa a atualização nos seguintes estágios:

1. Atualiza a instância de banco de dados de origem. As réplicas de leitura estão disponíveis durante esse estágio.
2. Atualiza as réplicas de leitura em paralelo, independentemente das janelas de manutenção de réplicas. O banco de dados de origem está disponível durante esse estágio.

Para atualizações da versão principal de réplicas de leitura entre regiões, o Amazon RDS executa ações adicionais:

- Gera um grupo de opções para a versão de destino automaticamente
- Copia todas as opções e as configurações de opção do grupo de opções original para o novo grupo de opções
- Associa a réplica de leitura entre regiões atualizada ao novo grupo de opções

Atualizações da versão principal do Oracle

Para realizar uma atualização de versão principal, modifique manualmente a instância de banco de dados. As atualizações de versões principais não ocorrem automaticamente.

> **Important**
>
> Teste completamente qualquer atualização para verificar se as aplicações funcionam corretamente antes de aplicá-la aos bancos de dados de produção. Para ter mais informações, consulte Testar uma atualização do Oracle DB.

Tópicos

- Versões compatíveis com atualizações principais
- Classes de instância compatíveis com atualizações principais
- Coletar estatísticas antes de atualizações principais
- Permitir atualizações principais

Versões compatíveis com atualizações principais

O Amazon RDS oferece suporte às atualizações da versão principal a seguir.

Versão atual	Atualização compatível
19.0.0.0 usando a arquitetura CDB	21.0.0.0

Uma atualização de versão principal do banco de dados Oracle deve atualizar para uma Release Update (RU) que foi lançada no mesmo mês ou posterior. Downgrades da versão principal não são compatíveis com nenhuma versão do Oracle Database.

Classes de instância compatíveis com atualizações principais

A instância de banco de dados Oracle pode ser executada em uma classe da instância de banco de dados não compatível com a versão para a qual você está atualizando. Nesse caso, migre a instância de banco de dados para uma classe de instância de banco de dados compatível antes da atualização. Para obter mais informações sobre as classes de instância de banco de dados

compatíveis de cada versão e edição do Amazon RDS for Oracle, consulte Classes de instâncias de banco de dados .

Coletar estatísticas antes de atualizações principais

Para realizar uma atualização de versão principal, a Oracle recomenda coletar estatísticas do otimizador sobre a instância de banco de dados que você está atualizando. Essa ação pode reduzir o tempo de inatividade da instância de banco de dados durante a atualização.

Para coletar estatísticas do otimizador, conecte-se à instância de banco de dados como o usuário mestre e execute o procedimento `DBMS_STATS.GATHER_DICTIONARY_STATS`, como no exemplo a seguir.

```
EXEC DBMS_STATS.GATHER_DICTIONARY_STATS;
```

Consulte mais informações em GATHER_DICTIONARY_STATS Procedure na documentação da Oracle.

Permitir atualizações principais

Uma atualização de versão principal do mecanismo pode ser incompatível com seu aplicativo. A atualização é irreversível. Se você especificar uma versão principal para o parâmetro EngineVersion diferente da versão principal atual, deverá permitir atualizações de versão principais.

Se atualizar uma versão principal usando o comando modify-db-instance da CLI, especifique `--allow-major-version-upgrade`. Essa configuração não é persistente, portanto, você deve especificar `--allow-major-version-upgrade` sempre que executar uma atualização principal. Este parâmetro não afeta as atualizações de versões secundárias do mecanismo. Para obter mais informações, consulte Atualizar a versão de mecanismo de uma instância de banco de dados.

Se você atualizar uma versão principal usando o console, não será necessário escolher uma opção para permitir a atualização. Em vez disso, o console exibe um aviso de que as atualizações principais são irreversíveis.

Atualizações de versões secundárias Oracle

No RDS para Oracle, uma atualização de versão secundária é uma atualização feita em uma versão principal do mecanismo de banco de dados. No RDS, uma versão secundária do mecanismo é uma atualização de versão (RU) ou um pacote de patches espaciais (SPB). Por exemplo, se sua instância

de banco de dados executa a versão principal do Oracle Database 21c e a versão secundária 21.0.0.0.ru-2024-10.rur-2024-10.r1, você pode atualizar o mecanismo de banco de dados para a versão secundária 21.0.0.0.ru-2025-01.rur-2025-01.r1. O RDS para Oracle não é compatível com downgrades de versões secundárias.

Você pode fazer upgrade do seu mecanismo de banco de dados para uma versão secundária manual ou automaticamente. Para saber como fazer upgrade manualmente, consulte Atualizar manualmente a versão do mecanismo. Para saber como configurar upgrades automáticos, consulte Atualizar automaticamente a versão do mecanismo espelho. Não importa se você faz upgrade manual ou automaticamente, o upgrade de versão secundária acarreta tempo de inatividade. Considere esse tempo de inatividade ao planejar suas atualizações.

> **Important**
>
> Teste completamente qualquer atualização para verificar se as aplicações funcionam corretamente antes de aplicá-la aos bancos de dados de produção. Para ter mais informações, consulte Testar uma atualização do Oracle DB.

Tópicos

- Atualizações de versão (RUs) e pacotes de patches espaciais (SPBs)
- Ativar as atualizações automáticas de versão secundária para Oracle
- Notificação de atualizações automáticas de versão secundária no RDS para Oracle
- Quando o RDS programa atualizações automáticas de versão secundária no RDS para Oracle
- Gerenciar uma atualização automática de versão secundária no RDS para Oracle

Atualizações de versão (RUs) e pacotes de patches espaciais (SPBs)

No RDS, uma atualização de versão (RU) é uma versão secundária trimestral do mecanismo que inclui correções de segurança, correções de erros e novos recursos para o Oracle Database. Um pacote de patches espaciais (SPB) é uma RU do mecanismo que inclui patches projetados para a opção Oracle Spatial. Por exemplo, o SPB denominado 19.0.0.0.ru-2025-01.spb-1.r1 inclui todos os patches na RU 19.0.0.0.ru-2025-01.rur-2025-01.r1 correspondente, além de patches específicos para o Spatial. Os SPBs são compatíveis somente com o Oracle Database 19c.

Quando a instância estiver configurada para atualizações automáticas de versões secundárias, as RUs e os SPBs estarão em caminhos de atualização separados. Normalmente, um SPB é lançado

duas a três semanas após a RU correspondente. A tabela a seguir mostra exemplos de versões secundárias do Oracle Database 19c.

Caminho de atualização de RU padrão	Caminho de atualização de SPB
19.0.0.0.ru-2025-01.rur-2025-01.r1	19.0.0.0.ru-2025-01.spb-1.r1
19.0.0.0.ru-2025-04.rur-2025-04.r1	19.0.0.0.ru-2025-04.spb-1.r1
19.0.0.0.ru-2025-07.rur-2025-07.r1	19.0.0.0.ru-2025-07.spb-1.r1
19.0.0.0.ru-2025-10.rur-2025-10.r1	19.0.0.0.ru-2025-10.spb-1.r1

Se sua instância de banco de dados estiver configurada para atualizações automáticas, ela estará no caminho de atualização correspondente à sua versão atual. Por exemplo, se a instância de banco de dados estiver executando a versão 19.0.0.0.ru-2025-01.rur-2025-01.r1 quando 19.0.0.0.ru-2025-04.rur-2025-04.r1 for lançada, ela será automaticamente atualizada para essa RU. Da mesma forma, se a instância de banco de dados estiver executando a versão 19.0.0.0.ru-2025-01.spb-1.r1 quando 19.0.0.0.ru-2025-04.spb-1.r1 for lançada, ela será automaticamente atualizada para esse SPB. A instância que estiver executando a versão 19.0.0.0.ru-2025-01.rur-2025-01.r1, que é uma RU, não será atualizada automaticamente para 19.0.0.0.ru-2025-04.spb-1.r1, que é um SPB em um caminho de atualização separado.

Você pode atualizar a instância de banco de dados para SPBs mesmo que ela não use patches espaciais, mas esses patches se aplicam somente ao Oracle Spatial. Você pode atualizar manualmente de uma RU para um SPB na mesma versão do mecanismo ou em uma versão posterior. Por exemplo, é possível atualizar a instância de 19.0.0.0.ru-2025-01.rur-2025-01.r1 para uma das seguintes versões do mecanismo:

- 19.0.0.0.ru-2025-01.spb-1.r1
- 19.0.0.0.ru-2025-04.spb-1.r1

Você pode atualizar a instância de um SPB para uma RU somente se a RU for uma versão posterior do mecanismo. Por exemplo, é possível atualizar do SPB 19.0.0.0.ru-2025-04.spb-1.r1 para a RU posterior 19.0.0.0.ru-2025-07.rur-2025-07.r1, mas não para a mesma RU 19.0.0.0.ru-2025-04.rur-2025-04.r1.

Se a instância de banco de dados estiver configurada para atualizações automáticas de versões secundárias e você fizer a atualização manualmente de uma RU para um SPB ou de um SPB para uma RU, o caminho de atualização automática será alterado. Suponha que você atualize manualmente da RU 19.0.0.0.ru-2025-01.rur-2025-01.r1 para o SPB 19.0.0.0.ru-2025-01.spb-1.r1. Sua próxima atualização automática de versão secundária será para a versão 19.0.0.0.ru-2025-04.spb-1.r1 do SPB.

Como os SPBs funcionam como RUs, as APIs do RDS para atualizar a instância para RUs e SPBs são idênticas. Os comandos a seguir demonstram a atualização para uma RU e para um SPB.

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --engine-version 19.0.0.0.ru-2025-01.rur-2025-01.r1

aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --engine-version 19.0.0.0.ru-2025-01.spb-1.r1
```

Para ter mais informações sobre como especificar a opção Oracle Spatial, consulte Como funcionam os pacotes de patches espaciais (SPBs). Para examinar uma lista de RUs e SPBs compatíveis com o Oracle Database 19c, consulte Amazon RDS for Oracle Database 19c (19.0.0.0).

Ativar as atualizações automáticas de versão secundária para Oracle

Em um upgrade automático de versão secundária, o RDS aplica a versão secundária mais recente disponível ao seu banco de dados Oracle sem intervenção manual. Uma instância de banco de dados do Amazon RDS para Oracle agenda seu upgrade durante a próxima janela de manutenção nas seguintes circunstâncias:

- A instância de banco de dados está com a opção Upgrade automático de versão secundária ativada.
- A instância de banco de dados não está executando a versão secundária mais recente do mecanismo de banco de dados.
- A instância de banco de dados não tem um upgrade de banco de dados pendente.

Para saber como ativar upgrades automáticos, consulte Atualizar automaticamente a versão do mecanismo espelho.

Notificação de atualizações automáticas de versão secundária no RDS para Oracle

O RDS publica um aviso prévio antes de começar a programar upgrades automáticos. Você pode encontrar a notificação na guia Manutenção e backups da página de detalhes do banco de dados. A mensagem tem o seguinte formato:

```
An automatic minor version upgrade to engine version will become available
 on availability-date and will be applied during a subsequent maintenance window.
```

A *availability-date* no aviso prévio é a data em que o RDS começa a programar upgrades para instâncias de banco de dados na Região da AWS. Não é a data em que o upgrade da instância de banco de dados está programado para ocorrer. Por exemplo, se a *availability-date* for 1.º de março, nessa data o RDS poderá programar o upgrade para 14 de abril.

Você também pode obter a data de disponibilidade do upgrade usando o comando `describe-pending-maintenance-actions` na AWS CLI, conforme mostrado no exemplo a seguir:

```
aws rds describe-pending-maintenance-actions

{
    "PendingMaintenanceActions": [
        {
            "ResourceIdentifier": "arn:aws:rds:us-east-1:123456789012:db:orclinst1",
            "PendingMaintenanceActionDetails": [
                {
                    "Action": "db-upgrade",
                    "Description": "Automatic minor version upgrade to
 21.0.0.0.ru-2024-07.rur-2024-07.r1",
                    "CurrentApplyDate": "2024-12-02T08:10:00Z",
                    "OptInStatus": "next-maintenance"
                }
            ]
        }, ...
```

A tabela a seguir detalha as opções disponíveis para cada tipo de mensagem de ação de manutenção pendente.

Mensagem de ação de manutenção pendente	Uma mensagem é exibida.	Elegível para ser aplicado na próxima janela de manutenção?	Elegível para ser aplicado imediatamente?	Elegível para que a aceitação seja desfeita?
Uma atualização automática de uma versão secundária para a *versão do mecanismo* estará disponível na *data de disponibilidade* e deverá ser aplicada durante uma janela de manutenção subsequente.	De quatro a seis semanas antes do agendamento das atualizações automáticas.	Sim	Sim	Sim
Atualização automática da versão secundária para a *versão do mecanismo*	Na *data de disponibilidade* ou após. O RDS aplica automaticamente essa atualização na próxima janela de manutenção da instância de banco de dados.	Sim	Sim	Não

Para ter mais informações sobre describe-pending-maintenance-actions, consulte a Referência de comandos da AWS CLI.

Quando o RDS programa atualizações automáticas de versão secundária no RDS para Oracle

Quando chega a data de disponibilidade dos upgrades automáticos, o RDS começa a programar os upgrades. Para a maioria das Regiões da AWS, o RDS programa um upgrade para a RU trimestral mais recente de quatro a seis semanas após a data ser disponibilizada. A data agendada varia

de acordo com a Região da AWS e outros fatores. Para ter mais informações sobre RUs e RURs, consulte Notas de lançamento do Amazon RDS para Oracle.

Quando o RDS programa o upgrade, a seguinte notificação aparece na guia Manutenção e backups da página de detalhes do banco de dados:

```
Automatic minor version upgrade to engine-version
```

A mensagem anterior indica que o RDS programou o upgrade do mecanismo de banco de dados para a próxima janela de manutenção.

Às vezes, uma nova versão secundária fica disponível antes que o RDS aplique uma versão secundária anterior. Por exemplo, a instância está executando a *minor-version-1* quando tanto a *minor-version-2* quanto a *minor-version-3* estão disponíveis como destinos de upgrade. Nessa situação, para evitar tempo de inatividade desnecessário para as instâncias de banco de dados, o RDS programa o upgrade automático da versão secundária para a versão mais recente, ignorando o upgrade para a versão anterior. Neste exemplo, o RDS faz upgrade da instância da *minor-version-1* diretamente para a *minor-version-3*.

Para garantir certa frequência de upgrades de versões secundárias, é possível atualizar as instâncias manualmente em vez de usar o mecanismo de upgrade automático. Para programar um upgrade para a próxima janela de manutenção, especifique `--no-apply-immediately` quando você fizer upgrade para uma versão secundária usando `modify-db-instance`. Para fazer upgrade imediatamente, especifique `--apply-immediately`. Para ter mais informações, consulte Atualizar manualmente a versão do mecanismo.

Gerenciar uma atualização automática de versão secundária no RDS para Oracle

Quando uma nova versão secundária se torna disponível, você pode fazer upgrade de sua instância de banco de dados para essa versão secundária O exemplo a seguir faz upgrade da instância de banco de dados chamada `orclinst1` imediatamente:

```
aws rds apply-pending-maintenance-action \
    --resource-identifier arn:aws:rds:us-east-1:123456789012:db:orclinst1 \
    --apply-action db-upgrade \
    --opt-in-type immediate
```

Para desistir de um upgrade automático de versão secundária que ainda não tenha sido programado, defina `--opt-in-type` como `undo-opt-in`, como no exemplo a seguir:

```
aws rds apply-pending-maintenance-action \
    --resource-identifier arn:aws:rds:us-east-1:123456789012:db:orclinst1 \
    --apply-action db-upgrade \
    --opt-in-type undo-opt-in
```

Se o RDS já tiver programado um upgrade para sua instância de banco de dados, você não poderá usar `apply-pending-maintenance-action` para cancelá-lo. No entanto, você poderá modificar sua instância de banco de dados e desativar o recurso de upgrade automático de versão secundária, e isso vai cancelar o upgrade.

Para saber como desativar upgrades automáticos de versão secundária, consulte Atualizar automaticamente a versão do mecanismo espelho. Para ter mais informações sobre apply-pending-maintenance-action, consulte a Referência de comandos da AWS CLI.

Considerações sobre atualizações de bancos de dados Oracle

Antes de atualizar a instância Oracle, analise as informações a seguir.

Tópicos

- Considerações sobre o Oracle Multitenant
- Considerações de grupos de opções
- Considerações de grupos de parâmetros
- Considerações sobre fuso horário
- Considerações sobre pacote de patches espaciais (SPB)

Considerações sobre o Oracle Multitenant

A tabela a seguir descreve as arquiteturas do Oracle Database compatíveis em versões diferentes.

Versão do Oracle Database	Status de suporte do RDS	Arquitetura
Oracle Database 21c	Compatível	Somente CDB
Oracle Database 19c	Compatível	CDB ou não CDB

A tabela a seguir descreve os caminhos de atualização compatíveis e incompatíveis.

Caminho de atualização	Compatível?
CDB para CDB	Sim
Não CDB para CDB	Não, mas é possível converter uma arquitetura não CDB em CDB e depois atualizá-la
CDB para não CDB	Não

Para obter mais informações sobre o Oracle Multitenant no RDS for Oracle, consulte Configuração de locatário único da arquitetura CDB.

Considerações de grupos de opções

Se sua instância de banco de dados usa um grupo de opções personalizado, às vezes o Amazon RDS não pode atribuir automaticamente um novo grupo de opções. Por exemplo, isso ocorre quando você faz a atualização para uma nova versão principal. Nesses casos, especifique um novo grupo de opções ao atualizar. Recomendamos que você crie um novo grupo de opções e adicione a ele as mesmas opções que o seu grupo de opções personalizadas existente.

Para obter mais informações, consulte Criar um grupo de opções ou Copiar um grupo de opções.

Se a instância de banco de dados usar um grupo de opções personalizado que contenha a opção as opções `APEX` e `APEX-DEV`, você pode reduzir o tempo de atualização algumas vezes. Para fazer isso, atualize sua versão do Oracle APEX ao mesmo tempo que sua instância de banco de dados. Para obter mais informações, consulte Atualizar a versão do Oracle APEX.

Considerações de grupos de parâmetros

Se a sua instância de banco de dados usar um grupo de parâmetros personalizado, em alguns casos, o Amazon RDS não poderá atribuir automaticamente a ela um novo grupo de parâmetros. Por exemplo, isso ocorre quando você faz a atualização para uma nova versão principal. Nesses casos, certifique-se de especificar um novo grupo de parâmetros ao atualizar. Recomendamos que você crie um novo parameter group e configure os parâmetros como no seu parameter group personalizado existente.

Para obter mais informações, consulte Criar um grupo de parâmetros de banco de dados no Amazon RDS ou Copiar um grupo de parâmetros de banco de dados no Amazon RDS.

Considerações sobre fuso horário

É possível usar a opção de fuso horário para alterar o fuso horário do sistema usado pela sua instância de banco de dados Oracle. Por exemplo, é possível alterar o fuso horário de uma instância de banco de dados para que ele seja compatível com um ambiente local ou um aplicativo herdado. A opção de fuso horário altera o fuso horário no nível do host. O Amazon RDS for Oracle atualiza automaticamente o fuso horário do sistema ao longo do ano. Para obter mais informações sobre o fuso horário do sistema, consulte Fuso horário da Oracle.

Quando uma instância de banco de dados Oracle é criada, o banco de dados define automaticamente o fuso horário do banco de dados. O fuso horário do banco de dados também é conhecido como fuso DST (horário de verão). O fuso horário do banco de dados é diferente do fuso horário do sistema.

Entre versões do Oracle Database, conjuntos de patches ou patches individuais podem incluir novas versões do DST. Esses patches refletem as alterações nas regras de transição para várias regiões de fuso horário. Por exemplo, um governo pode mudar a data de início do DST. Alterações às regras de DST podem afetar dados existentes do tipo de dados `TIMESTAMP WITH TIME ZONE`.

Se você atualizar uma instância do RDS for Oracle, o Amazon RDS não atualizará o fuso horário do banco de dados automaticamente. Para atualizar o arquivo de fuso horário automaticamente, você pode incluir a opção `TIMEZONE_FILE_AUTOUPGRADE` no grupo de opções associado à sua instância de banco de dados durante ou após a atualização da versão do mecanismo. Para obter mais informações, consulte Atualização automática do arquivo de fuso horário do Oracle.

Como alternativa, para atualizar o fuso horário do arquivo de banco de dados manualmente, crie uma nova instância de banco de dados Oracle que tenha o patch de DST desejado. No entanto, recomendamos atualizar o arquivo de fuso horário do banco de dados usando a opção `TIMEZONE_FILE_AUTOUPGRADE`.

Depois de atualizar o arquivo de fuso horário, migre os dados da instância atual para a nova instância. É possível migrar dados usando várias técnicas, incluindo as seguintes:

- AWS Database Migration Service
- Oracle GoldenGate
- Oracle Data Pump
- Exportação/importação original (suporte desativado para uso geral)

> **Note**
>
> Quando dados são migrados usando o Oracle Data Pump, o utilitário gera o erro ORA-39405 quando a versão de fuso horário de destino é inferior à versão do fuso horário de origem.

Para obter mais informações, consulte Restrições de TIMESTAMP WITH TIMEZONE na documentação da Oracle.

Considerações sobre pacote de patches espaciais (SPB)

No RDS para Oracle, uma atualização de versão (RU) é uma versão secundária do mecanismo que inclui correções de segurança, correções de erros e novos recursos para o Oracle Database. Um pacote de patches espaciais (SPB) é uma versão secundária do mecanismo que inclui patches projetados para a opção Oracle Spatial. Por exemplo, 19.0.0.0.ru-2025-01.spb-1.r1 é uma versão secundária do mecanismo que contém os patches de RU na versão do mecanismo 19.0.0.0.ru-2025-01.rur-2025-01.r1 e patches do Spatial.

Quando você atualizar seu banco de dados par SPBs, considere o seguinte:

- Os SPBs são compatíveis somente com o Oracle Database 19c.
- Normalmente, um SPB é lançado duas a três semanas após a RU trimestral correspondente.
- Você pode atualizar a instância de banco de dados para um SPB mesmo que ela não use a opção Oracle Spatial, mas os patches espaciais na versão do mecanismo se aplicam somente ao Oracle Spatial. Você pode criar uma instância em um SPB e instalar a opção Oracle Spatial posteriormente.
- Se você habilitar a atualização automática de versões secundárias para sua instância de banco de dados, o caminho da atualização dependerá de a instância estar usando um SPB ou uma RU naquele momento. Se a instância de banco de dados estiver usando um SPB, o RDS a atualizará automaticamente para o SPB mais recente. Se a instância de banco de dados estiver usando uma RU, o RDS a atualizará automaticamente para a RU mais recente.
- Você pode atualizar manualmente a instância de banco de dados de uma RU para um SPB somente se a versão do SPB do mecanismo for igual ou posterior à RU atual.
- Você pode atualizar manualmente sua instância de banco de dados de um SPB para uma RU somente se a RU for uma versão posterior.

Testar uma atualização do Oracle DB

Antes de fazer upgrade da instância de banco de dados para uma versão principal, teste exaustivamente o banco de dados e todas as aplicações que o acessam para atestar a compatibilidade com a nova versão. Recomendamos que você use o procedimento a seguir.

Para testar um upgrade de versão principal

1. Reveja a documentação de upgrade da Oracle para a nova versão do mecanismo de banco de dados para ver se existem problemas de compatibilidade que podem afetar seu banco de dados ou seus aplicativos. Para obter mais informações, consulte o Database Upgrade Guide, na documentação da Oracle.
2. Se a sua instância de banco de dados usar um grupo de opções personalizado, crie um novo grupo de opções compatível com a nova versão para a qual você está atualizando. Para obter mais informações, consulte Considerações de grupos de opções.
3. Se a sua instância de banco de dados usar um parameter group personalizado, crie um novo parameter group compatível com a nova versão para a qual você está atualizando. Para obter mais informações, consulte Considerações de grupos de parâmetros.
4. Crie um snapshot de banco de dados da instância de banco de dados a ser atualizada. Para obter mais informações, consulte Criar um snapshot de banco de dados para uma instância de banco de dados single-AZ para o Amazon RDS.
5. Restaure o snapshot de banco de dados para criar uma nova instância de banco de dados de teste. Para obter mais informações, consulte Restaurar uma instância de banco de dados.
6. Modifique essa nova instância de banco de dados de teste para atualizá-la para a nova versão, usando um dos seguintes métodos:

   - Console
   - AWS CLI
   - API do RDS
7. Realize testes:

   - Execute quantos testes de garantia de qualidade na instância de banco de dados atualizada forem necessários para garantir que o seu banco de dados e o seu aplicativo funcionem corretamente com a nova versão.
   - Implemente os novos testes necessários para avaliar o impacto de problemas de compatibilidade que você identificou na etapa 1.

- Teste todos os procedimentos, funções e triggers armazenados.
- Direcione versões de teste dos seus aplicativos à instância de banco de dados atualizada. Verifique se os aplicativos funcionam corretamente com a nova versão.
- Avalie o armazenamento usado pela instância atualizada para determinar se o upgrade requer armazenamento adicional. Talvez você precise escolher uma classe de instância maior para dar suporte à nova versão em produção. Para obter mais informações, consulte Classes de instâncias de banco de dados .

8. Se todos os testes forem aprovados, atualize a instância de banco de dados de produção. Recomendamos que você confirme se a instância de banco de dados está funcionando corretamente antes de permitir operações de gravação na instância de banco de dados.

Atualizar a versão de uma instância de banco de dados do RDS para Oracle

Para atualizar manualmente a versão do mecanismo de banco de dados de uma instância de banco de dados do RDS para Oracle, use o AWS Management Console, a AWS CLI ou a API do RDS. Para ter informações gerais sobre atualizações de bancos de dados, consulte Atualizar a versão de uma instância de banco de dados do RDS para Oracle. Para ter destinos de atualização válidos, use o comando describe-db-engine-versions da AWS CLI.

Console

Como atualizar a versão do mecanismo de uma instância de banco de dados do RDS para Oracle usando o console

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Databases (Bancos de dados) e a instância de banco de dados que você deseja atualizar.
3. Selecione Modify.
4. Em Versão do mecanismo de banco de dados, escolha uma versão posterior do banco de dados.
5. Escolha Continue (Continuar) e verifique o resumo de modificações. Entenda as implicações de uma atualização de versão de banco de dados. Não é possível converter uma instância de

banco de dados atualizada na versão anterior. Antes de continuar, não se esqueça de testar o banco de dados e a aplicação com a nova versão.

6. Decida quando agendar a atualização da instância de banco de dados. Para aplicar as alterações imediatamente, escolha Apply immediately. Escolher essa opção pode causar uma interrupção em alguns casos. Para ter mais informações, consulte Usar a configuração de programação de modificações.
7. Na página de confirmação, revise suas alterações. Se estiverem corretas, escolha Modify DB Instance (Modificar instância de banco de dados) para salvar suas alterações.

   Como alternativa, escolha Back (Voltar) para editar suas alterações ou escolha Cancel (Cancelar) para cancelar as alterações.

AWS CLI

Para atualizar a versão do mecanismo de uma instância de banco de dados do RDS para Oracle, use o comando modify-db-instance da CLI. Especifique os seguintes parâmetros:

- `--db-instance-identifier`: o nome da instância de banco de dados do RDS para Oracle.
- `--engine-version` – o número da versão do mecanismo de banco de dados para a qual será feita a atualização.

  Para obter informações sobre versões de mecanismo válidas, use o comando AWS CLI describe-db-engine-versions.
- `--allow-major-version-upgrade`: para atualizar a versão do mecanismo de banco de dados.
- `--no-apply-immediately` – para aplicar alterações durante a próxima janela de manutenção. Para aplicar as alterações imediatamente, use `--apply-immediately`.

Example

O exemplo a seguir atualiza uma instância do CDB denominada `myorainst` da versão atual de `19.0.0.0.ru-2024-01.rur-2024-01.r1` para a versão `21.0.0.0.ru-2024-04.rur-2024-04.r1`.

Para Linux, macOS ou Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier myorainst \
```

```
    --engine-version 21.0.0.0.ru-2024-04.rur-2024-04.r1 \
    --allow-major-version-upgrade \
    --no-apply-immediately
```

Para Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier myorainst ^
    --engine-version 21.0.0.0.ru-2024-04.rur-2024-04.r1 ^
    --allow-major-version-upgrade ^
    --no-apply-immediately
```

API do RDS

Para atualizar uma instância de banco de dados do RDS para Oracle, use a ação ModifyDBInstance. Especifique os seguintes parâmetros:

- `DBInstanceIdentifier` – o nome da instância de banco de dados. Por exemplo *`myorainst`*.
- `EngineVersion` – o número da versão do mecanismo de banco de dados para a qual será feita a atualização. Para obter informações sobre versões de mecanismo válidas, use a operação DescribeDBEngineVersions.
- `AllowMajorVersionUpgrade` – para permitir ou não uma atualização de versão principal. Para isso, defina o valor como `true`.
- `ApplyImmediately` – se desejar aplicar as alterações imediatamente ou durante a próxima janela de manutenção. Para aplicar as alterações imediatamente, defina o valor como `true`. Para aplicar alterações durante a próxima janela de manutenção, defina o valor como `false`.

Como atualizar para um snapshot de banco de dados Oracle

A atualização dos snapshots de banco de dados do Oracle no Amazon RDS garante que o banco de dados permaneça seguro e totalmente suportado. À medida que as versões mais antigas do Oracle chegarem ao fim do suporte de patches, você poderá atualizar qualquer snapshot de banco de dados manual vinculado a essas versões para evitar possíveis vulnerabilidades ou limitações de serviço. Para obter mais informações, consulte Oracle Engine Version Management.

O Amazon RDS comporta a atualização de snapshots em todas as Regiões da AWS.

Console

Para atualizar para um snapshot de banco de dados Oracle

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Snapshots e selecione o snapshot de banco de dados que você deseja atualizar.
3. Em Actions (Ações), selecione Upgrade snapshot (Atualizar snapshot). A página Upgrade snapshot (Atualizar snapshot) é exibida.
4. Escolha a New engine version (Nova versão do mecanismo) para a qual o snapshot será atualizado.
5. (Opcional) Em Option group (Grupo de opções), escolha o grupo de opções para o snapshot de banco de dados atualizado. As mesmas considerações de grupo de opções se aplicam ao atualizar um snapshot de banco de dados e uma instância de banco de dados. Para obter mais informações, consulte Considerações de grupos de opções.
6. Escolha Save changes (Salvar alterações) para salvar suas alterações.

   Durante o processo de atualização, todas as ações do snapshot serão desabilitadas para esse snapshot de banco de dados. Além disso, o status do snapshot de banco de dados muda de available (disponível) para upgrading (atualizando) e muda para active (ativo) após a conclusão. Se não for possível atualizar o snapshot de banco de dados devido a problemas de corrupção do snapshot, o status mudará para unavailable (indisponível). Não é possível recuperar o snapshot desse estado.

   > **Note**
   >
   > Se a atualização do snapshot de banco de dados falhar, o snapshot será revertido para o estado original com a versão original.

AWS CLI

Para atualizar um snapshot de banco de dados Oracle usando a AWS CLI, chame o comando modify-db-snapshot com os seguintes parâmetros:

- `--db-snapshot-identifier` – o nome do snapshot de banco de dados.

- `--engine-version` – a versão de atualização do snapshot.

Talvez também seja necessário incluir o parâmetro a seguir. As mesmas considerações de grupo de opções se aplicam ao atualizar um snapshot de banco de dados e uma instância de banco de dados. Para obter mais informações, consulte Considerações de grupos de opções.

- `--option-group-name` – o grupo de opções para o snapshot de banco de dados atualizado.

Example

O exemplo a seguir atualiza um snapshot de banco de dados.

Para Linux, macOS ou Unix:

```
aws rds modify-db-snapshot \
    --db-snapshot-identifier mydbsnapshot \
    --engine-version 19.0.0.0.ru-2020-10.rur-2020-10.r1 \
    --option-group-name default:oracle-se2-19
```

Para Windows:

```
aws rds modify-db-snapshot ^
    --db-snapshot-identifier mydbsnapshot ^
    --engine-version 19.0.0.0.ru-2020-10.rur-2020-10.r1 ^
    --option-group-name default:oracle-se2-19
```

API do RDS

Para atualizar um snapshot de banco de dados Oracle usando a API do Amazon RDS, chame a operação ModifyDBSnapshot com os seguintes parâmetros:

- `DBSnapshotIdentifier` – o nome do snapshot de banco de dados.
- `EngineVersion` – a versão de atualização do snapshot.

Você também pode precisar incluir o parâmetro `OptionGroupName`. As mesmas considerações de grupo de opções se aplicam ao atualizar um snapshot de banco de dados e uma instância de banco de dados. Para obter mais informações, consulte Considerações de grupos de opções.

Usar software de terceiros com a instância de banco de dados do Oracle

É possível hospedar uma instância de banco de dados do RDS para Oracle que seja compatível com ferramentas e software de terceiros.

Tópicos

- Usar o Oracle GoldenGate com o Amazon RDS para Oracle
- Usar o Oracle Repository Creation Utility no RDS for Oracle
- Configurando o Oracle Connection Manager em uma instância do Amazon EC2
- Instalar um Siebel Database no Oracle no Amazon RDS

Usar o Oracle GoldenGate com o Amazon RDS para Oracle

O Oracle GoldenGate coleta, replica e gerencia dados transacionais entre bancos de dados. Ele é um processo de captura de dados de alterações (CDC) baseado em log e um pacote de software de replicação usado com bancos de dados para sistemas de processamento de transações online (OLTP). O Oracle GoldenGate cria arquivos de trilha que contêm os dados alterados mais recentes do banco de dados de origem. Depois, ele envia esses arquivos ao servidor, onde um processo converte o arquivo de trilha em SQL padrão para ser aplicado ao banco de dados de destino.

O Oracle GoldenGate com RDS para Oracle é compatível com seguintes recursos:

- Replicação de banco de dados ativo-ativo
- Recuperação de desastres
- Proteção de dados
- Replicação em regiões e entre regiões
- Migração e upgrades com tempo de inatividade zero
- Replicação de dados entre uma instância de banco de dados do RDS para Oracle e um banco de dados que não seja Oracle

> **Note**
>
> Para obter uma lista de bancos de dados compatíveis, consulte Oracle Fusion Middleware Supported System Configurations (Configurações do sistema compatíveis com o Oracle Fusion Middleware) na documentação da Oracle.

Você pode usar o Oracle GoldenGate com RDS para Oracle para atualizar para as versões principais do banco de dados Oracle. Por exemplo, é possível usar o Oracle GoldenGate para atualizar de um banco de dados on-premises do Oracle Database 11g para o Oracle Database 19c em uma instância de banco de dados do Amazon RDS.

Tópicos

- Versões e opções de licenciamento com suporte para o Oracle GoldenGate
- Requisitos e limitações do Oracle GoldenGate
- Arquitetura do Oracle GoldenGate
- Configurar o Oracle GoldenGate

- Trabalhar com os utilitários EXTRACT e REPLICAT do Oracle GoldenGate
- Monitorar o Oracle GoldenGate
- Solução de problemas do Oracle GoldenGate

Versões e opções de licenciamento com suporte para o Oracle GoldenGate

É possível usar a Standard Edition 2 (SE2) ou a Enterprise Edition (EE) do RDS para Oracle com Oracle GoldenGate versão 12c e posterior. É possível usar os seguintes recursos do Oracle GoldenGate:

- O Oracle GoldenGate Remote Capture (extração) é compatível.
- A captura (extração) é compatível com instâncias de banco de dados do RDS para Oracle que usam a arquitetura tradicional de banco de dados não CDB. A captura remota de PDB do Oracle GoldenGate é compatível com CDBs que executam o Oracle Database 21c ou o Oracle Database 19c versão 19.0.0.0.ru-2024-04.rur-2024-04.r1 ou posterior.
- O Oracle GoldenGate Remote Delivery (replicat) é compatível com o RDS para instâncias de banco de dados Oracle que usam arquiteturas não CDB ou CDB. A entrega remota é compatível com a replicação integrada, replicação paralela, replicação coordenada e replicação clássica.
- O RDS da Oracle é compatível com as arquiteturas Clássica e de Microsserviço do Oracle GoldenGate.
- A replicação de valores do Oracle GoldenGate DDL e Sequence é compatível com o uso do modo de captura integrado.

Você é responsável por gerenciar o licenciamento (BYOL) do Oracle GoldenGate para usá-lo com o Amazon RDS em todas as Regiões da AWS. Para ter mais informações, consulte Opções de licenciamento do RDS para Oracle.

Requisitos e limitações do Oracle GoldenGate

Ao trabalhar do Oracle GoldenGate e RDS for Oracle, considere os seguintes requisitos e limitações:

- Você é responsável por configurar e gerenciar o Oracle GoldenGate para usar com o RDS para Oracle.
- Você é responsável por configurar uma versão do Oracle GoldenGate certificada com os bancos de dados de origem e de destino. Para obter mais informações, consulte Configurações do sistema compatíveis com o Oracle Fusion Middleware na documentação da Oracle.

- Você pode usar o Oracle GoldenGate em vários ambientes da AWSdiferentes para diversos casos de uso. Se você tiver um problema de suporte relacionado ao Oracle GoldenGate, entre em contato com os Serviços de Suporte da Oracle.
- É possível usar o Oracle GoldenGate em instâncias de banco de dados do RDS para Oracle que usam a Oracle Transparent Data Encryption (TDE). Para manter a integridade dos dados replicados, configure a criptografia no hub do Oracle GoldenGate usando volumes criptografados do Amazon EBS ou criptografia de arquivos de trilha. Configure também a criptografia para dados enviados entre o hub do Oracle GoldenGate e as instâncias de banco de dados de fonte e destino. As instâncias de banco de dados do RDS for Oracle são compatíveis com a criptografia com Oracle Secure Sockets Layer ou Oracle Native Network Encryption.

Arquitetura do Oracle GoldenGate

A arquitetura do GoldenGate para uso com o Amazon RDS consiste nos seguintes módulos desacoplados:

Fonte do banco de dados

A fonte do banco de dados pode ser um banco de dados Oracle no local, um banco de dados Oracle em uma instância do Amazon EC2 ou um banco de dados Oracle em uma instância de banco de dados do Amazon RDS.

Hub do Oracle GoldenGate

Um hub do Oracle GoldenGate move as informações de transação da fonte do banco de dados para o banco de dados de destino. O hub pode ser um dos seguintes:

- Uma instância do Amazon EC2 com o Oracle Database e o Oracle GoldenGate instalados
- Uma instalação da Oracle on-premises.

É possível ter mais de um hub do Amazon EC2. Recomendamos que você use dois hubs se utilizar o Oracle GoldenGate para replicação entre regiões.

Bancos de dados de destino:

O banco de dados de destino pode estar em uma instância de banco de dados do Amazon RDS, em uma instância do Amazon EC2 ou no local.

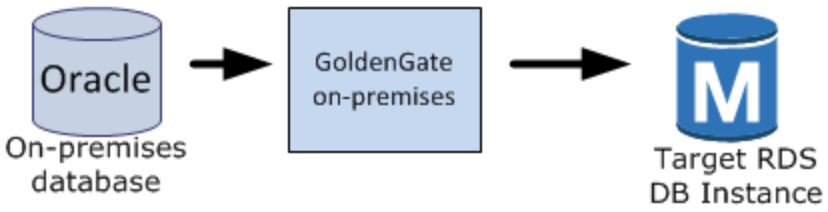
As seções a seguir descrevem cenários comuns para o Oracle GoldenGate no Amazon RDS.

Tópicos

- Fonte do banco de dados on-premises e hub do Oracle GoldenGate
- Fonte do banco de dados on-premises e hub do Amazon EC2
- Fonte do banco de dados do Amazon RDS e hub do Amazon EC2
- Fonte do banco de dados do Amazon EC2 e hub do Amazon EC2
- Hubs do Amazon EC2 em diferentes regiões da AWS

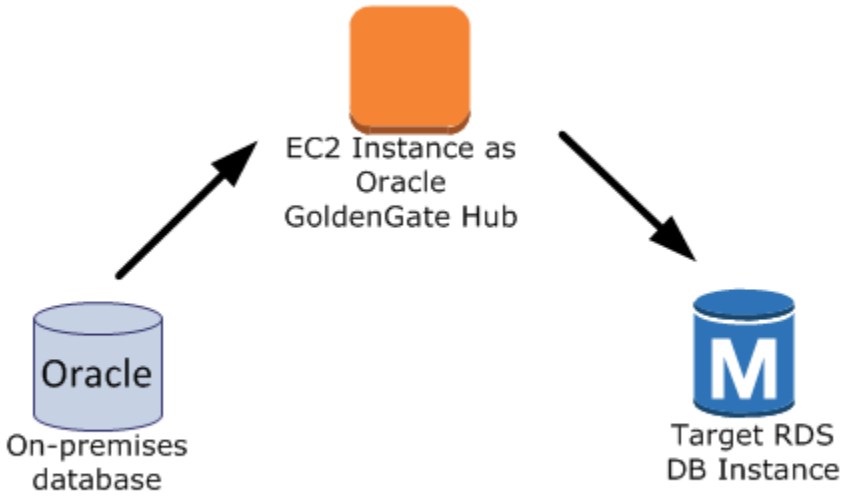
Fonte do banco de dados on-premises e hub do Oracle GoldenGate

Nesse cenário, uma fonte do banco de dados Oracle on-premises e um hub do Oracle GoldenGate on-premises fornecem dados para uma instância de banco de dados do Amazon RDS de destino.



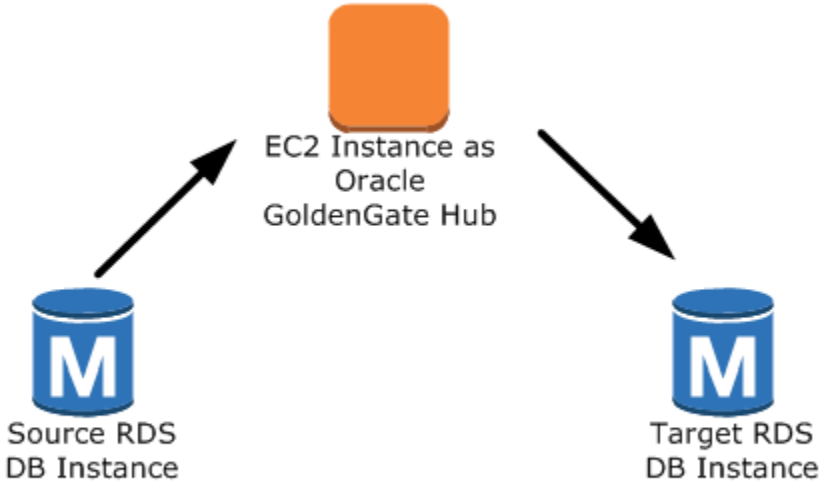
Fonte do banco de dados on-premises e hub do Amazon EC2

Nesse cenário, um banco de dados Oracle on-premises atua como a fonte do banco de dados. É conectado a um hub de instância do Amazon EC2. Este hub fornece dados para uma instância de banco de dados de destino do RDS for Oracle.



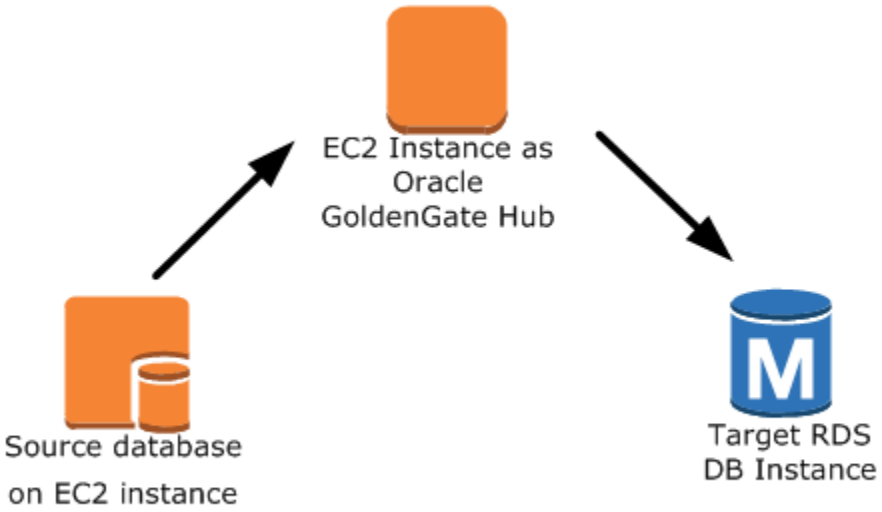
Fonte do banco de dados do Amazon RDS e hub do Amazon EC2

Nesse cenário, uma instância de banco de dados do RDS for Oracle atua como fonte do banco de dados. É conectado a um hub de instância do Amazon EC2. Este hub fornece dados para uma instância de banco de dados de destino do RDS for Oracle.



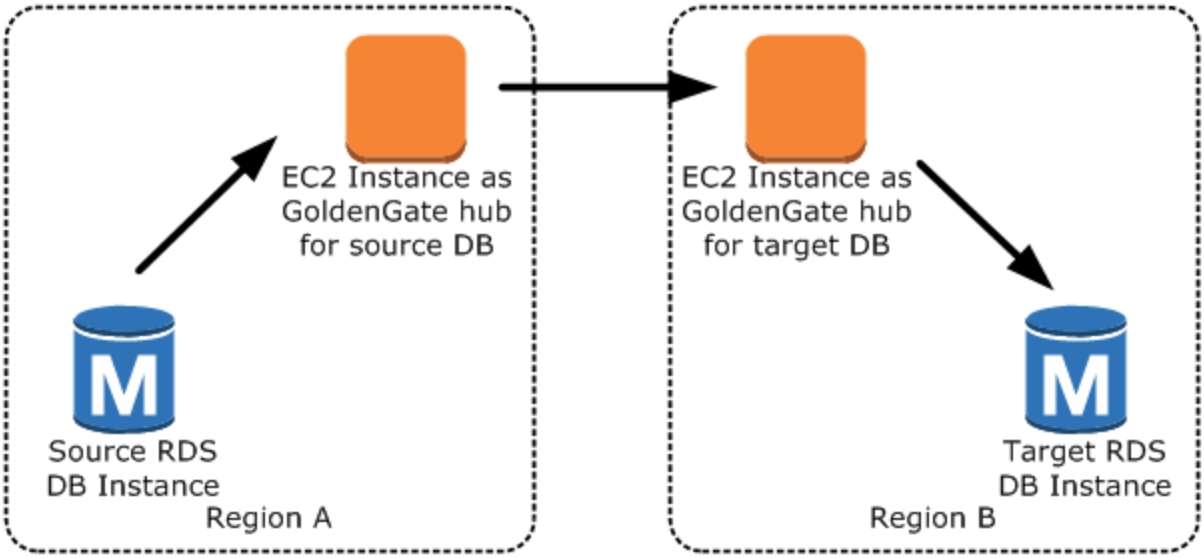
Fonte do banco de dados do Amazon EC2 e hub do Amazon EC2

Nesse cenário, um banco de dados Oracle em uma instância do Amazon EC2 atua como a fonte do banco de dados. É conectado a um hub de instância do Amazon EC2. Este hub fornece dados para uma instância de banco de dados de destino do RDS for Oracle.



Hubs do Amazon EC2 em diferentes regiões da AWS

Nesse cenário, um banco de dados Oracle em uma instância de banco de dados do Amazon RDS está conectado a um hub de instância do Amazon EC2 na mesma região da AWS. O hub está conectado a um hub da instância do Amazon EC2 em outra região da AWS. Esse segundo hub fornece dados para a instância de banco de dados de destino do RDS for Oracle na mesma região da AWS como o segundo hub da instância do Amazon EC2.



> **Note**
>
> Quaisquer problemas que afetem o funcionamento do Oracle GoldenGate em ambiente on-premises também afetarão a execução do Oracle GoldenGate na AWS. É altamente recomendável que você monitore o hub do Oracle GoldenGate para garantir que `EXTRACT` e `REPLICAT` sejam retomados se ocorrer um failover. Como o hub do Oracle GoldenGate é executado em uma instância do Amazon EC2, o Amazon RDS não gerencia o hub do Oracle GoldenGate e não pode garantir que ele esteja em execução.

Configurar o Oracle GoldenGate

Para configurar o Oracle GoldenGate usando o Amazon RDS, configure o hub em uma instância do Amazon EC2 e, depois, configure os bancos de dados de origem e de destino. As seções a seguir fornecem um exemplo de como configurar o Oracle GoldenGate para uso com o Amazon RDS para Oracle.

Tópicos

- Configurar um hub do Oracle GoldenGate no Amazon EC2
- Configurar um banco de dados de origem para uso com o Oracle GoldenGate no Amazon RDS
- Configurar um banco de dados de destino para uso com o Oracle GoldenGate no Amazon RDS

Configurar um hub do Oracle GoldenGate no Amazon EC2

Para criar um hub do Oracle GoldenGate em uma instância do Amazon EC2, crie primeiro uma instância do Amazon EC2 com uma instalação completa do cliente do Oracle RDBMS. A instância do Amazon EC2 também deve ter o software do Oracle GoldenGate instalado. As versões do software Oracle GoldenGate dependem das versões dos bancos de dados de origem e de destino. Para obter mais informações sobre a instalação do Oracle GoldenGate, consulte a documentação do Oracle GoldenGate.

A instância do Amazon EC2 que serve como o hub do GoldenGate armazena e processa as informações de transação da fonte do banco de dados em arquivos de trilha. Para que esse processo seja aceito, atenda às seguintes condições:

- Você alocou armazenamento suficiente para os arquivos de trilha.

- A instância do Amazon EC2 tem poder de processamento suficiente para gerenciar a quantidade de dados.
- Além disso, certifique-se de que a instância do EC2 tenha memória suficiente para armazenar as informações de transação antes delas serem gravadas no arquivo de trilha.

Como definir um hub de configuração de arquitetura clássica do Oracle GoldenGate em uma instância do Amazon EC2

1. Crie os subdiretórios no diretório do Oracle GoldenGate.

   No shell da linha de comando do Amazon EC2, inicie o `ggsci`, o interpretador de comandos do Oracle GoldenGate. O comando `CREATE SUBDIRS` cria subdiretórios no diretório `/gg` para arquivos de parâmetros, relatórios e ponto de verificação.

   ```
   prompt$ cd /gg
   prompt$ ./ggsci

   GGSCI> CREATE SUBDIRS
   ```

2. Configure o arquivo `mgr.prm`.

   O exemplo a seguir adiciona linhas ao arquivo `$GGHOME/dirprm/mgr.prm`.

   ```
   PORT 8199
   PurgeOldExtracts ./dirdat/*, UseCheckpoints, MINKEEPDAYS 5
   ```

3. Inicie o gerente.

   O exemplo a seguir inicia o `ggsci` e executa o comando `start mgr`.

   ```
   GGSCI> start mgr
   ```

O hub do Oracle GoldenGate agora está pronto para uso.

Configurar um banco de dados de origem para uso com o Oracle GoldenGate no Amazon RDS

Conclua as tarefas a seguir para configurar um banco de dados de origem a ser usado com o Oracle GoldenGate.

Etapas de configuração

- Etapa 1: Ativar o registro em log complementar no banco de dados de origem
- Etapa 2: Definir o parâmetro de inicialização ENABLE_GOLDENGATE_REPLICATION como true
- Etapa 3: Definir o período de retenção de log no banco de dados de origem
- Etapa 4: Criar uma conta do usuário do Oracle GoldenGate no banco de dados de origem
- Etapa 5: Conceder privilégios ao usuário no banco de dados de origem
- Etapa 6: Adicionar um alias TNS ao banco de dados de origem

Etapa 1: Ativar o registro em log complementar no banco de dados de origem

Para ativar o registro complementar mínimo em nível de banco de dados, execute o seguinte procedimento PL/SQL:

```
EXEC rdsadmin.rdsadmin_util.alter_supplemental_logging(p_action => 'ADD')
```

Etapa 2: Definir o parâmetro de inicialização ENABLE_GOLDENGATE_REPLICATION como true

Quando você define o parâmetro de inicialização `ENABLE_GOLDENGATE_REPLICATION` como `true`, ele permite que os serviços de banco de dados sejam compatíveis com a replicação lógica. Se o banco de dados de origem estiver em uma instância de banco de dados do Amazon RDS, tenha um grupo de parâmetros atribuído à instância de banco de dados com o parâmetro de inicialização `ENABLE_GOLDENGATE_REPLICATION` definido como `true`. Para obter mais informações sobre o parâmetro de inicialização `ENABLE_GOLDENGATE_REPLICATION`, consulte a documentação do banco de dados Oracle.

Etapa 3: Definir o período de retenção de log no banco de dados de origem

Configure o banco de dados de origem para reter os redo logs arquivados. Considere as seguintes diretrizes:

- Especifique a duração da retenção de log em horas. O valor mínimo é uma hora.
- Defina a duração para exceder qualquer tempo de inatividade possível, qualquer período de comunicação possível e qualquer período de problemas de rede possível com a instância de origem. Essa duração permite que o Oracle GoldenGate recupere logs da instância de origem de acordo com a necessidade.
- Verifique se você tem armazenamento suficiente na instância para os arquivos.

Por exemplo, defina o período de retenção para logs redo arquivados como 24 horas.

```
EXEC rdsadmin.rdsadmin_util.set_configuration('archivelog retention hours',24)
```

Se você não tiver a retenção de logs ativada, ou se o valor de retenção for muito pequeno, receberá uma mensagem de erro semelhante à seguinte.

```
2022-03-06 06:17:27  ERROR   OGG-00446  error 2 (No such file or directory)
opening redo log /rdsdbdata/db/GGTEST3_A/onlinelog/o1_mf_2_9k4bp1n6_.log for sequence
 1306
Not able to establish initial position for begin time 2022-03-06 06:16:55.
```

Como sua instância de banco de dados retém seus redo logs arquivados, verifique se você tem espaço suficiente para os arquivos. Para ver quanto espaço você usou nas últimas *num_hours* horas, use a consulta a seguir, substituindo *num_hours* pelo número de horas.

```
SELECT SUM(BLOCKS * BLOCK_SIZE) BYTES FROM V$ARCHIVED_LOG
   WHERE NEXT_TIME>=SYSDATE-num_hours/24 AND DEST_ID=1;
```

Etapa 4: Criar uma conta do usuário do Oracle GoldenGate no banco de dados de origem

O Oracle GoldenGate é executado como usuário de banco de dados e requer os privilégios apropriados para acessar os redo logs e redo logs arquivados do banco de dados de origem. Para concedê-los, crie uma conta do usuário no banco de dados de origem. Para obter mais informações sobre as permissões para uma conta de usuário do GoldenGate, consulte a documentação do Oracle.

As instruções a seguir criam uma conta do usuário denominada `oggadm1`.

```
CREATE TABLESPACE administrator;
CREATE USER oggadm1  IDENTIFIED BY "password"
   DEFAULT TABLESPACE ADMINISTRATOR TEMPORARY TABLESPACE TEMP;
ALTER USER oggadm1 QUOTA UNLIMITED ON administrator;
```

> **Note**
>
> Especifique uma senha diferente do prompt mostrado aqui como prática recomendada de segurança.

Etapa 5: Conceder privilégios ao usuário no banco de dados de origem

Nessa tarefa, você concede os privilégios de conta necessários para usuários do banco de dados em seu banco de dados de origem.

Como conceder privilégios de conta no banco de dados de origem

1. Conceda os privilégios necessários à conta de usuário do Oracle GoldenGate usando o comando SQL `grant` e o procedimento `grant_sys_object` do `rdsadmin.rdsadmin_util`. As instruções a seguir concedem privilégios a um usuário denominado `oggadm1`.

```
GRANT CREATE SESSION, ALTER SESSION TO oggadm1;
GRANT RESOURCE TO oggadm1;
GRANT SELECT ANY DICTIONARY TO oggadm1;
GRANT FLASHBACK ANY TABLE TO oggadm1;
GRANT SELECT ANY TABLE TO oggadm1;
GRANT SELECT_CATALOG_ROLE TO rds_master_user_name WITH ADMIN OPTION;
EXEC rdsadmin.rdsadmin_util.grant_sys_object ('DBA_CLUSTERS', 'OGGADM1');
GRANT EXECUTE ON DBMS_FLASHBACK TO oggadm1;
GRANT SELECT ON SYS.V_$DATABASE TO oggadm1;
GRANT ALTER ANY TABLE TO oggadm1;
```

2. Conceda os privilégios necessários a uma conta do usuário para ser um administrador do Oracle GoldenGate. Execute o programa PL/SQL a seguir.

```
EXEC rdsadmin.rdsadmin_dbms_goldengate_auth.grant_admin_privilege (
    grantee                 => 'OGGADM1',
    privilege_type          => 'capture',
    grant_select_privileges => true,
    do_grants               => TRUE);
```

   Para revogar privilégios, use o procedimento `revoke_admin_privilege` no mesmo pacote.

Etapa 6: Adicionar um alias TNS ao banco de dados de origem

Adicione a entrada a seguir a `$ORACLE_HOME/network/admin/tnsnames.ora` no Oracle Home para ser usada pelo processo EXTRACT. Para obter mais informações sobre o arquivo `tnsnames.ora`, consulte a documentação da Oracle.

```
OGGSOURCE=
   (DESCRIPTION=
```

```
        (ENABLE=BROKEN)
        (ADDRESS_LIST=
            (ADDRESS=(PROTOCOL=TCP)(HOST=goldengate-source.abcdef12345.us-
west-2.rds.amazonaws.com)(PORT=8200)))
        (CONNECT_DATA=(SERVICE_NAME=ORCL))
    )
```

Configurar um banco de dados de destino para uso com o Oracle GoldenGate no Amazon RDS

Nessa tarefa, você configura uma instância de banco de dados de destino para uso com o Oracle GoldenGate.

Etapas de configuração

- Etapa 1: Definir o parâmetro de inicialização ENABLE_GOLDENGATE_REPLICATION como true
- Etapa 2: Criar uma conta do usuário do Oracle GoldenGate no banco de dados de destino
- Etapa 3: Conceder privilégios de conta no banco de dados de destino
- Etapa 4: Adicionar um alias TNS ao banco de dados de destino

Etapa 1: Definir o parâmetro de inicialização ENABLE_GOLDENGATE_REPLICATION como true

Quando você define o parâmetro de inicialização `ENABLE_GOLDENGATE_REPLICATION` como `true`, ele permite que os serviços de banco de dados sejam compatíveis com a replicação lógica. Se o banco de dados de origem estiver em uma instância de banco de dados do Amazon RDS, tenha um grupo de parâmetros atribuído à instância de banco de dados com o parâmetro de inicialização `ENABLE_GOLDENGATE_REPLICATION` definido como `true`. Para obter mais informações sobre o parâmetro de inicialização `ENABLE_GOLDENGATE_REPLICATION`, consulte a documentação do banco de dados Oracle.

Etapa 2: Criar uma conta do usuário do Oracle GoldenGate no banco de dados de destino

O Oracle GoldenGate é executado como um usuário do banco de dados e requer os privilégios de banco de dados apropriados. Para garantir que ele tenha esses privilégios, crie uma conta do usuário no banco de dados de destino.

A instrução a seguir cria um usuário denominado `oggadm1`.

```
CREATE TABLESPSACE administrator;
CREATE USER oggadm1  IDENTIFIED BY "password"
   DEFAULT TABLESPACE administrator
   TEMPORARY TABLESPACE temp;
```

```
ALTER USER oggadm1 QUOTA UNLIMITED ON administrator;
```

> **Note**
>
> Especifique uma senha diferente do prompt mostrado aqui como prática recomendada de segurança.

Etapa 3: Conceder privilégios de conta no banco de dados de destino

Nessa tarefa, você concede os privilégios de conta necessários para usuários do banco de dados em seu banco de dados de destino.

Como conceder privilégios de conta no banco de dados de destino

1. Conceda os privilégios necessários à conta do usuário do Oracle GoldenGate no banco de dados de destino. No exemplo a seguir, conceda privilégios a `oggadm1`.

```
GRANT CREATE SESSION        TO oggadm1;
GRANT ALTER SESSION         TO oggadm1;
GRANT CREATE CLUSTER        TO oggadm1;
GRANT CREATE INDEXTYPE      TO oggadm1;
GRANT CREATE OPERATOR       TO oggadm1;
GRANT CREATE PROCEDURE      TO oggadm1;
GRANT CREATE SEQUENCE       TO oggadm1;
GRANT CREATE TABLE          TO oggadm1;
GRANT CREATE TRIGGER        TO oggadm1;
GRANT CREATE TYPE           TO oggadm1;
GRANT SELECT ANY DICTIONARY TO oggadm1;
GRANT CREATE ANY TABLE      TO oggadm1;
GRANT ALTER ANY TABLE       TO oggadm1;
GRANT LOCK ANY TABLE        TO oggadm1;
GRANT SELECT ANY TABLE      TO oggadm1;
GRANT INSERT ANY TABLE      TO oggadm1;
GRANT UPDATE ANY TABLE      TO oggadm1;
GRANT DELETE ANY TABLE      TO oggadm1;
```

2. Conceda os privilégios necessários a uma conta do usuário para ser um administrador do Oracle GoldenGate. Execute o programa PL/SQL a seguir.

```
EXEC rdsadmin.rdsadmin_dbms_goldengate_auth.grant_admin_privilege (
    grantee                 => 'OGGADM1',
```

```
    privilege_type          => 'apply',
    grant_select_privileges => true,
    do_grants               => TRUE);
```

Para revogar privilégios, use o procedimento `revoke_admin_privilege` no mesmo pacote.

Etapa 4: Adicionar um alias TNS ao banco de dados de destino

Adicione a entrada a seguir a `$ORACLE_HOME/network/admin/tnsnames.ora` no Oracle Home para ser usada pelo processo `REPLICAT`. Para bancos de dados multilocação Oracle, o alias TNS deve apontar para o nome do serviço do PDB. Para obter mais informações sobre o arquivo `tnsnames.ora`, consulte a documentação da Oracle.

```
OGGTARGET=
    (DESCRIPTION=
        (ENABLE=BROKEN)
        (ADDRESS_LIST=
            (ADDRESS=(PROTOCOL=TCP)(HOST=goldengate-target.abcdef12345.us-
west-2.rds.amazonaws.com)(PORT=8200)))
        (CONNECT_DATA=(SERVICE_NAME=ORCL))
    )
```

Trabalhar com os utilitários EXTRACT e REPLICAT do Oracle GoldenGate

Os utilitários `EXTRACT` e `REPLICAT` do Oracle GoldenGate trabalham juntos para manter os bancos de dados de origem e destino sincronizados via replicação de transações incrementais usando arquivos de trilha. Todas as alterações que ocorrem no banco de dados de origem são automaticamente detectadas por `EXTRACT`, depois formatadas e transferidas para arquivos de trilha no Oracle GoldenGate on-premises ou no hub de instâncias do Amazon EC2. Após a conclusão da carga inicial, os dados são lidos desses arquivos e replicados para o banco de dados de destino pelo utilitário `REPLICAT`.

Executar o utilitário EXTRACT do Oracle GoldenGate

O utilitário `EXTRACT` recupera, converte e gera dados do banco de dados de origem para arquivos de trilha. O processo básico é o seguinte:

1. `EXTRACT` coloca os detalhes da transação em fila na memória ou em armazenamento de disco temporário.
2. O banco de dados de origem confirma a transação.

3. EXTRACT grava os detalhes da transação em um arquivo de trilha.
4. O arquivo de trilha roteia esses detalhes para o Oracle GoldenGate on-premises ou o hub da instância do Amazon EC2 e depois para o banco de dados de destino.

As etapas a seguir iniciam o utilitário EXTRACT, capturam os dados do `EXAMPLE.TABLE` no banco de dados de origem `OGGSOURCE` e criam os arquivos de trilha.

Como executar o utilitário EXTRACT

1. Configure o arquivo de parâmetro EXTRACT no hub do Oracle GoldenGate (on-premises ou na instância do Amazon EC2). A listagem a seguir mostra um exemplo do arquivo de parâmetro EXTRACT denominado `$GGHOME/dirprm/eabc.prm`.

```
EXTRACT EABC

USERID oggadm1@OGGSOURCE, PASSWORD "my-password"
EXTTRAIL /path/to/goldengate/dirdat/ab

IGNOREREPLICATES
GETAPPLOPS
TRANLOGOPTIONS EXCLUDEUSER OGGADM1

TABLE EXAMPLE.TABLE;
```

2. No hub do Oracle GoldenGate, faça login no banco de dados de origem e execute a interface de linha de comando do Oracle GoldenGate `ggsci`. O exemplo a seguir mostra o formato para login.

```
dblogin oggadm1@OGGSOURCE
```

3. Adicione dados de transação para ativar o registro em log complementar para a tabela de banco de dados.

```
add trandata EXAMPLE.TABLE
```

4. Usando a linha de comando `ggsci`, ative o utilitário EXTRACT usando os comandos a seguir.

```
add extract EABC tranlog, INTEGRATED tranlog, begin now
add exttrail /path/to/goldengate/dirdat/ab
   extract EABC,
```

```
   MEGABYTES 100
```

5. Registre o utilitário EXTRACT no banco de dados para que os logs de arquivamento não sejam excluídos. Essa tarefa permite que você recupere transações antigas, não confirmadas, se necessário. Para registrar o utilitário EXTRACT no banco de dados, use o comando a seguir.

```
register EXTRACT EABC, DATABASE
```

6. Inicie o utilitário EXTRACT com o comando a seguir.

```
start EABC
```

Executar o utilitário REPLICAT do Oracle GoldenGate

O utilitário REPLICAT “envia por push” informações de transação nos arquivos de trilha ao banco de dados de destino.

As etapas a seguir ativam e iniciam o utilitário REPLICAT para que ele possa replicar os dados capturados na tabela EXAMPLE.TABLE no banco de dados de destino OGGTARGET.

Como executar o utilitário REPLICATE

1. Configure o arquivo de parâmetro REPLICAT no hub do Oracle GoldenGate (on-premises ou na instância do EC2). A listagem a seguir mostra um exemplo do arquivo de parâmetro REPLICAT denominado $GGHOME/dirprm/rabc.prm.

```
REPLICAT RABC

USERID oggadm1@OGGTARGET, password "my-password"

ASSUMETARGETDEFS
MAP EXAMPLE.TABLE, TARGET EXAMPLE.TABLE;
```

> **Note**
>
> Especifique uma senha diferente do prompt mostrado aqui como prática recomendada de segurança.

2. Faça login no banco de dados de destino e execute a interface de linha de comando do Oracle GoldenGate (`ggsci`). O exemplo a seguir mostra o formato para login.

```
dblogin userid oggadm1@OGGTARGET
```

3. Usando a linha de comando `ggsci`, adicione uma tabela de verificação. O usuário indicado deve ser a conta de usuário do Oracle GoldenGate, e não o proprietário do esquema da tabela de destino. O exemplo a seguir cria um ponto de verificação denominado `gg_checkpoint`.

```
add checkpointtable oggadm1.oggchkpt
```

4. Para habilitar o utilitário `REPLICAT`, use o seguinte comando:

```
add replicat RABC EXTTRAIL /path/to/goldengate/dirdat/ab CHECKPOINTTABLE
 oggadm1.oggchkpt
```

5. Inicie o utilitário `REPLICAT` usando o comando a seguir.

```
start RABC
```

Monitorar o Oracle GoldenGate

Ao usar o Oracle GoldenGate para replicação, certifique-se de que o processo do Oracle GoldenGate esteja ativo e em execução e que os bancos de dados de origem e destino estejam sincronizados. Você pode usar as seguintes ferramentas de monitoramento:

- O Amazon CloudWatch é um serviço de monitoramento usado nesse padrão para monitorar os logs de erros do GoldenGate.
- O Amazon SNS é um serviço de notificação de mensagens usado nesse padrão para enviar notificações por e-mail.

Para obter instruções detalhadas, consulte Monitorar os logs do Oracle GoldenGate usando o Amazon CloudWatch.

Solução de problemas do Oracle GoldenGate

Esta seção explica os problemas mais comuns ao usar o Oracle GoldenGate com o Amazon RDS para Oracle.

Tópicos

- Erro ao abrir um redo log online
- O Oracle GoldenGate parece estar devidamente configurado, mas a replicação não está funcionando
- REPLICAT integrado lento devido a consulta no SYS. “_DBA_APPLY_CDR_INFO”

Erro ao abrir um redo log online

Configure os bancos de dados para reter os redo logs arquivados. Considere as seguintes diretrizes:

- Especifique a duração da retenção de log em horas. O valor mínimo é uma hora.
- Defina a duração para exceder qualquer tempo de inatividade possível, qualquer período de comunicação possível e qualquer período de problemas de rede possível com a instância de banco de dados de origem. Essa duração permite que o Oracle GoldenGate recupere logs da instância de banco de dados de origem de acordo com a necessidade.
- Verifique se você tem armazenamento suficiente na instância para os arquivos.

Se você não tiver a retenção de logs ativada, ou se o valor de retenção for muito pequeno, receberá uma mensagem de erro semelhante à seguinte.

```
2022-03-06 06:17:27  ERROR   OGG-00446  error 2 (No such file or directory)
opening redo log /rdsdbdata/db/GGTEST3_A/onlinelog/o1_mf_2_9k4bp1n6_.log for sequence
 1306
Not able to establish initial position for begin time 2022-03-06 06:16:55.
```

O Oracle GoldenGate parece estar devidamente configurado, mas a replicação não está funcionando

Para tabelas pré-existentes, você precisa especificar o SCN do qual o Oracle GoldenGate trabalha.

Como corrigir esse problema

1. Faça login no banco de dados de origem e execute a interface de linha de comando do Oracle GoldenGate (`ggsci`). O exemplo a seguir mostra o formato para login.

   ```
   dblogin userid oggadm1@OGGSOURCE
   ```

2. Usando a linha de comando `ggsci`, configure o SCN inicial para o processo `EXTRACT`. O exemplo a seguir define o SCN como 223274 para `EXTRACT`.

```
ALTER EXTRACT EABC SCN 223274
start EABC
```

3. Faça login no banco de dados de destino. O exemplo a seguir mostra o formato para login.

```
dblogin userid oggadm1@OGGTARGET
```

4. Usando a linha de comando `ggsci`, configure o SCN inicial para o processo `REPLICAT`. O exemplo a seguir define o SCN como 223274 para `REPLICAT`.

```
start RABC atcsn 223274
```

REPLICAT integrado lento devido a consulta no SYS. “_DBA_APPLY_CDR_INFO”

O Oracle GoldenGate Conflict Detection and Resolution (CDR) fornece rotinas básicas de resolução de conflitos. Por exemplo, o CDR pode resolver um conflito exclusivo para uma instrução `INSERT`.

Quando o CDR resolve um conflito, ele pode inserir registros na tabela de exceção `_DBA_APPLY_CDR_INFO` temporariamente. O `REPLICAT` integrado exclui esses registros posteriormente. Em um cenário raro, o `REPLICAT` integrado pode processar um grande número de colisões, mas um novo `REPLICAT` integrado não o substitui. Em vez de serem removidas, as linhas `_DBA_APPLY_CDR_INFO` existentes ficam órfãs. Todos os novos processos `REPLICAT` integrados abrandam porque estão consultando linhas órfãs no `_DBA_APPLY_CDR_INFO`.

Para remover todas as linhas de `_DBA_APPLY_CDR_INFO`, use o procedimento `rdsadmin.rdsadmin_util.truncate_apply$_cdr_info` do Amazon RDS. Este procedimento é lançado como parte da versão de outubro de 2020 e atualização de patch. O procedimento está disponível nas seguintes versões do banco de dados:

- Version 21.0.0.0.ru-2022-01.rur-2022-01.r1 e posterior
- Version 19.0.0.0.ru-2020-10.rur-2020-10.r1 e posterior

O exemplo a seguir trunca a tabela `_DBA_APPLY_CDR_INFO`.

```
SET SERVEROUTPUT ON SIZE 2000
EXEC rdsadmin.rdsadmin_util.truncate_apply$_cdr_info;
```

Usar o Oracle Repository Creation Utility no RDS for Oracle

Você pode usar o Amazon RDS para hospedar uma instância de banco de dados do RDS for Oracle que contém os esquemas para oferecer compatibilidade com seus componentes do Oracle Fusion Middleware. Para poder usar os componentes do Fusion Middleware, você deve criar e preencher esquemas para eles no seu banco de dados. Você cria e preenche os esquemas usando o Oracle Repository Creation Utility (RCU).

Versões e opções de licenciamento compatíveis para RCU

O Amazon RDS não oferece suporte para o Oracle Repository Creation Utility (RCU) versão 12c somente. Você pode usar o RCU nas seguintes configurações:

- RCU 12c com Oracle Database 21c
- RCU 12c com Oracle Database 19c

Antes de poder usar o RCU, faça o seguinte:

- Obtenha uma licença para o Oracle Fusion Middleware.
- Siga as diretrizes de licenciamento da Oracle para o banco de dados Oracle que hospeda o repositório. Para obter mais informações, consulte o Manual do usuário de informações sobre o licenciamento do Oracle Fusion Middleware na documentação da Oracle.

O Fusion MiddleWare é compatível com repositórios no Oracle Database Enterprise Edition e Standard Edition 2. A Oracle recomenda a Enterprise Edition para instalações de produção que exigem particionamento e instalações que exigem a reconstrução do índice online.

Antes de criar sua instância do RDS for Oracle, confirme a versão do banco de dados Oracle que você precisa usar para oferecer compatibilidade com os componentes que deseja implantar. Para encontrar os requisitos para os componentes e as versões do Fusion Middleware que deseja implantar, use a Matriz de Certificação. Para obter mais informações, consulte Configurações do sistema compatíveis com o Oracle Fusion Middleware na documentação da Oracle.

O Amazon RDS oferece suporte para atualizações de versões do banco de dados Oracle conforme necessário. Para ter mais informações, consulte Atualizar a versão de mecanismo de uma instância de banco de dados.

Requisitos e limitações do RCU

Para usar o RCU, você precisa de um Amazon VPC. Sua instância de banco de dados do Amazon RDS precisa estar disponível somente para os componentes do Fusion Middleware, e não para a internet pública. Portanto, hospede sua instância de banco de dados do Amazon RDS em uma sub-rede privada, fornecendo melhor segurança. Você também precisa de uma instância de banco de dados do RDS for Oracle. Para ter mais informações, consulte Criar uma instância de banco de dados Oracle e conectar-se a ela.

Você pode armazenar os esquemas para qualquer componente do Fusion Middleware na sua instância de banco de dados do Amazon RDS. Os esquemas a seguir foram verificados para instalar corretamente:

- Analytics (ACTIVITIES)
- Audit Services (IAU)
- Audit Services Append (IAU_APPEND)
- Audit Services Viewer (IAU_VIEWER)
- Discussions (DISCUSSIONS)
- Metadata Services (MDS)
- Oracle Business Intelligence (BIPLATFORM)
- Oracle Platform Security Services (OPSS)
- Portal and Services (WEBCENTER)
- Portlet Producers (PORTLET)
- Service Table (STB)
- SOA Infrastructure (SOAINFRA)
- User Messaging Service (UCSUMS)
- WebLogic Services (WLS)

Diretrizes para usar o RCU

Veja a seguir algumas recomendações para trabalhar com a sua instância de banco de dados neste cenário:

- Recomendamos que você use o Multi-AZ para cargas de trabalho de produção. Para mais informações sobre como trabalhar com várias zonas de disponibilidade, consulte Regiões, zonas de disponibilidade e Local Zones.
- Para segurança adicional, a Oracle recomenda que você use o Transparent Data Encryption (TDE) para criptografar seus dados em repouso. Se você possui uma licença do Enterprise Edition que inclui a opção de segurança avançada, poderá habilitar a criptografia em repouso usando a opção TDE. Para obter mais informações, consulte Oracle Transparent Data Encryption.

  O Amazon RDS também fornece uma opção de criptografia em repouso para todas as edições de banco de dados. Para obter mais informações, consulte Criptografar recursos do Amazon RDS.
- Configure seus security groups de VPC para permitir a comunicação entre seus servidores de aplicativos e sua instância de banco de dados do Amazon RDS. Os servidores de aplicativos que hospedam os componentes do Fusion Middleware podem estar no Amazon EC2 ou no ambiente local.

Executando o RCU

Para criar e preencher os esquemas para dar suporte aos seus componentes do Fusion Middleware, use o Oracle Repository Creation Utility (RCU). Você pode executar o RCU de diferentes maneiras.

Tópicos

- Executar RCU usando a linha de comando em uma etapa
- Executar o RCU usando a linha de comando em várias etapas
- Executar o RCU no modo interativo

Executar RCU usando a linha de comando em uma etapa

Se você não precisa editar nenhum dos seus esquemas antes de os preencher, pode executar o RCU em uma única etapa. Caso contrário, consulte a seção a seguir para executar o RCU em várias etapas.

Você pode executar o RCU no modo silencioso usando o parâmetro da linha de comando `-silent`. Quando você executa o RCU no modo silencioso, pode evitar inserir senhas na linha de comando mediante a criação de um arquivo de texto contendo as senhas. Crie um arquivo de texto com a senha para `dbUser` na primeira linha e a senha para cada componente nas linhas subsequentes. Você especifica o nome do arquivo de senha como o último parâmetro para o comando RCU.

Example

O exemplo a seguir cria e preenche esquemas para o componente SOA Infrastructure (e suas dependências) em uma única etapa.

Para Linux, macOS ou Unix:

```
export ORACLE_HOME=/u01/app/oracle/product/12.2.1.0/fmw
export JAVA_HOME=/usr/java/jdk1.8.0_65
${ORACLE_HOME}/oracle_common/bin/rcu \
-silent \
-createRepository \
-connectString ${dbhost}:${dbport}:${dbname} \
-dbUser ${dbuser} \
-dbRole Normal \
-honorOMF \
-schemaPrefix ${SCHEMA_PREFIX} \
-component MDS \
-component STB \
-component OPSS \
-component IAU \
-component IAU_APPEND \
-component IAU_VIEWER \
-component UCSUMS \
-component WLS \
-component SOAINFRA \
-f < /tmp/passwordfile.txt
```

Para obter mais informações, consulte Execução do utilitário de criação de repositórios a partir da linha de comando na documentação da Oracle.

Executar o RCU usando a linha de comando em várias etapas

Para editar manualmente seus scripts de esquema, execute o RCU em várias etapas:

1. Execute o RCU no modo Prepare Scripts for System Load (Preparar scripts para o carregamento do sistema) usando o parâmetro da linha de comando `-generateScript` para criar os scripts para os seus esquemas.
2. Edite e execute manualmente o script gerad `script_systemLoad.sql`.
3. Execute o RCU novamente no modo Perform Product Load (Realizar o carregamento do produto) usando o parâmetro da linha de comando `-dataLoad` para preencher os esquemas.

4. Execute o script de limpeza gerado `script_postDataLoad.sql`.

Para executar o RCU no modo silencioso, especifique o parâmetro da linha de comando `-silent`. Quando você executa o RCU no modo silencioso, pode evitar digitar senhas na linha de comando, criando um arquivo de texto contendo as senhas. Crie um arquivo de texto com a senha para `dbUser` na primeira linha e a senha para cada componente nas linhas subsequentes. Especifique o nome do arquivo de senha como o último parâmetro para o comando do RCU.

Example

O exemplo a seguir cria scripts de esquema para o componente SOA Infrastructure (e suas dependências).

Para Linux, macOS ou Unix:

```
export ORACLE_HOME=/u01/app/oracle/product/12.2.1.0/fmw
export JAVA_HOME=/usr/java/jdk1.8.0_65
${ORACLE_HOME}/oracle_common/bin/rcu \
-silent \
-generateScript \
-connectString ${dbhost}:${dbport}:${dbname} \
-dbUser ${dbuser} \
-dbRole Normal \
-honorOMF \
[-encryptTablespace true] \
-schemaPrefix ${SCHEMA_PREFIX} \
-component MDS \
-component STB \
-component OPSS \
-component IAU \
-component IAU_APPEND \
-component IAU_VIEWER \
-component UCSUMS \
-component WLS \
-component SOAINFRA \
-scriptLocation /tmp/rcuscripts \
-f < /tmp/passwordfile.txt
```

Agora, você pode editar o script gerado, conectar-se à sua instância de banco de dados Oracle e executar o script. O script gerado se chama `script_systemLoad.sql`. Para obter informações

sobre como se conectar à sua instância de banco de dados Oracle, consulte Etapa 3: Conectar seu cliente SQL a uma instância de banco de dados Oracle..

O exemplo a seguir preenche os esquemas do componente SOA Infrastructure (e suas dependências).

Para Linux, macOS ou Unix:

```
export JAVA_HOME=/usr/java/jdk1.8.0_65
${ORACLE_HOME}/oracle_common/bin/rcu \
-silent \
-dataLoad \
-connectString ${dbhost}:${dbport}:${dbname} \
-dbUser ${dbuser} \
-dbRole Normal \
-honorOMF \
-schemaPrefix ${SCHEMA_PREFIX} \
-component MDS \
-component STB \
-component OPSS \
-component IAU \
-component IAU_APPEND \
-component IAU_VIEWER \
-component UCSUMS \
-component WLS \
-component SOAINFRA \
-f < /tmp/passwordfile.txt
```

Para terminar, você se conecta à sua instância de banco de dados Oracle e executa o script de limpeza. O nome do script é `script_postDataLoad.sql`.

Para obter mais informações, consulte Execução do utilitário de criação de repositórios a partir da linha de comando na documentação da Oracle.

Executar o RCU no modo interativo

Para usar a interface de usuário gráfica do RCU, execute o RCU no modo interativo. Inclua o parâmetro `-interactive` e omita o parâmetro `-silent`. Para obter mais informações, consulte Noções básicas sobre as telas do utilitário de criação de repositórios na documentação da Oracle.

Example

O exemplo a seguir inicia o RCU no modo interativo e pré-preenche as informações de conexão.

Para Linux, macOS ou Unix:

```
export ORACLE_HOME=/u01/app/oracle/product/12.2.1.0/fmw
export JAVA_HOME=/usr/java/jdk1.8.0_65
${ORACLE_HOME}/oracle_common/bin/rcu \
-interactive \
-createRepository \
-connectString ${dbhost}:${dbport}:${dbname} \
-dbUser ${dbuser} \
-dbRole Normal
```

Solução de problemas do RCU

Esteja ciente dos seguintes problemas.

Tópicos

- Oracle Managed Files (OMF)
- Privilégios de objeto
- Enterprise Scheduler Service

Oracle Managed Files (OMF)

O Amazon RDS usa arquivos de dados OMF para simplificar o gerenciamento do armazenamento. Você pode personalizar atributos de espaço de tabela, como o gerenciamento do tamanho e da extensão. No entanto, se você especificar um nome de arquivo de dados ao executar o RCU, o código do espaço de tabela falhará com `ORA-20900`. É possível usar o RCU com o OMF das seguintes maneiras:

- No RCU 12.2.1.0 e versões posteriores, use o parâmetro de linha de comando `-honorOMF`.
- No RCU 12.1.0.3 e versões posteriores, use várias etapas e edite o script gerado. Para ter mais informações, consulte Executar o RCU usando a linha de comando em várias etapas.

Privilégios de objeto

Como o Amazon RDS é um serviço gerenciado, você não tem acesso completo `SYSDBA` à sua instância de banco de dados do RDS para Oracle. No entanto, o RCU 12c oferece suporte a usuários com privilégios menores. Na maioria dos casos, o privilégio de usuário mestre é suficiente para criar repositórios.

A conta principal pode conceder diretamente privilégios que já foram concedidos `WITH GRANT OPTION`. Em alguns casos, ao tentar conceder privilégios de objeto SYS, o RCU pode falhar com `ORA-01031`. Você pode tentar novamente e executar o procedimento armazenado `rdsadmin_util.grant_sys_object`, conforme mostrado no exemplo a seguir:

```
BEGIN
  rdsadmin.rdsadmin_util.grant_sys_object('GV_$SESSION','MY_DBA','SELECT');
END;
/
```

Se você tentar conceder privilégios SYS no objeto SCHEMA_VERSION_REGISTRY, a operação poderá falhar com `ORA-20199: Error in rdsadmin_util.grant_sys_object`. Você pode qualificar a tabela SCHEMA_VERSION_REGISTRY$ e a visualização SCHEMA_VERSION_REGISTRY com o nome do proprietário do esquema, que é SYSTEM, e repetir a operação. Como alternativa, você pode criar um sinônimo. Faça login como usuário principal e execute as seguintes instruções:

```
CREATE OR REPLACE VIEW SYSTEM.SCHEMA_VERSION_REGISTRY
  AS SELECT * FROM SYSTEM.SCHEMA_VERSION_REGISTRY$;
CREATE OR REPLACE PUBLIC SYNONYM SCHEMA_VERSION_REGISTRY FOR
 SYSTEM.SCHEMA_VERSION_REGISTRY;
CREATE OR REPLACE PUBLIC SYNONYM SCHEMA_VERSION_REGISTRY$ FOR SCHEMA_VERSION_REGISTRY;
```

Enterprise Scheduler Service

Quando você usa o RCU para remover um repositório do Enterprise Scheduler Service, o RCU pode falhar com `Error: Component drop check failed`.

Configurando o Oracle Connection Manager em uma instância do Amazon EC2

O Oracle Connection Manager (CMAN) é um servidor de proxy que encaminha solicitações de conexão para servidores de banco de dados ou outros servidores proxy. Você pode usar o CMAN para configurar o seguinte:

Controle de acesso

Você pode criar regras que filtram solicitações de clientes especificadas pelo usuário e aceitam outras.

Multiplexing de sessão

Você pode canalizar várias sessões de clientes por meio de uma conexão de rede para um destino de servidor compartilhado.

Normalmente, o CMAN reside em um host separado do servidor de banco de dados e dos hosts do cliente. Para obter mais informações, consulte Configuring Oracle Connection Manager (Configurar o Oracle Connection Manager) na documentação do Oracle Database.

Tópicos

- Versões e opções de licenciamento compatíveis para CMAN
- Requisitos e limitações do CMAN
- Configurando o CMAN

Versões e opções de licenciamento compatíveis para CMAN

O CMAN oferece suporte à Enterprise Edition de todas as versões do Oracle Database que o Amazon RDS oferece suporte. Para obter mais informações, consulte Versões do RDS para Oracle.

Você pode instalar o Oracle Connection Manager em um host separado do host em que o Oracle Database está instalado. Você não precisa de uma licença separada para o host que executa o CMAN.

Requisitos e limitações do CMAN

Para fornecer uma experiência totalmente gerenciada, o Amazon RDS restringe o acesso ao sistema operacional. Você não pode modificar parâmetros de banco de dados que exigem acesso ao sistema

operacional. Assim, o Amazon RDS não oferece suporte a recursos do CMAN que exigem que você faça login no sistema operacional.

Configurando o CMAN

Ao configurar o CMAN, você executa a maior parte do trabalho fora do banco de dados do RDS for Oracle.

Tópicos

- Etapa 1: configurar o CMAN em uma instância do Amazon EC2 na mesma VPC da instância do RDS for Oracle
- Etapa 2: configurar parâmetros de banco de dados para o CMAN
- Etapa 3: associar a sua instância de banco de dados ao grupo de parâmetros.

Etapa 1: configurar o CMAN em uma instância do Amazon EC2 na mesma VPC da instância do RDS for Oracle

Para saber como configurar o CMAN, siga as instruções detalhadas na postagem do blog Configurando e usando o Oracle Connection Manager no Amazon EC2 para Amazon RDS for Oracle.

Etapa 2: configurar parâmetros de banco de dados para o CMAN

Para recursos do CMAN, como Modo Traffic Director e multiplexação de sessão, defina o parâmetro `REMOTE_LISTENER` do endereço da instância do CMAN em um grupo de parâmetros de banco de dados. Considere o seguinte cenário:

- A instância CMAN reside em um host com endereço IP `10.0.159.100` e usa porta `1521`.
- Os bancos de dados `orcla`, `orclb`, e `orclc` residem em instâncias de banco de dados do RDS for Oracle separadas.

A tabela a seguir mostra como definir o valor `REMOTE_LISTENER`. O valor `LOCAL_LISTENER` é definido automaticamente pelo Amazon RDS.

Nome da instância de banco de dados	IP da instância de banco de dados	Valor do ouvinte local (definido automaticamente)	Valor de ouvinte remoto (definido por usuário)
orcla	10.0.159.200	(address= (protocol=tcp) (host=10.0.159.200) (port=1521))	10.0.159.100:1521
orclb	10.0.159.300	(address= (protocol=tcp) (host=10.0.159.300) (port=1521))	10.0.159.100:1521
orclc	10.0.159.400	(address= (protocol=tcp) (host=10.0.159.400) (port=1521))	10.0.159.100:1521

Etapa 3: associar a sua instância de banco de dados ao grupo de parâmetros.

Crie ou modifique sua instância de banco de dados para usar o grupo de parâmetros configurado no Etapa 2: configurar parâmetros de banco de dados para o CMAN. Para obter mais informações, consulte Associar um grupo de parâmetros de banco de dados a uma instância de banco de dados no Amazon RDS.

Instalar um Siebel Database no Oracle no Amazon RDS

Você pode usar o Amazon RDS para hospedar um Siebel Database em uma instância de banco de dados Oracle. O Siebel Database faz parte da arquitetura de aplicativos Siebel Customer Relationship Management (CRM). Para ver uma ilustração, consulte Arquitetura genérica do Siebel Business Application.

Use o tópico a seguir para ajudar a configurar o Siebel Database em uma instância de banco de dados Oracle no Amazon RDS. Você também pode descobrir como usar a Amazon Web Services para dar suporte aos outros componentes exigidos pela arquitetura de aplicativos Siebel CRM.

> **Note**
>
> Para instalar um Siebel Database no Oracle no Amazon RDS, você precisa usar a conta de usuário mestre. Você não precisa do privilégio SYSDBA. O privilégio de usuário mestre é suficiente. Para obter mais informações, consulte Privilégios da conta de usuário mestre.

Licenciamento e versões

Para instalar um Siebel Database no Amazon RDS, você deve usar sua própria licença Oracle Database e sua própria licença Siebel. Você deve ter a licença Oracle Database apropriada (com with Software Update License and Support) para a classe da instância de banco de dados e a edição do Oracle Database. Para obter mais informações, consulte Opções de licenciamento do RDS para Oracle.

O Oracle Database Enterprise Edition é a única edição certificada pela Siebel para esse cenário. O Amazon RDS oferece suporte ao Siebel CRM versão 15.0 ou 16.0.

O Amazon RDS oferece suporte para atualizações de versões do banco de dados. Para obter mais informações, consulte Atualizar a versão de mecanismo de uma instância de banco de dados.

Antes de começar

Antes de começar, você precisa de uma Amazon VPC. Como a instância de banco de dados do Amazon RDS precisa estar disponível somente para o Siebel Enterprise Server, e não para a Internet pública, a instância de banco de dados do Amazon RDS é hospedada em uma sub-rede privada, oferecendo maior segurança. Para obter informações sobre como criar uma Amazon VPC para uso com o Siebel CRM, consulte Criar uma instância de banco de dados Oracle e conectar-se a ela.

Antes de começar, você também precisa de uma instância de banco de dados Oracle. Para obter informações sobre como criar uma instância de banco de dados Oracle para uso com o Siebel CRM, consulte Criar uma instância de banco de dados do Amazon RDS.

Instalar e configurar um Siebel Database

Depois de criar sua instância de banco de dados Oracle, você pode instalar o Siebel Database. Você instala o banco de dados criando contas de proprietário e administrador de tabelas, instalando procedimentos armazenados e funções e executando o Siebel Database Configuration Wizard. Para obter mais informações, consulte Instalação do Siebel Database no RDBMS.

Para executar o Siebel Database Configuration Wizard, você precisa usar a conta de usuário mestre. Você não precisa do privilégio SYSDBA. O privilégio de usuário mestre é suficiente. Para obter mais informações, consulte Privilégios da conta de usuário mestre.

Usar outros recursos do Amazon RDS com um Siebel Database

Depois de criar sua instância de banco de dados Oracle, você pode usar recursos adicionais do Amazon RDS para ajudá-lo a personalizar seu Siebel Database.

Coletar estatísticas com a opção Oracle Statspack

Você pode adicionar recursos à instância de banco de dados com o uso de opções em grupos de opções de banco de dados. Quando criou sua instância de banco de dados Oracle, você usou o grupo de opções de banco de dados padrão. Se quiser adicionar recursos ao seu banco de dados, você pode criar um novo grupo de opções para a sua instância de banco de dados.

Se você deseja coletar estatísticas de desempenho sobre o seu Siebel Database, pode adicionar o recurso Oracle Statspack. Para obter mais informações, consulte Oracle Statspack.

Algumas alterações de opções são aplicadas imediatamente e outras são aplicadas durante a próxima janela de manutenção da instância de banco de dados. Para obter mais informações, consulte Trabalhar com grupos de opções. Depois de criar um grupo de opções personalizado, modifique sua instância de banco de dados para anexá-lo. Para obter mais informações, consulte Modificar uma instância de banco de dados do Amazon RDS.

Ajuste de performance com parâmetros

Você gerencia sua configuração de mecanismo de banco de dados com o uso dos parâmetros em um parameter group de banco de dados. Quando criou sua instância de banco de dados Oracle, você usou o parameter group de banco de dados padrão. Se quiser personalizar sua configuração

de banco de dados, você pode criar um novo parameter group para a sua instância de banco de dados.

Quando você altera um parâmetro, dependendo do tipo de parâmetro, as alterações são aplicadas imediatamente ou após a reinicialização manual da instância de banco de dados. Para obter mais informações, consulte Grupos de parâmetros para Amazon RDS. Depois de criar um parameter group personalizado, modifique sua instância de banco de dados para anexá-lo. Para obter mais informações, consulte Modificar uma instância de banco de dados do Amazon RDS.

Para otimizar sua instância de banco de dados Oracle para o Siebel CRM, você pode personalizar determinados parâmetros. A tabela a seguir mostra algumas configurações de parâmetros recomendadas. Para obter mais informações sobre como ajustar a performance do Siebel CRM, consulte o Siebel CRM Performance Tuning Guide (Guia de ajuste da performance do Siebel CRM).

Nome do parâmetro	Valor padrão	Orientação para a performance ideal do Siebel CRM
`_always_semi_join`	`CHOOSE`	`OFF`
`_b_tree_bitmap_plans`	`TRUE`	`FALSE`
`_like_with_bind_as_equality`	`FALSE`	`TRUE`
`_no_or_expansion`	`FALSE`	`FALSE`
`_optimizer_join_sel_sanity_check`	`TRUE`	`TRUE`
`_optimizer_max_permutations`	2000	100

Nome do parâmetro	Valor padrão	Orientação para a performance ideal do Siebel CRM
`_optimizer_sortmerge_join_enabled`	`TRUE`	`FALSE`
`_partition_view_enabled`	`TRUE`	`FALSE`
`open_cursors`	`300`	No mínimo **`2000`**.

Criar snapshots

Depois de criar seu Siebel Database, você pode copiar o banco de dados usando os recursos de snapshot do Amazon RDS. Para obter mais informações, consulte Criar um snapshot de banco de dados para uma instância de banco de dados single-AZ para o Amazon RDS e Restaurar uma instância de banco de dados.

Suporte para outros componentes do Siebel CRM

Além do seu Siebel Database, você também pode usar a Amazon Web Services para dar suporte aos outros componentes da sua arquitetura de aplicativos Siebel CRM. Você pode encontrar mais informações sobre o suporte fornecido pela Amazon AWS para componentes adicionais do Siebel CRM na tabela a seguir.

Componente do Siebel CRM	Amazon AWS Support
Siebel Enterprise (com um ou mais Siebel Servers)	Você pode hospedar seus Siebel Servers em instâncias do Amazon Elastic Compute Cloud (Amazon EC2). Você pode usar o Amazon EC2 para iniciar quantos servidores virtuais forem necessários. Usando o Amazon EC2, você pode escalar facilmente para cima ou para baixo de forma a lidar com mudanças nos requisitos. Para obter mais informações, consulte O que é o Amazon EC2?

Componente do Siebel CRM	Amazon AWS Support
	Você pode colocar seus servidores na mesma VPC com sua instância de banco de dados e usar o security group da VPC para acessar o banco de dados. Para obter mais informações, consulte Trabalhar com uma instância de banco de dados em uma VPC.
Servidores da web (com Siebel Web Server Extensions)	Você pode instalar vários Servidores da Web em várias instâncias do EC2. Você pode usar o Elastic Load Balancing para distribuir o tráfego de entrada entre as instâncias. Para obter mais informações, consulte O que é o Elastic Load Balancing?
Siebel Gateway Name Server	Você pode hospedar seu Siebel Gateway Name Server em uma instância do EC2. Em seguida, pode colocar seu servidor na mesma VPC com a instância de banco de dados e usar o security group da VPC para acessar o banco de dados. Para obter mais informações, consulte Trabalhar com uma instância de banco de dados em uma VPC.

Notas de lançamento do mecanismo Oracle Database

As atualizações feitas nas instâncias de banco de dados do Amazon RDS for Oracle as mantêm atualizadas. Caso aplique atualizações, fique tranquilo porque a instância de banco de dados está executando uma versão do software de banco de dados que foi submetida a testes pela Oracle e pela Amazon. Não oferecemos suporte à aplicação de patches únicos em instâncias de banco de dados individuais do RDS para Oracle.

Você pode especificar qualquer versão atualmente compatível do Oracle Database ao criar uma instância de banco de dados. Você pode especificar a versão principal, como o Oracle Database 19c, e qualquer versão secundária compatível para a versão principal especificada. Se nenhuma versão for especificada, o Amazon RDS usará uma versão compatível por padrão, geralmente a versão mais recente. Se uma versão principal for especificada, mas uma versão secundária não, o Amazon RDS usará por padrão um lançamento recente da versão principal que você especificou. Para ver uma lista de versões compatíveis e padrões para instâncias de banco de dados recém-criadas, use o comando da AWS CLI `describe-db-engine-versions`.

Para obter detalhes sobre as versões do Oracle Database compatíveis com o Amazon RDS, consulte Notas de lançamento do Amazon RDS para Oracle.

Amazon RDS para PostgreSQL

O Amazon RDS oferece suporte a instâncias de banco de dados executando várias versões do PostgreSQL. Para obter uma lista das versões disponíveis, consulte Versões disponíveis do banco de dados PostgreSQL.

Você pode criar instâncias de banco de dados e snapshot de banco de dados, além de restaurações e backup pontuais. As instâncias de banco de dados que executam o PostgreSQL oferecem suporte a implantações multi-AZ, réplicas de leitura, IOPS provisionadas e podem ser criadas dentro de uma nuvem privada virtual (VPC). Você também pode usar o Secure Socket Layer (SSL) para conectar-se a uma instância de banco de dados executando PostgreSQL.

Antes de criar uma instância de banco de dados, conclua as etapas em Configuração do ambiente do Amazon RDS.

Você pode usar qualquer aplicação cliente padrão SQL para executar comandos para a instância no computador do cliente. Dentre essas aplicações, estão pgAdmin, uma ferramenta popular de administração e de desenvolvimento de código aberto para PostgreSQL, ou o psql, um utilitário de linha de comando que faz parte da instalação do PostgreSQL. Para oferecer uma experiência de serviço gerenciada, o Amazon RDS não fornece acesso ao host para as instâncias de banco de dados. Ele também restringe o acesso a determinados procedimentos e tabelas do sistema que exigem privilégios avançados. O Amazon RDS oferece suporte a acesso a bancos de dados em uma instância de banco de dados que usa qualquer aplicação cliente SQL padrão. O Amazon RDS não permite o acesso direto do host a uma instância de banco de dados usando Telnet ou Secure Shell (SSH).

O Amazon RDS para PostgreSQL é compatível com vários padrões do setor. Por exemplo, você pode usar os bancos de dados do Amazon RDS para PostgreSQL para criar aplicativos em conformidade com a HIPAA e para armazenar informações relacionadas à saúde. Isso inclui o armazenamento de informações de saúde protegidas sob um Business Associate Agreement (BAA – Acordo de Associação Comercial) concluído com a AWS. O Amazon RDS para PostgreSQL também atende aos requisitos de segurança do Federal Risk and Authorization Management Program (FedRAMP). O Amazon RDS para PostgreSQL recebeu uma autoridade provisória da Joint Authorization Board (JAB) da FedRAMP para operar (P-ATO) na linha de base ALTA da FedRAMP nas regiões da AWS GovCloud (US). Para obter mais informações sobre padrões de conformidade com suporte, consulte Conformidade da nuvem AWS.

Para importar os dados PostgreSQL em uma instância de banco de dados, siga as informações na seção Como importar dados no PostgreSQL do Amazon RDS.

> **Important**
>
> Se você encontrar algum problema com a instância de banco de dados do RDS para PostgreSQL, o atendente do AWS Support poderá precisar de mais informações sobre a integridade dos bancos de dados. O objetivo é garantir que o AWS Support receba as informações necessárias o mais rápido possível.
> É possível usar o PG Collector para ajudar a reunir informações valiosas do banco de dados em um arquivo HTML consolidado. Para ter mais informações sobre o PG Collector, como executá-lo e como baixar o relatório HTML, consulte PG Collector.
> Após a conclusão bem-sucedida e salvo indicação em contrário, o script exibe a saída em um formato HTML legível. O script foi projetado para excluir desse HTML quaisquer dados ou detalhes de segurança que possam comprometer sua empresa. Ele também não faz modificações no banco de dados ou no ambiente. No entanto, se houver informações no HTML que você não se sinta confortável para compartilhar, fique à vontade para removê-las antes de fazer upload do HTML. Quando o HTML estiver aceitável, faça upload dele utilizando a seção de anexos nos detalhes do caso de suporte.

Tópicos

- Tarefas comuns de gerenciamento do Amazon RDS para PostgreSQL
- Trabalhar com o ambiente de visualização de banco de dados
- Versões disponíveis do banco de dados PostgreSQL
- Conceitos básicos sobre o processo de lançamento incremental do RDS para PostgreSQL
- Versões de extensões do PostgreSQL compatíveis
- Trabalhar com recursos do PostgreSQL compatíveis com o Amazon RDS para PostgreSQL
- Conectar a uma instância de banco de dados executando o mecanismo de banco de dados do PostgreSQL
- Proteger conexões com o RDS para PostgreSQL com SSL/TLS
- Usar a autenticação Kerberos com o Amazon RDS para PostgreSQL
- Usar um servidor DNS personalizado para acesso à rede de saída
- Atualizações do mecanismo de banco de dados do RDS para PostgreSQL

- Atualizar uma versão do mecanismo de snapshot de banco de dados PostgreSQL
- Trabalhar com réplicas de leitura do Amazon RDS para PostgreSQL
- Melhorar a performance das consultas para o RDS para PostgreSQL com leituras otimizadas pelo Amazon RDS
- Como importar dados no PostgreSQL do Amazon RDS
- Exportar dados de uma instância de banco de dados do RDS para PostgreSQL para o Amazon S3
- Invocar uma função do AWS Lambda de uma instância de banco de dados do RDS for PostgreSQL
- Tarefas comuns de DBA do Amazon RDS para PostgreSQL
- Ajustar com eventos de espera do RDS para PostgreSQL
- Ajustar o RDS para PostgreSQL com insights proativos do Amazon DevOps Guru
- Usar extensões PostgreSQL com o Amazon RDS para PostgreSQL
- Trabalhar com os invólucros de dados externos compatíveis do Amazon RDS for PostgreSQL
- Trabalhar com Trusted Language Extensions para PostgreSQL

Tarefas comuns de gerenciamento do Amazon RDS para PostgreSQL

Estas são tarefas de gerenciamento comuns que você executa com uma instância de banco de dados do Amazon RDS para PostgreSQL, com links para documentação relevante de cada tarefa.

Área de tarefa	Documentação relevante
Configuração do Amazon RDS para o primeiro uso Antes de criar sua instância de banco de dados, certifique-se de concluir alguns pré-requisitos. Por exemplo, as instância s de banco de dados são criadas por padrão com um firewall que impede acessá-las. Então é necessário criar um grupo de segurança com os endereços IP corretos e uma configuração de rede para acessar a instância de banco de dados.	Configuração do ambiente do Amazon RDS
Noções básicas sobre instâncias de banco de dados do Amazon RDS	Classes de instâncias de banco de dados

Área de tarefa	Documentação relevante
Se você estiver criando uma instância de banco de dados para fins de produção, deve entender como classes de instâncias, tipos de armazenamento e IOPS provisionadas funcionam no Amazon RDS.	Tipos de armazenamento do Amazon RDS Armazenamento SSD de IOPS provisionadas
Encontrar versões disponíveis do PostgreSQL O Amazon RDS oferece suporte a várias versões do PostgreSQL.	Versões disponíveis do banco de dados PostgreSQL
Configuração do suporte à alta disponibilidade e failover Uma instância de banco de dados de produção deve usar implantações multi-AZ. As implantações multi-AZ oferecem maior disponibilidade, durabilidade de dados e tolerância a falhas para instâncias de banco de dados.	Configurar e gerenciar uma implantação multi-AZ para o Amazon RDS
Noções básicas sobre a rede da Amazon Virtual Private Cloud (VPC) Se a sua conta AWS tiver uma VPC padrão, a instância de banco de dados será criada automaticamente dentro da VPC padrão. Em alguns casos, sua conta pode não ter uma VPC padrão e você pode querer uma instância de banco de dados em uma VPC. Nesses casos, crie a VPC e os grupos de sub-redes antes de criar a instância de banco de dados.	Trabalhar com uma instância de banco de dados em uma VPC
Importar dados para o PostgreSQL do Amazon RDS Você pode usar várias ferramentas diferentes para importar os dados para sua instância de banco de dados PostgreSQL no Amazon RDS.	Como importar dados no PostgreSQL do Amazon RDS

Área de tarefa	Documentação relevante
Configurar réplicas de leitura somente leitura (primárias e em espera) O RDS para PostgreSQL oferece suporte às réplicas de leitura na mesma região da AWS e em uma região da AWS diferente da instância primária.	Trabalhar com réplicas de leitura de instância de banco de dados Trabalhar com réplicas de leitura do Amazon RDS para PostgreSQL Criar uma réplica de leitura em uma Região da AWS diferente
Noções básicas sobre grupos de segurança Por padrão, as instâncias de banco de dados são criadas com um firewall que impede o acesso a elas. Para fornecer acesso por meio desse firewall, você edita as regras de entrada do grupo de segurança de VPC associado à VPC que hospeda a instância de banco de dados.	Controlar acesso com grupos de segurança
Configuração de grupos de parâmetros e recursos Para alterar os parâmetros padrão para sua instância de banco de dados, crie um grupo de parâmetros de banco de dados personalizado e altere as configurações dele. Se você fizer isso antes de criar sua instância de banco de dados, é possível escolher seu grupo de parâmetros de banco de dados personalizado ao criar a instância.	Grupos de parâmetros para Amazon RDS
Conectar à instância de banco de dados PostgreSQL Após criar um grupo de segurança e associá-lo a uma instância de banco de dados, você poderá se conectar à instância de banco de dados usando qualquer aplicação de cliente SQL padrão, como `psql` ou `pgAdmin`.	Conectar a uma instância de banco de dados executando o mecanismo de banco de dados do PostgreSQL Usar o SSL com uma instância de banco de dados PostgreSQL

Área de tarefa	Documentação relevante
Backup e restauração da instância de banco de dados Você pode configurar a instância de banco de dados para fazer backups automáticos ou snapshots manuais e, depois, restaurar instâncias a partir dos backups ou snapshots.	Backup, restauração e exportação de dados
Monitorar atividades e da performance da instância de banco de dados Monitore uma instância de banco de dados PostgreSQL usando as métricas, eventos e o monitoramento avançado do Amazon RDS do CloudWatch.	Visualizar métricas no console do Amazon RDS Visualizar eventos do Amazon RDS
Atualização da versão do banco de dados PostgreSQL Você pode fazer atualizações das versões principal e secundári a para sua instância de banco de dados PostgreSQL.	Atualizações do mecanismo de banco de dados do RDS para PostgreSQL Escolher uma versão principal para uma atualização do RDS para PostgreSQL
Trabalhar com arquivos de log Você pode acessar os arquivos de log de sua instância de banco de dados PostgreSQL.	Arquivos de log do banco de dados do RDS para PostgreSQL
Noções básicas sobre as práticas recomendadas para instância s de banco de dados PostgreSQL Encontre algumas de práticas recomendadas para trabalhar com o PostgreSQL no Amazon RDS.	Práticas recomendadas para trabalhar com PostgreSQL

Veja a seguir uma lista de outras seções deste guia que podem ajudar você a entender e usar recursos importantes do RDS para PostgreSQL:

- Noções básicas de perfis e permissões do PostgreSQL
- Controlar o acesso de usuários ao banco de dados PostgreSQL

- Trabalhar com parâmetros na instância de banco de dados do RDS para PostgreSQL
- Entender os mecanismos de registro compatíveis com o RDS para PostgreSQL
- Trabalhar com o autovacuum do PostgreSQL no Amazon RDS para PostgreSQL
- Usar um servidor DNS personalizado para acesso à rede de saída

Trabalhar com o ambiente de visualização de banco de dados

A comunidade do PostgreSQL lança novas versões e novas extensões do PostgreSQL continuamente. Isso dá aos usuários do PostgreSQL a oportunidade de experimentar uma nova versão do PostgreSQL mais cedo. Para saber mais sobre o processo de lançamento beta da comunidade do PostgreSQL, consulte Beta Information (Informações sobre a versão beta) na documentação do PostgreSQL. Da mesma forma, o Amazon RDS disponibiliza determinadas versões beta do PostgreSQL como versões prévias. Isso permite que você crie instâncias de banco de dados usando a versão prévia e teste seus recursos no ambiente de visualização do banco de dados.

As instâncias de banco de dados do RDS para PostgreSQL no Database Preview Environment são semelhantes às outras instâncias do RDS para PostgreSQL em termos de funcionalidade. No entanto, você não pode usar uma versão prévia para produção.

Lembre-se das seguintes limitações importantes:

- Todas as instâncias de banco de dados são excluídas 60 dias após serem criadas, juntamente com backups e snapshots.
- Só é possível criar uma instância de banco de dados em uma virtual private cloud (VPC) com base no serviço da Amazon VPC.
- Você só pode usar o SSD de uso geral e o armazenamento em SSD IOPS provisionadas.
- Você não pode obter ajuda do AWS Support com instâncias de banco de dados. Em vez disso, você pode postar suas perguntas na comunidade de perguntas e respostas gerenciadas do AWS, AWSre:Post.
- Não é possível copiar um snapshot de uma instância de banco de dados para um ambiente de produção.

As opções a seguir são compatíveis com a visualização.

- Você só pode criar instâncias de banco de dados com o uso dos tipos M6i, R6i, M6g, M5, T3, R6g e R5. Para obter mais informações sobre as classes de instância do RDS, consulte Classes de instâncias de banco de dados .
- Você pode usar implantações single-AZ e multi-AZ.
- Você pode usar funções padrão de despejo e carregamento do PostgreSQL para exportar ou importar bancos de dados para o Database Preview Environment.

Tópicos

- Atributos não compatíveis com o ambiente de visualização de banco de dados
- PostgreSQL versão 17 no Ambiente de Pré-visualização do Banco de Dados
- Criar uma nova instância de banco de dados no ambiente de visualização de banco de dados

Atributos não compatíveis com o ambiente de visualização de banco de dados

Os atributos a seguir não estão disponíveis no ambiente de visualização de banco de dados:

- Cópia de snapshots entre regiões
- Réplicas de leitura entre regiões

PostgreSQL versão 17 no Ambiente de Pré-visualização do Banco de Dados

> **Note**
>
> Esta é a documentação prévia do Amazon RDS PostgreSQL versão 17. Está sujeita a alteração.

O PostgreSQL versão 17.0 já está disponível no Ambiente de Pré-visualização do Banco de Dados do Amazon RDS. O PostgreSQL versão 17.0 contém várias melhorias que estão descritas na seguinte documentação do PostgreSQL: PostgreSQL 17 Released!

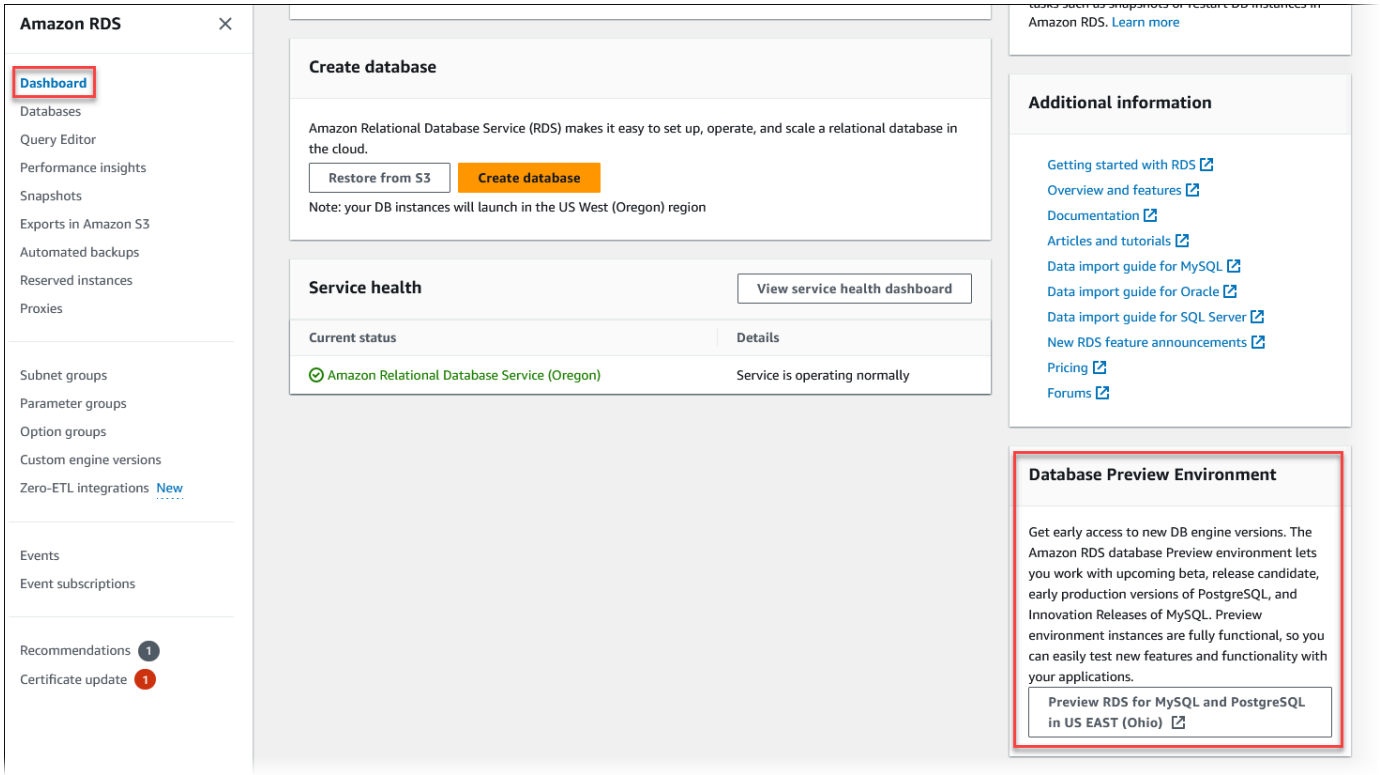
Para obter informações sobre o ambiente de visualização de banco de dados, consulte the section called “Trabalhar com o ambiente de visualização de banco de dados”. Para acessar o ambiente de visualização do console, selecione https://console.aws.amazon.com/rds-preview/.

Criar uma nova instância de banco de dados no ambiente de visualização de banco de dados

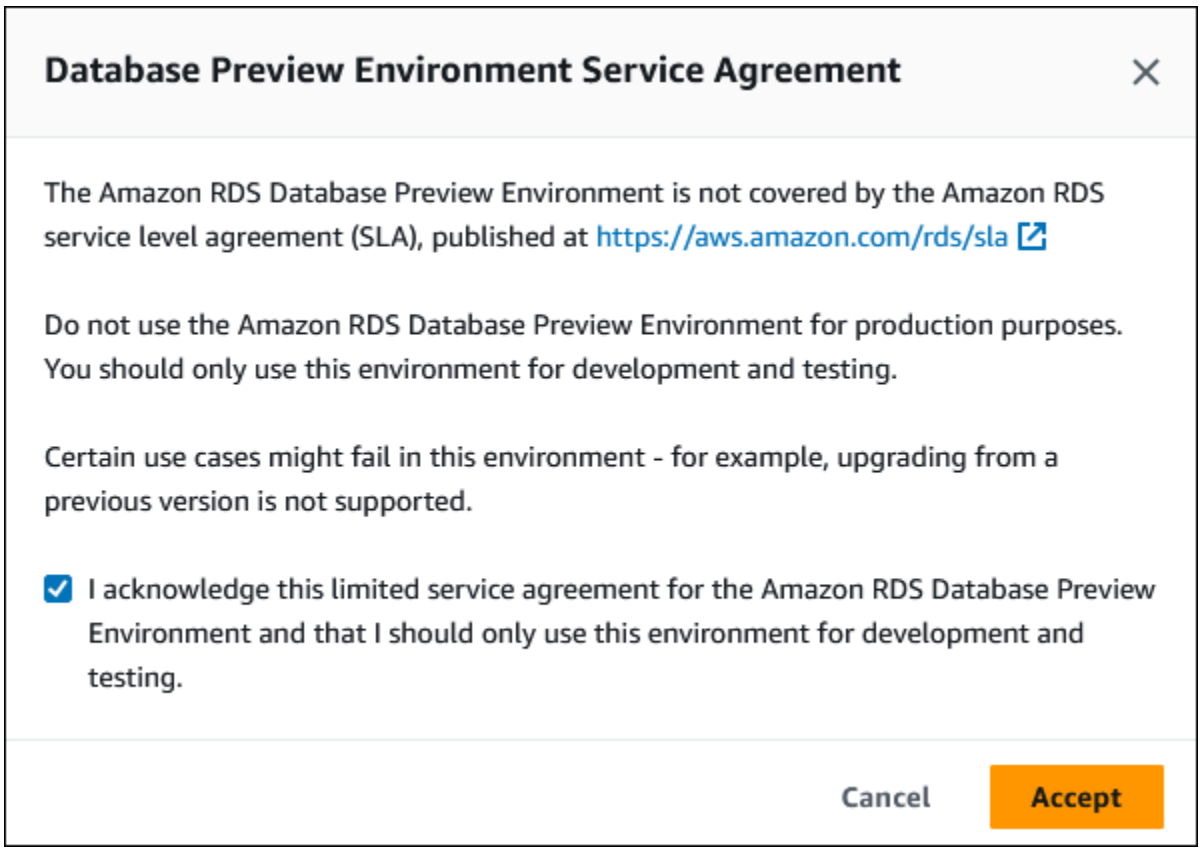
Use o procedimento a seguir para criar uma instância de banco de dados no ambiente de visualização.

Para criar uma instância de banco de dados no ambiente de visualização de banco de dados

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. Escolha Dashboard (Painel) no painel de navegação.
3. Na página Dashboard (Painel), localize a seção Database Preview Environment (Ambiente de visualização do banco de dados) na página Dashboard (Painel), conforme mostrado na imagem a seguir.



Você também pode navegar diretamente para o Ambiente de visualização do banco de dados. Antes de continuar, você deve reconhecer e aceitar as limitações.



4. Para criar a instância de banco de dados do RDS para PostgreSQL, siga o mesmo processo de criação de qualquer instância de banco de dados do Amazon RDS. Para obter mais informações, consulte o procedimento Console em Criar uma instância de banco de dados.

Para criar uma instância no ambiente de visualização do banco de dados usando a API do RDS ou a AWS CLI, use o endpoint a seguir.

```
rds-preview.us-east-2.amazonaws.com
```

Versões disponíveis do banco de dados PostgreSQL

O Amazon RDS oferece suporte a instâncias de banco de dados executando várias edições do PostgreSQL. Você pode especificar qualquer versão do PostgreSQL disponível atualmente ao criar uma nova instância de banco de dados. Você pode especificar a versão principal (como o PostgreSQL 14) e qualquer versão secundária disponível para a versão principal especificada. Se nenhuma versão for especificada, o Amazon RDS usará uma versão disponível por padrão, geralmente a versão mais recente. Se uma versão principal for especificada, mas uma versão secundária não, o Amazon RDS usará por padrão um lançamento recente da versão principal que você especificou.

Para ver uma lista de versões disponíveis, bem como os padrões para instâncias de banco de dados recém-criadas, use o comando da AWS CLI `describe-db-engine-versions`. Por exemplo, para exibir a versão padrão do mecanismo PostgreSQL, use o seguinte comando:

```
aws rds describe-db-engine-versions --default-only --engine postgres
```

Para obter detalhes sobre as versões do PostgreSQL compatíveis com o Amazon RDS, consulte "Amazon RDS for PostgreSQL Release Notes" (Notas de lançamento do Amazon RDS para PostgreSQL). Você também pode visualizar informações sobre datas de suporte para as versões principais do mecanismo executando o comando describe-db-major-engine-versionsAWS CLI ou usando a operação de API DescribeDBMajorEngineVersions do RDS.

Se ainda não estiver tudo pronto para fazer upgrade manualmente para uma nova versão do mecanismo principal antes da data de término do suporte padrão do RDS, o Amazon RDS inscreverá automaticamente seus bancos de dados no Suporte estendido do Amazon RDS após a data de término do suporte padrão do RDS. Assim, você poderá continuar executando o RDS para PostgreSQL versão 11 e posterior. Para obter mais informações, consulte Suporte estendido do Amazon RDS com Amazon RDS e Preço do Amazon RDS.

Versões obsoletas do Amazon RDS for PostgreSQL

Observe as seguintes versões obsoletas:

- O RDS para PostgreSQL 10 foi descontinuado em fevereiro de 2023.
- O RDS para PostgreSQL 9.6 foi descontinuado em março de 2022.
- O RDS para PostgreSQL 9.5 foi descontinuado em março de 2021.

Para saber mais sobre a política de descontinuação do RDS para PostgreSQL, consulte Perguntas frequentes sobre o Amazon RDS. Para obter mais informações sobre as versões do PostgreSQL, consulte Versionamento de política na documentação do PostgreSQL.

Conceitos básicos sobre o processo de lançamento incremental do RDS para PostgreSQL

O RDS para PostgreSQL oferece correções de segurança, melhorias de performance e novos recursos por meio de versões incrementais, mantendo a compatibilidade com versões secundárias. Essas versões são rotuladas como R1, R2, R3 e assim por diante.

Convenção de nomenclatura de versões de lançamento

- R1 é a versão inicial de uma versão secundária. Ocasionalmente, ela inclui novos recursos, extensões ou atualizações de extensões existentes.
- As versões de lançamento subsequentes (R2, R3 e posteriores) incluem:
  - Atualizações de segurança
  - Melhorias de performance
  - Correções de bugs
  - Atualizações de extensão

Vantagens do processo de lançamento incremental do RDS para PostgreSQL.

O processo de lançamento incremental oferece as seguintes vantagens:

- Adoção rápida de novas versões da comunidade do PostgreSQL e, ao mesmo tempo, gerenciamento separado dos aprimoramentos específicos do RDS por meio de versões subsequentes. Isso simplifica o processo de lançamento e garante a entrega mais rápida de atualizações essenciais.
- Acesso a correções de bugs, novos recursos e atualizações de segurança e de extensões, mantendo a compatibilidade com a versão secundária do PostgreSQL.

Gerenciamento de atualizações de versão

O Amazon RDS notifica você sobre novos lançamentos incrementais por meio de ações de manutenção pendentes no AWS Management Console. Você pode atualizar o banco de dados usando um destes métodos:

- Habilite atualizações automáticas durante as janelas de manutenção programada.
- Aplique atualizações manualmente por meio de ações de manutenção pendentes.
- Use implantações azul/verde com replicação física para minimizar o tempo de inatividade. Para obter mais informações, consulte Implantações azul/verde do Amazon RDS oferecem suporte à atualização de versões secundárias do RDS para PostgreSQL.

Antes de atualizar o banco de dados, considere os seguintes pontos principais:

- É necessário reinicializar o banco de dados para fazer atualizações, a menos que você use implantações azul/verde com replicação física.
- Algumas versões incrementais são obrigatórias, particularmente aquelas que incluem correções de segurança.

Para obter mais informações sobre como atualizar sua instância de banco de dados do Amazon RDS, consulte Extensões confiáveis do PostgreSQL e apply-pending-maintenance-action.

Versões de extensões do PostgreSQL compatíveis

O RDS para PostgreSQL é compatível com várias extensões do PostgreSQL. A comunidade PostgreSQL às vezes se refere a elas como módulos. Extensões expandem a funcionalidade fornecida pelo mecanismo PostgreSQL. Você pode encontrar uma lista de extensões às quais o Amazon RDS oferece suporte no grupo de parâmetros de banco de dados padrão para essa versão do PostgreSQL. Você também pode ver a lista de extensões atuais que usam o `psql` exibindo o parâmetro `rds.extensions` conforme o exemplo a seguir.

```
SHOW rds.extensions;
```

> **Note**
>
> Os parâmetros adicionados em uma versão secundária podem ser exibidos incorretamente ao usar o parâmetro `rds.extensions` em `psql`.

A partir do RDS para PostgreSQL 13, determinadas extensões podem ser instaladas por usuários do banco de dados que não sejam o `rds_superuser`. Elas são conhecidas como extensões confiáveis. Para saber mais, consulte Extensões confiáveis do PostgreSQL.

Determinadas versões do RDS para PostgreSQL oferecem suporte ao parâmetro `rds.allowed_extensions`. Esse parâmetro permite que um `rds_superuser` limite as extensões que podem ser instaladas na instância de banco de dados do RDS para PostgreSQL. Para ter mais informações, consulte Restringir a instalação de extensões do PostgreSQL.

Para listas das versões e extensões do PostgreSQL compatíveis com cada versão disponível do RDS para PostgreSQL, consulte “PostgreSQL extensions supported on Amazon RDS” (Extensões do PostgreSQL compatíveis com o Amazon RDS) em “Amazon RDS for PostgreSQL Release Notes” (Notas de lançamento do Amazon RDS para PostgreSQL).

Restringir a instalação de extensões do PostgreSQL

Você pode restringir quais extensões podem ser instaladas em uma instância de banco de dados PostgreSQL. Por padrão, esse parâmetro não está definido, portanto, qualquer extensão compatível pode ser adicionada se o usuário tiver permissões para isso. Para fazer isso, defina o parâmetro `rds.allowed_extensions` para uma string de nomes de extensão separados por vírgulas. Ao adicionar uma lista de extensões a esse parâmetro, você identifica explicitamente as extensões que

a instância de banco de dados do RDS para PostgreSQL pode usar. Somente essas extensões podem ser instaladas na instância de banco de dados PostgreSQL.

A string padrão para o `rds.allowed_extensions` parâmetro é '*', o que significa que qualquer extensão disponível para a versão do motor pode ser instalada. Alterar o parâmetro `rds.allowed_extensions` não requer uma reinicialização do banco de dados porque é um parâmetro dinâmico.

O mecanismo de instância de banco de dados PostgreSQL deve ser uma das seguintes versões para você usar o parâmetro `rds.allowed_extensions`:

- Todas as versões 16 do PostgreSQL
- PostgreSQL 15 e todas as versões posteriores
- PostgreSQL 14 e todas as versões posteriores
- PostgreSQL 13.3 e versões secundárias posteriores
- PostgreSQL 12.7 e versões secundárias posteriores

Para ver quais instalações de extensão são permitidas, use o comando psql a seguir.

```
postgres=> SHOW rds.allowed_extensions;
 rds.allowed_extensions
------------------------
 *
```

Se uma extensão foi instalada antes de ser deixada de fora da lista no parâmetro `rds.allowed_extensions`, a extensão ainda pode ser usada normalmente, e comandos como `ALTER EXTENSION` e `DROP EXTENSION` continuarão a funcionar. No entanto, depois que uma extensão é restrita, `CREATE EXTENSION` os comandos para a extensão restrita falharão.

Instalação de dependências de extensão com `CREATE EXTENSION CASCADE` também são restritas. A extensão e suas dependências devem ser especificadas em `rds.allowed_extensions`. Se uma instalação de dependência de extensão falhar, a instrução `CREATE EXTENSION CASCADE` inteira falhará.

Se uma extensão não estiver incluída no parâmetro `rds.allowed_extensions`, você verá um erro como o mostrado a seguir, caso tente instalá-la.

```
ERROR: permission denied to create extension "extension-name"
```

```
HINT: This extension is not specified in "rds.allowed_extensions".
```

Extensões confiáveis do PostgreSQL

Para instalar a maioria das extensões do PostgreSQL são necessários privilégios de `rds_superuser`. O PostgreSQL 13 introduziu as extensões confiáveis, o que reduz a necessidade de privilégios de `rds_superuser` para usuários comuns. Com esse recurso, os usuários podem instalar diversas extensões, se tiverem o privilégio `CREATE` no banco de dados atual, em vez de precisarem ter a função `rds_superuser`. Para obter mais informações, consulte o comando SQL CREATE EXTENSION na documentação do PostgreSQL.

A seguir listamos as extensões que podem ser instaladas por um usuário que tem o privilégio `CREATE` no banco de dados atual e não necessitam da função `rds_superuser`:

- bool_plperl
- btree_gin
- btree_gist
- citext
- cube
- dict_int
- fuzzystrmatch
- hstore
- intarray
- isn
- jsonb_plperl
- ltree
- pg_trgm
- pgcrypto
- plperl
- plpgsql
- pltcl
- tablefunc
- tsm_system_rows

- tsm_system_time
- unaccent
- uuid-ossp

Para listas das versões e extensões do PostgreSQL compatíveis com cada versão disponível do RDS para PostgreSQL, consulte PostgreSQL extensions supported on Amazon RDS (Extensões do PostgreSQL compatíveis com o Amazon RDS) em “Amazon RDS for PostgreSQL Release Notes” (Notas de lançamento do Amazon RDS para PostgreSQL).

Trabalhar com recursos do PostgreSQL compatíveis com o Amazon RDS para PostgreSQL

O Amazon RDS para PostgreSQL oferece suporte a muitos dos recursos mais comuns do PostgreSQL. Por exemplo, o PostgreSQL tem um recurso autovacuum que executa manutenção de rotina no banco de dados. O recurso de autovacuum está ativo por padrão. Embora você possa desativar esse recurso, é altamente recomendável que você o mantenha ativado. Compreender esse recurso e o que você pode fazer para garantir que ele funcione como deveria é uma tarefa básica de qualquer DBA. Para obter mais informações sobre o autovacuum, consulte Trabalhar com o autovacuum do PostgreSQL no Amazon RDS para PostgreSQL. Para saber mais sobre outras tarefas comuns do DBA, consulte Tarefas comuns de DBA do Amazon RDS para PostgreSQL.

O RDS para PostgreSQL também oferece suporte a extensões que adicionam funcionalidades importantes à instância de banco de dados. Por exemplo, você pode usar a extensão PostGIS para trabalhar com dados espaciais ou usar a extensão pg_cron para programar a manutenção de dentro da instância. Para obter mais informações sobre as extensões PostgreSQL, consulte Usar extensões PostgreSQL com o Amazon RDS para PostgreSQL.

Os invólucros de dados externos são um tipo específico de extensão projetado para permitir que sua instância de banco de dados do RDS para PostgreSQL funcione com outros bancos de dados comerciais ou tipos de dados. Para obter mais informações sobre invólucros de dados externos compatíveis com o RDS para PostgreSQL, consulte Trabalhar com os invólucros de dados externos compatíveis do Amazon RDS for PostgreSQL.

A seguir, você pode encontrar informações sobre mais alguns recursos compatíveis com o RDS para PostgreSQL.

Tópicos

- Tipos de dados personalizados e enumerações com o RDS para PostgreSQL
- Acionadores de eventos para RDS para PostgreSQL
- Páginas grandes para RDS para PostgreSQL
- Executar replicação lógica para o Amazon RDS para PostgreSQL
- Disco de RAM para o stats_temp_directory
- Tablespaces para RDS para PostgreSQL
- Agrupamentos do RDS para PostgreSQL para EBCDIC e outras migrações de mainframe

- Gerenciar sincronização de slots lógicos do RDS para PostgreSQL

Tipos de dados personalizados e enumerações com o RDS para PostgreSQL

O PostgreSQL é compatível com a criação de tipos de dados personalizados e o trabalho com enumerações. Para obter mais informações sobre como criar e trabalhar com enumerações e outros tipos de dados, consulte Enumerated types (Tipos enumerados) na documentação do PostgreSQL.

Veja a seguir como criar um tipo como uma enumeração e, em seguida, inserir valores em uma tabela.

```
CREATE TYPE rainbow AS ENUM ('red', 'orange', 'yellow', 'green', 'blue', 'purple');
CREATE TYPE
CREATE TABLE t1 (colors rainbow);
CREATE TABLE
INSERT INTO t1 VALUES ('red'), ( 'orange');
INSERT 0 2
SELECT * from t1;
colors
--------
red
orange
(2 rows)
postgres=> ALTER TYPE rainbow RENAME VALUE 'red' TO 'crimson';
ALTER TYPE
postgres=> SELECT * from t1;
colors
---------
crimson
orange
(2 rows)
```

Acionadores de eventos para RDS para PostgreSQL

Todas as versões atuais do PostgreSQL suportam acionadores de eventos, assim como todas as versões disponíveis do RDS para PostgreSQL. Você pode usar a conta de usuário principal (padrão, `postgres`) para criar, modificar, renomear e excluir acionadores de eventos. Os acionadores de eventos estão no nível da instância do banco de dados, portanto, podem ser aplicados a todos os bancos de dados em uma instância.

Por exemplo, o código a seguir cria um acionador de evento que imprime o usuário atual no final de cada comando de linguagem de definição de dados (DDL).

```
CREATE OR REPLACE FUNCTION raise_notice_func()
    RETURNS event_trigger
    LANGUAGE plpgsql AS
$$
BEGIN
    RAISE NOTICE 'In trigger function: %', current_user;
END;
$$;

CREATE EVENT TRIGGER event_trigger_1
    ON ddl_command_end
EXECUTE PROCEDURE raise_notice_func();
```

Para obter mais informações sobre os triggers de eventos do PostgreSQL, consulte Triggers de eventos na documentação do PostgreSQL.

Há várias limitações de uso para os acionadores de eventos do PostgreSQL no Amazon RDS. Incluindo o seguinte:

- Não é possível criar gatilhos de eventos em réplicas de leitura. No entanto, você pode criar triggers de eventos na origem de uma réplica de leitura. Os acionadores de eventos serão copiados para a réplica de leitura. Os triggers de eventos na réplica de leitura não são acionados nela quando há mudanças provenientes da origem. No entanto, se a réplica de leitura for promovida, os acionadores de eventos existentes serão ativados quando ocorrerem operações do banco de dados.
- Para realizar uma atualização de versão principal da instância de banco de dados do PostgreSQL que usa acionadores de eventos, exclua os acionadores antes de atualizar a instância.

Páginas grandes para RDS para PostgreSQL

Páginas grandes são um recurso de gerenciamento de memória que reduz a sobrecarga quando uma instância de banco de dados está trabalhando com grandes blocos contíguos de memória, como os usados por buffers compartilhados. Esse recurso PostgreSQL é compatível com todas as versões do RDS para PostgreSQL atualmente disponíveis. As páginas grandes são alocadas à aplicação usando chamadas de memória compartilhada para a memória compartilhada `mmap` ou `SYSV`. O RDS para PostgreSQL comporta tamanhos de página de 4 KB e 2 MB.

Você pode ativar ou desativar páginas muito grandes alterando o valor do parâmetro `huge_pages`. O recurso é ativado por padrão para todas as classes de instância de banco de dados que não sejam classes de instância de banco de dados micro, pequenas e médias.

O RDS para PostgreSQL usa páginas enormes com base na memória compartilhada disponível. Se a instância de banco de dados não puder usar páginas enormes devido a restrições de memória compartilhada, o Amazon RDS impedirá que a instância de banco de dados seja iniciada. Nesse caso, o Amazon RDS define o status da instância de banco de dados como um estado de parâmetros incompatíveis. Nesse caso, configure o parâmetro `huge_pages` como `off` para permitir que o Amazon RDS inicie a instância de banco de dados.

O parâmetro `shared_buffers` é essencial para configurar o grupo de memória compartilhada, necessário para usar páginas grandes. O valor padrão para o parâmetro `shared_buffers` usa uma macro de parâmetros de banco de dados. Essa macro define uma porcentagem do total de 8 KB de páginas que estão disponíveis para a memória da instância de banco de dados. Quando você usa páginas enormes, elas estão localizadas com as páginas enormes. O Amazon RDS coloca uma instância de banco de dados em um estado de parâmetros incompatível se os parâmetros da memória compartilhada estão configurados para exigir mais de 90% da memória da instância de banco de dados.

Para saber mais sobre o gerenciamento de memória do PostgreSQL, consulte Consumo de recursos na documentação do PostgreSQL.

Executar replicação lógica para o Amazon RDS para PostgreSQL

A partir da versão 10.4, o RDS para PostgreSQL é compatível com a sintaxe SQL de publicação e assinatura, que foi introduzida pela primeira vez no PostgreSQL 10. Para saber mais, consulte Logical replication (Replicação lógica) na documentação do PostgreSQL.

> **Note**
>
> Além do recurso nativo de replicação lógica do PostgreSQL introduzido no PostgreSQL 10, o RDS para PostgreSQL também é compatível com a extensão `pglogical`. Para obter mais informações, consulte Usar pglogical para sincronizar dados entre instâncias.

A seguir, você pode encontrar informações sobre como configurar a replicação lógica de uma instância de banco de dados do RDS para PostgreSQL.

Tópicos

- Considerações sobre a replicação lógica e a decodificação lógica
- Como trabalhar com slots de replicação lógica
- Replicar dados em nível de tabela usando replicação lógica

Considerações sobre a replicação lógica e a decodificação lógica

O RDS para PostgreSQL oferece suporte a transmissão de alterações de Write-Ahead Log (WAL – Log de gravação antecipada) usando slots de replicação lógica. Ele também permite o uso de decodificação lógica. Você pode configurar slots de replicação lógica em sua instância e transmitir alterações no banco de dados por meio desses slots para um cliente, como `pg_recvlogical`. Você cria slots de replicação lógica no nível do banco de dados. Esses slots são compatíveis com conexões de replicação para um único banco de dados.

Os clientes mais comuns para replicação lógica do PostgreSQL são o AWS Database Migration Service ou um host gerenciado personalizado em uma instância do Amazon EC2. O slot de replicação lógica não tem informações sobre o receptor do transmissão. Além disso, não é exigido que o destino seja um banco de dados de réplica. Se você configurar um slot de replicação lógica e não fizer a leitura no slot, os dados poderão ser gravados no armazenamento da instância de banco de dados e lotá-lo rapidamente.

Ative a replicação lógica do PostgreSQL e a descodificação lógica no Amazon RDS com um parâmetro, um tipo de conexão de replicação e uma função de segurança. O cliente da descodificação lógica pode ser qualquer cliente que possa estabelecer uma conexão de replicação a um banco de dados em uma instância de banco de dados PostgreSQL.

Como ativar a descodificação lógica de uma instância de banco de dados do RDS para PostgreSQL

1. Verifique se a conta de usuário que você está usando tem as seguintes funções:
   - A função `rds_superuser` para que você possa ativar a replicação lógica
   - A função `rds_replication` atribui as permissões necessárias para gerenciar slots lógicos e transmitir dados usando slots lógicos.
2. Defina o parâmetro estático `rds.logical_replication` como 1. Como parte da aplicação desse parâmetro, defina também os parâmetros `wal_level`, `max_wal_senders`, `max_replication_slots` e `max_connections`. Essas alterações de parâmetros podem

aumentar a geração de WALs. Portanto, configure o parâmetro `rds.logical_replication` quando estiver usando slots lógicos.

3. Reinicialize a instância de banco de dados para que o parâmetro estático `rds.logical_replication` tenha efeito.
4. Crie um slot de replicação lógica conforme explicado na próxima seção. Esse processo requer que você especifique um plug-in de decodificação. Atualmente, o RDS para PostgreSQL aceita os plug-ins de saída test_decoding e wal2json fornecidos com o PostgreSQL.

Para obter mais informações sobre a descodificação lógica do PostgreSQL, consulte a documentação do PostgreSQL.

Como trabalhar com slots de replicação lógica

Você pode usar comandos SQL para trabalhar com slots lógicos. Por exemplo, o comando a seguir cria um slot lógico denominado `test_slot` usando o plug-in de saída padrão `test_decoding` do PostgreSQL.

```
SELECT * FROM pg_create_logical_replication_slot('test_slot', 'test_decoding');
slot_name    | xlog_position
-----------------+---------------
regression_slot | 0/16B1970
(1 row)
```

Para listar slots lógicos, use o seguinte comando.

```
SELECT * FROM pg_replication_slots;
```

Para descartar um slot lógico, use o seguinte comando.

```
SELECT pg_drop_replication_slot('test_slot');
pg_drop_replication_slot
-----------------------
(1 row)
```

Para obter mais exemplos sobre como trabalhar com slots lógicos de replicação, consulte "Logical decoding examples" (Exemplos de decodificação lógica) na documentação do PostgreSQL.

Após criar um slot de replicação lógica, você pode iniciar o streaming. O exemplo a seguir mostra como a decodificação lógica é controlada sobre o protocolo de replicação de streaming. Este

exemplo usa o programa pg_recvlogical, incluído na distribuição do PostgreSQL. Para fazer Isso, a autenticação do cliente deve estar configurada para permitir conexões de replicação.

```
pg_recvlogical -d postgres --slot test_slot -U postgres
    --host -instance-name.111122223333.aws-region.rds.amazonaws.com
    -f -  --start
```

Para ver o conteúdo da visualização `pg_replication_origin_status`, consulte a função `pg_show_replication_origin_status`.

```
SELECT * FROM pg_show_replication_origin_status();
local_id | external_id | remote_lsn | local_lsn
----------+-------------+------------+-----------
(0 rows)
```

Replicar dados em nível de tabela usando replicação lógica

Você pode usar a replicação lógica para replicar dados das tabelas de origem para as tabelas de destino no RDS para PostgreSQL. A replicação lógica primeiro executa um carregamento inicial dos dados existentes das tabelas de origem e, em seguida, continua a replicar as alterações em andamento.

1. Criar as tabelas de origem

   Conecte-se ao banco de dados de origem na instância de banco de dados do RDS para PostgreSQL:

   ```
   source=> CREATE TABLE testtab (slno int primary key);
   CREATE TABLE
   ```

2. Inserir dados nas tabelas de origem

   ```
   source=> INSERT INTO testtab VALUES (generate_series(1,1000));
   INSERT 0 1000
   ```

3. Criar uma publicação para tabelas de origem

   - Crie uma publicação para tabelas de origem:

```
source=> CREATE PUBLICATION testpub FOR TABLE testtab;
CREATE PUBLICATION
```

   - Use uma consulta SELECT para verificar os detalhes da publicação que foi criada:

```
source=> SELECT * FROM pg_publication;
  oid   | pubname | pubowner | puballtables | pubinsert | pubupdate | pubdelete |
 pubtruncate | pubviaroot
--------+---------+----------+--------------+-----------+-----------+-----------
+-------------+------------
 115069 | testpub |    16395 | f            | t         | t         | t         |
 t           | f
(1 row)
```

   - Verifique se as tabelas de origem foram adicionadas à publicação:

```
source=> SELECT * FROM pg_publication_tables;
pubname | schemaname | tablename
---------+------------+-----------
 testpub | public     | testtab
(1 rows)
```

   - Para replicar todas as tabelas em um banco de dados, use:

```
CREATE PUBLICATION testpub FOR ALL TABLES;
```

   - Se a publicação já tiver sido criada para uma tabela individual e você precisar adicionar uma nova tabela, execute a seguinte consulta para adicionar novas tabelas à publicação existente:

```
ALTER PUBLICATION <publication_name> add table <new_table_name>;
```

4. Conectar-se ao banco de dados de destino e criar tabelas de destino

   - Conecte-se ao banco de dados de destino na instância de banco de dados de destino. Crie as tabelas de destino com o mesmo nome das tabelas de origem:

```
target=> CREATE TABLE testtab (slno int primary key);
CREATE TABLE
```

   - Verifique se não há dados presentes nas tabelas de destino executando uma consulta SELECT nas tabelas de destino:

```
target=> SELECT count(*) FROM testtab;
 count
-------
     0
(1 row)
```

5. Criar e verificar a assinatura no banco de dados de destino

   - Crie a assinatura no banco de dados de destino:

```
target=> CREATE SUBSCRIPTION testsub
CONNECTION 'host=<source RDS/host endpoint> port=5432 dbname=<source_db_name>
 user=<user> password=<password>'
PUBLICATION testpub;
NOTICE:  Created replication slot "testsub" on publisher
CREATE SUBSCRIPTION
```

   - Use uma consulta SELECT para verificar se a assinatura está habilitada:

```
target=> SELECT oid, subname, subenabled, subslotname, subpublications FROM
 pg_subscription;
  oid  | subname | subenabled | subslotname | subpublications
-------+---------+------------+-------------+-----------------
```

```
 16434 | testsub | t          | testsub     | {testpub}
(1 row)
```

- Quando a assinatura é criada, ela carrega todos os dados das tabelas de origem para as tabelas de destino. Execute uma consulta SELECT nas tabelas de destino para verificar se os dados iniciais foram carregados:

```
target=> SELECT count(*) FROM testtab;
 count
-------
  1000
(1 row)
```

6. Verificar o slot de replicação no banco de dados de origem

   A criação de uma assinatura no banco de dados de destino cria um slot de replicação no banco de dados de origem. Verifique os detalhes do slot de replicação executando a seguinte consulta SELECT no banco de dados de origem:

```
source=> SELECT * FROM pg_replication_slots;

slot_name |  plugin  | slot_type | datoid | database | temporary | active |
 active_pid | xmin | catalog_xmin | restart_lsn | confirmed_flush_lsn | wal_status
 | safe_wal_size
----------+----------+-----------+--------+----------+-----------+--------
+------------+------+--------------+-------------+---------------------
+------------+---------------
testsub   | pgoutput | logical   | 115048 | source   | f         | t      |
 846 |      |         6945 | 58/B4000568 | 58/B40005A0         | reserved   |
(1 row)
```

7. Testar a replicação

   - Teste se as alterações de dados nas tabelas de origem estão sendo replicadas para as tabelas de destino inserindo linhas nas tabelas de origem:

```
source=> INSERT INTO testtab VALUES(generate_series(1001,2000));
INSERT 0 1000

source=> SELECT count(*) FROM testtab;
 count
-------
  2000
(1 row)
```

- Verifique o número de linhas nas tabelas de destino para confirmar se novas inserções estão sendo replicadas:

```
target=> SELECT count(*) FROM testtab;
 count
-------
  2000
(1 row)
```

8. Atualizar a assinatura depois de adicionar tabelas

   - Quando você adiciona novas tabelas a uma publicação existente, é obrigatório atualizar a assinatura para que as alterações entrem em vigor:

```
ALTER SUBSCRIPTION <subscription_name> REFRESH PUBLICATION;
```

   - Esse comando busca informações ausentes da tabela no publicador e inicia a replicação das tabelas que foram adicionadas às publicações assinadas desde que a assinatura foi criada ou atualizada pela última vez.

Disco de RAM para o stats_temp_directory

Você pode usar o parâmetro `rds.pg_stat_ramdisk_size` do RDS para PostgreSQL para especificar a memória do sistema alocada a um disco RAM para armazenar o `stats_temp_directory` do PostgreSQL. O parâmetro de disco de RAM só está disponível no RDS para PostgreSQL 14 e versões anteriores.

Mediante certas workloads, definir este parâmetro pode melhorar a performance e diminuir os requisitos de E/S. Para obter mais informações sobre como usar o `stats_temp_directory`, consulte a documentação do PostgreSQL.

Para configurar um disco RAM para o parâmetro `stats_temp_directory`, configure o parâmetro `rds.pg_stat_ramdisk_size` como um valor literal inteiro no grupo de parâmetros usado pela instância de banco de dados. Esse parâmetro denota MB, portanto, você deve usar um valor inteiro. Expressões, fórmulas e funções não são válidas para o parâmetro `rds.pg_stat_ramdisk_size`. Reinicialize a instância de banco de dados para que o novo valor entre em vigor. Para obter informações sobre como configurar parâmetros, consulte Grupos de parâmetros para Amazon RDS.

Por exemplo, o seguinte comando da AWS CLI define o parâmetro do disco de RAM para 256 MB.

```
aws rds modify-db-parameter-group \
    --db-parameter-group-name pg-95-ramdisk-testing \
    --parameters "ParameterName=rds.pg_stat_ramdisk_size, ParameterValue=256,
 ApplyMethod=pending-reboot"
```

Depois de reiniciar, execute o seguinte comando para ver o status de `stats_temp_directory`:

```
postgres=> SHOW stats_temp_directory;
```

O comando deve retornar um resultado parecido com o exemplo a seguir.

```
stats_temp_directory
----------------------------
/rdsdbramdisk/pg_stat_tmp
(1 row)
```

Tablespaces para RDS para PostgreSQL

O RDS para PostgreSQL oferece suporte a tablespaces para compatibilidade. Como todo o armazenamento está em um único volume lógico, você não pode usar tablespaces para divisão ou isolamento de E/S. Nossos benchmarks e experiência indicam que um único volume lógico é a melhor configuração para a maioria dos casos de uso.

Para criar e usar tablespaces com sua instância de banco de dados do RDS para PostgreSQL é necessário a função `rds_superuser`. A sua conta de usuário principal da instância de banco de

dados do RDS para PostgreSQL (nome padrão, `postgres`) é membro dessa função. Para obter mais informações, consulte Noções básicas de perfis e permissões do PostgreSQL.

Se você especificar um nome de arquivo ao criar um espaço de tabela, o prefixo de caminho será `/rdsdbdata/db/base/tablespace`. O exemplo a seguir coloca arquivos de espaço de tabela em `/rdsdbdata/db/base/tablespace/data`. Este exemplo pressupõe que um usuário `dbadmin` (função) existe e que lhe foi concedido a função `rds_superuser` necessária para trabalhar com tablespaces.

```
postgres=> CREATE TABLESPACE act_data
  OWNER dbadmin
  LOCATION '/data';
CREATE TABLESPACE
```

Para saber mais sobre tablespaces do PostgreSQL, consulte Tablespaces na documentação do PostgreSQL.

Agrupamentos do RDS para PostgreSQL para EBCDIC e outras migrações de mainframe

O RDS para PostgreSQL versões 10 e posteriores inclui a ICU versão 60.2, que é baseada no Unicode 10.0 e inclui agrupamentos do Unicode Common Locale Data Repository, CLDR 32. Essas bibliotecas de internacionalização de software garantem que as codificações de caracteres sejam apresentadas de forma consistente, independentemente do sistema operacional ou da plataforma. Para obter mais informações sobre o Unicode CLDR-32, consulte a "CLDR 32 Release Note" (Nota de lançamento do CLDR 32) no site do Unicode CLDR. Você pode saber mais sobre os componentes de internacionalização do Unicode (ICU) no site do Comitê Técnico do ICU (ICU-TC). Para obter informações sobre o ICU-60, consulte Baixar o ICU 60.

A partir da versão 14.3, o RDS para PostgreSQL também inclui agrupamentos que ajudam na integração e conversão de dados de sistemas baseados em EBCDIC. O código de intercâmbio decimal codificado binário estendido ou codificação EBCDIC é comumente usada pelos sistemas operacionais de mainframe. Esses agrupamentos fornecidos pelo Amazon RDS são definidos de forma restrita para classificar somente os caracteres Unicode que são mapeados diretamente para páginas de código EBCDIC. Os caracteres são classificados em ordem de pontos de código EBCDIC para permitir a validação dos dados após a conversão. Esses agrupamentos não incluem formulários desnormalizados nem incluem caracteres Unicode que não são mapeados diretamente para um caractere na página de código EBCDIC de origem.

Os mapeamentos de caracteres entre páginas de código EBCDIC e pontos de código Unicode são baseados em tabelas publicadas pela IBM. O conjunto completo é disponibilizado pela IBM como um arquivo compactado para download. O RDS para PostgreSQL usou esses mapeamentos com as ferramentas fornecidas pelo ICU para criar os agrupamentos listados nas tabelas desta seção. Os nomes do agrupamento incluem um idioma e um país, conforme exigido pelo ICU. No entanto, as páginas de código EBCDIC não especificam idiomas, e algumas páginas de código EBCDIC abrangem vários países. Isso significa que a parte do idioma e do país dos nomes de agrupamento na tabela é arbitrária e não precisa corresponder à localidade atual. Em outras palavras, o número da página de código é a parte mais importante do nome do agrupamento nesta tabela. Você pode usar qualquer um dos agrupamentos listados nas tabelas a seguir em qualquer banco de dados do RDS para PostgreSQL.

- Unicode to EBCDIC collations table: algumas ferramentas de migração de dados de mainframe usam internamente LATIN1 ou LATIN9 para codificar e processar dados. Essas ferramentas usam esquemas de ida e volta para preservar a integridade dos dados e oferecer suporte à conversão reversa. Os agrupamentos nesta tabela podem ser usados por ferramentas que processam dados usando a codificação LATIN1, que não exige tratamento especial.
- Unicode to LATIN9 collations table: você pode usar esses agrupamentos em qualquer banco de dados do RDS para PostgreSQL.

Na tabela a seguir, você encontra agrupamentos disponíveis no RDS para PostgreSQL que mapeiam páginas de código EBCDIC para pontos de código Unicode. Recomendamos usar os agrupamentos nesta tabela para o desenvolvimento de aplicações que exijam classificação com base na ordem das páginas de código da IBM.

Nome do agrupamento PostgreSQL	Descrição do mapeamento da página de código e ordem de classificação
da-DK-cp277-x-icu	Os caracteres Unicode que são mapeados diretamente para a página de código IBM EBCDIC 277 (por tabelas de conversão) são classificados na ordem de pontos de código IBM CP 277
de-DE-cp273-x-icu	Os caracteres Unicode que são mapeados diretamente para a página de código IBM

Nome do agrupamento PostgreSQL	Descrição do mapeamento da página de código e ordem de classificação
	EBCDIC 273 (por tabelas de conversão) são classificados na ordem de pontos de código IBM CP 273
en-GB-cp285-x-icu	Os caracteres Unicode que são mapeados diretamente para a página de código IBM EBCDIC 285 (por tabelas de conversão) são classificados na ordem de pontos de código IBM CP 285
en-US-cp037-x-icu	Os caracteres Unicode que são mapeados diretamente para a página de código IBM EBCDIC 037 (por tabelas de conversão) são classificados na ordem de pontos de código IBM CP 37
es-ES-cp284-x-icu	Os caracteres Unicode que são mapeados diretamente para a página de código IBM EBCDIC 284 (por tabelas de conversão) são classificados na ordem de pontos de código IBM CP 284
fi-FI-cp278-x-icu	Os caracteres Unicode que são mapeados diretamente para a página de código IBM EBCDIC 278 (por tabelas de conversão) são classificados na ordem de pontos de código IBM CP 278
fr-FR-cp297-x-icu	Os caracteres Unicode que são mapeados diretamente para a página de código IBM EBCDIC 297 (por tabelas de conversão) são classificados na ordem de pontos de código IBM CP 297

Nome do agrupamento PostgreSQL	Descrição do mapeamento da página de código e ordem de classificação
it-IT-cp280-x-icu	Os caracteres Unicode que são mapeados diretamente para a página de código IBM EBCDIC 280 (por tabelas de conversão) são classificados na ordem de pontos de código IBM CP 280
nl-BE-cp500-x-icu	Os caracteres Unicode que são mapeados diretamente para a página de código IBM EBCDIC 500 (por tabelas de conversão) são classificados na ordem de pontos de código IBM CP 500

O Amazon RDS fornece um conjunto de agrupamentos adicionais que classificam pontos de código Unicode mapeados para caracteres LATIN9 usando as tabelas publicadas pela IBM, na ordem dos pontos de código originais, de acordo com a página de código EBCDIC dos dados de origem.

Nome do agrupamento PostgreSQL	Descrição do mapeamento da página de código e ordem de classificação
da-DK-cp1142m-x-icu	Caracteres Unicode mapeados para caractere s LATIN9 originalmente convertidos da página de código IBM EBCDIC 1142 (por tabelas de conversão) são classificados na ordem de pontos de código IBM CP 1142
de-DE-cp1141m-x-icu	Caracteres Unicode mapeados para caractere s LATIN9 originalmente convertidos da página de código IBM EBCDIC 1141 (por tabelas de conversão) são classificados na ordem de pontos de código IBM CP 1141
en-GB-cp1146m-x-icu	Caracteres Unicode mapeados para caractere s LATIN9 originalmente convertidos da página de código IBM EBCDIC 1146 (por tabelas de

Nome do agrupamento PostgreSQL	Descrição do mapeamento da página de código e ordem de classificação
	conversão) são classificados na ordem de pontos de código IBM CP 1146
en-US-cp1140m-x-icu	Caracteres Unicode mapeados para caracteres LATIN9 originalmente convertidos da página de código IBM EBCDIC 1140 (por tabelas de conversão) são classificados na ordem de pontos de código IBM CP 1140
es-ES-cp1145m-x-icu	Caracteres Unicode mapeados para caracteres LATIN9 originalmente convertidos da página de código IBM EBCDIC 1145 (por tabelas de conversão) são classificados na ordem de pontos de código IBM CP 1145
fi-FI-cp1143m-x-icu	Caracteres Unicode mapeados para caracteres LATIN9 originalmente convertidos da página de código IBM EBCDIC 1143 (por tabelas de conversão) são classificados na ordem de pontos de código IBM CP 1143
fr-FR-cp1147m-x-icu	Caracteres Unicode mapeados para caracteres LATIN9 originalmente convertidos da página de código IBM EBCDIC 1147 (por tabelas de conversão) são classificados na ordem de pontos de código IBM CP 1147
it-IT-cp1144m-x-icu	Caracteres Unicode mapeados para caracteres LATIN9 originalmente convertidos da página de código IBM EBCDIC 1144 (por tabelas de conversão) são classificados na ordem de pontos de código IBM CP 1144

Nome do agrupamento PostgreSQL	Descrição do mapeamento da página de código e ordem de classificação
nl-BE-cp1148m-x-icu	Caracteres Unicode mapeados para caracteres LATIN9 originalmente convertidos da página de código IBM EBCDIC 1148 (por tabelas de conversão) são classificados na ordem de pontos de código IBM CP 1148

A seguir, você pode encontrar um exemplo de como usar um agrupamento do RDS para PostgreSQL.

```
db1=> SELECT pg_import_system_collations('pg_catalog');
 pg_import_system_collations
-----------------------------
                          36
db1=> SELECT '¤' < 'a' col1;
 col1
------
 t
db1=> SELECT '¤' < 'a' COLLATE "da-DK-cp277-x-icu" col1;
 col1
------
 f
```

Recomendamos usar os agrupamentos na Unicode to EBCDIC collations table e na Unicode to LATIN9 collations table para desenvolvimento de aplicações que exijam classificação com base na ordenação das páginas de código da IBM. Os seguintes agrupamentos (com o sufixo “b”) também são visíveis em `pg_collation`, mas são destinados ao uso por ferramentas de integração e migração de dados de mainframe em AWS que mapeiam páginas de código com mudanças específicas de pontos de código e exigem tratamento especial em agrupamento. Ou seja, o uso dos agrupamentos a seguir não é recomendado.

- da-DK-277b-x-icu
- da-DK-1142b-x-icu
- de-DE-cp273b-x-icu
- de-DE-cp1141b-x-icu

- en-GB-cp1146b-x-icu
- en-GB-cp285b-x-icu
- en-US-cp037b-x-icu
- en-US-cp1140b-x-icu
- es-ES-cp1145b-x-icu
- es-ES-cp284b-x-icu
- fi-FI-cp1143b-x-icu
- fr-FR-cp1147b-x-icu
- fr-FR-cp297b-x-icu
- it-IT-cp1144b-x-icu
- it-IT-cp280b-x-icu
- nl-BE-cp1148b-x-icu
- nl-BE-cp500b-x-icu

Para saber mais sobre a migração de aplicações de ambientes de mainframe para a AWS, consulte O que é o AWS Mainframe Modernization?

Para obter mais informações sobre como gerenciar agrupamentos no PostgreSQL, consulte "Collation Support" (Compatibilidade com agrupamentos) na documentação do PostgreSQL.

Gerenciar sincronização de slots lógicos do RDS para PostgreSQL

A partir da comunidade PostgreSQL 17, um novo recurso para sincronizar automaticamente os slots de replicação lógica dos servidores primários para os servidores em espera foi introduzido por meio do parâmetro `sync_replication_slots` ou da função `pg_sync_replication_slots()` relacionada, que sincroniza manualmente os slots na execução.

Esses recursos estão disponíveis a partir do RDS para PostgreSQL 17. Uma configuração típica terá uma instância primária e sua réplica de leitura, bem como um assinante de replicação lógica para a primária.

Garanta que a assinatura seja criada com a opção de failover definida como verdadeira:

```
CREATE SUBSCRIPTION subname CONNECTION 'host=...' PUBLICATION pubname WITH (failover =
 true);
```

Isso cria um slot lógico no publicador com o failover habilitado.

```
postgres=> SELECT slot_name, slot_type, failover FROM pg_catalog.pg_replication_slots;
 slot_name | slot_type | failover
-----------+-----------+----------
 subname   | logical   | t
(1 row)
```

Ao habilitar a sincronização de slots, todos os slots de replicação lógica de failover na primária são criados automaticamente nos standbys físicos e sincronizados periodicamente. Garante que os seguintes valores foram definidos por meio de grupos de parâmetros:

- `rds.logical_replication` deve ser 1 para permitir a replicação lógica.
- `hot_standby_feedback` deve ser 1 no standby.
- `rds.logical_slot_sync_dbname` no standby deve ser definido como um nome de banco de dados válido.

  O valor padrão do parâmetro é `postgres`. Se a instância de publicação lógica tiver o banco de dados `postgres`, o parâmetro padrão não precisará ser alterado.
- `synchronized_standby_slots` na primária deve ser configurado para o slot de replicação física do standby que deve ser sincronizado.
- `sync_replication_slots` deve ser 1 para habilitar a sincronização automática.

Com um slot de assinatura habilitado para failover e os valores dos parâmetros acima, quando um standby é promovido, o assinante pode alterar sua assinatura dessa instância recém-promovida e continuar a replicação lógica sem problemas.

Conectar a uma instância de banco de dados executando o mecanismo de banco de dados do PostgreSQL

Depois que o Amazon RDS provisiona a sua instância de banco de dados, você pode usar qualquer aplicação cliente padrão SQL para se conectar à instância. Para que você possa se conectar à sua instância de banco de dados, ela precisa estar disponível e acessível. O que determina se você pode se conectar à instância de fora da VPC é a maneira como você criou a instância de banco de dados do Amazon RDS:

- Se você tiver criado sua instância de banco de dados como pública, dispositivos e instâncias do Amazon EC2 fora da VPC poderão se conectar ao seu banco de dados.
- Se você tiver criado sua instância de banco de dados como privada, somente instâncias e dispositivos do Amazon EC2 dentro da Amazon VPC poderão se conectar ao seu banco de dados.

Para verificar se sua instância de banco de dados é pública ou privada, use o AWS Management Console para visualizar a guia Connectivity & security (Conectividade e segurança) para sua instância. Em Security (Segurança), você pode encontrar o valor “Publicly accessible” (Acessível ao público), com No (Não) para privado e Yes (Sim) para público.

Para saber mais sobre diferentes configurações do Amazon RDS e da Amazon VPC e como elas afetam a acessibilidade, consulte Cenários para acessar uma instância de banco de dados em uma VPC.

Sumário

- Instalar o cliente psql
- Localizar as informações de conexão para uma instância de banco de dados do RDS para PostgreSQL
- Usar pgAdmin para se conectar a uma instância de banco de dados RDS for PostgreSQL
- Usar psql para se conectar à sua instância de banco de dados RDS for PostgreSQL
- Conectar-se ao RDS para PostgreSQL com o driver JDBC da Amazon Web Services (AWS)
- Conectar-se ao RDS para PostgreSQL com o driver Python da Amazon Web Services (AWS)
- Solucionar problemas de conexões com a sua instância RDS for PostgreSQL
  - Erro – FATAL: o banco de dados nome não existe
  - Erro – Não foi possível conectar-se ao servidor: a conexão expirou por exceder o tempo limite.

- Erros com regras de acesso a grupos de segurança

Instalar o cliente psql

Para se conectar à sua instância de banco de dados por meio de uma instância do EC2, você pode instalar um cliente PostgreSQL na instância do EC2. Para instalar a versão mais recente do cliente psql no Amazon Linux 2023, execute o seguinte comando:

```
sudo dnf install postgresql<version number>
```

Para instalar a versão mais recente do cliente psql no Amazon Linux 2, execute o seguinte comando:

```
sudo yum install -y postgresql
```

Para instalar a versão mais recente do cliente psql no Ubuntu, execute o seguinte comando:

```
sudo apt install -y postgresql-client
```

Localizar as informações de conexão para uma instância de banco de dados do RDS para PostgreSQL

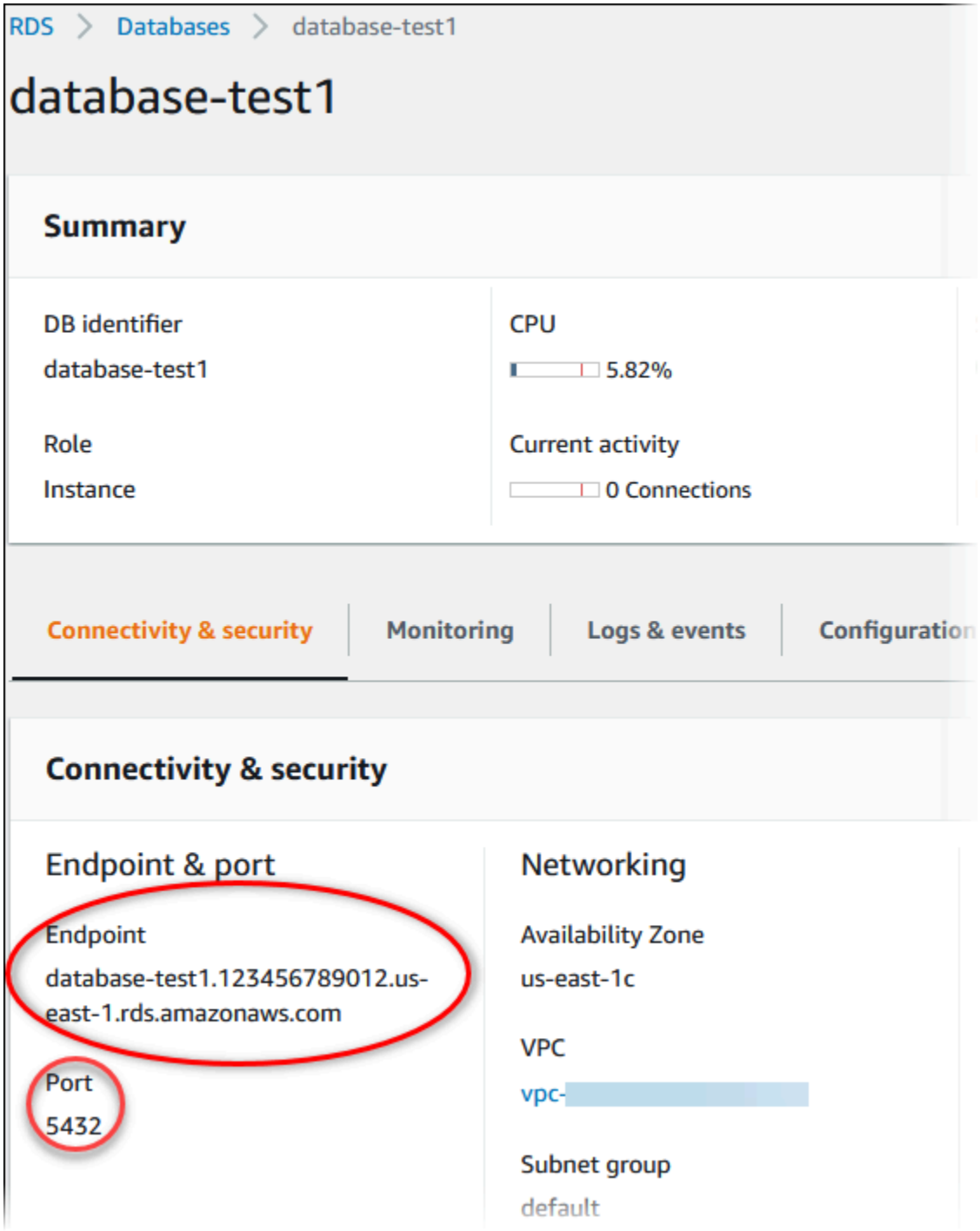
Se a instância de banco de dados estiver disponível e acessível, você poderá se conectar fornecendo as seguintes informações à aplicação cliente SQL:

- O endpoint da instância de banco de dados, que serve como o nome do host (nome DNS) da instância.
- A porta na qual a instância de banco de dados está ouvindo. Para o PostgreSQL, a porta padrão é 5432.
- O nome e senha do usuário da instância de banco de dados. O "nome de usuário primário" padrão do PostgreSQL é `postgres`.
- O nome e a senha do banco de dados (nome do banco de dados).

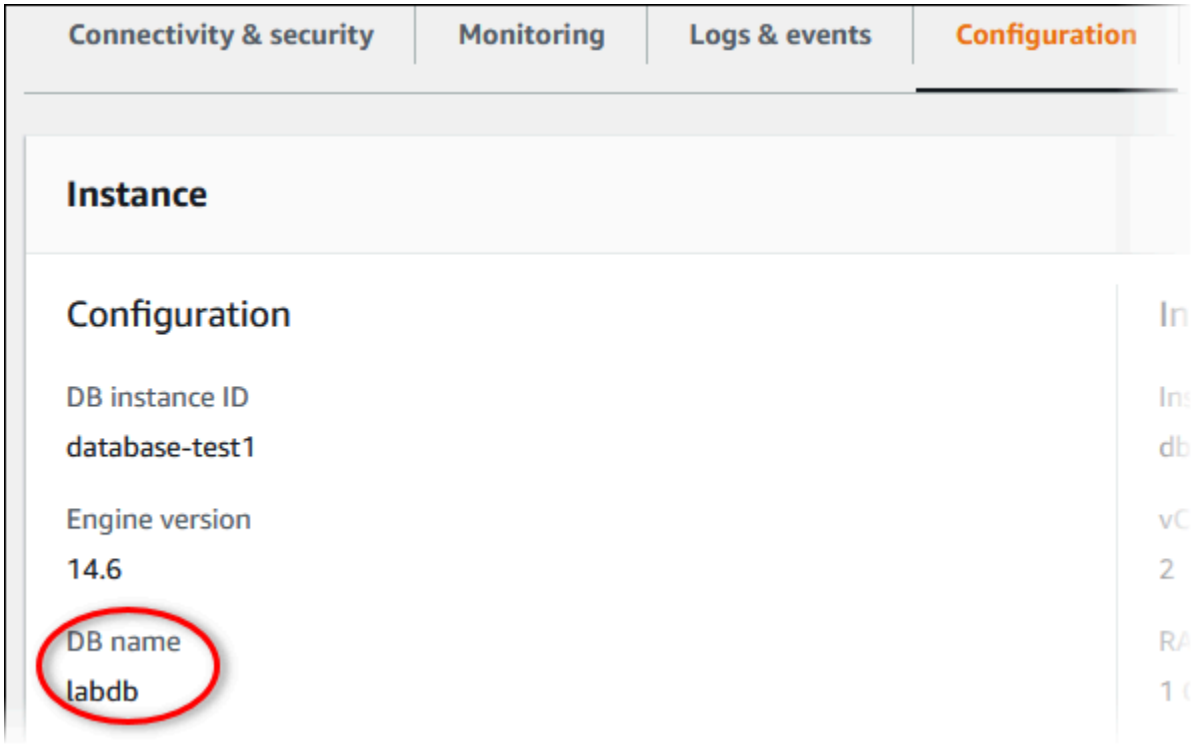
Você pode obter esses detalhes usando o comando AWS Management Console, o comando describe-db-instances da AWS CLI ou a operação DescribeDBInstances da API do Amazon RDS.

Para localizar o endpoint, o número da porta e o nome do banco de dados utilizando o AWS Management Console

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. Abra o console do RDS e escolha Databases (Bancos de dados) para exibir uma lista das instâncias de banco de dados.
3. Escolha o nome da instância de banco de dados do PostgreSQL para exibir os detalhes.
4. Na guia Connectivity & security (Conectividade e segurança), copie o endpoint. Além disso, anote o número da porta. Você precisará do endpoint e do número da porta para conectar-se à instância de banco de dados.



5. Na guia Configuration (Configuração), anote o nome do banco de dados. Se você tiver criado um banco de dados quando criou a instância do RDS for PostgreSQL, verá o nome listado abaixo do nome do banco de dados. Se você não tiver criado um banco de dados, o nome do banco de dados exibirá um traço (-).



A seguir, veja dois jeitos de se conectar a uma instância de banco de dados PostgreSQL. O primeiro exemplo usa o pgAdmin, uma popular ferramenta de administração e desenvolvimento de código aberto para o PostgreSQL. O segundo exemplo usa psql, um utilitário de linha de comando que faz parte de uma instalação do PostgreSQL.

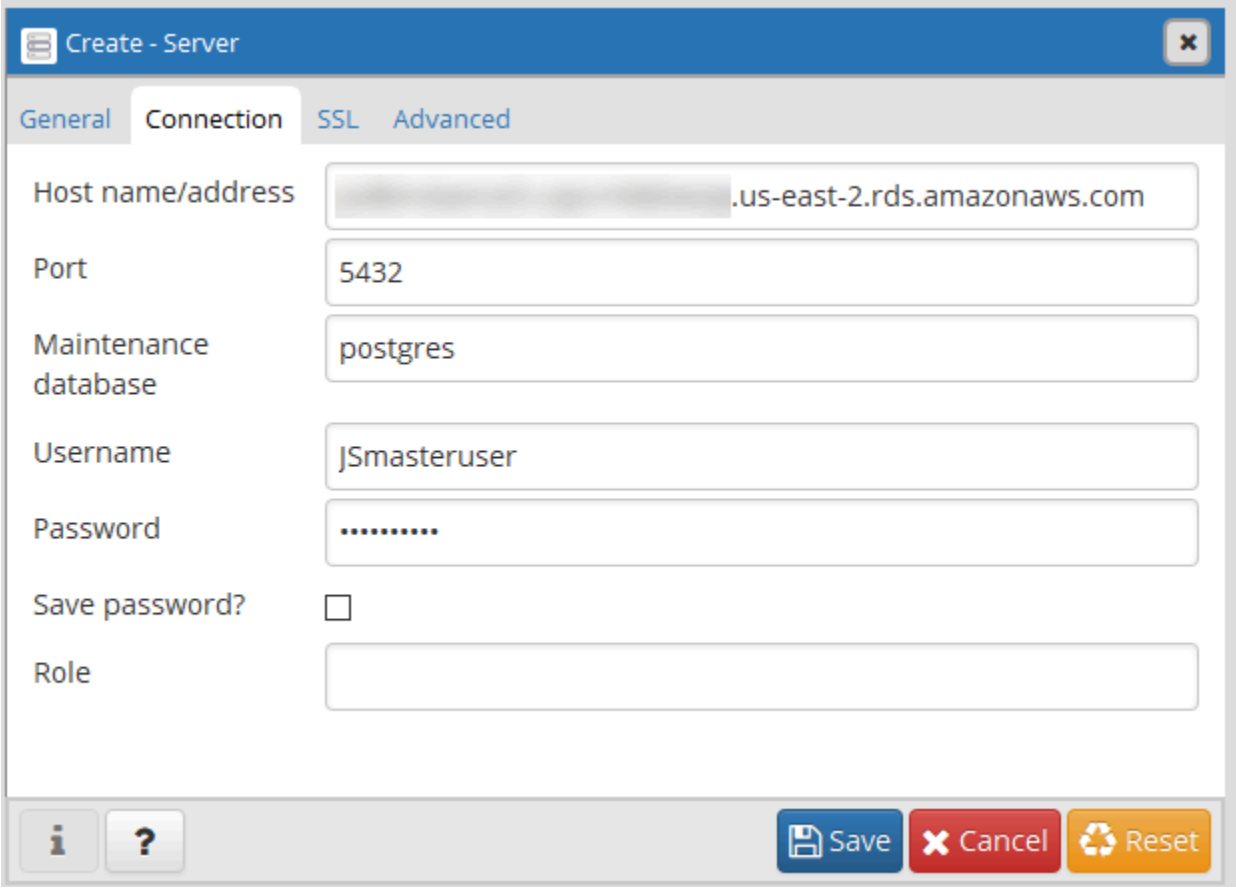
Usar pgAdmin para se conectar a uma instância de banco de dados RDS for PostgreSQL

É possível utilizar a ferramenta de código aberto pgAdmin para se conectar a uma instância de banco de dados RDS for PostgreSQL. Você pode baixar e instalar pgAdmin em http://www.pgadmin.org/ sem ter uma instância local do PostgreSQL no seu computador cliente.

Para se conectar a uma instância de banco de dados RDS for PostgreSQL utilizando pgAdmin

1. Inicie a aplicação pgAdmin no computador cliente.
2. Na guia Dashboard (Painel), escolha Add New Server (Adicionar novo servidor).
3. Na caixa de diálogo Create - Server (Criar - Servidor), digite um nome na guia Geral para identificar o servidor no pgAdmin.
4. Na guia Connection (Conexão), digite as informações a seguir de sua instância de banco de dados:

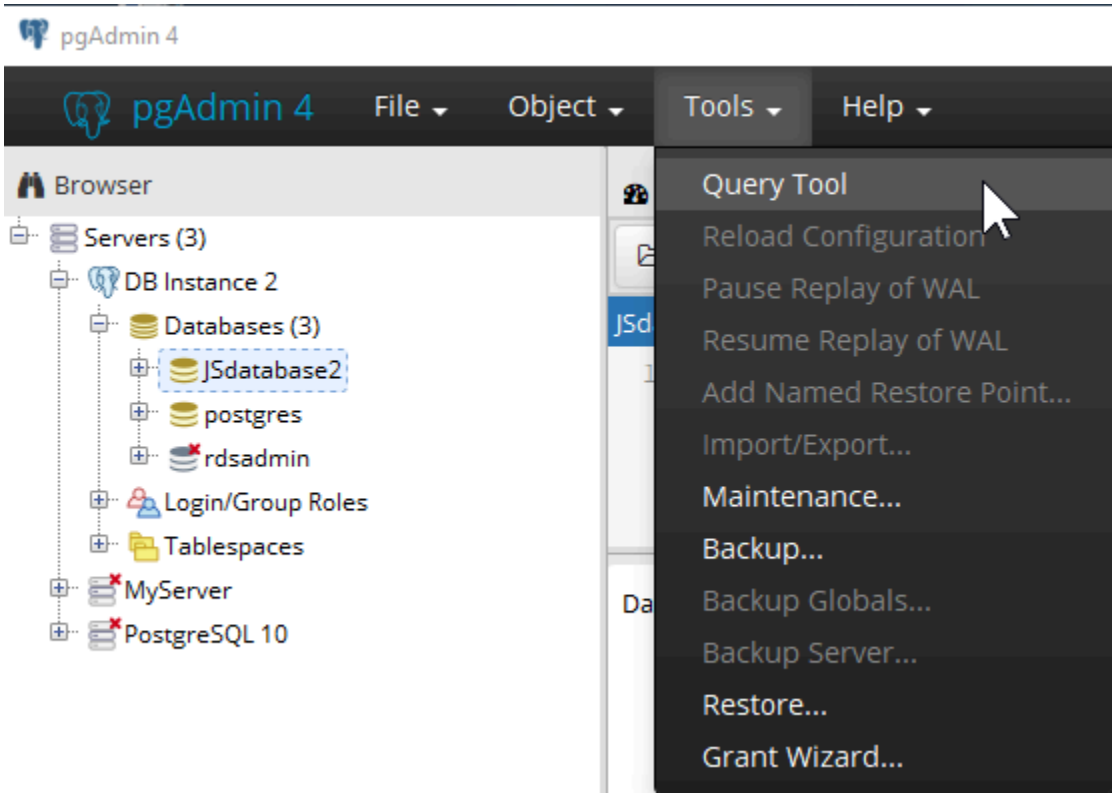
- Para Host, digite o endpoint, por exemplo `mypostgresql.c6c8dntfzzhgv0.us-east-2.rds.amazonaws.com`.
- Para Port (Porta), digite a porta designada.
- Em Username (Nome de usuário), digite o nome de usuário que você inseriu ao criar a instância de banco de dados (se você alterou o padrão do "nome de usuário primário", `postgres`).
- Para Password (Senha), digite a senha que você inseriu quando criou a instância de base de dados.



5. Escolha Salvar.

   Se você tiver quaisquer problemas para se conectar, consulte Solucionar problemas de conexões com a sua instância RDS for PostgreSQL.

6. Para acessar um banco de dados no navegador de pgAdmin, expanda Servers (Servidores), a instância de banco de dados, e Databases (Bancos de dados). Escolha o nome do banco de dados da instância de banco de dados.



7. Para abrir um painel onde você pode inserir comandos SQL, escolha Tools (Ferramentas), Query Tool (Ferramenta de consulta).

Usar psql para se conectar à sua instância de banco de dados RDS for PostgreSQL

É possível utilizar uma instância local do utilitário de linha de comando psql para conectar-se a uma instância de banco de dados RDS for PostgreSQL. Você precisa do PostgreSQL ou do cliente psql instalado em seu computador de cliente.

Você pode baixar o cliente PostgreSQL no site do PostgreSQL. Siga as instruções específicas da versão de seu sistema operacional para instalar o psql.

Para se conectar à sua instância de banco de dados RDS for PostgreSQL utilizando psql, você precisa fornecer as informações do host (DNS), as credenciais de acesso e o nome do banco de dados.

Use um dos formatos a seguir para se conectar a uma instância de banco de dados RDS for PostgreSQL. Ao se conectar, você terá de inserir uma senha. Para trabalhos ou scripts de lotes, use a opção `--no-password`. Essa opção é definida para toda a sessão.

> **Note**
>
> Uma tentativa de conexão com `--no-password` falha quando o servidor requer autenticação de senha e uma senha não está disponível de outras fontes. Para ter mais informações, consulte a documentação do psql.

Se esta é a primeira vez que você está se conectando a essa instância de banco de dados, ou se você ainda não criou um banco de dados para essa instância RDS para PostgreSQL, será possível conectar-se ao banco de dados Postgres utilizando o “nome de usuário primário” e a senha.

Para Unix, use o formato a seguir.

```
psql \
   --host=<DB instance endpoint> \
   --port=<port> \
   --username=<master username> \
   --password \
   --dbname=<database name>
```

Para Windows, use o formato a seguir.

```
psql ^
   --host=<DB instance endpoint> ^
   --port=<port> ^
   --username=<master username> ^
   --password ^
   --dbname=<database name>
```

Por exemplo, o comando a seguir se conecta a um banco de dados chamado `mypgdb` em uma instância de banco de dados PostgreSQL chamada `mypostgresql` usando credenciais fictícias.

```
psql --host=mypostgresql.c6c8mwvfdgv0.us-west-2.rds.amazonaws.com --port=5432 --
username=awsuser --password --dbname=mypgdb
```

Conectar-se ao RDS para PostgreSQL com o driver JDBC da Amazon Web Services (AWS)

O driver JDBC da Amazon Web Services (AWS) foi projetado como um wrapper JDBC avançado. Esse wrapper é complementar e amplia a funcionalidade do driver JDBC existente. O driver é compatível de forma intercambiável com o driver pgJDBC.

Para instalar o driver JDBC da AWS, anexe o arquivo .jar do driver JDBC da AWS (localizado na aplicação `CLASSPATH`) e mantenha referências ao respectivo driver da comunidade. Atualize o respectivo prefixo do URL de conexão da seguinte forma:

- `jdbc:postgresql://` para `jdbc:aws-wrapper:postgresql://`

Consulte mais informações sobre o driver JDBC da AWS e siga as instruções para usá-lo em Amazon Web Services (AWS) JDBC Driver GitHub repository.

Conectar-se ao RDS para PostgreSQL com o driver Python da Amazon Web Services (AWS)

O driver Python da Amazon Web Services (AWS) foi projetado como um wrapper Python avançado. Esse wrapper é complementar e amplia a funcionalidade do driver Psycopg de código aberto. O driver Python da AWS é compatível com as versões 3.8 e posterior do Python. É possível instalar o pacote `aws-advanced-python-wrapper` usando o comando `pip`, bem como os pacotes de código aberto `psycopg`.

Para ter mais informações sobre o driver Python da AWS e instruções completas para usá-lo, consulte Amazon Web Services (AWS) Python Driver GitHub repository.

Solucionar problemas de conexões com a sua instância RDS for PostgreSQL

Tópicos

- Erro – FATAL: o banco de dados nome não existe
- Erro – Não foi possível conectar-se ao servidor: a conexão expirou por exceder o tempo limite.
- Erros com regras de acesso a grupos de segurança

Erro – FATAL: o banco de dados *nome* não existe

Se você receber um erro como `FATAL: database name does not exist` ao tentar se conectar, tente usar o nome padrão do banco de dados postgres para a opção `--dbname`.

Erro – Não foi possível conectar-se ao servidor: a conexão expirou por exceder o tempo limite.

Se você não puder conectar-se à instância do banco de dados, o erro mais comum é `Could not connect to server: Connection timed out.` Se receber esse erro, verifique o seguinte:

- Verifique se o nome do host usado é o endpoint da instância de banco de dados e se o número da porta usado está correto.
- Certifique-se de que a acessibilidade pública da instância de banco de dados esteja definida como Yes (Sim) para permitir conexões externas. Para modificar a configuração Public access (Acesso público), consulte Modificar uma instância de banco de dados do Amazon RDS.
- Garante que o usuário que está se conectando ao banco de dados tenha acesso CONNECT a ele. Você pode usar a consulta a seguir para fornecer acesso de conexão ao banco de dados.

```
GRANT CONNECT ON DATABASE database name TO username;
```

- Verifique se o grupo de segurança atribuído à instância do banco de dados tem as regras para permitir o acesso através de qualquer firewall que sua conexão possa passar. Por exemplo, se a instância de banco de dados foi criada usando a porta padrão 5432, sua empresa pode empregar regras de firewall bloqueando conexões a essa porta a partir de dispositivos externos da empresa.

  Para corrigir isso, modifique a instância de banco de dados para usar uma porta diferente. Além disso, certifique-se de que o grupo de segurança aplicado à instância do banco de dados permite conexões com a nova porta. Para modificar a configuração Database port (Porta do banco de dados), consulte Modificar uma instância de banco de dados do Amazon RDS.
- Verifique se a porta que você está tentando usar já está ocupada por uma instância local do PostgreSQL ou por outro serviço em execução no computador. Por exemplo, se você tiver um banco de dados do PostgreSQL local em execução na mesma porta (o padrão é 5432), isso poderá impedir uma conexão bem-sucedida com a instância de banco de dados do RDS para PostgreSQL. Certifique-se de que a porta esteja livre ou tente se conectar com um número de porta diferente, se possível.
- Consulte também Erros com regras de acesso a grupos de segurança.

Erros com regras de acesso a grupos de segurança

O problema de comunicação mais comum é com as regras de acesso de grupo de segurança atribuídas à instância do banco de dados. Se você usou o grupo de segurança padrão quando criou a instância de banco de dados, é provável que o grupo de segurança não tenha as regras que lhe permitem acessar a instância.

Para a conexão funcionar, o grupo de segurança que você atribuiu à instância de banco de dados em sua criação quando o criou deve permitir o acesso à instância de banco de dados. Por exemplo, se a instância de banco de dados foi criada em uma VPC, ela deverá ter um grupo de segurança da VPC que autorize as conexões. Verifique se a instância de banco de dados foi criada usando um grupo de segurança que não autoriza conexões do dispositivo ou da instância do Amazon EC2 em que o aplicativo está sendo executado.

É possível adicionar ou editar uma regra de entrada no grupo de segurança. Em Source (Origem), selecionar My IP (Meu IP) concede acesso à instância de banco de dados do endereço IP detectado no navegador. Para ter mais informações, consulte Fornecer acesso à instância de banco de dados na VPC criando um grupo de segurança.

De forma alternativa, se a instância de banco de dados tiver sido criada fora de uma VPC, ela deverá ter um grupo de segurança de banco de dados que autorize essas conexões.

Para ter mais informações sobre grupos de segurança do Amazon RDS, consulte Controlar acesso com grupos de segurança.

Proteger conexões com o RDS para PostgreSQL com SSL/TLS

O RDS para PostgreSQL é compatível com a criptografia Secure Sockets Layer (SSL) para instâncias de banco de dados PostgreSQL. Por meio do SSL, você pode criptografar uma conexão do PostgreSQL entre seus aplicativos e suas instâncias de banco de dados PostgreSQL. Você também pode forçar todas as conexões à instância de banco de dados PostgreSQL a usarem SSL. O RDS para PostgreSQL também é compatível com Transport Layer Security (TLS), o protocolo sucessor do SSL.

Para saber mais sobre o Amazon RDS e a proteção de dados, incluindo criptografia de conexões usando SSL/TLS, consulte Proteção de dados no Amazon RDS.

Tópicos

- Usar o SSL com uma instância de banco de dados PostgreSQL
- Atualizar aplicações para conexão com instâncias de banco de dados PostgreSQL usando novos certificados SSL/TLS

Usar o SSL com uma instância de banco de dados PostgreSQL

O Amazon RDS oferece suporte à criptografia de Secure Socket Layer (SSL) para instâncias de banco de dados PostgreSQL. Por meio do SSL, você pode criptografar uma conexão do PostgreSQL entre seus aplicativos e suas instâncias de banco de dados PostgreSQL. Por padrão, o RDS para PostgreSQL usa e espera que todos os clientes se conectem usando SSL/TLS, mas você também pode exigir isso. O RDS para PostgreSQL é compatível com o Transport Layer Security (TLS) versões 1.1, 1.2 e 1.3.

Para obter informações gerais sobre o suporte para SSL e bancos de dados PostgreSQL, consulte Suporte para SSL na documentação do PostgreSQL. Para obter informações sobre como usar uma conexão SSL via JDBC, consulte Configurar o cliente na documentação do PostgreSQL.

O suporte a SSL está disponível em todas as regiões da AWS para o PostgreSQL. O Amazon RDS cria um certificado SSL para sua instância de banco de dados do PostgreSQL quando essa instância é criada. Se você habilitar a verificação de certificado SSL, o certificado SSL incluirá o endpoint da instância de banco de dados como o nome comum (CN) do certificado SSL para se proteger contra ataques de falsificação.

Tópicos

- Conectar-se à instância de banco de dados PostgreSQL via SSL
- Exigir uma conexão SSL para uma instância de banco de dados PostgreSQL
- Determinar o status de conexão SSL
- Suítes de cifras SSL no RDS para PostgreSQL

Conectar-se à instância de banco de dados PostgreSQL via SSL

Para se conectar à instância de banco de dados PostgreSQL via SSL

1. Baixe o certificado.

   Para obter informações sobre como baixar certificados, consulte Usar SSL/TLS para criptografar uma conexão com uma instância ou um cluster de banco de dados.
2. Conecte-se à sua instância de banco de dados PostgreSQL via SSL.

   Quando você se conecta usando SSL, seu cliente pode escolher se deseja verificar a cadeia de certificados. Se os seus parâmetros de conexão especificarem `sslmode=verify-ca` ou `sslmode=verify-full`, seu cliente exigirá que os certificados de CA do RDS estejam no armazenamento confiável ou sejam referenciados no URL da conexão. Esse requisito tem o objetivo de verificar a cadeia de certificados que assina o seu certificado de banco de dados.

   Quando um cliente, como psql ou JDBC, é configurado com suporte para SSL, primeiro ele tenta se conectar ao banco de dados via SSL por padrão. Se esse cliente não puder se conectar via SSL, ele voltará a se conectar sem SSL. O modo `sslmode` padrão utilizado é diferente entre os clientes baseados em libpq (como o psql) e o JDBC. Os clientes JDBC e baseados em libpq usam `prefer` como padrão.

   Use o parâmetro `sslrootcert` para fazer referência ao certificado, por exemplo `sslrootcert=rds-ssl-ca-cert.pem`.

Veja a seguir um exemplo de uso do `psql` para se conectar a uma instância de banco de dados do PostgreSQL usando SSL com verificação de certificado.

```
$ psql "host=db-name.555555555555.ap-southeast-1.rds.amazonaws.com
    port=5432 dbname=testDB user=testuser sslrootcert=rds-ca-rsa2048-g1.pem
 sslmode=verify-full"
```

Exigir uma conexão SSL para uma instância de banco de dados PostgreSQL

Você pode exigir que conexões estabelecidas a sua instância de banco de dados PostgreSQL usem SSL por meio do parâmetro `rds.force_ssl`. O valor padrão do parâmetro `rds.force_ssl` é 1 (ativado) para o RDS para PostgreSQL versão 15 e posterior. Para todas as outras versões principais do RDS para PostgreSQL 14 e posterior, o valor padrão desse parâmetro é 0 (desativado). Você pode definir o parâmetro `rds.force_ssl` como 1 (ativado) para exigir SSL/TLS para conexões com o seu cluster de banco de dados. Você pode definir o parâmetro `rds.force_ssl` como 1 (ativado) para exigir o SSL para conexões a sua instância de banco de dados.

Para alterar o valor desse parâmetro, você precisa criar um grupo de parâmetros de banco de dados personalizado. Depois precisará alterar o valor de `rds.force_ssl` em seu grupo de parâmetros de banco de dados personalizado para 1 para ativar esse recurso. Se você preparar o grupo de parâmetros de banco de dados personalizado antes de criar sua instância de banco de dados do RDS para PostgreSQL, poderá escolhê-lo (em vez de um grupo de parâmetros padrão) durante o processo de criação. Se você fizer isso depois que a instância de banco de dados do RDS para PostgreSQL já estiver em execução, será necessário reinicializar a instância para que ela use o grupo de parâmetros personalizado. Para obter mais informações, consulte Grupos de parâmetros para Amazon RDS.

Quando o recurso `rds.force_ssl` está ativo em sua instância de banco de dados, as tentativas de conexão que não estão usando SSL são rejeitadas com a seguinte mensagem:

```
$ psql -h db-name.555555555555.ap-southeast-1.rds.amazonaws.com port=5432 dbname=testDB
 user=testuser
psql: error: FATAL: no pg_hba.conf entry for host "w.x.y.z", user "testuser", database
 "testDB", SSL off
```

Determinar o status de conexão SSL

O status criptografado da conexão é mostrado no banner de login ao se conectar à instância de banco de dados:

```
Password for user master:
psql (10.3)
SSL connection (cipher: DHE-RSA-AES256-SHA, bits: 256)
Type "help" for help.
postgres=>
```

Você também pode carregar a extensão `sslinfo` e chamar a função `ssl_is_used()` para determinar se o SSL está sendo usado. A função retornará `t` se a conexão estiver usando o SSL, caso contrário, retornará `f`.

```
postgres=> CREATE EXTENSION sslinfo;
CREATE EXTENSION
postgres=> SELECT ssl_is_used();
ssl_is_used
---------
t
(1 row)
```

Para obter informações mais detalhadas, use a seguinte consulta para obter informações de `pg_settings`:

```
SELECT name as "Parameter name", setting as value, short_desc FROM pg_settings WHERE
 name LIKE '%ssl%';
             Parameter name             |                  value                  |
                  short_desc
----------------------------------------+-----------------------------------------
+-----------------------------------------------------
 ssl                                    | on                                      |
 Enables SSL connections.
 ssl_ca_file                            | /rdsdbdata/rds-metadata/ca-cert.pem     |
 Location of the SSL certificate authority file.
 ssl_cert_file                          | /rdsdbdata/rds-metadata/server-cert.pem |
 Location of the SSL server certificate file.
 ssl_ciphers                            | HIGH:!aNULL:!3DES                       |
 Sets the list of allowed SSL ciphers.
 ssl_crl_file                           |                                         |
 Location of the SSL certificate revocation list file.
 ssl_dh_params_file                     |                                         |
 Location of the SSL DH parameters file.
 ssl_ecdh_curve                         | prime256v1                              |
 Sets the curve to use for ECDH.
 ssl_key_file                           | /rdsdbdata/rds-metadata/server-key.pem  |
 Location of the SSL server private key file.
 ssl_library                            | OpenSSL                                 |
 Name of the SSL library.
 ssl_max_protocol_version               |                                         |
 Sets the maximum SSL/TLS protocol version to use.
```

```
 ssl_min_protocol_version               | TLSv1.2                                 |
 Sets the minimum SSL/TLS protocol version to use.
 ssl_passphrase_command                 |                                         |
 Command to obtain passphrases for SSL.
 ssl_passphrase_command_supports_reload | off                                     |
 Also use ssl_passphrase_command during server reload.
 ssl_prefer_server_ciphers              | on                                      |
 Give priority to server ciphersuite order.
(14 rows)
```

Você também pode coletar todas as informações sobre o uso de SSL da instância de banco de dados do RDS para PostgreSQL por processo, cliente e aplicação usando a seguinte consulta:

```
SELECT datname as "Database name", usename as "User name", ssl, client_addr,
 application_name, backend_type
   FROM pg_stat_ssl
   JOIN pg_stat_activity
   ON pg_stat_ssl.pid = pg_stat_activity.pid
   ORDER BY ssl;
 Database name | User name | ssl |  client_addr   |    application_name    |
 backend_type
---------------+-----------+-----+----------------+------------------------
+------------------------------
               |           | f   |                |                        | autovacuum
 launcher
               | rdsadmin  | f   |                |                        | logical
 replication launcher
               |           | f   |                |                        | background
 writer
               |           | f   |                |                        |
 checkpointer
               |           | f   |                |                        | walwriter
 rdsadmin      | rdsadmin  | t   | 127.0.0.1      |                        | client
 backend
 rdsadmin      | rdsadmin  | t   | 127.0.0.1      | PostgreSQL JDBC Driver | client
 backend
 postgres      | postgres  | t   | 204.246.162.36 | psql                   | client
 backend
(8 rows)
```

Para identificar a criptografia usada para sua conexão SSL, você pode consultar da seguinte forma:

```
postgres=> SELECT ssl_cipher();
```

```
ssl_cipher
--------------------
DHE-RSA-AES256-SHA
(1 row)
```

Para saber mais sobre a opção `sslmode`, consulte Funções de controle de conexão com o banco de dados na documentação do PostgreSQL.

Suítes de cifras SSL no RDS para PostgreSQL

O parâmetro de configuração do PostgreSQL, ssl_ciphers, especifica as categorias de suítes de cifras permitidas para conexões SSL com o banco de dados ao usar o TLS 1.2 e anterior.

No RDS para PostgreSQL 16 e versões posteriores, você pode modificar o parâmetro `ssl_ciphers` para usar valores específicos das suítes de cifras permitidas. Esse é um parâmetro dinâmico que não exige a reinicialização da instância do banco de dados. Para visualizar as suítes de cifras permitidas, use o console do Amazon RDS ou o seguinte comando da AWS CLI:

```
aws rds describe-db-parameters --db-parameter-group-name <your-parameter-group> --
region <region> --endpoint-url <endpoint-url> --output json | jq '.Parameters[] |
 select(.ParameterName == "ssl_ciphers")'
```

A tabela a seguir lista as suítes de cifras padrão e as suítes de cifras permitidas para versões que aceitam configurações personalizadas.

Versão do mecanismo PostgreSQL	Valores padrão da suíte de ssl_ciphers	Valores personalizados permitidos da suíte de ssl_ciphers

17	`HIGH:!aNULL:!3DES`	`TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384`
		`TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256`
		`TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256`

Versão do mecanismo PostgreSQL	Valores padrão da suíte de ssl_ciphers	Valores personalizados permitidos da suíte de ssl_ciphers
		TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
16	HIGH:!aNULL:!3DES	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
15	HIGH:!aNULL:!3DES	Não é possível usar ssl_ciphers personalizados
14	HIGH:!aNULL:!3DES	Não é possível usar ssl_ciphers personalizados
13	HIGH:!aNULL:!3DES	Não é possível usar ssl_ciphers personalizados

Versão do mecanismo PostgreSQL	Valores padrão da suíte de ssl_ciphers	Valores personalizados permitidos da suíte de ssl_ciphers
12	`HIGH:!aNULL:!3DES`	Não é possível usar ssl_ciphers personalizados
11.4 e versões secundárias posteriores	`HIGH:MEDIUM:+3DES:!aNULL:!RC4`	Não é possível usar ssl_ciphers personalizados
11.1, 11.2	`HIGH:MEDIUM:+3DES:!aNULL`	Não é possível usar ssl_ciphers personalizados
10.9 e versões secundárias posteriores	`HIGH:MEDIUM:+3DES:!aNULL:!RC4`	Não é possível usar ssl_ciphers personalizados
10.7 e versões secundárias anteriores	`HIGH:MEDIUM:+3DES:!aNULL`	Não é possível usar ssl_ciphers personalizados

Para configurar todas as conexões de instância e usar a suíte de cifras `TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384`, modifique seu grupo de parâmetros conforme mostrado no seguinte exemplo:

```
aws rds modify-db-parameter-group --db-parameter-group-name <your-parameter-group> --
parameters
 "ParameterName='ssl_ciphers',ParameterValue='TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384',ApplyMet
```

Este exemplo usa uma cifra ECDSA, que exige que sua instância use uma autoridade de certificação com criptografia de curva elíptica (ECC) para estabelecer uma conexão. Para obter informações sobre as autoridades de certificação oferecidas pelo Amazon RDS, consulte Certificate authorities.

Você pode verificar as cifras em uso por meio dos métodos descritos em Determinar o status de conexão SSL.

As cifras podem ter nomes diferentes dependendo do contexto:

- As cifras da lista de permissões que você pode configurar em seu grupo de parâmetros são referidas com o respectivo nome da Internet Assigned Numbers Authority (IANA).

- Os banners de logon `sslinfo` e `psql` se referem às cifras usando o respectivo nome do OpenSSL.

Por padrão, o valor de `ssl_max_protocol_version` no RDS para PostgreSQL 16 e versões posteriores é TLS v1.3. Você deve definir o valor desse parâmetro como TLS v1.2, pois o TLS v1.3 não usa as configurações de cifra especificadas no parâmetro `ssl_ciphers`. Quando você define o valor como TLS v1.2, as conexões usam somente as cifras que você define em `ssl_ciphers`.

```
aws rds modify-db-parameter-group --db-parameter-group-name <your-parameter-group> --
parameters
 "ParameterName='ssl_max_protocol_version',ParameterValue='TLSv1.2',ApplyMethod=immediate"
```

Para garantir que as conexões do banco de dados usem SSL, defina `rds.force_ssl parameter` como 1 no seu grupo de parâmetros. Para obter mais informações sobre parâmetros e grupos de parâmetros, consulte Grupos de parâmetros para Amazon RDS.

Atualizar aplicações para conexão com instâncias de banco de dados PostgreSQL usando novos certificados SSL/TLS

Os certificados usados para Secure Socket Layer ou TLS Transport Layer Security (SSL/TLS) geralmente têm uma vida útil definida. Quando os provedores de serviços atualizam seus certificados de Autoridade de Certificação (CA), os clientes devem atualizar suas aplicações para usar os novos certificados. A seguir, você pode encontrar informações sobre como determinar se suas aplicações cliente usam SSL/TLS para conectar-se à sua instância de banco de dados Amazon RDS for PostgreSQL. Você também pode encontrar informações sobre como conferir se essas aplicações verificam o certificado do servidor quando elas se conectam.

> **Note**
>
> Uma aplicação cliente configurada para verificar o certificado do servidor antes da conexão SSL/TLS deve ter um certificado válido de uma CA no repositório de confiança do cliente. Atualize o repositório de confiança do cliente quando houver novos certificados.

Depois de atualizar seus certificados de CA nos armazenamentos confiáveis do aplicativo cliente, você pode fazer o rodízio dos certificados nas suas instâncias de banco de dados. É altamente recomendável testar esses procedimentos em um ambiente não voltado à produção antes de implementá-los em seus ambientes de produção.

Para ter mais informações sobre a mudança de certificados, consulte Alternar o certificado SSL/TLS. Para ter mais informações sobre como fazer download de certificados, consulte Usar SSL/TLS para criptografar uma conexão com uma instância ou um cluster de banco de dados. Para obter informações sobre o uso de SSL/TLS com instâncias de bancos de dados PostgreSQL, consulte Usar o SSL com uma instância de banco de dados PostgreSQL.

Tópicos

- Determinar se as aplicações estão se conectando a instâncias de banco de dados PostgreSQL usando SSL
- Determinar se um cliente requer verificação de certificado para se conectar
- Atualizar o armazenamento confiável de aplicações
- Usar conexões SSL/TLS para diferentes tipos de aplicações

Determinar se as aplicações estão se conectando a instâncias de banco de dados PostgreSQL usando SSL

Verifique a configuração da instância de banco de dados para obter o valor do parâmetro `rds.force_ssl`. Por padrão, o parâmetro `rds.force_ssl` é definido como `0` (desativado) para instâncias de banco de dados usando versões do PostgreSQL anteriores à 15. Por padrão, `rds.force_ssl` é definido como 1 (ativo) para instâncias de banco de dados usando o PostgreSQL versão 15 e versões principais posteriores. Se o parâmetro `rds.force_ssl` estiver definido como 1 (ativado), os clientes serão obrigados a usar SSL/TLS para conexões. Para ter mais informações sobre parameter groups, consulte Grupos de parâmetros para Amazon RDS.

Se você estiver usando o RDS PostgreSQL versão 9.5 ou versão posterior principal e `rds.force_ssl` não estiver definido como 1 (ativado), consulte a visualização `pg_stat_ssl` para verificar conexões usando SSL. Por exemplo, a consulta a seguir retorna apenas conexões SSL e informações sobre os clientes que usam SSL.

```
SELECT datname, usename, ssl, client_addr
  FROM pg_stat_ssl INNER JOIN pg_stat_activity ON pg_stat_ssl.pid =
 pg_stat_activity.pid
  WHERE ssl is true and usename<>'rdsadmin';
```

Somente as linhas que usarem conexões SSL/TLS serão exibidas com informações sobre a conexão. Esta é uma saída de exemplo.

```
 datname  | username | ssl | client_addr
----------+---------+-----+-------------
 benchdb  | pgadmin | t   | 53.95.6.13
 postgres | pgadmin | t   | 53.95.6.13
(2 rows)
```

Esta consulta exibe apenas as conexões atuais no momento da consulta. A ausência de resultados não indica que nenhum aplicativo esteja usando conexões SSL. Outras conexões SSL podem ser estabelecidas em um momento diferente.

Determinar se um cliente requer verificação de certificado para se conectar

Quando um cliente, como psql ou JDBC, é configurado com suporte para SSL, primeiro ele tenta se conectar ao banco de dados via SSL por padrão. Se esse cliente não puder se conectar via SSL, ele voltará a se conectar sem SSL. O modo padrão `sslmode` utilizado para clientes baseados em libpq (como o psql) e o JDBC é definido como `prefer`. O certificado no servidor é verificado apenas quando `sslrootcert` é fornecido com `sslmode` definido como `verify-ca` ou `verify-full`. Um erro será lançado se o certificado for inválido.

Use `PGSSLROOTCERT` para verificar o certificado com a variável de ambiente `PGSSLMODE`, com `PGSSLMODE` definido como `verify-ca` ou `verify-full`.

```
PGSSLMODE=verify-full PGSSLROOTCERT=/fullpath/ssl-cert.pem psql -h
 pgdbidentifier.cxxxxxxxx.us-east-2.rds.amazonaws.com -U masteruser -d postgres
```

Use o argumento `sslrootcert` para verificar o certificado com `sslmode` no formato de string de conexão, com `sslmode` definido como `verify-ca` ou `verify-full` para verificar o certificado.

```
psql "host=pgdbidentifier.cxxxxxxxx.us-east-2.rds.amazonaws.com sslmode=verify-full
 sslrootcert=/full/path/ssl-cert.pem user=masteruser dbname=postgres"
```

Por exemplo, no caso anterior, se você estiver usando um certificado raiz inválido, verá um erro semelhante ao seguinte no seu cliente.

```
psql: SSL error: certificate verify failed
```

Atualizar o armazenamento confiável de aplicações

Para obter informações sobre como atualizar o armazenamento de confiança para aplicações PostgreSQL, consulte Secure TCP/IP connections with SSL na documentação do PostgreSQL.

Para obter informações sobre como baixar o certificado raiz, consulte Usar SSL/TLS para criptografar uma conexão com uma instância ou um cluster de banco de dados.

Para obter exemplos de scripts que importam certificados, consulte Script de exemplo para importar certificados para o seu armazenamento confiável.

> **Note**
>
> Ao atualizar o armazenamento confiável, é possível reter certificados mais antigos, além de adicionar os novos certificados.

Usar conexões SSL/TLS para diferentes tipos de aplicações

Veja a seguir informações sobre o uso de conexões SSL/TLS para diferentes tipos de aplicativos:

- psql

  O cliente é chamado da linha de comandos especificando opções como uma string de conexão ou como variáveis de ambiente. Para conexões SSL/TLS, as opções relevantes são `sslmode` (variável de ambiente `PGSSLMODE`), `sslrootcert` (variável de ambiente `PGSSLROOTCERT`).

  Para conhecer a lista completa de opções, consulte Palavras-chave de parâmetros na documentação do PostgreSQL. Para conhecer a lista completa de variáveis de ambiente, consulte Variáveis de ambiente na documentação do PostgreSQL.

- pgAdmin

  Esse cliente baseado em navegador é uma interface mais amigável para conectar-se a um banco de dados PostgreSQL.

  Para obter informações sobre como configurar conexões, consulte a Documentação de pgAdmin.

- JDBC

  O JDBC permite conexões de banco de dados com aplicativos Java.

Para obter informações gerais sobre como conectar-se a um banco de dados PostgreSQL com JDBC, consulte Connecting to the database (Conectar-se ao banco de dados) na documentação do driver do PostgreSQL. Para obter informações sobre como conectar-se com SSL/TLS, consulte Configuring the client (Configurar o cliente) na documentação do PostgreSQL.

- Python

  Uma biblioteca Python popular para conectar-se a bancos de dados PostgreSQL é `psycopg2`.

  Para obter informações sobre como usar o `psycopg2`, consulte a documentação de psycopg2. Para um breve tutorial sobre como conectar-se a um banco de dados PostgreSQL, consulte Tutorial sobre psycopg2. Você pode encontrar informações sobre as opções aceitas pelo comando de conexão em O conteúdo do módulo psycopg2.

> **Important**
>
> Depois de determinar que suas conexões com o banco de dados usam SSL/TLS e ter atualizado o armazenamento confiável de aplicações, você poderá atualizar o banco de dados para usar os certificados rds-ca-rsa2048-g1. Para obter instruções, consulte a etapa 3 em Atualizar o certificado CA modificando a instância ou o cluster de banco de dados.

Usar a autenticação Kerberos com o Amazon RDS para PostgreSQL

É possível usar o Kerberos para autenticar usuários quando eles se conectam à sua instância de banco de dados executando o PostgreSQL. Para fazer isso, configure sua instância de banco de dados para usar o AWS Directory Service for Microsoft Active Directory para autenticação Kerberos. O AWS Directory Service for Microsoft Active Directory também é chamado de AWS Managed Microsoft AD. É um recurso disponível com o AWS Directory Service. Para saber mais, consulte What is AWS Directory Service? (O que é o ?) no Guia de administração do AWS Directory Service.

Para iniciar, crie um diretório AWS Managed Microsoft AD para armazenar credenciais de usuário. Depois, forneça à instância de banco de dados PostgreSQL o domínio do Active Directory e outras informações. Quando os usuários são autenticados com a instância de banco de dados PostgreSQL, as solicitações de autenticação são encaminhadas para o diretório AWS Managed Microsoft AD.

Manter todas as suas credenciais no mesmo diretório pode economizar tempo e esforço. Há um lugar centralizado para armazenar e gerenciar credenciais para várias instâncias de banco de dados. O uso de um diretório também pode melhorar o perfil de segurança geral.

Além disso, é possível acessar credenciais de seu próprio Microsoft Active Directory on-premises. Para fazer isso, crie uma relação de domínio confiável para que o diretório AWS Managed Microsoft AD confie no Microsoft Active Directory on-premises. Dessa forma, seus usuários podem acessar as instâncias de do PostgreSQL com a mesma experiência de autenticação única (SSO) do Windows, como quando acessam workloads na sua rede on-premises.

Um banco de dados pode usar autenticação por senha ou autenticação por senha com autenticação do Kerberos ou do AWS Identity and Access Management (IAM). Para ter mais informações sobre a autenticação do IAM, consulte Autenticação do banco de dados do IAM para MariaDB, MySQL e PostgreSQL.

Tópicos

- Disponibilidade de região e versão
- Visão geral da autenticação Kerberos para instâncias de banco de dados PostgreSQL
- Configurar a autenticação Kerberos para instâncias de banco de dados do PostgreSQL
- Gerenciar uma instância de banco de dados do RDS para PostgreSQL em um domínio do Active Directory
- Conectar-se ao PostgreSQL com a autenticação Kerberos

Disponibilidade de região e versão

A disponibilidade e a compatibilidade de recursos variam entre versões específicas de cada mecanismo de banco de dados e entre Regiões da AWS. Para ter mais informações sobre a disponibilidade de versões e regiões do RDS para PostgreSQL com autenticação de Kerberos, consulte Regiões e mecanismos de banco de dados compatíveis com a autenticação Kerberos no Amazon RDS.

Visão geral da autenticação Kerberos para instâncias de banco de dados PostgreSQL

Para configurar a autenticação Kerberos para uma instância de banco de dados PostgreSQL, siga as etapas a seguir, descritas em mais detalhes posteriormente:

1. Use AWS Managed Microsoft AD para criar um diretório do AWS Managed Microsoft AD. É possível usar o AWS Management Console, a AWS CLI ou a API do AWS Directory Service para criar o diretório. Certifique-se de abrir as portas de saída relevantes no grupo de segurança do diretório para que o diretório possa se comunicar com a instância.
2. Crie uma função que forneça ao Amazon RDS acesso para fazer chamadas para o diretório AWS Managed Microsoft AD. Para fazer isso, crie um perfil do AWS Identity and Access Management (IAM) que use a política gerenciada do IAM `AmazonRDSDirectoryServiceAccess`.

   Para o perfil do IAM permitir acesso, o endpoint do AWS Security Token Service (AWS STS) deve estar ativado na região da AWS correta da conta da AWS. Os endpoints do AWS STS são ativados por padrão em todas as Regiões da AWS e é possível usá-los sem ter que tomar medidas adicionais. Para ter mais informações, consulte Ativar e desativar o AWS STS em uma AWS região da no Manual do usuário do IAM.
3. Crie e configure usuários no diretório AWS Managed Microsoft AD usando as ferramentas do Microsoft Active Directory. Para ter mais informações sobre como criar usuários em seu Active Directory, consulte Gerenciar usuários e grupos no Microsoft AD gerenciado pela AWS noGuia de administração doAWS Directory Service.
4. Se você planeja localizar o diretório e a instância de Bancos de Dados em contas da AWS ou nuvens privadas virtuais (VPCs) diferentes, configure o emparelhamento de VPCs. Para ter mais informações, consulte O que é emparelhamento de VPC? no Amazon VPC Peering Guide.
5. Crie ou modifique uma instância de banco de dados PostgreSQL no console, na CLI ou na API do RDS usando um dos seguintes métodos:
   - Criar uma instância de banco de dados do Amazon RDS

- Modificar uma instância de banco de dados do Amazon RDS
- Restaurar uma instância de banco de dados
- Restaurar uma instância de banco de dados para um momento especificado no Amazon RDS

É possível localizar a instância na mesma Amazon Virtual Private Cloud (VPC) que o diretório ou em uma conta da AWS ou VPC diferente. Ao criar ou modificar a instância de banco de dados PostgreSQL, faça o seguinte:

- Forneça o identificador de domínio (identificador `d-*`) que foi gerado quando você criou seu diretório.
- Forneça o nome do perfil do IAM criado.
- Certifique-se de que o grupo de segurança da instância de banco de dados possa receber o tráfego de entrada do grupo de segurança do diretório.

6. Use as credenciais de usuário mestre do RDS para conectar-se à instância de de banco de dados PostgreSQL. Crie o usuário no PostgreSQL para ser identificado externamente. Usuários identificados externamente podem fazer login na instância de banco de dados PostgreSQL usando a autenticação Kerberos.

Configurar a autenticação Kerberos para instâncias de banco de dados do PostgreSQL

Use o AWS Directory Service for Microsoft Active Directory (AWS Managed Microsoft AD) para configurar a autenticação Kerberos para uma instância de banco de dados PostgreSQL. Para configurar a autenticação Kerberos, execute as etapas a seguir.

Tópicos

- Etapa 1: Criar um diretório usando o AWS Managed Microsoft AD
- Etapa 2: (Opcional) Criar uma relação de confiança entre o Active Directory on-premises e o AWS Directory Service
- Etapa 3: Criar um perfil do IAM para que o Amazon RDS acesse o AWS Directory Service
- Etapa 4: Criar e configurar usuários
- Etapa 5: Ativar o tráfego entre VPCs entre o diretório e a instância de banco de dados
- Etapa 6: Criar ou modificar uma instância de banco de dados PostgreSQL
- Etapa 7: Criar usuários do PostgreSQL para suas entidades principais do Kerberos
- Etapa 8: Configurar um cliente PostgreSQL

Etapa 1: Criar um diretório usando o AWS Managed Microsoft AD

O AWS Directory Service cria um Active Directory totalmente gerenciado na Nuvem AWS. Quando você cria um diretório AWS Managed Microsoft AD, o AWS Directory Service cria dois controladores de domínio e servidores DNS para você. Os servidores do diretório são criados em sub-redes diferentes em uma VPC. Essa redundância ajuda a garantir que o diretório permaneça acessível mesmo se ocorrer uma falha.

Ao criar um diretório AWS Managed Microsoft AD, o AWS Directory Service realiza as seguintes tarefas em seu nome:

- Configura um Active Directory dentro da VPC.
- Cria de uma conta de administrador do diretório com o nome de usuário `Admin` e a senha especificada. Use essa conta para gerenciar seu diretório.

  > **Important**
  >
  > Certifique-se de salvar essa senha. O AWS Directory Service não armazena essa senha e não é possível recuperá-la ou redefini-la.

- Cria um grupo de segurança para os controladores do diretório. O grupo de segurança deve permitir a comunicação com a instância de banco de dados PostgreSQL.

Quando você lança o AWS Directory Service for Microsoft Active Directory, a AWS cria uma Unidade organizacional (OU) que contém todos os objetos do diretório. Essa UO, que tem o nome de NetBIOS que você digitou ao criar o diretório, está localizada na raiz do domínio. A raiz do domínio é controlada e de propriedade da AWS.

A conta `Admin`, que foi criada com o diretório AWS Managed Microsoft AD, tem permissões para as atividades administrativas mais comuns da UO:

- Criar, atualizar ou excluir usuários
- Adicionar recursos ao domínio, como servidores de arquivos ou de impressão, e atribuir permissões para esses recursos aos usuários na UO
- Criar OUs adicionais e contêineres
- Delegar autoridade
- Restaurar objetos excluídos da Lixeira do Active Directory

- Execute os módulos Active Directory e Domain Name Service (DNS) para Windows PowerShell no Active Directory Web Service

A conta `Admin` também possui direitos para executar as seguintes atividades de domínio:

- Gerenciar configurações de DNS (adicionar, remover ou atualizar registros, zonas e encaminhadores)
- Visualizar logs de eventos de DNS
- Visualizar logs de eventos de segurança

Como criar um diretório com AWS Managed Microsoft AD

1. No painel de navegação do console do AWS Directory Service, escolha Directories (Diretórios) e escolha Set up directory (Configurar diretório).
2. Escolha AWS Managed Microsoft AD. O AWS Managed Microsoft AD é a única opção atualmente compatível para uso com o Amazon RDS.
3. Escolha Next (Próximo).
4. Na página Enter directory information (Inserir informações do diretório), forneça as seguintes informações:

   Edição

   Escolha a edição que atenda às suas necessidades.

   Nome do DNS do diretório

   O nome completo do diretório, como **`corp.example.com`**.

   Nome de NetBIOS do diretório

   O nome curto opcional do diretório, como `CORP`.

   Descrição do diretório

   Uma descrição opcional do diretório.

   Senha do Admin

   A senha do administrador do diretório. O processo de criação do diretório cria uma conta de administrador com o nome de usuário `Admin` e essa senha.

A senha do administrador do diretório não pode incluir a palavra "admin". A senha diferencia letras maiúsculas de minúsculas e deve ter entre 8 e 64 caracteres. Ela também precisa conter pelo menos um caractere de três das quatro categorias a seguir:

- Letras minúsculas (a–z)
- Letras maiúsculas (A–Z)
- Números (0–9)
- Caracteres não alfanuméricos (~!@#$%^&*_-+=`|\(){}[]:;"'<>,.?/)

Confirmar senha

Digite a senha do administrador novamente.

> **Important**
>
> Salve essa senha. O AWS Directory Service não armazena essa senha e não é possível recuperá-la ou redefini-la.

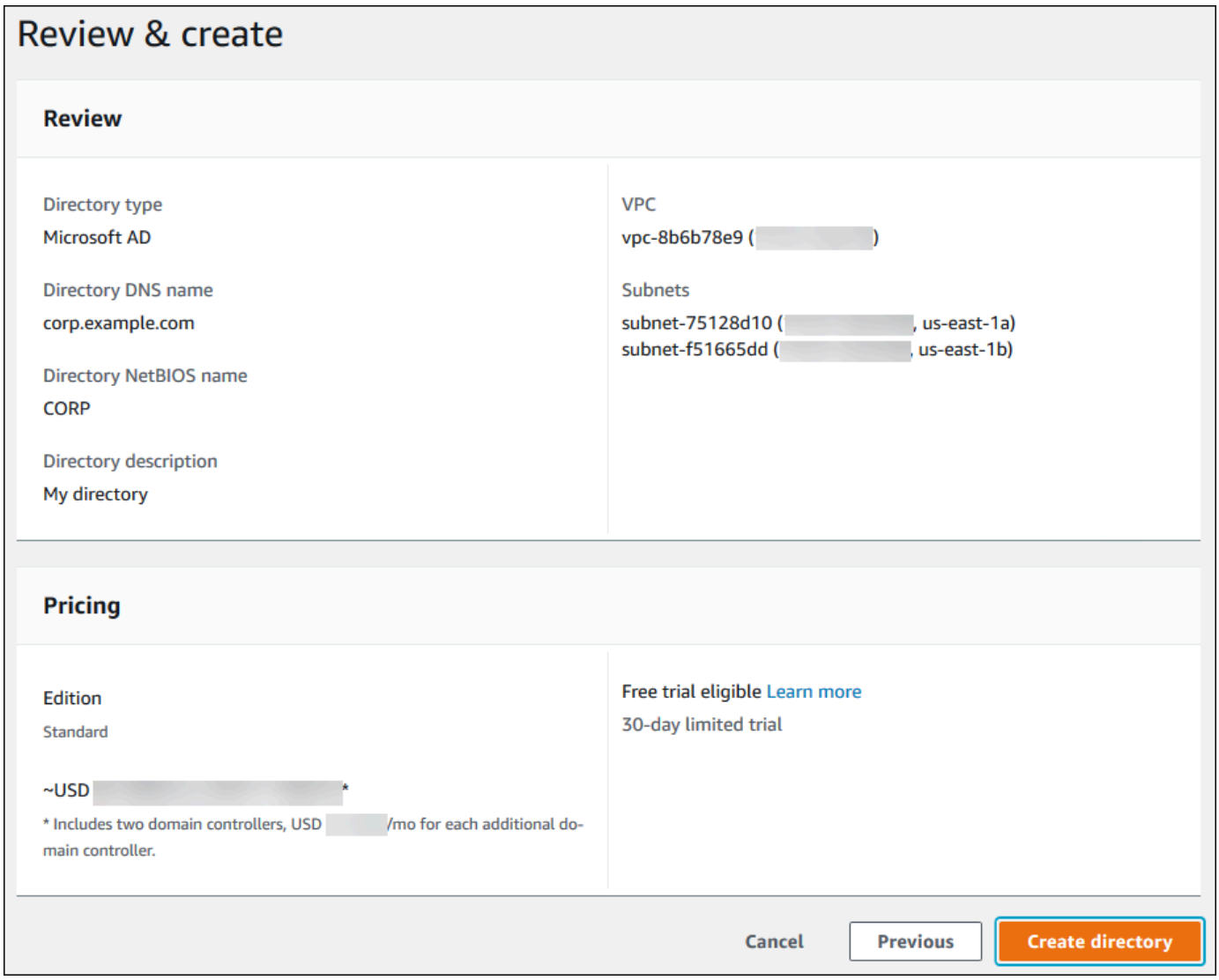
5. Escolha Next (Próximo).
6. Na página Choose VPC and subnets (Selecionar VPC e sub-redes), forneça as seguintes informações:

   VPC

   Escolha a VPC do diretório. É possível criar a instância de banco de dados PostgreSQL nessa mesma VPC ou em outra VPC.

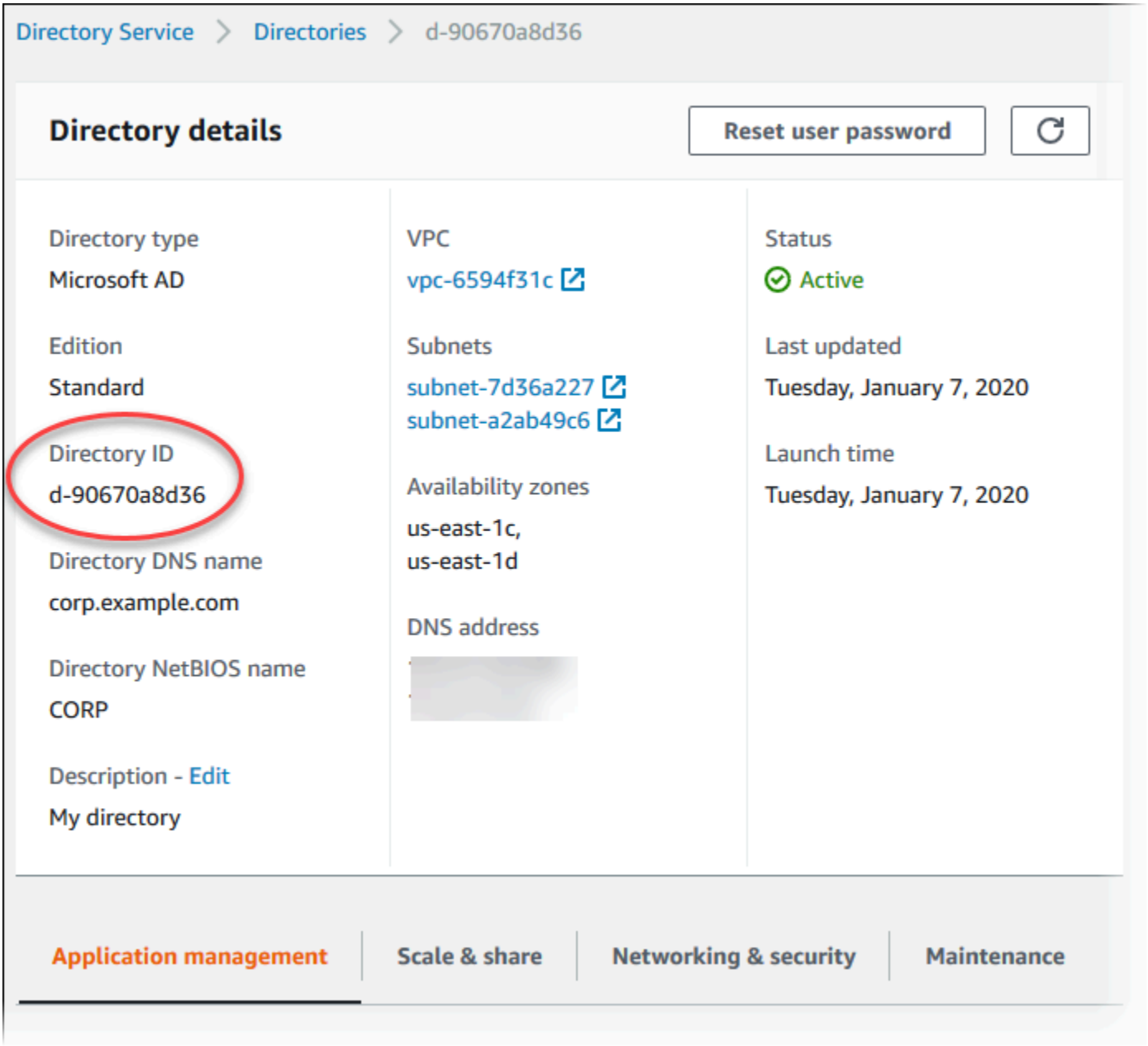
   Sub-redes

   Escolha as sub-redes para os servidores do diretório. As duas sub-redes deve estar em diferentes zonas de disponibilidade.
7. Escolha Next (Próximo).
8. Analise as informações do diretório. Se alterações forem necessárias, escolha Previous (Anterior) e faça as alterações. Quando as informações estiverem corretas, escolha Create directory (Criar diretório).



A criação do diretório leva vários minutos. Depois que o diretório tiver sido criado com sucesso, o valor de Status muda para Active (Ativo).

Para visualizar informações sobre o diretório, escolha o ID do diretório na listagem de diretórios. Anote o valor do Directory ID (ID do diretório). Esse valor será necessário ao criar ou modificar a instância de banco de dados PostgreSQL.



Etapa 2: (Opcional) Criar uma relação de confiança entre o Active Directory on-premises e o AWS Directory Service

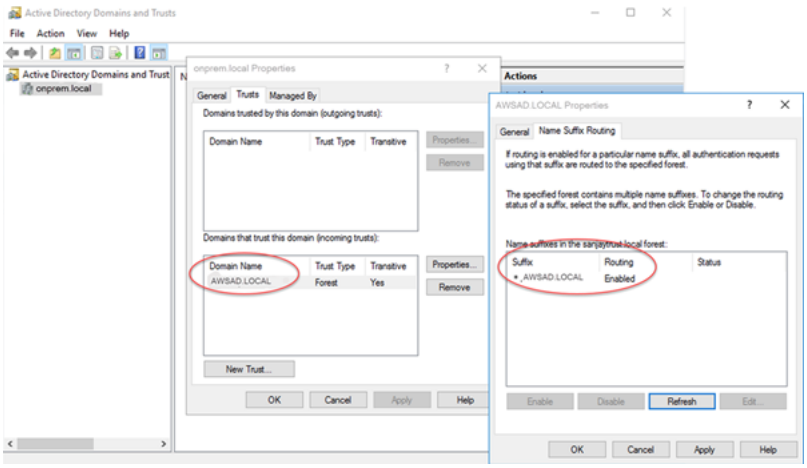
Se você não planeja usar seu próprio Microsoft Active Directory on-premises, vá para Etapa 3: Criar um perfil do IAM para que o Amazon RDS acesse o AWS Directory Service.

Para obter a autenticação Kerberos usando o Active Directory local, é necessário criar uma relação de domínio confiável usando uma confiança de floresta entre o Microsoft Active Directory on-premises e o diretório AWS Managed Microsoft AD (criado em Etapa 1: Criar um diretório usando o AWS Managed Microsoft AD). A relação de confiança pode ser unidirecional, onde o diretório AWS

Managed Microsoft AD confia no Microsoft Active Directory on-premises. A confiança também pode ser bidirecional, onde os dois Active Directories confiam um no outro. Para ter mais informações sobre como configurar confianças usando o AWS Directory Service, consulte Quando criar uma relação de confiança no Guia de administração do AWS Directory Service.

> **Note**
>
> Se você usar um Microsoft Active Directory on-premises, os clientes do Windows se conectam usando o nome de domínio do AWS Directory Service no endpoint em vez de rds.amazonaws.com. Para saber mais, consulte Conectar-se ao PostgreSQL com a autenticação Kerberos.

Verifique se o nome de domínio do Microsoft Active Directory on-premises inclui um roteamento de sufixo DNS que corresponde à relação de confiança recém-criada. A captura de tela a seguir mostra um exemplo.



Etapa 3: Criar um perfil do IAM para que o Amazon RDS acesse o AWS Directory Service

Para que o Amazon RDS chame o AWS Directory Service para você, sua conta da AWS precisa de um perfil do IAM que use a política gerenciada `AmazonRDSDirectoryServiceAccess` do IAM. Essa função permite que o Amazon RDS faça chamadas para o AWS Directory Service.

Quando você cria uma instância de banco de dados usando o AWS Management Console e a conta do usuário do console tiver a permissão `iam:CreateRole`, o console criará o perfil do IAM necessário automaticamente. Nesse caso, o nome da função é `rds-directoryservice-kerberos-access-role`. Caso contrário, é necessário criar a função do IAM manualmente. Ao

criar essa função do IAM, escolha `Directory Service` e associe a AWS política gerenciada da `AmazonRDSDirectoryServiceAccess` a ela.

Para ter mais informações sobre como criar funções do IAM para um serviço, consulte o tópico sobre como Criar uma função para delegar permissões a um serviço da AWS, no Guia do usuário do IAM.

> **Note**
>
> O perfil do IAM usado para a autenticação do Windows para RDS para Microsoft SQL Server não pode ser usado para o Amazon RDS para PostgreSQL.

Como alternativa ao uso da política gerenciada pelo `AmazonRDSDirectoryServiceAccess`, você pode criar políticas com as permissões exigidas. Nesse caso, o perfil do IAM deve ter a política de confiança do IAM a seguir.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "directoryservice.rds.amazonaws.com",
          "rds.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

A função também deve ter a política de perfil do IAM a seguir.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "ds:DescribeDirectories",
        "ds:AuthorizeApplication",
        "ds:UnauthorizeApplication",
        "ds:GetAuthorizedApplicationDetails"
      ],
    "Effect": "Allow",
    "Resource": "*"
    }
  ]
}
```

Para as Regiões da AWS pelas quais você optou, use entidades principais de serviço específicas da região nas políticas de confiança de perfil do IAM. Ao criar uma política de confiança para serviços nessas regiões, especifique o código da região na entidade principal do serviço.

O seguinte exemplo mostra uma política de confiança que inclui entidades principais de serviço específicas da região:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "directoryservice.rds.REGION-CODE.amazonaws.com",
          "rds.REGION-CODE.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
```

```
    }
  ]
}
```

Substitua REGION-CODE pelo código da sua região específica. Por exemplo, use as seguintes entidades principais de serviço para a região Ásia-Pacífico (Melbourne):

```
"Service": [
  "directoryservice.rds.ap-southeast-4.amazonaws.com",
  "rds.ap-southeast-4.amazonaws.com"
]
```

Etapa 4: Criar e configurar usuários

Você pode criar usuários com a ferramenta Usuários e computadores do Active Directory. Essa é uma das ferramentas do Active Directory Domain Services e do Active Directory Lightweight Directory Services. Para obter mais informações, consulte Add Users and Computers to the Active Directory domain (Adicionar usuários e computadores ao domínio do Active Directory) na documentação da Microsoft. Nesse caso, os usuários são indivíduos ou outras entidades, como seus computadores que fazem parte do domínio e cujas identidades estão sendo mantidas no diretório.

Para criar usuários em um diretório do AWS Directory Service, é necessário estar conectado a uma instância do Amazon EC2 baseada no Windows que seja membro do diretório do AWS Directory Service. Ao mesmo tempo, é necessário estar conectado como um usuário que tenha privilégios para criar usuários. Para obter mais informações, consulte Criar um usuário no Guia de administração do AWS Directory Service.

Etapa 5: Ativar o tráfego entre VPCs entre o diretório e a instância de banco de dados

Se você planeja localizar o diretório e a instância de banco de dados na mesma VPC, ignore esta etapa e prossiga para Etapa 6: Criar ou modificar uma instância de banco de dados PostgreSQL.

Se você planejar localizar o diretório e a instância de Bancos de Dados em VPCs diferentes, configure o tráfego entre VPCs usando o emparelhamento de VPCs ou o AWS Transit Gateway.

O procedimento a seguir habilita o tráfego entre VPCs usando o emparelhamento de VPCs. Siga as instruções em O que é o emparelhamento de VPCs? no Guia de emparelhamento do Amazon Virtual Private Cloud.

Como habilitar o tráfego entre VPCs usando o emparelhamento de VPCs

1. Configure regras apropriadas de roteamento de VPC para garantir que o tráfego de rede possa fluir em ambos os sentidos.
2. Certifique-se de que o grupo de segurança da instância de banco de dados possa receber o tráfego de entrada do grupo de segurança do diretório.
3. Garanta que não há nenhuma regra na lista de controle de acesso (ACL) de rede para bloquear o tráfego.

Se uma conta diferente da AWS for proprietária do diretório, é necessário compartilhá-lo.

Como compartilhar o diretório entre contas da AWS.

1. Inicie o compartilhamento do diretório com a conta da AWS na qual a instância de Bancos de Dados será criada seguindo as instruções em Tutorial: Compartilhar o diretório da AWS Managed Microsoft AD para ingresso perfeito no domínio do EC2 no Guia de administração do AWS Directory Service.
2. Faça login no console do AWS Directory Service usando a conta para a instância de banco de dados e garanta que o domínio tenha o status SHARED antes de prosseguir.
3. Enquanto estiver conectado ao console do AWS Directory Service usando a conta da instância de banco de dados, observe o valor do Directory ID (ID do diretório). Use esse ID do diretório para associar a instância de banco de dados ao domínio.

Etapa 6: Criar ou modificar uma instância de banco de dados PostgreSQL

Crie ou modifique uma instância de banco de dados PostgreSQL para usar com seu diretório. É possível usar o console, a CLI ou a API do RDS para associar uma instância de banco de dados a um diretório. Você pode fazer isso por meio de uma das seguintes maneiras:

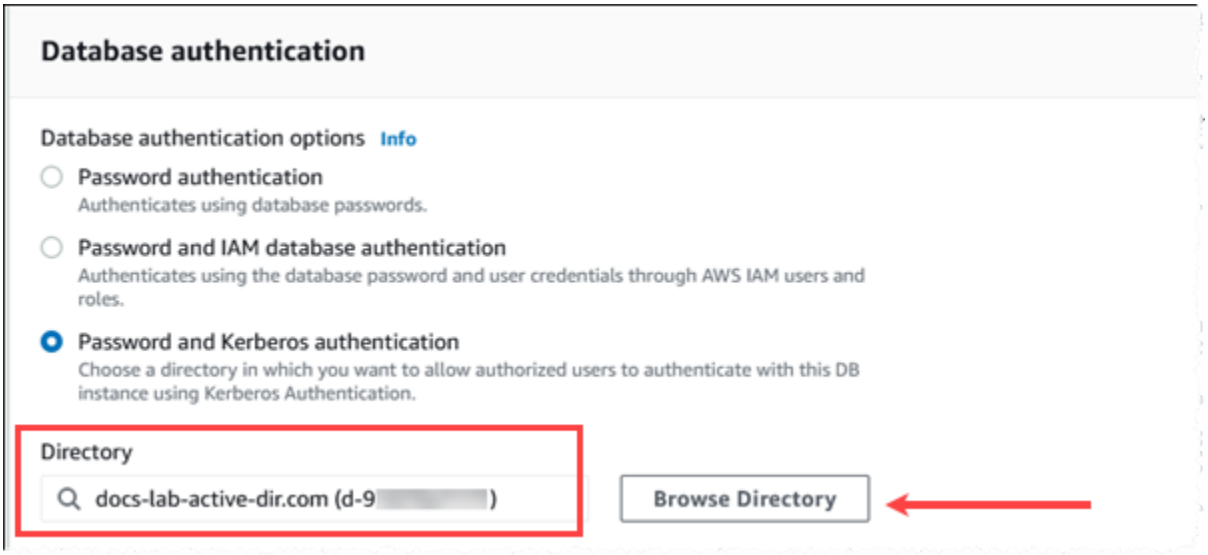
- Crie uma instância de banco de dados PostgreSQL usando o console, o comando create-db-instance da CLI ou a operação de API CreateDBInstance do RDS. Para obter instruções, consulte Criar uma instância de banco de dados do Amazon RDS.
- Modifique uma instância de banco de dados PostgreSQL existente usando o console, o comando modify-db-instance da CLI ou a operação de API ModifyDBInstance do RDS. Para obter instruções, consulte Modificar uma instância de banco de dados do Amazon RDS.

- Restaure uma instância de banco de dados PostgreSQL de um snapshot de banco de dados usando o console, o comando restore-db-instance-from-db-snapshot da CLI ou a operação de API RestoreDBInstanceFromDBSnapshot do RDS. Para obter instruções, consulte Restaurar uma instância de banco de dados.
- Restaure uma instância de banco de dados PostgreSQL em um determinado momento usando o console, o comando restore-db-instance-to-point-in-time da CLI ou a operação de API RestoreDBInstanceToPointInTime do RDS. Para obter instruções, consulte Restaurar uma instância de banco de dados para um momento especificado no Amazon RDS.

A autenticação Kerberos só é compatível com instâncias de de banco de dados PostgreSQL em uma VPC. A instância de banco de dados pode estar na mesma VPC do diretório ou em uma VPC diferente. A instância de banco de dados deve usar um grupo de segurança que permita entrada e saída na VPC do diretório para que a instância de banco de dados possa se comunicar com o diretório.

Console

Ao usar o console para criar, modificar ou restaurar uma instância de banco de dados, escolha Password and Kerberos authentication (Senha e autenticação Kerberos) na seção Database authentication (Autenticação de banco de dados). Em seguida, escolha Browse Directory (Procurar diretório). Selecione o diretório ou escolha Create a new directory (Criar um novo diretório) para usar o Directory Service.



AWS CLI

Quando você usa a AWS CLI, são necessários os seguintes parâmetros para que a instância de banco de dados possa usar o diretório criado:

- Para o parâmetro `--domain`, use o identificador de domínio (identificador "d-*") gerado quando o diretório foi criado.
- Para o parâmetro `--domain-iam-role-name`, use a função criada que usa a política gerenciada `AmazonRDSDirectoryServiceAccess` do IAM.

Por exemplo, o comando da CLI a seguir modifica uma instância de banco de dados para usar um diretório.

```
aws rds modify-db-instance --db-instance-identifier mydbinstance --domain d-Directory-ID --domain-iam-role-name role-name
```

> **Important**
>
> Se você modificar uma instância de banco de dados para habilitar a autenticação Kerberos, reinicialize a instância de banco de dados após a alteração.

Etapa 7: Criar usuários do PostgreSQL para suas entidades principais do Kerberos

Nesse ponto, sua instância de banco de dados do RDS para PostgreSQL é unida ao domínio AWS Managed Microsoft AD. Os usuários que você criou no diretório em Etapa 4: Criar e configurar usuários precisam ser configurados como usuários do banco de dados do PostgreSQL e receber privilégios para fazer login no banco de dados. Você faz isso fazendo login como usuário do banco de dados com privilégios `rds_superuser`. Por exemplo, se você aceitou os padrões ao criar sua instância de banco de dados do PostgreSQL, use `postgres`, conforme mostrado nas etapas a seguir.

Como criar usuários de banco de dados do PostgreSQL para entidades principais do Kerberos

1. Use o `psql` para conectar-se ao endpoint da instância de banco de dados de seu RDS para PostgreSQL usando `psql`. O exemplo a seguir usa a conta `postgres` padrão para a função `rds_superuser`.

```
psql --host=cluster-instance-1.111122223333.aws-region.rds.amazonaws.com --
port=5432 --username=postgres --password
```

2. Crie um nome de usuário do banco de dados para cada entidade principal do Kerberos (nome de usuário do Active Directory) que você deseja que tenha acesso ao banco de dados. Use o nome de usuário canônico (identidade) conforme definido na instância do Active Directory, ou seja, uma letra minúscula `alias` (nome de usuário no Active Directory) e o nome em maiúscula do domínio do Active Directory para esse nome de usuário. O nome de usuário do Active Directory é um usuário autenticado externamente, portanto, use aspas ao redor do nome conforme mostrado a seguir.

```
postgres=> CREATE USER "username@CORP.EXAMPLE.COM" WITH LOGIN;
CREATE ROLE
```

3. Conceda a função `rds_ad` ao usuário do banco de dados.

```
postgres=> GRANT rds_ad TO "username@CORP.EXAMPLE.COM";
GRANT ROLE
```

Depois de concluir a criação de todos os usuários do PostgreSQL para suas identidades de usuário do Active Directory, os usuários podem acessar a instância de banco de dados do RDS para PostgreSQL usando suas credenciais do Kerberos.

É necessário que os usuários do banco de dados que se autenticam usando o Kerberos estejam fazendo isso nas máquinas cliente que sejam membros do domínio do Active Directory.

Os usuários do banco de dados aos quais foi concedida a função `rds_ad` não podem ter também a função `rds_iam`. Isso também se aplica a assinaturas aninhadas. Para obter mais informações, consulte Autenticação do banco de dados do IAM para MariaDB, MySQL e PostgreSQL.

Etapa 8: Configurar um cliente PostgreSQL

Para configurar um cliente PostgreSQL, execute as seguintes etapas:

- Crie um arquivo krb5.conf (ou equivalente) para apontar para o domínio.
- Verifique se o tráfego pode fluir entre o host do cliente e o AWS Directory Service. Use um utilitário de rede, como o Netcat, para o seguinte:
  - Verifique o tráfego pelo DNS da porta 53.

  - Verifique o tráfego pelo TCP/UDP da porta 53 e do Kerberos, que inclui as portas 88 e 464 do AWS Directory Service.
- Verifique se o tráfego pode fluir entre o host do cliente e a instância de banco de dados pela porta do banco de dados. Por exemplo, use psql para conectar e acessar o banco de dados.

Veja a seguir um exemplo de conteúdo krb5.conf para o AWS Managed Microsoft AD.

```
[libdefaults]
 default_realm = EXAMPLE.COM
[realms]
 EXAMPLE.COM = {
  kdc = example.com
  admin_server = example.com
 }
[domain_realm]
 .example.com = EXAMPLE.COM
 example.com = EXAMPLE.COM
```

Veja a seguir um exemplo de conteúdo krb5.conf para o Microsoft Active Directory on-premises.

```
[libdefaults]
 default_realm = EXAMPLE.COM
[realms]
 EXAMPLE.COM = {
  kdc = example.com
  admin_server = example.com
 }
 ONPREM.COM = {
  kdc = onprem.com
  admin_server = onprem.com
 }
[domain_realm]
 .example.com = EXAMPLE.COM
 example.com = EXAMPLE.COM
 .onprem.com = ONPREM.COM
 onprem.com = ONPREM.COM
 .rds.amazonaws.com = EXAMPLE.COM
 .amazonaws.com.cn = EXAMPLE.COM
 .amazon.com = EXAMPLE.COM
```

Gerenciar uma instância de banco de dados do RDS para PostgreSQL em um domínio do Active Directory

É possível usar o console, a CLI ou a API do RDS para gerenciar a instância de banco de dados e suas relações com o Microsoft Active Directory. Por exemplo, é possível associar um Microsoft Active Directory para habilitar a autenticação Kerberos. Também é possível remover a associação de um Microsoft Active Directory para desabilitar a autenticação Kerberos. Também é possível mover uma instância de banco de dados para a autenticação externa por um Microsoft Active Directory para outro.

Por exemplo, usando a CLI, é possível fazer o seguinte:

- Para tentar habilitar a autenticação Kerberos novamente para uma assinatura com falha, use o comando da CLI modify-db-instance . Especifique o ID do diretório da associação atual para a opção `--domain`.
- Para desabilitar a autenticação Kerberos em uma instância de banco de dados, use o comando da CLI modify-db-instance . Especifique `none` para a opção `--domain`.
- Para mover uma instância de banco de dados de um domínio para outro, use o comando da CLI modify-db-instance. Especifique o identificador de domínio do novo domínio para a opção `--domain`.

Compreensão da associação de domínio

Depois de criar ou modificar sua instância de banco de dados, ele se torna membro do domínio. É possível visualizar o status da associação do domínio no console ou executando o comando da CLI describe-db-instances. O status da instância de banco de dados pode ser um dos seguintes:

- `kerberos-enabled` – a instância de banco de dados que tem a autenticação Kerberos habilitada.
- `enabling-kerberos`: a AWS está no processo de habilitar a autenticação Kerberos nessa instância de bancos de dados.
- `pending-enable-kerberos` – a habilitação da autenticação Kerberos está pendente nessa instância de banco de dados.
- `pending-maintenance-enable-kerberos`: a AWS tentará habilitar a autenticação Kerberos na instância de bancos de dados durante a próxima janela de manutenção programada.

- `pending-disable-kerberos` – a desabilitação da autenticação Kerberos está pendente nessa instância de banco de dados.
- `pending-maintenance-disable-kerberos`: a AWS tentará desabilitar a autenticação Kerberos na instância de banco de dados durante a próxima janela de manutenção programada.
- `enable-kerberos-failed`: um problema de configuração impediu que a AWS habilitasse a autenticação Kerberos na instância de banco de dados. Corrija o problema de configuração antes de emitir o comando novamente para modificar a instância de banco de dados.
- `disabling-kerberos`: a AWS está no processo de desabilitar a autenticação Kerberos nessa instância de bancos de dados.

Uma solicitação para habilitar a autenticação Kerberos pode falhar por conta de um novo problema de conectividade de rede ou de um perfil do IAM incorreto. Em alguns casos, poderá haver falha na tentativa de habilitar a autenticação Kerberos quando você criar ou modificar uma instância de banco de dados. Nesse caso, verifique se você está usando o perfil do IAM correto e modifique a instância de banco de dados para ingressar no domínio.

> **Note**
>
> Somente a autenticação Kerberos com o RDS para PostgreSQL envia tráfego para os servidores DNS do domínio. Todas as outras solicitações de DNS são tratadas como acesso à rede de saída nas instâncias de banco de dados que executam o PostgreSQL. Para obter mais informações sobre o acesso à rede de saída com o RDS para PostgreSQL, consulte Usar um servidor DNS personalizado para acesso à rede de saída.

Conectar-se ao PostgreSQL com a autenticação Kerberos

Você pode se conectar ao PostgreSQL com autenticação Kerberos com a interface pgAdmin ou com uma interface de linha de comando, como psql. Para obter mais informações sobre a conexão, consulte Conectar a uma instância de banco de dados executando o mecanismo de banco de dados do PostgreSQL . Para saber como obter o endpoint, o número da porta e outros detalhes necessários para a conexão, consulte Conectar-se à instância de banco de dados PostgreSQL.

> **Note**
>
> A autenticação e a criptografia GSSAPI no PostgreSQL são implementadas pela biblioteca Kerberos `libkrb5.so`. Recursos como `postgres_fdw` e `dblink` também dependem dessa mesma biblioteca para conexões de saída com autenticação ou criptografia Kerberos.

pgAdmin

Para usar o pgAdmin para conectar-se ao PostgreSQL com autenticação Kerberos, execute as seguintes etapas:

1. Inicie a aplicação pgAdmin no computador cliente.
2. Na guia Dashboard (Painel), escolha Add New Server (Adicionar novo servidor).
3. Na caixa de diálogo Criar - servidor, insira um nome na guia Geral para identificar o servidor no pgAdmin.
4. Na guia Connection (Conexão), insira as seguintes informações do banco de dados do RDS para PostgreSQL.
   - Em Host, insira o endpoint da Instância de banco de dados do RDS para PostgreSQL. Um endpoint é semelhante ao seguinte:

     ```
     RDS-DB-instance.111122223333.aws-region.rds.amazonaws.com
     ```

     Para se conectar a um Microsoft Active Directory on-premises de um cliente Windows, use o nome de domínio do AWS Managed Active Directory em vez de `rds.amazonaws.com` no endpoint do host. Por exemplo, suponha que o nome de domínio do Managed Active Directory da AWS seja `corp.example.com`. Depois, em Host, o endpoint seria especificado da seguinte forma:

     ```
     RDS-DB-instance.111122223333.aws-region.corp.example.com
     ```

   - Em Porta, insira a porta designada.
   - Em Banco de dados de manutenção, insira o nome do banco de dados inicial ao qual o cliente se conectará.
   - Em Nome de usuário, insira o nome de usuário que você inseriu para a autenticação Kerberos em Etapa 7: Criar usuários do PostgreSQL para suas entidades principais do Kerberos .

5. Escolha Salvar.

Psql

Para usar o psql para conectar-se ao PostgreSQL com autenticação Kerberos, execute as seguintes etapas:

1. Em um prompt de comando, execute o comando a seguir.

   ```
   kinit username
   ```

   Substitua *username* pelo nome de usuário. No prompt, insira a senha armazenada no Microsoft Active Directory para o usuário.

2. Se a instância de banco de dados PostgreSQL estiver usando uma VPC acessível publicamente, coloque um endereço IP para o endpoint do da instância de banco de dados em seu arquivo `/etc/hosts` no cliente do EC2. Por exemplo, os comandos a seguir obtêm o endereço IP e o colocam no arquivo `/etc/hosts`.

   ```
   % dig +short PostgreSQL-endpoint.AWS-Region.rds.amazonaws.com
   ;; Truncated, retrying in TCP mode.
   ec2-34-210-197-118.AWS-Region.compute.amazonaws.com.
   34.210.197.118

   % echo " 34.210.197.118  PostgreSQL-endpoint.AWS-Region.rds.amazonaws.com" >> /etc/
   hosts
   ```

   Se você estiver usando um Microsoft Active Directory on-premises de um cliente Windows, será necessário se conectar usando um endpoint especializado. Em vez de usar o domínio `rds.amazonaws.com` da Amazon no endpoint do host, use o nome de domínio do Managed Active Directory da AWS.

   Por exemplo, suponha que o nome de domínio do Managed Active Directory da AWS seja `corp.example.com`. Use o formato *PostgreSQL-endpoint.AWS-Region.corp.example.com* para o endpoint e coloque-o no arquivo `/etc/hosts`.

   ```
   % echo " 34.210.197.118  PostgreSQL-endpoint.AWS-Region.corp.example.com" >> /etc/
   hosts
   ```

3. Use o comando psql a seguir para fazer login em uma instância de banco de dados PostgreSQL com integração ao Active Directory.

```
psql -U username@CORP.EXAMPLE.COM -p 5432 -h PostgreSQL-endpoint.AWS-
Region.rds.amazonaws.com postgres
```

Para fazer login no cluster de banco de dados PostgreSQL de um cliente Windows usando um Active Directory on-premises, use o seguinte comando psql com o nome de domínio da etapa anterior (`corp.example.com`):

```
psql -U username@CORP.EXAMPLE.COM -p 5432 -h PostgreSQL-endpoint.AWS-
Region.corp.example.com postgres
```

Usar um servidor DNS personalizado para acesso à rede de saída

O RDS para PostgreSQL é compatível com o acesso à rede de saída nas instâncias de banco de dados e permite a resolução Domain Name Service (DNS) de um servidor DNS de propriedade do cliente. Você pode resolver apenas nomes de domínio totalmente qualificados a partir da sua instância de banco de dados do RDS para PostgreSQL por meio do seu servidor DNS personalizado.

Tópicos

- Ativar resolução DNS personalizada
- Desativar resolução DNS personalizada
- Configuração de um servidor DNS personalizado

Ativar resolução DNS personalizada

Para ativar a resolução DNS na VPC do cliente, primeiro associe um grupo de parâmetros de banco de dados personalizado à instância do RDS para PostgreSQL. Em seguida, ative o parâmetro `rds.custom_dns_resolution` definindo-o como 1 e reinicie a instância de banco de dados para que as alterações sejam aplicadas.

Desativar resolução DNS personalizada

Para desativar a resolução de DNS na VPC do cliente, primeiro desative o parâmetro `rds.custom_dns_resolution` do grupo de parâmetros de banco de dados personalizado definindo-o como 0. Em seguida, reinicie a instância de banco de dados para que as alterações entrem em vigor.

Configuração de um servidor DNS personalizado

Após a configuração do seu servidor de nomes DNS personalizado, demora até 30 minutos para propagar as alterações na sua instância de banco de dados. Depois que as alterações são propagadas para sua a instância de banco de dados, todo o tráfego de rede de saída que requer uma pesquisa de DNS consulta o servidor DNS por meio da porta 53.

> **Note**
>
> Se você não configurar um servidor DNS personalizado e se `rds.custom_dns_resolution` estiver definido como 1, os hosts serão resolvidos usando

uma zona privada do Amazon Route 53. Para obter mais informações, consulte Como trabalhar com zonas hospedadas privadas.

Para configurar um servidor DNS personalizado para a instância de banco de dados do RDS para PostgreSQL

1. Nas opções de Protocolo de Configuração Dinâmica de Host (DHCP) anexadas à sua VPC, defina a opção `domain-name-servers` como o endereço IP do seu servidor de nomes DNS. Para obter mais informações, consulte Conjuntos de opções de DHCP.

   > **Note**
   >
   > A opção `domain-name-servers` aceita até quatro valores, mas sua instância de banco de dados do Amazon RDS usa apenas o primeiro valor.

2. Certifique-se de que seu servidor DNS possa resolver todas as consultas de pesquisa, incluindo nomes DNS públicos, nomes DNS privados do Amazon EC2 e nomes DNS específicos do cliente. Se o tráfego de rede de saída contiver pesquisas de DNS que o seu servidor DNS não pode manipular, este deverá ter provedores de DNS upstream configurados.
3. Configure seu servidor DNS para produzir respostas de protocolo UDP de 512 bytes ou menos.
4. Configure seu servidor DNS para produzir respostas de protocolo TCP de 1.024 bytes ou menos.
5. Configure seu servidor DNS para permitir tráfego de entrada das suas instâncias de banco de dados do Amazon RDS pela porta 53. Se o seu servidor DNS estiver em uma Amazon VPC, esta deverá ter um grupo de segurança que contenha regras de entrada que permitam tráfego UDP e TCP na porta 53. Se o servidor DNS não estiver em uma Amazon VPC, ele deverá ter configurações de firewall apropriadas para permitir o tráfego de entrada UDP e TCP na porta 53.

   Para obter mais informações, consulte Grupos de segurança para sua VPC e Adição e remoção de regras.
6. Configurar a VPC da sua instância de banco de dados do Amazon RDS para permitir o tráfego de saída por meio da porta 53. Sua VPC deve ter um grupo de segurança que contém regras de saída que permitem o tráfego TCP e UDP na porta 53.

   Para obter mais informações, consulte Grupos de segurança para a VPC e Adicionar e remover regras no Guia do usuário da Amazon VPC.

7. Certifique-se de que o caminho de roteamento entre a instância de banco de dados Amazon RDS e o servidor DNS esteja configurado corretamente para permitir o tráfego DNS.

   Também, se a instância de banco de dados Amazon RDS e o servidor DNS não estiverem na mesma VPC, verifique se uma conexão de emparelhamento está configurada entre eles. Para obter mais informações, consulte O que é emparelhamento de VPC? no Guia de emparelhamento da Amazon VPC.

Atualizações do mecanismo de banco de dados do RDS para PostgreSQL

Há dois tipos de atualizações que você pode gerenciar para o banco de dados do PostgreSQL:

- Atualizações de sistema operacionais: ocasionalmente, o Amazon RDS pode precisar atualizar o sistema operacional subjacente do banco de dados para aplicar correções de segurança ou alterações no sistema operacional. Decida quando o Amazon RDS aplicará atualizações de SO usando o console do RDS, a AWS Command Line Interface (AWS CLI) ou a API do RDS. Para obter mais informações sobre atualizações de SO, consulte Aplicar atualizações para uma instância de banco de dados.
- Atualizações de mecanismos de banco de dados: quando o Amazon RDS comporta uma nova versão de um mecanismo de banco de dados, você pode atualizar os de bancos de dados para essa nova versão.

Nesse contexto, banco de dados é uma instância de banco de dados do RDS para PostgreSQL ou um cluster de banco de dados multi-AZ.

Há dois tipos de atualizações do mecanismo de bancos de dados do PostgreSQL: de versão principal e de versão secundária.

Atualizações da versão principal

As atualizações da versão principal podem conter as alterações de banco de dados incompatíveis com os aplicativos existentes. Por isso, você deve realizar manualmente as atualizações de versões principais dos bancos de dados. Você pode iniciar uma atualização de versão principal modificando manualmente a instância de banco de dados ou o cluster de banco de dados multi-AZ. Antes de realizar uma atualização de versão principal, recomendamos seguir as etapas descritas em Escolher uma versão principal para uma atualização do RDS para PostgreSQL.

O Amazon RDS gerencia os upgrades de versões principais multi-AZ das seguintes formas:

- Implantação de instância de banco de dados multi-AZ: o Amazon RDS faz upgrade simultaneamente da instância primária e de todas as instâncias de espera. O banco de dados pode ficar indisponível por vários minutos enquanto o upgrade é concluído.
- Implantação de cluster de banco de dados multi-AZ: o Amazon RDS faz upgrade simultaneamente das instâncias de gravador e leitor. O banco de dados pode ficar indisponível por vários minutos enquanto o upgrade é concluído.

Se você fizer upgrade de uma instância de banco de dados com réplicas de leitura na região, o Amazon RDS fará upgrade das réplicas e da instância de banco de dados primária.

O Amazon RDS não atualiza réplicas de leitura de clusters de banco de dados multi-AZ. Se você realizar uma atualização de versão principal de um cluster de banco de dados multi-AZ, o estado da replicação das réplicas de leitura mudará para Encerrado. Você deve excluir e recriar as réplicas de leitura manualmente após a conclusão da atualização.

> **Tip**
>
> É possível minimizar o tempo de inatividade necessário para a atualização da versão principal utilizando uma implantação azul/verde. Para ter mais informações, consulte Usar implantações azul/verde para atualizações de banco de dados.

Atualizações de versões secundárias

Por outro lado, atualizações de versões secundárias incluem apenas alterações compatíveis com versões anteriores dos aplicativos existentes. Você pode iniciar uma atualização de versão secundária manualmente modificando o banco de dados. Ou é possível habilitar a opção Upgrade automático de versões secundárias ao criar ou modificar um banco de dados. Isso significa que o Amazon RDS atualiza automaticamente o banco de dados depois de testar e aprovar a nova versão.

O Amazon RDS gerencia os upgrades de versões secundárias multi-AZ das seguintes formas:

- Implantação de instância de banco de dados multi-AZ: o Amazon RDS faz upgrade simultaneamente da instância primária e de todas as instâncias de espera. O banco de dados pode ficar indisponível por vários minutos enquanto o upgrade é concluído.
- Implantação de cluster de banco de dados multi-AZ: o Amazon RDS faz upgrade das instâncias de banco de dados de leitor, uma por vez. Depois, uma das instâncias de banco de dados de leitor passa a ser a nova instância de banco de dados de gravador. Depois, o Amazon RDS atualiza a antiga instância de gravador (que agora é uma instância de leitor). Os clusters de banco de dados multi-AZ normalmente reduzem o tempo de inatividade de atualizações de versões secundárias para aproximadamente 35 segundos. Quando usado com o RDS Proxy, é possível reduzir ainda mais o tempo de inatividade para um segundo ou menos. Para ter mais informações, consulte Amazon RDS Proxy. Como alternativa, é possível usar um proxy de banco de dados de código aberto, como ProxySQL, PgBouncer ou driver avançado de encapsulamento JDBC da AWS.

Se o banco de dados tiver réplicas de leitura, primeiro faça upgrade de todas elas antes de fazer upgrade da instância ou do cluster de origem.

Para ter mais informações, consulte Atualizações automáticas da versão secundária para o RDS para PostgreSQL. Para obter informações sobre como executar manualmente uma atualização de versão secundária, consulte Atualizar manualmente a versão do mecanismo.

Para ter mais informações sobre versões de mecanismos de banco de dados e a política de descontinuação dessas versões, consulte Versões de mecanismos de banco de dados nas Perguntas frequentes sobre o Amazon RDS.

Tópicos

- Considerações sobre atualizações do PostgreSQL
- Encontrar destinos de atualização válidos
- Números de versão do PostgreSQL
- Números de versão do RDS no RDS para PostgreSQL
- Escolher uma versão principal para uma atualização do RDS para PostgreSQL
- Como atualizar a versão principal do RDS para PostgreSQL
- Atualizações automáticas da versão secundária para o RDS para PostgreSQL
- Atualizar extensões do PostgreSQL em bancos de dados do RDS para PostgreSQL

Considerações sobre atualizações do PostgreSQL

Para fazer a atualização dos bancos de dados com segurança, o Amazon RDS usa o utilitário `pg_upgrade`, descrito na documentação do PostgreSQL

Se o período de retenção de backup for maior que 0, o Amazon RDS usará dois snapshots de banco de dados durante o processo de upgrade. O primeiro snapshot de banco de dados é do banco de dados antes de qualquer alteração de atualização ter sido feita. Se a atualização apresentar falha para seus bancos de dados, você poderá restaurar esse snapshot para criar um banco de dados que execute a versão antiga. O segundo snapshot de banco de dados é tirado após a conclusão do upgrade. Esses snapshots de banco de dados são excluídos automaticamente quando o período de retenção do backup expira.

> **Note**
>
> O Amazon RDS só tirará snapshots de banco de dados durante o processo de atualização se você tiver definido o período de retenção de backup para o banco de dados como um número maior que 0. Para modificar o período de retenção de backup para uma instância de banco de dados, consulte the section called “Modificação de uma instância de banco de dados”. Não é possível configurar um período de retenção de backup personalizado para um cluster de banco de dados multi-AZ.

Ao fazer uma atualização de versão principal de uma instância de banco de dados, qualquer réplica de leitura na região também será atualizada automaticamente. Após o início do fluxo de trabalho de atualização, as réplicas de leitura aguardam até que a `pg_upgrade` seja concluída com êxito na instância de banco de dados primária. Depois disso, a atualização da instância de banco de dados primária aguarda a conclusão das atualizações da réplica de leitura. Você enfrenta uma interrupção até que a atualização seja concluída. Quando você realiza uma atualização de versão principal de um cluster de banco de dados multi-AZ, o estado da replicação das réplicas de leitura muda para Encerrado.

Depois que uma atualização for concluída, não será possível reverter para a versão anterior do mecanismo de banco de dados. Se quiser retornar à versão anterior, restaure o snapshot de banco de dados que foi tirado antes da atualização para criar um banco de dados.

Encontrar destinos de atualização válidos

Ao usar o AWS Management Console para atualizar um banco de dados, ele exibe os destinos de atualização válidos para o banco de dados. Também é possível usar o seguinte comando da AWS CLI para identificar os destinos de atualização válidos para um banco de dados:

Para Linux, macOS ou Unix:

```
aws rds describe-db-engine-versions \
  --engine postgres \
  --engine-version version-number \
  --query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" --
output text
```

Para Windows:

```
aws rds describe-db-engine-versions ^
  --engine postgres ^
  --engine-version version-number ^
  --query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" --
output text
```

Por exemplo, para identificar os destinos de atualização válidos para um banco de dados PostgreSQL versão 16.1, execute o seguinte comando da AWS CLI:

Para Linux, macOS ou Unix:

```
aws rds describe-db-engine-versions \
  --engine postgres \
  --engine-version 16.1 \
  --query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" --
output text
```

Para Windows:

```
aws rds describe-db-engine-versions ^
  --engine postgres ^
  --engine-version 16.1 ^
  --query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" --
output text
```

Números de versão do PostgreSQL

A sequência de numeração da versão do mecanismo de banco de dados do PostgreSQL é a seguinte:

- Para versões 10 e posteriores do PostgreSQL, o número da versão do mecanismo é no formato major.minor. O número da versão principal é a parte inteira do número da versão. O número da versão secundária é a parte fracionária do número da versão.

  Uma atualização de versão principal aumenta a parte inteira do número da versão, como atualizar de 10.minor para 11.minor.
- Para versões do PostgreSQL anteriores à 10, o número da versão do mecanismo é no formato major.major.minor. O número da versão principal do mecanismo é o número inteiro e a primeira parte fracionária do número da versão. Por exemplo, a 9.6 é uma versão principal. O número da

versão secundária é a terceira parte do número da versão. Por exemplo, para a versão 9.6.12, 12 é o número da versão secundária.

Uma atualização de versão principal aumenta a parte principal do número da versão. Por exemplo, uma atualização de 9.6.12 para 11.14 é uma atualização de versão principal, em que 9.6 e 11 são os números da versão principal.

Consulte informações sobre a numeração de versão do Suporte estendido do RDS em Nomenclatura da versão do Suporte estendido do Amazon RDS.

Números de versão do RDS no RDS para PostgreSQL

Os números de versão do RDS usam o esquema de nomenclatura *major*.*minor*.*patch*. Uma versão do patch do RDS inclui correções de bugs importantes adicionadas a uma versão secundária após o lançamento. Consulte informações sobre a numeração de versão do Suporte estendido do RDS em Nomenclatura da versão do Suporte estendido do Amazon RDS.

Para identificar o número da versão do Amazon RDS do banco de dados, você deve primeiro criar a extensão `rds_tools` usando o seguinte comando:

```
CREATE EXTENSION rds_tools;
```

A partir do lançamento do PostgreSQL versão 15.2-R2, você pode descobrir o número de versão do RDS do banco de dados do RDS para PostgreSQL com a seguinte consulta SQL:

```
postgres=> SELECT rds_tools.rds_version();
```

Por exemplo, consultar um banco de dados do RDS para PostgreSQL 15.2 retorna o seguinte:

```
rds_version
----------------
 15.2.R2
(1 row)
```

Escolher uma versão principal para uma atualização do RDS para PostgreSQL

As atualizações para versões principais podem conter alterações incompatíveis com as versões anteriores do banco de dados. A nova funcionalidade pode fazer com que suas aplicações existentes

parem de funcionar corretamente. Por esse motivo, o Amazon RDS não aplica atualizações para versões principais automaticamente. Para realizar uma atualização de versão principal, modifique manualmente o banco de dados. Teste completamente qualquer atualização para verificar se as aplicações funcionam corretamente antes de aplicá-la aos bancos de dados de produção. Ao fazer uma atualização de versão principal do PostgreSQL, recomendamos que você siga as etapas descritas em Como atualizar a versão principal do RDS para PostgreSQL.

Ao atualizar a implantação de uma instância de banco de dados single-AZ ou multi-AZ do PostgreSQL para a próxima versão principal, todas as réplicas de leitura associadas ao banco de dados também serão atualizadas para a próxima versão principal. Em alguns casos, você pode pular para uma versão principal mais alta ao atualizar. Se a atualização ignorar uma versão principal, as réplicas de leitura também serão atualizadas para essa versão principal de destino. As atualizações para a versão 11 que ignoram outras versões principais têm certas limitações. Você pode encontrar os detalhes nas etapas descritas em Como atualizar a versão principal do RDS para PostgreSQL.

A maioria das extensões do PostgreSQL não é atualizada durante uma atualização de mecanismo do PostgreSQL. Elas devem ser atualizadas separadamente. Para obter mais informações, consulte Atualizar extensões do PostgreSQL em bancos de dados do RDS para PostgreSQL.

Você pode descobrir quais versões principais estão disponíveis para o banco de dados do RDS para PostgreSQL executando a seguinte consulta da AWS CLI:

```
aws rds describe-db-engine-versions --engine postgres  --engine-version your-version
 --query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" --
output text
```

A tabela a seguir resume os resultados dessa consulta para todas as versões disponíveis. Um asterisco (*) no número da versão significa que ela foi descontinuada. Se a versão atual não for compatível, recomendamos que você atualize para o destino de atualização da versão secundária mais recente ou para um dos outros destinos de atualização disponíveis para essa versão.

Versão de origem atual	Destinos de atualização
17.5	Nenhum
17.4	17.5
17.3	17.5, 17.4

Versão de origem atual	Destinos de atualização
17.2	17.5, 17.4, 17.3
17,1	17.5, 17.4, 17.3 , 17.2
16.9	17.5
16.8	17.5, 17.4
16.7	17.5, 17.4, 17.3 16.8
16.6	17.5, 17.4, 17.3 , 17.2 16.8, 16.7
16.5	17.5, 17.4, 17.3, 17.2, 17.1 16.8, 16.7, 16.6
16.4	17.5, 17.4, 17.3, 17.2, 17.1 16.8, 16.7, 16.6, 16.5
16.3	17.5, 17.4, 17.3, 17.2, 17.1 16.8, 16.7, 16.6, 16.5, 16.4
16.2	17.5, 17.4, 17.3, 17.2, 17.1 16.8, 16.7, 16.6, 16.5, 16.4, 16.3
16.1	17.5, 17.4, 17.3, 17.2, 17.1 16.8, 16.7, 16.6, 16.5, 16.4, 16.3, 16.2
15.13	17.5 16.9

Versão de origem atual	Destinos de atualização
15.12	17.4 16.9, 16.8 15.13
15.11	17.3 16.9, 16.8, 16.7 15.13, 15.12
15.10	17.2 16.9, 16.8, 16.7, 16.6 15.13, 15.12, 15.11
15.9	17.1 16.9, 16.8, 16.7, 16.6, 16.5 15.13, 15.12, 15.11, 15.10
15.8	16.9, 16.8, 16.7, 16.6, 16.5, 16.4 15.13, 15.12, 15.11, 15.10, 15.9
15.7	16.9, 16.8, 16.7, 16.6, 16.5, 16.4, 16.3 15.13, 15.12, 15.11, 15.10, 15.9, 15.8
15.6	16.9, 16.8, 16.7, 16.6, 16.5, 16.4, 16.3, 16.2 15.13, 15.12, 15.11, 15.10, 15.9, 15.8, 15.7

Versão de origem atual	Destinos de atualização
15.5	16.9, 16.8, 16.7, 16.6, 16.5, 16.4, 16.3, 16.2, 16.1 15.13, 15.12, 15.11, 15.10, 15.9, 15.8, 15.7, 15.6
15.4	16.9, 16.8, 16.7, 16.6, 16.5, 16.4, 16.3, 16.2, 16.1 15.13, 15.12, 15.11, 15.10, 15.9, 15.8, 15.7, 15.6, 15.5
15.3*	16.9, 16.8, 16.7, 16.6, 16.5, 16.4, 16.3, 16.2, 16.1 15.13, 15.12, 15.11, 15.10, 15.9, 15.8, 15.7, 15.6, 15.5, 15.4
15.2	16.9, 16.8, 16.7, 16.6, 16.5, 16.4, 16.3, 16.2, 16.1 15.13, 15.12, 15.11, 15.10, 15.9, 15.8, 15.7, 15.6, 15.5, 15.4
14.18	17.5 16.9 15.13
14.17	17.4 16.8 15.13, 15.12 14.18

Versão de origem atual	Destinos de atualização
14.16	17.3 16.7 15.13, 15.12, 15.11 14.18, 14.17
14.15	17.2 16.6 15.13, 15.12, 15.11, 15.10 14.18, 14.17, 14.16
14.14	17.1 16.5 15.13, 15.12, 15.11, 15.10, 15.9 14.18, 14.17, 14.16, 14.15
14.13	16.4 15.13, 15.12, 15.11, 15.10, 15.9, 15.8 14.18, 14.17, 14.16, 14.15, 14.14
14.12	16.3 15.13, 15.12, 15.11, 15.10, 15.9, 15.8, 15.7 14.18, 14.17, 14.16, 14.15, 14.14, 14.13

Versão de origem atual	Destinos de atualização
14.11	16.2 15.13, 15.12, 15.11, 15.10, 15.9, 15.8, 15.7, 15.6 14.18, 14.17, 14.16, 14.15, 14.14, 14.13, 14.12
14.10	16.1 15.13, 15.12, 15.11, 15.10, 15.9, 15.8, 15.7, 15.6, 15.5 14.18, 14.17, 14.16, 14.15, 14.14, 14.13, 14.12, 14.11
14.9	15.13, 15.12, 15.11, 15.10, 15.9, 15.8, 15.7, 15.6, 15.5, 15.4 14.18, 14.17, 14.16, 14.15, 14.14, 14.13, 14.12, 14.11, 14.10
14.8*	15.13, 15.12, 15.11, 15.10, 15.9, 15.8, 15.7, 15.6, 15.5, 15.4 14.18, 14.17, 14.16, 14.15, 14.14, 14.13, 14.12, 14.11, 14.10, 14.9
14.7.*	15.13, 15.12, 15.11, 15.10, 15.9, 15.8, 15.7, 15.6, 15.5, 15.4 14.18, 14.17, 14.16, 14.15, 14.14, 14.13, 14.12, 14.11, 14.10, 14.9
14.6*	15.13, 15.12, 15.11, 15.10, 15.9, 15.8, 15.7, 15.6, 15.5, 15.4 14.18, 14.17, 14.16, 14.15, 14.14, 14.13, 14.12, 14.11, 14.10, 14.9

Versão de origem atual	Destinos de atualização
14.5*	15.13, 15.12, 15.11, 15.10, 15.9, 15.8, 15.7, 15.6, 15.5, 15.4 14.18, 14.17, 14.16, 14.15, 14.14, 14.13, 14.12, 14.11, 14.10, 14.9
14.4*	15.13, 15.12, 15.11, 15.10, 15.9, 15.8, 15.7, 15.6, 15.5, 15.4 14.18, 14.17, 14.16, 14.15, 14.14, 14.13, 14.12, 14.11, 14.10, 14.9
14.3*	15.13, 15.12, 15.11, 15.10, 15.9, 15.8, 15.7, 15.6, 15.5, 15.4 14.18, 14.17, 14.16, 14.15, 14.14, 14.13, 14.12, 14.11, 14.10, 14.9
14.2*	15.13, 15.12, 15.11, 15.10, 15.9, 15.8, 15.7, 15.6, 15.5, 15.4 14.18, 14.17, 14.16, 14.15, 14.14, 14.13, 14.12, 14.11, 14.10, 14.9
14.1*	15.13, 15.12, 15.11, 15.10, 15.9, 15.8, 15.7, 15.6, 15.5, 15.4 14.18, 14.17, 14.16, 14.15, 14.14, 14.13, 14.12, 14.11, 14.10, 14.9
13.21	17.5 16.9 15.13 14.17

Versão de origem atual	Destinos de atualização
13.20	17.4 16.8 15.12 14.18, 14.17 13.21
13.19	17.3 16.7 15.11 14.18, 14.17, 14.16 13.21, 13.20
13.18	16.6 15.10 14.18, 14.17, 14.15 13.21, 13.20, 13.19
13.17	16.5 15.9 14.18, 14.17, 14.15, 14.14 13.21, 13.20, 13.19, 13.18

Versão de origem atual	Destinos de atualização
13.16	16.4 15.8 14.18, 14.17, 14.15, 14.14, 14.13 13.21, 13.20, 13.19, 13.18, 13.17
13.15	16.3 15.8, 15.7 14.18, 14.17, 14.15, 14.14, 14.13, 14.12 13.21, 13.20, 13.19, 13.18, 13.17, 13.16
13.14	16.2 15.6 14.18, 14.17, 14.15, 14.14, 14.13, 14.12, 14.11 13.21, 13.20, 13.19, 13.18, 13.17, 13.16, 13.15
13.13	16.1 15.5 14.18, 14.17, 14.15, 14.14, 14.13, 14.12, 14.11, 14.10 13.21, 13.20, 13.19, 13.18, 13.17, 13.16, 13.15, 13.14

Versão de origem atual	Destinos de atualização
13.12	15.4 14.18, 14.17, 14.15, 14.14, 14.13, 14.12, 14.11, 14.10, 14.9 13.21, 13.20, 13.19, 13.18, 13.17, 13.16, 13.15, 13.14, 13.13
13.11	15.3 14.18, 14.17, 14.15, 14.14, 14.13, 14.12, 14.11, 14.10, 14.9 13.21, 13.20, 13.19, 13.18, 13.17, 13.16, 13.15, 13.14, 13.13, 13.12
13.10	15.2 14.18, 14.17, 14.15, 14.14, 14.13, 14.12, 14.11, 14.10, 14.9 13.21, 13.20, 13.19, 13.18, 13.17, 13.16, 13.14, 13.13, 13.12, 13.11
13.9	14.18, 14.17, 14.15, 14.14, 14.13, 14.12, 14.11, 14.10, 14.9 13.21, 13.20, 13.19, 13.18, 13.17, 13.16, 13.14, 13.13, 13.12, 13.11
13.8	14.18, 14.17, 14.1514.14, 14.13, 14.12, 14.11, 14.10, 14.9 13.21, 13.20, 13.19, 13.18, 13.17, 13.16, 13.14, 13.13, 13.12, 13.11

Versão de origem atual	Destinos de atualização
13.7	14.18, 14.17, 14.15, 14.14, 14.13, 14.12, 14.11, 14.10, 14.9 13.21, 13.20, 13.19, 13.18, 13.17, 13.16, 13.14, 13.13, 13.11
13.6*	14.18, 14.17, 14.15, 14.14, 14.13, 14.12, 14.11, 14.10, 14.9 13.21, 13.20, 13.19, 13.18, 13.17, 13.16, 13.14, 13.13, 13.11
13.5*	14.18, 14.17, 14.15, 14.14, 14.13, 14.12, 14.11, 14.10, 14.9 13.21, 13.20, 13.19, 13.18, 13.17, 13.16, 13.14, 13.13, 13.11
13.4*	14.18, 14.17, 14.15, 14.14, 14.13, 14.12, 14.11, 14.10, 14.9 13.21, 13.20, 13.19, 13.18, 13.17, 13.16, 13.14, 13.13, 13.11
13.3*	14.18, 14.17, 14.15, 14.14, 14.13, 14.12, 14.11, 14.10, 14.9 13.21, 13.20, 13.19, 13.18, 13.17, 13.16, 13.14, 13.13, 13.11
13.2*, 13.1*	14.18, 14.17, 14.15, 14.14, 14.13, 14.12, 14.11, 14.10, 14.9 13.21, 13.20, 13.19, 13.18, 13.17, 13.16, 13.14, 13.13, 13.11

Versão de origem atual	Destinos de atualização
12.22	17.2 16.6 15.10 14.15 13.18
12.21	17.1 16.5 15.9 14.14 13.18, 13.17 12.22
12.20	16.4 15.8 14.13 13.18, 13.17, 13.16 12.22, 12.21
12.19	16.3 15.7 14.12 13.18, 13.17, 13.16, 13.15 12.22, 12.21, 12.20

Versão de origem atual	Destinos de atualização
12.18	16.2 15.6 14.11 13.18, 13.17, 13.16, 13.15, 13.14 12.22, 12.21, 12.20, 12.19
12.17	16.1 15.5 14.10 13.18, 13.17, 13.16, 13.15, 13.14, 13.13 12.22, 12.21, 12.20, 12.19, 12.18
12.16	15.4 14.9 13.18, 13.17, 13.16, 13.15, 13.14, 13.13, 13.12 12.22, 12.21, 12.20, 12.19, 12.18, 12.17
12.15	15.3 14.8 13.18, 13.17, 13.16, 13.15, 13.14, 13.13, 13.12, 13.11 12.22, 12.21, 12.2012.19, 12.18, 12.17, 12.16

Versão de origem atual	Destinos de atualização
12.14	15.2 14.7 13.18, 13.17, 13.16, 13.15, 13.14, 13.13, 13.12, 13.11 12.22, 12.21, 12.20, 12.19, 12.18, 12.17, 12.16, 12.15
12.13	14.6 13.18, 13.17, 13.16, 13.15, 13.14, 13.13, 13.12, 13.11 12.22, 12.21, 12.20, 12.19, 12.18, 12.17, 12.16, 12.15
12.12*	14.5 13.18, 13.17, 13.16, 13.15, 13.14, 13.13, 13.12, 13.11 12.22, 12.21, 12.20, 12.19, 12.18, 12.17, 12.16, 12.15
12.11	14,4 13.18, 13.17, 13.16, 13.15, 13.14, 13.13, 13.12, 13.11 12.22, 12.21, 12.20, 12.19, 12.18, 12.17, 12.16, 12.15

Versão de origem atual	Destinos de atualização
12.10*	14.2 13.18, 13.17, 13.16, 13.15, 13.14, 13.13, 13.12, 13.11 12.22, 12.21, 12.20, 12.19, 12.18, 12.17, 12.16, 12.15
12.9*	14.1 13.18, 13.17, 13.16, 13.15, 13.14, 13.13, 13.12, 13.11 12.22, 12.21, 12.20, 12.19, 12.18, 12.17, 12.16, 12.15
12.8*	13.18, 13.17, 13.16, 13.15, 13.14, 13.13, 13.12, 13.11 12.22, 12.21, 12.20, 12.19, 12.18, 12.17, 12.16, 12.15
12.7*	13.18, 13.17, 13.16, 13.15, 13.14, 13.13, 13.12, 13.11 12.22, 12.21, 12.20, 12.19, 12.18, 12.17, 12.16, 12.15
12.6*, 12.5*, 12.4*, 12.3*, 12.2*	13.18, 13.17, 13.16, 13.15, 13.14, 13.13, 13.12, 13.11 12.22, 12.21, 12.20, 12.19, 12.18, 12.17, 12.16, 12.15

Versão de origem atual	Destinos de atualização
11.22	16.1 15.5 14.10 13.13 12.17 11.22-RDS.20240418

* Esta versão não é mais compatível.

Como atualizar a versão principal do RDS para PostgreSQL

Recomendamos o seguinte processo ao realizar uma atualização de versão principal em um banco de dados do Amazon RDS para PostgreSQL:

1. Tenha um grupo de parâmetros compatível com a versão pronto para uso – Se você estiver usando um grupo de parâmetros personalizado, terá duas opções. Você pode especificar um grupo de parâmetros padrão para a nova versão do mecanismo de banco de dados. Ou você pode criar seu próprio grupo de parâmetros personalizado para a nova versão do mecanismo de banco de dados. Para ter mais informações, consulte the section called “Grupos de parâmetros” e the section called “Grupos de parâmetros de cluster de banco de dados”.
2. Verifique se há classes de banco de dados não compatíveis: confirme se a classe de instância do banco de dados é compatível com a versão do PostgreSQL para a qual você está atualizando. Para ter mais informações, consulte Mecanismos de banco de dados compatíveis para classes de instância de banco de dados.
3. Verifique o uso sem suporte:
   - Transações preparadas – Confirme ou reverta todas as transações preparadas abertas antes de tentar uma atualização.

     Você pode usar a consulta a seguir para verificar se não há transações preparadas já abertas no banco de dados.

```
SELECT count(*) FROM pg_catalog.pg_prepared_xacts;
```

- Tipos de dados reg* – Remova todos os usos dos tipos de dados reg* antes de tentar fazer uma atualização. Exceto por `regtype` e `regclass`, você não pode atualizar os tipos de dados reg*. O utilitário `pg_upgrade` não pode persistir esse tipo de dados, que é usado pelo Amazon RDS para fazer a atualização.

  Para verificar se não há nenhum uso de tipos de dados reg* sem suporte, use a consulta a seguir para cada banco de dados.

```
SELECT count(*) FROM pg_catalog.pg_class c, pg_catalog.pg_namespace n,
 pg_catalog.pg_attribute a
  WHERE c.oid = a.attrelid
      AND NOT a.attisdropped
      AND a.atttypid IN ('pg_catalog.regproc'::pg_catalog.regtype,
                         'pg_catalog.regprocedure'::pg_catalog.regtype,
                         'pg_catalog.regoper'::pg_catalog.regtype,
                         'pg_catalog.regoperator'::pg_catalog.regtype,
                         'pg_catalog.regconfig'::pg_catalog.regtype,
                         'pg_catalog.regdictionary'::pg_catalog.regtype)
      AND c.relnamespace = n.oid
      AND n.nspname NOT IN ('pg_catalog', 'information_schema');
```

4. Verifique se há bancos de dados inválidos:

   - Garanta que não haja bancos de dados inválidos. A coluna `datconnlimit` no catálogo `pg_database` inclui um valor de `-2` para marcar como inválidos bancos de dados que foram interrompidos durante uma operação `DROP DATABASE`.

     Use a seguinte consulta para verificar se há bancos de dados inválidos:

```
SELECT datname FROM pg_database WHERE datconnlimit = - 2;
```

   - A consulta anterior retorna nomes de banco de dados inválidos. É possível usar `DROP DATABASE` *`invalid_db_name`*`;` para descartar bancos de dados inválidos. Também é possível usar o seguinte comando para descartar bancos de dados inválidos:

```
SELECT 'DROP DATABASE ' || quote_ident(datname) || ';' FROM pg_database WHERE
 datconnlimit = -2 \gexec
```

Consulte mais informações sobre bancos de dados inválidos em Noções básicas sobre o comportamento do autovacuum com bancos de dados inválidos.

5. Gerencie os slots de replicação lógica: a atualização não será possível se a instância tiver algum slot de replicação lógica. Os slots de replicação lógica são normalmente usados para migração do AWS DMS e para replicar tabelas do banco de dados para data lakes, ferramentas de BI e outros destinos. Antes de atualizar, certifique-se de saber a finalidade de qualquer slot de replicação lógica que esteja em uso e confirme se não há problema em excluí-los. Se os slots de replicação lógica ainda estiverem sendo usados, você não deve excluí-los e não poderá continuar com a atualização.

   Se os slots de replicação lógica não forem necessários, você poderá excluí-los usando o seguinte SQL:

   ```
   SELECT * FROM pg_replication_slots WHERE slot_type NOT LIKE 'physical';
   SELECT pg_drop_replication_slot(slot_name);
   ```

   As configurações de replicação lógica que usam a extensão `pglogical` também precisam ter slots descartados para uma atualização bem-sucedida da versão principal. Para obter informações sobre como identificar e descartar slots criados utilizando a extensão `pglogical`, consulte Gerenciar slots de replicação lógica para RDS para PostgreSQL.

6. Gerencie as réplicas de leitura: a atualização de uma implantação de instância de banco de dados single-AZ ou multi-AZ também atualiza as réplicas de leitura na região e a instância de banco de dados primária. O Amazon RDS não atualiza réplicas de leitura de clusters de banco de dados multi-AZ.

   Não é possível atualizar as réplicas de leitura separadamente. Se fosse possível, isso poderia levar a situações em que os bancos de dados primário e de réplica têm diferentes versões principais do PostgreSQL. Contudo, as atualizações de réplicas de leitura podem aumentar o tempo de inatividade da instância de banco de dados primária. Para impedir uma atualização de réplica de leitura, promova a réplica para uma instância independente ou exclua-a antes de iniciar o processo de atualização.

O processo de atualização recria o grupo de parâmetros da réplica de leitura com base no grupo de parâmetros atual da réplica de leitura. Você só pode aplicar um grupo de parâmetros personalizado a uma réplica de leitura após a conclusão da atualização modificando a réplica de leitura. Para obter mais informações sobre réplicas de leitura, consulte Trabalhar com réplicas de leitura do Amazon RDS para PostgreSQL.

7. Faça um backup – recomendamos que você faça um backup antes de atualizar a versão principal. Assim, você terá um ponto de restauração conhecido para seu banco de dados. Se o período de retenção de backup for maior que 0, o processo de atualização criará snapshots do banco de dados antes e depois da atualização. Para alterar o período de retenção de backup, consulte Modificar uma instância de banco de dados do Amazon RDS e the section called “Modificar um cluster de banco de dados multi-AZ”.

   Para realizar um backup manualmente, consulte the section called “Criar um snapshot de banco de dados para uma instância de banco de dados de uma única zona de disponibilidade” e the section called “Criar um snapshot de cluster de banco de dados Multi-AZ”.

8. Atualizar determinadas extensões antes de uma atualização da versão principal: se você planeja ignorar uma versão principal com a atualização, é necessário atualizar determinadas extensões antes de executar a atualização da versão principal. Por exemplo, fazer upgrade das versões 9.5.x ou 9.6.x para uma versão 11.x ignora uma versão principal. As extensões a serem atualizadas incluem PostGIS e extensões relacionadas para processamento de dados espaciais.

   - `address_standardizer`
   - `address_standardizer_data_us`
   - `postgis_raster`
   - `postgis_tiger_geocoder`
   - `postgis_topology`

   Você não pode atualizar diretamente para a versão 17 do PostgreSQL se estiver usando `rdkit` versão 4.6.0 e anterior e o PostgreSQL versão 16 e anterior, devido à incompatibilidade `rdkit`. Confira abaixo as opções de atualização:

   - Se você estiver usando a versão 13 ou anterior do PostgreSQL, precisará primeiro realizar uma atualização da versão principal para a versão 14.14 ou versões 14 posteriores, 15.9 ou versões 15 posteriores, ou 16.5 ou versões 16 posteriores e, em seguida, realizar a atualização para o PostgreSQL 17.
   - Se você estiver usando a versão 14, 15 ou 16 do PostgreSQL, precisará realizar uma atualização de versão secundária para 14.14 ou versões 14 posteriores, 15.9 ou versões 15

posteriores, ou 16.5 ou versões 16 posteriores e, em seguida, atualizar para a versão 17 do PostgreSQL.

Execute o comando a seguir para cada extensão que você está usando:

```
ALTER EXTENSION PostgreSQL-extension UPDATE TO 'new-version';
```

Para ter mais informações, consulte Atualizar extensões do PostgreSQL em bancos de dados do RDS para PostgreSQL. Para saber mais sobre como fazer upgrade do PostGIS, consulte Etapa 6: Atualize a extensão PostGIS.

9. Descartar determinadas extensões antes da atualização da versão principal – uma atualização que avança de uma versão principal para a versão 11.x não é compatível com a atualização da extensão `pgRouting`. O upgrade das versões 9.4.x, 9.5.x ou 9.6.x para as versões 11.x ignora uma versão principal. É seguro descartar a extensão `pgRouting` e reinstalá-la em uma versão compatível após a atualização. Para obter versões de extensão para as quais você pode atualizar, consulte Versões de extensões do PostgreSQL compatíveis.

   As extensões `tsearch2` e `chkpass` não são mais compatíveis com o PostgreSQL versões 11 ou posterior. Se você estiver atualizando para a versão 11.x, descarte as extensões `tsearch2` e `chkpass` antes da atualização.

10. Libere os tipos de dados desconhecidos – libere os tipos de dados `unknown` dependendo da versão de destino.

    O PostgreSQL versão 10 descontinuou o suporte para tipos de dados `unknown`. Se um banco de dados da versão 9.6 utilizar o tipo de dados `unknown`, uma atualização para uma versão 10 exibirá uma mensagem de erro como a seguinte:

    ```
    Database instance is in a state that cannot be upgraded: PreUpgrade checks failed:
    The instance could not be upgraded because the 'unknown' data type is used in user
     tables.
    Please remove all usages of the 'unknown' data type and try again."
    ```

    Para localizar o tipo de dados `unknown` no banco de dados para remover a coluna incorreta ou para alterar para um tipo de dados compatível, use o seguinte SQL:

    ```
    SELECT DISTINCT data_type FROM information_schema.columns WHERE data_type ILIKE
     'unknown';
    ```

11. Realizar uma simulação – é altamente recomendável testar uma atualização da versão principal em uma cópia do banco de dados de produção antes de fazer a atualização no banco de dados de produção. Você pode monitorar os planos de execução no banco de dados de teste duplicado para detectar possíveis regressões do plano de execução e avaliar a performance. Para criar uma instância de teste duplicada, restaure o banco de dados a partir de um snapshot recente ou faça uma restauração point-in-time para o último momento restaurável.

    Para obter mais informações, consulte the section called “Restauração a partir de um snapshot” ou the section called “Recuperação para um ponto no tempo”. Para clusters de banco de dados multi-AZ, consulte the section called “Restaurar a partir de um snapshot para cluster de banco de dados multi-AZ” ou the section called “Restaurar um cluster de banco de dados multi-AZ para um horário especificado”.

    Para obter detalhes sobre como realizar a atualização, consulte the section called “Atualizar manualmente a versão do mecanismo”.

    Ao atualizar um banco de dados versão 9.6 para a versão 10, esteja ciente de que o PostgreSQL 10 habilita consultas paralelas por padrão. Teste o impacto do paralelismo antes da atualização alterando o parâmetro `max_parallel_workers_per_gather` no banco de dados de teste para 2.

    > **Note**
    >
    > O valor padrão para o parâmetro `max_parallel_workers_per_gather` no grupo de parâmetros de banco de dados `default.postgresql10` é 2.

    Para obter mais informações, consulte o tópico sobre Parallel Query na documentação do PostgreSQL. Para desabilitar o paralelismo na versão 10, defina o parâmetro `max_parallel_workers_per_gather` para 0.

    Durante a atualização da versão principal, os bancos de dados `public` e `template1` e o esquema de `public` em cada banco de dados são renomeados temporariamente. Esses objetos aparecerão nos logs com o nome original e uma string aleatória anexada. A string é anexada para que as configurações personalizadas, como `locale` e `owner`, sejam preservadas durante a atualização da versão principal. Quando a atualização for concluída, os objetos serão renomeados de volta com seus nomes originais.

> **Note**
>
> Durante o processo de atualização da versão principal, não é possível fazer uma restauração para um ponto no tempo da instância de banco de dados ou do cluster de banco de dados multi-AZ. Depois que o Amazon RDS realizar a atualização, ele usará um backup automático do banco de dados. Você pode fazer uma restauração para um ponto no tempo anterior à atualização e posterior ao backup automático do banco de dados.

12. Se uma atualização falhar com erros de procedimento de pré-verificação, resolva os problemas – durante o processo de atualização da versão principal, o Amazon RDS para PostgreSQL executa primeiro um procedimento de pré-verificação para identificar quaisquer problemas que possam causar falha na atualização. O procedimento de pré-verificação verifica todas as possíveis condições incompatíveis em todos os bancos de dados da instância.

    Se a pré-verificação encontrar um problema, ele criará um evento de log indicando que a pré-verificação da atualização falhou Os detalhes do processo de pré-verificação estão em um log de atualização chamado `pg_upgrade_precheck.log` para todos os bancos de dados de um banco de dados. O Amazon RDS acrescenta a data e a hora ao nome de arquivo. Para obter mais informações sobre como visualizar logs, consulte Monitorar arquivos de log do Amazon RDS.

    Se uma atualização de réplica de leitura falhar na pré-verificação, a replicação na réplica de leitura com falha será interrompida e a réplica de leitura ficará com estado de encerrada. Exclua a réplica de leitura e recrie uma nova com base na instância de banco de dados primária atualizada.

    Resolva todos os problemas identificados no log de pré-verificação e tente fazer a atualização da versão principal novamente. Veja a seguir um exemplo de log de pré-verificação.

```
----------------------------------------------------------------------
Upgrade could not be run on Wed Apr 4 18:30:52 2018
----------------------------------------------------------------------
The instance could not be upgraded from 9.6.11 to 10.6 for the following reasons.
Please take appropriate action on databases that have usage incompatible with the
 requested major engine version upgrade and try the upgrade again.

* There are uncommitted prepared transactions. Please commit or rollback all prepared
 transactions.* One or more role names start with 'pg_'. Rename all role names that
 start with 'pg_'.
```

```
* The following issues in the database 'my"million$"db' need to be corrected before
 upgrading:** The ["line","reg*"] data types are used in user tables. Remove all
 usage of these data types.
** The database name contains characters that are not supported by RDS for
 PostgreSQL. Rename the database.
** The database has extensions installed that are not supported on the target
 database version. Drop the following extensions from your database: ["tsearch2"].

* The following issues in the database 'mydb' need to be corrected before
 upgrading:** The database has views or materialized views that depend on
 'pg_stat_activity'. Drop the views.
```

13. Se uma atualização de réplica de leitura falhar durante a atualização do banco de dados, resolva o problema. Uma réplica de leitura com falha será colocada no estado `incompatible-restore` e a replicação será encerrada no banco de dados. Exclua a réplica de leitura e recrie uma nova com base na instância de banco de dados primária atualizada.

> **Note**
>
> O Amazon RDS não atualiza réplicas de leitura para clusters de banco de dados multi-AZ. Se você realizar uma atualização de versão principal em um cluster de banco de dados multi-AZ, o estado da replicação das réplicas de leitura mudará para Encerrado.

   Uma atualização de réplica de leitura pode falhar pelos seguintes motivos:

   - Não foi possível acompanhar a instância de banco de dados primária, mesmo após a espera.
   - Ela estava em um estado de ciclo de vida incompatível ou terminal, como armazenamento esgotado, restauração incompatível e assim por diante.
   - Quando a atualização da instância de banco de dados primária foi iniciada, havia uma atualização de versão secundária separada em execução na réplica de leitura.
   - A réplica de leitura usou parâmetros incompatíveis.
   - A réplica de leitura não pôde se comunicar com a instância de banco de dados primária para sincronizar a pasta de dados.

14. Atualize o banco de dados de produção: quando a simulação da atualização da versão principal é realizada com êxito, você pode atualizar o banco de dados de produção com segurança. Para ter mais informações, consulte Atualizar manualmente a versão do mecanismo.

15. Execute a operação `ANALYZE` para atualizar a tabela `pg_statistic`. Você deve fazer isso para cada banco de dados em todos os bancos de dados do PostgreSQL. As estatísticas do otimizador

não são transferidas durante uma atualização de versão principal, portanto, você precisa gerar novamente todas as estatísticas para evitar problemas de performance. Execute o comando sem nenhum parâmetro para gerar estatísticas para todas as tabelas regulares no banco de dados atual da seguinte forma:

```
ANALYZE VERBOSE;
```

O sinalizador `VERBOSE` é opcional, mas usá-lo mostra o progresso. Para obter mais informações, consulte ANALYZE na documentação do PostgreSQL.

> **Note**
>
> Execute ANALYZE em seu sistema após a atualização para evitar problemas de performance.

Após a conclusão da atualização da versão principal, recomendamos o seguinte:

- Uma atualização do PostgreSQL não atualiza nenhuma extensão do PostgreSQL. Para atualizar extensões, consulte Atualizar extensões do PostgreSQL em bancos de dados do RDS para PostgreSQL.
- Como opção, use o Amazon RDS para visualizar dois logs que o utilitário `pg_upgrade` produz. Esses logs são `pg_upgrade_internal.log` e `pg_upgrade_server.log`. O Amazon RDS acrescenta a data e a hora ao nome de arquivo desses logs. Você pode visualizar esses logs como visualiza qualquer outro log. Para obter mais informações, consulte Monitorar arquivos de log do Amazon RDS.

  Você também pode fazer upload dos logs de atualização para o Amazon CloudWatch Logs. Para obter mais informações, consulte Publicação de logs do PostgreSQL no Amazon CloudWatch Logs.
- Para verificar se tudo funciona como esperado, teste o aplicativo no banco de dados atualizado com uma workload semelhante. Verificada a atualização, é possível excluir essa instância de teste.

Atualizações automáticas da versão secundária para o RDS para PostgreSQL

Se habilitar a opção Upgrade automático de versões secundárias ao criar ou modificar uma instância de banco de dados ou um cluster de banco de dados multi-AZ, o banco de dados poderá ser atualizado automaticamente.

Para cada versão principal do RDS para PostgreSQL, uma versão secundária é designada pelo RDS como a versão de atualização automática. Depois que uma versão secundária foi testada e aprovada pelo Amazon RDS, a atualização de versão secundária ocorre automaticamente durante a janela de manutenção. O RDS não define automaticamente versões secundárias lançadas mais recentemente como a versão de atualização automática. Antes de o RDS designar uma versão de atualização automática, diversos critérios são considerados, como estes:

- Problemas de segurança conhecidos
- Bugs na versão de comunidade do PostgreSQL
- Estabilidade geral de frota desde que a versão secundária foi lançada.

É possível utilizar o comando da AWS CLI a seguir para determinar a versão atual secundária de destino de atualização automática para determinada versão secundária do PostgreSQL em uma Região da AWS específica.

Para Linux, macOS ou Unix:

```
aws rds describe-db-engine-versions \
--engine postgres \
--engine-version minor-version \
--region region \
--query "DBEngineVersions[*].ValidUpgradeTarget[*].
{AutoUpgrade:AutoUpgrade,EngineVersion:EngineVersion}" \
--output text
```

Para Windows:

```
aws rds describe-db-engine-versions ^
--engine postgres ^
--engine-version minor-version ^
--region region ^
```

```
--query "DBEngineVersions[*].ValidUpgradeTarget[*].
{AutoUpgrade:AutoUpgrade,EngineVersion:EngineVersion}" ^
--output text
```

Por exemplo, o comando da AWS CLI a seguir determina o destino de atualização secundária automática para a versão 16.1 secundária do PostgreSQL na Região da AWS Leste dos EUA (Ohio) (us-east-2).

Para Linux, macOS ou Unix:

```
aws rds describe-db-engine-versions \
--engine postgres \
--engine-version 16.1 \
--region us-east-2 \
--query "DBEngineVersions[*].ValidUpgradeTarget[*].
{AutoUpgrade:AutoUpgrade,EngineVersion:EngineVersion}" \
--output table
```

Para Windows:

```
aws rds describe-db-engine-versions ^
--engine postgres ^
--engine-version 16.1 ^
--region us-east-2 ^
--query "DBEngineVersions[*].ValidUpgradeTarget[*].
{AutoUpgrade:AutoUpgrade,EngineVersion:EngineVersion}" ^
--output table
```

A saída é semelhante à seguinte.

```
---------------------------------
|    DescribeDBEngineVersions    |
+--------------+-----------------+
|  AutoUpgrade |  EngineVersion  |
+--------------+-----------------+
|  False       |  16.2           |
|  True        |  16.3           |
|  False       |  16.4           |
|  False       |  16.5           |
|  False       |  16.6           |
|  False       |  17.1           |
```

```
|  False       |  17.2           |
+--------------+-----------------+
```

Neste exemplo, o valor de `AutoUpgrade` é `True` para o PostgreSQL versão 16.3. Portanto, o destino de atualização secundária automática é o PostgreSQL versão 16.3, que está destacado na saída.

Um banco de dados do PostgreSQL será atualizado automaticamente durante a janela de manutenção, caso os seguintes critérios sejam atendidos:

- O banco de dados tem a opção Upgrade automático de versões secundárias habilitada.
- O banco de dados está executando uma versão de mecanismo de banco de dados secundária que é anterior à versão secundária de atualização automática.

Para ter mais informações, consulte Atualizar automaticamente a versão do mecanismo espelho.

> **Note**
>
> Uma atualização do PostgreSQL não atualiza extensões do PostgreSQL. Para atualizar extensões, consulte Atualizar extensões do PostgreSQL em bancos de dados do RDS para PostgreSQL.

Atualizar extensões do PostgreSQL em bancos de dados do RDS para PostgreSQL

Uma atualização de mecanismo do PostgreSQL não atualiza a maior parte das extensão do PostgreSQL. Para atualizar uma extensão após uma atualização de versão, use o comando `ALTER EXTENSION UPDATE`.

> **Note**
>
> Para obter informações sobre como atualizar a extensão PostGIS, consulte Gerenciar dados espaciais com a extensão PostGIS (Etapa 6: Atualize a extensão PostGIS).
> Para atualizar a extensão `pg_repack`, solte a extensão e crie a versão no banco de dados atualizado. Para obter mais informações, consulte a instalação do pg_repack na documentação do `pg_repack`.

Para atualizar uma extensão, use o seguinte comando.

```
ALTER EXTENSION extension_name UPDATE TO 'new_version';
```

Para obter a lista de versões com suporte das extensões do PostgreSQL, consulte Versões de extensões do PostgreSQL compatíveis.

Para listar as extensões instaladas no momento, use o catálogo pg_extension do PostgreSQL no comando a seguir.

```
SELECT * FROM pg_extension;
```

Para visualizar uma lista das versões de extensão específicas disponíveis para a instalação, use a visualização pg_available_extension_versions do PostgreSQL no comando a seguir.

```
SELECT * FROM pg_available_extension_versions;
```

Atualizar uma versão do mecanismo de snapshot de banco de dados PostgreSQL

Com o Amazon RDS, você pode criar um snapshot de banco de dados do volume de armazenamento de sua instância de banco de dados PostgreSQL. Quando você cria um snapshot de banco de dados, ele é baseado na versão do mecanismo usada pela instância do Amazon RDS. É possível fazer a atualização da versão do mecanismo para os snapshots de banco de dados.

Após restaurar um snapshot de banco de dados atualizado para uma nova versão do mecanismo, certifique-se de testar se a atualização foi bem-sucedida. Para obter mais informações sobre uma atualização de versão principal, consulte Atualizações do mecanismo de banco de dados do RDS para PostgreSQL. Para saber como restaurar um snapshot de banco de dados, consulte Restaurar uma instância de banco de dados.

Você pode atualizar snapshots de banco de dados manuais criptografados ou não.

Para visualizar as versões de mecanismo disponíveis para o snapshot de banco de dados do RDS para PostgreSQL, use o exemplo da AWS CLI a seguir.

```
aws rds describe-db-engine-versions --engine postgres  --engine-version example-
engine-version --query "DBEngineVersions[*].ValidUpgradeTarget[*].
{EngineVersion:EngineVersion}" --output text --include-all
```

Consulte mais informações sobre as versões de mecanismo disponíveis para snapshots de banco de dados do RDS para PostgreSQL em Escolher uma versão principal para uma atualização do RDS para PostgreSQL.

> **Note**
>
> Não é possível atualizar snapshots de banco de dados automatizados criados durante o processo de backup automatizado.

Console

Para atualizar um snapshot de banco de dados

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.

2. No painel de navegação, selecione Snapshots.
3. Escolha o snapshot que você deseja atualizar.
4. Em Actions (Ações), selecione Upgrade snapshot (Atualizar snapshot). A página Upgrade snapshot (Atualizar snapshot) é exibida.
5. Escolha a opção New engine version (Nova versão de mecanismo) para a qual atualizar.
6. Escolha Save changes (Salvar alterações) para atualizar o snapshot.

   Durante o processo de atualização, todas as ações do snapshot serão desabilitadas para esse snapshot de banco de dados. Além disso, o status do snapshot de banco de dados muda de available (disponível) para upgrading (atualizando) e muda para active (ativo) após a conclusão. Se não for possível atualizar o snapshot de banco de dados devido a problemas de corrupção do snapshot, o status mudará para unavailable (indisponível). Não é possível recuperar o snapshot desse estado.

   > **Note**
   >
   > Se a atualização do snapshot de banco de dados falhar, o snapshot será revertido para o estado original com a versão original.

AWS CLI

Para atualizar um snapshot de banco de dados para uma nova versão do mecanismo de banco de dados, use o comando modify-db-snapshot da AWS CLI.

Parâmetros

- `--db-snapshot-identifier` – o identificador do snapshot de banco de dados a ser atualizado. O identificador deve ser um nome de recurso da Amazon (ARN) exclusivo. Para obter mais informações, consulte Nomes de recurso da Amazon (ARNs) no Amazon RDS.
- `--engine-version` – a versão do mecanismo para a qual será feita a atualização do snapshot de banco de dados.

Example

Para Linux, macOS ou Unix:

```
aws rds modify-db-snapshot \
```

```
    --db-snapshot-identifier my_db_snapshot \
    --engine-version new_version
```

Para Windows:

```
aws rds modify-db-snapshot ^
    --db-snapshot-identifier my_db_snapshot ^
    --engine-version new_version
```

API do RDS

Para atualizar um snapshot de banco de dados para uma nova versão do mecanismo de banco de dados, chame a operação ModifyDBSnapshot da API do Amazon RDS.

- `DBSnapshotIdentifier` – o identificador do snapshot de banco de dados a ser atualizado. O identificador deve ser um nome de recurso da Amazon (ARN) exclusivo. Para obter mais informações, consulte Nomes de recurso da Amazon (ARNs) no Amazon RDS.
- `EngineVersion` – a versão do mecanismo para a qual será feita a atualização do snapshot de banco de dados.

Trabalhar com réplicas de leitura do Amazon RDS para PostgreSQL

É possível escalar as leituras da instância de banco de dados do Amazon RDS para PostgreSQL adicionando réplicas de leitura à instância. Assim como acontece com outros mecanismos de banco de dados do Amazon RDS, o RDS para PostgreSQL usa os mecanismos de replicação nativos do PostgreSQL para manter as réplicas de leitura atualizadas com as alterações no banco de dados de origem. Para obter informações gerais sobre as réplicas de leitura e o Amazon RDS, consulte Trabalhar com réplicas de leitura de instância de banco de dados.

A seguir, você pode encontrar informações específicas sobre como trabalhar com réplicas de leitura com o RDS para PostgreSQL.

Decodificação lógica em uma réplica de leitura

O RDS para PostgreSQL comporta a replicação lógica em espera com o PostgreSQL 16.1. Isso permite criar uma decodificação lógica a partir de um modo de espera somente leitura que reduza a carga na instância de banco de dados primária. É possível ter maior disponibilidade para as aplicações que precisam sincronizar dados em vários sistemas. Esse recurso melhora a performance do data warehouse e da análise de dados.

Além disso, os slots de replicação em um modo de espera específico mantêm a promoção desse modo de espera para um primário. Isso significa que, no caso de um failover de uma instância de banco de dados primária ou da promoção de uma instância de espera para a nova primária, os slots de replicação serão mantidos e os antigos assinantes em espera não serão afetados.

Como criar a decodificação lógica em uma réplica de leitura

1. Ativar a replicação lógica: para criar a decodificação lógica em espera, é necessário ativar a replicação lógica na instância de banco de dados de origem e na réplica física. Para obter mais informações, consulte Configuração de réplica de leitura com o PostgreSQL.

    - Para ativar a replicação lógica para uma instância de banco de dados do RDS para PostgreSQL recém-criada, crie um grupo de parâmetros de banco de dados personalizado e defina o parâmetro estático `rds.logical_replication` como `1`. Depois, associe esse grupo de parâmetros de banco de dados à instância de banco de dados de origem e à réplica de leitura física. Para obter mais informações, consulte Associar um grupo de parâmetros de banco de dados a uma instância de banco de dados no Amazon RDS.

- Para ativar a replicação lógica para uma instância de banco de dados do RDS para PostgreSQL existente: modifique o grupo de parâmetros personalizado do banco de dados da instância de banco de dados de origem e à réplica de leitura física para definir o parâmetro estático `rds.logical_replication` como 1. Para obter mais informações, consulte Modificar parâmetros em um grupo de parâmetros de banco de dados no Amazon RDS.

> **Note**
>
> É necessário reinicializar a instância de banco de dados para aplicar essas alterações de parâmetros.

É possível usar a consulta a seguir para verificar os valores para `wal_level` e `rds.logical_replication` na instância de banco de dados de origem e na réplica de leitura física.

```
Postgres=>SELECT name,setting FROM pg_settings WHERE name IN
 ('wal_level','rds.logical_replication');

 name                    | setting
-------------------------+---------
 rds.logical_replication | on
 wal_level               | logical
(2 rows)
```

2. Criar uma tabela no banco de dados de origem: conecte-se ao banco de dados na instância de banco de dados de origem. Para obter mais informações, consulte Conectar a uma instância de banco de dados executando o mecanismo de banco de dados do PostgreSQL.

   Use as seguintes consultas para criar uma tabela no banco de dados de origem e inserir valores:

```
Postgres=>CREATE TABLE LR_test (a int PRIMARY KEY);
CREATE TABLE
```

```
Postgres=>INSERT INTO LR_test VALUES (generate_series(1,10000));
INSERT 0 10000
```

3. Criar uma publicação para a tabela de origem: use a consulta a seguir para criar uma publicação para a tabela na instância de banco de dados de origem.

```
Postgres=>CREATE PUBLICATION testpub FOR TABLE LR_test;
CREATE PUBLICATION
```

Use uma consulta SELECT para verificar os detalhes da publicação que foi criada na instância de banco de dados de origem e na instância de réplica de leitura física.

```
Postgres=>SELECT * from pg_publication;

oid    | pubname | pubowner | puballtables | pubinsert | pubupdate | pubdelete |
 pubtruncate | pubviaroot
-------+---------+----------+--------------+-----------+-----------+-----------
+-------------+------------
 16429 | testpub |    16413 | f            | t         | t         | t         | t
          | f
(1 row)
```

4. Criar uma assinatura a partir da instância de réplica lógica: crie outra instância de banco de dados do RDS para PostgreSQL como a instância de réplica lógica. Verifique se a VPC está configurada corretamente para garantir que essa instância de réplica lógica possa acessar a instância de réplica de leitura física. Para obter mais informações, consulte Amazon VPC e Amazon RDS. Se a instância de banco de dados de origem estiver ociosa, poderão ocorrer problemas de conectividade e a primária não enviará os dados para o modo de espera.

```
Postgres=>CREATE SUBSCRIPTION testsub CONNECTION 'host=Physical replica host name
 port=port
                dbname=source_db_name user=user password=password'
 PUBLICATION testpub;
NOTICE:  created replication slot "testsub" on publisher
CREATE SUBSCRIPTION
```

```
Postgres=>CREATE TABLE LR_test (a int PRIMARY KEY);
CREATE TABLE
```

Use uma consulta SELECT para verificar os detalhes da assinatura na instância de réplica lógica.

```
Postgres=>SELECT oid,subname,subenabled,subslotname,subpublications FROM
 pg_subscription;

oid    | subname | subenabled | subslotname | subpublications
-------+---------+------------+-------------+-----------------
 16429 | testsub | t          | testsub     | {testpub}
(1 row)
postgres=> select count(*) from LR_test;
 count
-------
 10000
(1 row)
```

5. Inspecionar o estado do slot de replicação lógica: somente é possível ver o slot de replicação física na instância de banco de dados de origem.

```
Postgres=>select slot_name, slot_type, confirmed_flush_lsn from
 pg_replication_slots;

slot_name                                  | slot_type | confirmed_flush_lsn
-------------------------------------------+-----------+---------------------
 rds_us_west_2_db_dhqfsmo5wbbjqrn3m6b6ivdhu4 | physical  |
(1 row)
```

No entanto, na instância de réplica de leitura, é possível ver o slot de replicação lógica e o valor `confirmed_flush_lsn` muda à medida que a aplicação consome ativamente as alterações lógicas.

```
Postgres=>select slot_name, slot_type, confirmed_flush_lsn from
 pg_replication_slots;

slot_name | slot_type | confirmed_flush_lsn
----------+-----------+---------------------
 testsub   | logical   | 0/500002F0
(1 row)
```

```
Postgres=>select slot_name, slot_type, confirmed_flush_lsn from
 pg_replication_slots;
```

```
slot_name | slot_type | confirmed_flush_lsn
-----------+-----------+--------------------
 testsub   | logical   | 0/5413F5C0
(1 row)
```

Limitações de réplicas de leitura com o PostgreSQL

As limitações de réplicas de leitura do PostgreSQL são as seguintes:

- As réplicas de leitura do PostgreSQL são somente leitura. Embora uma réplica de leitura não seja uma instância de banco de dados gravável, é possível promovê-la para se tornar uma instância de banco de dados do RDS para PostgreSQL autônoma. No entanto, o processo é irreversível.
- Não será possível criar uma réplica de leitura de outra réplica de leitura se a instância de banco de dados do RDS para PostgreSQL estiver executando uma versão do PostgreSQL anterior à 14.1. O RDS para PostgreSQL é compatível com réplicas de leitura em cascata somente no RDS para PostgreSQL versão 14.1 e versões superiores. Para obter mais informações, consulte Usar réplicas de leitura em cascata com o RDS para PostgreSQL.
- Se você promover uma réplica de leitura do PostgreSQL, ela se tornará uma instância de banco de dados gravável. Ela deixa de receber arquivos de log de gravação antecipada (WAL) de uma instância de banco de dados de origem e não é mais uma instância somente leitura. É possível criar réplicas de leitura da instância de banco de dados promovida da mesma forma como você faz para qualquer instância de banco de dados do RDS para PostgreSQL. Para obter mais informações, consulte Promoção de uma réplica de leitura a uma instância de banco de dados autônoma.
- Se você promover uma réplica de leitura do PostgreSQL de dentro de uma cadeia de replicação (uma série de réplicas de leitura em cascata), todas as réplicas de leitura posteriores existentes continuarão recebendo arquivos WAL da instância promovida, automaticamente. Para obter mais informações, consulte Usar réplicas de leitura em cascata com o RDS para PostgreSQL.
- A réplica de leitura do PostgreSQL associada mostrará um atraso de replicação de até 5 minutos se não houver nenhuma transação de usuário em execução na instância do banco de dados de origem. O atraso da réplica é calculado como `currentTime - lastCommitedTransactionTimestamp`, o que significa que quando nenhuma transação está sendo processada, o valor do atraso da réplica aumenta por um período até que o segmento do log de gravação antecipada (WAL) mude. Por padrão, o RDS para PostgreSQL alterna o segmento do WAL a cada 5 minutos, o que resulta em um registro de transação e uma diminuição no atraso relatado.

- Não é possível ativar backups automatizados de réplicas de leitura do PostgreSQL de versões do RDS para PostgreSQL anteriores à 14.1. Os backups automatizados de réplicas de leitura são compatíveis apenas com o RDS para PostgreSQL 14.1 e versões superiores. Para o RDS para PostgreSQL 13 e versões anteriores, crie um instantâneo de uma réplica de leitura para criar um backup dele.
- A recuperação a um ponto anterior no tempo (PITR) não é compatível com réplicas de leitura. Você pode usar PITR somente com uma instância primária (gravadora), não com uma réplica de leitura. Para saber mais, consulte Restaurar uma instância de banco de dados para um momento especificado no Amazon RDS.
- As réplicas de leitura para PostgreSQL versões 12 e anteriores são reiniciadas automaticamente durante a janela de manutenção de 60 a 90 dias para aplicar a rotação de senhas. Se a réplica perder a conexão com a origem antes da reinicialização programada, ela ainda será reinicializada para retomar a replicação. Para as versões 13 e posteriores do PostgreSQL, as réplicas de leitura podem sofrer breves desconexões e reconexões de replicação durante o processo de rotação de senhas.

Configuração de réplica de leitura com o PostgreSQL

O RDS para PostgreSQL usa a replicação de streaming nativo do PostgreSQL para criar uma cópia somente leitura de uma instância de banco de dados de origem. Essa instância de banco de dados da réplica de leitura é uma replicação física criada assincronamente da instância de banco de dados de origem. Ela é criada por uma conexão especial que transmite dados de log de gravação antecipada (WAL) da instância de banco de dados de origem e a réplica de leitura. Para obter mais informações, consulte Streaming Replication (Replicação em streaming) na documentação do PostgreSQL.

O PostgreSQL faz streaming de forma assíncrona das alterações do banco de dados para essa conexão segura conforme são feitas na instância de banco de dados de origem. É possível criptografar comunicações de aplicações de clientes com a instância de banco de dados de origem ou qualquer réplica de leitura definindo o parâmetro `ssl` como 1. Para obter mais informações, consulteUsar o SSL com uma instância de banco de dados PostgreSQL

O PostgreSQL usa um perfil de replicação para realizar a replicação em streaming. O perfil é privilegiado, mas não pode ser usado para modificar nenhum dado. O PostgreSQL usa um processo único para gerenciar a replicação.

É possível criar uma réplica de leitura do PostgreSQL sem afetar as operações ou os usuários da instância de banco de dados de origem. O Amazon RDS define os parâmetros e as permissões necessários na instância de banco de dados de origem e na réplica de leitura sem afetar o serviço. Um snapshot é criado da instância de banco de dados de origem, e esse snapshot é usado para criar a réplica de leitura. Se você excluir a réplica de leitura em algum momento futuro, não ocorrerá nenhuma interrupção.

Você pode criar até quinze réplicas de leitura de uma instância de banco de dados de origem na mesma região. Desde o RDS para PostgreSQL 14.1, também é possível criar até três níveis de réplica de leitura em uma cadeia (cascata) de uma instância de banco de dados de origem. Para obter mais informações, consulte Usar réplicas de leitura em cascata com o RDS para PostgreSQL. Em todos os casos, a instância de banco de dados de origem precisa ter backups automatizados configurados. Faça isso definindo o período de retenção do backup na instância de banco de dados como qualquer valor diferente de 0. Para obter mais informações, consulteComo criar uma réplica de leitura

É possível criar réplicas de leitura para a instância de banco de dados do RDS para PostgreSQL na mesma Região da AWS que a instância de banco de dados de origem. Isso é conhecido como replicação na região. Não é possível criar uma réplica de leitura em uma Regiões da AWS diferente daquela da instância de banco de dados de origem. Isso é conhecido como Replicação entre regiões. Para obter mais informações sobre como configurar réplicas de leitura entre regiões, consulte Criar uma réplica de leitura em uma Região da AWS diferente. Os vários mecanismos compatíveis com o processo de replicação na região e entre regiões diferem ligeiramente de acordo com a versão do RDS para PostgreSQL, como explicado em Como a replicação de streaming funciona para diferentes versões do RDS para PostgreSQL.

Para que a replicação funcione efetivamente, cada réplica de leitura deve ter a mesma quantidade de recursos de computação e de armazenamento que a instância de banco de dados de origem. Se você dimensionar a instância de banco de dados de origem, dimensione as réplicas de leitura também.

O Amazon RDS substituirá todos os parâmetros incompatíveis em uma réplica de leitura se eles impedirem que a réplica de leitura seja iniciada. Por exemplo, suponha que o valor do parâmetro `max_connections` seja mais alto na instância do banco de dados de origem do que na réplica de leitura. Nesse caso, o Amazon RDS atualiza o parâmetro na réplica de leitura para ser o mesmo valor que o da instância de banco de dados de origem.

As réplicas de leitura do RDS para PostgreSQL têm acesso a bancos de dados externos que estão disponíveis por meio de wrappers de dados externos (FDWs) na instância de banco de dados de

origem. Por exemplo, suponha que sua instância de banco de dados do RDS para PostgreSQL esteja usando o wrapper do `mysql_fdw` para acessar dados do RDS para o MySQL. Nesse caso, as réplicas de leitura também podem acessar esses dados. Outros FDWs compatíveis incluem `oracle_fdw`, `postgres_fdw` e `tds_fdw`. Para obter mais informações, consulteTrabalhar com os invólucros de dados externos compatíveis do Amazon RDS for PostgreSQL

Usar réplicas de leitura do RDS para PostgreSQL com configurações multi-AZ

É possível criar uma réplica de leitura de uma instância de banco de dados single-AZ ou multi-AZ. É possível usar implantações multi-AZ para melhorar a durabilidade e a disponibilidade de dados essenciais. Uma réplica em espera é uma réplica de leitura dedicada que pode assumir a workload em caso de falha do banco de dados de origem. Não é possível utilizar a réplica em espera para atender ao tráfego de leitura. No entanto, é possível criar réplicas de leitura de instâncias de banco de dados multi-AZ de alto tráfego para descarregar consultas somente leitura. Para saber mais sobre implantações multi-AZ, consulte Implantações de instâncias de banco de dados multi-AZ para o Amazon RDS.

Se a instância de banco de dados de origem de uma implantação multi-AZ sofrer failover sobre a de espera, todas as réplicas de leitura associadas passarão a usar a réplica de espera (não a primária) como a origem de replicação. As réplicas de leitura podem precisar de reinicialização, de acordo com a versão do RDS para PostgreSQL, da seguinte forma:

- PostgreSQL 13 e versões superiores: a reinicialização não é necessária. As réplicas de leitura são sincronizadas automaticamente com a nova primária. No entanto, em alguns casos, a aplicação cliente pode armazenar os detalhes do DNS (Domain Name Service) em cache para suas réplicas de leitura. Nesse caso, defina o valor do tempo de vida (TTL) como menos de 30 segundos. Isso impede que a réplica de leitura se mantenha em um endereço IP obsoleto (e, portanto, impede que ela sincronize com a nova primária). Para saber mais sobre essa e outras práticas recomendadas, consulte Diretrizes operacionais básicas do Amazon RDS.
- PostgreSQL 12 e todas as versões anteriores: as réplicas de leitura são reiniciadas automaticamente após um failover para a réplica em espera porque a em espera (agora primária) tem um endereço IP e um nome de instância diferentes. A reinicialização sincroniza a réplica de leitura com a nova primária.

Para saber mais sobre failover, consulte Fazer failover de uma instância de banco de dados multi-AZ para o Amazon RDS. Para saber mais sobre como as réplicas de leitura funcionam em uma implantação multi-AZ, consulte Trabalhar com réplicas de leitura de instância de banco de dados.

Para oferecer suporte a failover de uma réplica de leitura, é possível criar a réplica de leitura como uma instância de banco de dados multi-AZ para que o Amazon RDS crie uma em espera de sua réplica em outra zona de disponibilidade (AZ). Você pode criar a réplica de leitura como uma instância de banco de dados multi-AZ independentemente de o banco de dados de origem ser ou não uma instância de banco de dados multi-AZ.

Usar réplicas de leitura em cascata com o RDS para PostgreSQL

Desde a versão 14.1, o RDS para PostgreSQL é compatível com réplicas de leitura em cascata. Com réplicas de leitura em cascata, é possível dimensionar leituras sem adicionar sobrecarga à instância de banco de dados do RDS para PostgreSQL de origem. As atualizações do log WAL não são enviadas pela instância de banco de dados de origem a cada réplica de leitura. Em vez disso, cada réplica de leitura em uma série em cascata envia atualizações de log WAL para a próxima réplica de leitura da série. Isso reduz a carga na instância de banco de dados de origem.

Com réplicas de leitura em cascata, a instância de banco de dados do RDS para PostgreSQL envia dados de WAL para a primeira réplica de leitura da cadeia. Essa réplica de leitura envia dados de WAL para a segunda réplica na cadeia e assim por diante. O resultado final é que todas as réplicas de leitura na cadeia têm as alterações da instância de banco de dados do RDS para PostgreSQL, mas sem a sobrecarga apenas na instância de banco de dados de origem.

É possível criar uma série de até três réplicas de leitura em uma cadeia de uma instância de banco de dados de origem do RDS para PostgreSQL. Por exemplo, suponha que você tenha uma instância de banco de dados do RDS para PostgreSQL 14.1, `rpg-db-main`. Você pode fazer o seguinte:

- Começando com `rpg-db-main`, crie a primeira réplica de leitura na cadeia, `read-replica-1`.
- Na `read-replica-1`, crie a próxima réplica de leitura na cadeia, `read-replica-2`.
- Finalmente, na `read-replica-2`, crie a terceira réplica de leitura na cadeia, `read-replica-3`.

Não é possível criar outra réplica de leitura além dessa terceira réplica de leitura em cascata na série de `rpg-db-main`. Uma série completa de instâncias de uma instância de banco de dados de origem do RDS para PostgreSQL até o final de uma série de réplicas de leitura em cascata pode consistir em, no máximo, quatro instâncias de banco de dados.

Para que as réplicas de leitura em cascata funcionem, ative os backups automáticos no RDS para PostgreSQL. Crie a réplica de leitura primeiro e ative os backups automáticos na instância de banco de dados do RDS para PostgreSQL. O processo é igual ao de outros mecanismos de banco de dados do Amazon RDS. Para obter mais informações, consulte Como criar uma réplica de leitura

Como em qualquer réplica de leitura, é possível promover uma réplica de leitura que faz parte de uma cascata. A promoção de uma réplica de leitura de uma cadeia de réplicas de leitura remove essa réplica da cadeia. Por exemplo, suponha que você queira mover parte da workload da instância de banco de dados `rpg-db-main` para uma nova instância para uso somente pelo departamento de contabilidade. Pressupondo a cadeia com três réplicas de leitura do exemplo, você decide promover `read-replica-2`. A cadeia é afetada da seguinte forma:

- A promoção de `read-replica-2` a remove da cadeia de replicação.
  - Ela agora é uma instância de banco de dados de leitura/gravação completa.
  - Ela continua replicando para `read-replica-3`, da mesma forma como estava fazendo antes da promoção.
- A `rpg-db-main` continua a replicar para a `read-replica-1`.

Para obter mais informações sobre como promover réplicas de leitura, consulte Promoção de uma réplica de leitura a uma instância de banco de dados autônoma.

> **Note**
>
> - O RDS para PostgreSQL não aceita atualizações de versão principal para réplicas em cascata. Antes de realizar uma atualização de versão principal, você precisa remover as réplicas em cascata. Você pode recriá-las depois de concluir a atualização na instância de banco de dados de origem e nas réplicas de primeiro nível.
> - Para réplicas de leitura em cascata, o RDS para PostgreSQL é compatível com quinze réplicas de leitura para cada instância de banco de dados de origem no primeiro nível de replicação e cinco réplicas de leitura para cada instância de banco de dados de origem no segundo e no terceiro níveis de replicação.

Criar réplicas de leitura em cascata entre regiões com o RDS para PostgreSQL

O RDS para PostgreSQL é compatível com réplicas de leitura em cascata entre regiões. É possível criar uma réplica entre regiões com base na instância de banco de dados de origem e, depois, criar réplicas da mesma região com base nela. Você também pode criar uma réplica da mesma região com base na instância de banco de dados de origem e, depois, criar réplicas entre regiões com base nela.

Criar uma réplica entre regiões e, depois, criar réplicas na mesma região

É possível usar uma instância de banco de dados do RDS para PostgreSQL com a versão 14.1 ou posterior, `rpg-db-main`, para fazer o seguinte:

1. Comece com `rpg-db-main` (US-EAST-1) e crie a primeira réplica de leitura entre regiões na cadeia, `read-replica-1` (US-WEST-2).
2. Usando a primeira região cruzada `read-replica-1` (US-WEST-2), crie a segunda réplica de leitura na cadeia, `read-replica-2` (US-WEST-2).
3. Usando `read-replica-2`, crie a terceira réplica de leitura na cadeia, `read-replica-3` (US-WEST-2).

Criar uma réplica na mesma região e, depois, criar réplicas entre regiões

É possível usar uma instância de banco de dados do RDS para PostgreSQL com a versão 14.1 ou posterior, `rpg-db-main`, para fazer o seguinte:

1. Começando com `rpg-db-main` (US-EAST-1), crie a primeira réplica de leitura na cadeia, `read-replica-1` (US-EAST-1).
2. Usando `read-replica-1` (US-EAST-1), crie a primeira réplica de leitura entre regiões na cadeia, `read-replica-2` (US-WEST-2).
3. Usando `read-replica-2` (US-WEST-2), crie a terceira réplica de leitura na cadeia, `read-replica-3` (US-WEST-2).

Limitações na criação de réplicas de leitura entre regiões

- Uma cadeia de réplicas de banco de dados em cascata entre regiões pode abranger no máximo duas regiões, com no máximo quatro níveis. Os quatro níveis incluem a origem do banco de dados e três réplicas de leitura.

Vantagens de usar réplicas de leitura em cascata

- Melhor escalabilidade de leitura: ao distribuir consultas de leitura em várias réplicas, a replicação em cascata ajuda a equilibrar a carga. Isso melhora o desempenho, especialmente em aplicações com muita leitura, reduzindo a pressão sobre o banco de dados do gravador.

- Distribuição geográfica: as réplicas em cascata podem estar em diferentes localizações geográficas. Isso reduz a latência para usuários localizados longe do banco de dados principal e fornece uma réplica de leitura local, aprimorando o desempenho e a experiência do usuário.
- Alta disponibilidade e recuperação de desastres: no caso de uma falha no servidor primário, as réplicas podem ser promovidas para primárias, garantindo a continuidade. A replicação em cascata aprimora ainda mais isso fornecendo várias camadas de opções de failover, o que melhora a resiliência geral do sistema.
- Flexibilidade e crescimento modular: à medida que o sistema cresce, novas réplicas podem ser adicionadas em diferentes níveis, sem exigir uma reconfiguração importante do banco de dados principal. Essa abordagem modular permite o crescimento escalável e gerenciável da configuração de replicação.

Prática recomendada para usar réplicas de leitura entre regiões

- Antes de promover uma réplica, crie réplicas adicionais. Isso economizará tempo e proporcionará um processamento eficiente da workload.

Como a replicação de streaming funciona para diferentes versões do RDS para PostgreSQL

Conforme discutido em Configuração de réplica de leitura com o PostgreSQL, o RDS para PostgreSQL usa o protocolo de replicação de streaming nativo do PostgreSQL para enviar dados de WAL da instância de banco de dados de origem. Ele envia dados de WAL de origem para as réplicas de leitura na região e entre regiões. Com a versão 9.4, o PostgreSQL introduziu slots de replicação física como um mecanismo de suporte para o processo de replicação.

Um slot de replicação física impede que uma instância de banco de dados de origem remova dados de WAL antes que sejam consumidos por todas as réplicas de leitura. Cada réplica de leitura possui seu próprio slot físico na instância de banco de dados de origem. O slot monitora o WAL mais antigo (por número de sequência lógica, LSN) que pode ser necessário para a réplica. Depois que todos os slots e conexões de banco de dados progrediram além de um determinado WAL (LSN), esse LSN se torna um candidato para remoção no próximo ponto de verificação.

O Amazon RDS usa o Amazon S3 para arquivar dados de WAL. Para réplicas de leitura na região, é possível usar esses dados arquivados para recuperar a réplica de leitura quando necessário. Um

exemplo de quando isso pode ser necessário é se a conexão entre o banco de dados de origem e a réplica de leitura for interrompida por qualquer motivo.

Na tabela a seguir, é possível encontrar um resumo das diferenças entre as versões do PostgreSQL e os mecanismos de suporte para réplicas de leitura na região e entre regiões usados pelo RDS para PostgreSQL.

Versão	Na região	Entre regiões
PostgreSQL 14.1 e versões posteriores	• Slots de replicação • Arquivo do Amazon S3	• Slots de replicação
PostgreSQL 13 e versões anteriores	• Arquivo do Amazon S3	• Slots de replicação

Para obter mais informações, consulteMonitoração e ajuste do processo de replicação

Noções básicas dos parâmetros que controlam a replicação do PostgreSQL

Os parâmetros a seguir afetam o processo de replicação e determinam como as réplicas de leitura permanecem atualizadas com a instância de banco de dados de origem:

max_wal_senders

O parâmetro `max_wal_senders` especifica o número máximo de conexões com as quais a instância de banco de dados de origem é compatível ao mesmo tempo por meio do protocolo de replicação de streaming.

O valor padrão varia para versões do RDS para PostgreSQL:

- Para as versões 13, 14 e 15, o valor padrão é 20.
- Para as versões 16 e posteriores, o valor padrão é 35.

Esse parâmetro deve ser definido como um número um pouco maior que o número real de réplicas de leitura. Se esse parâmetro for definido muito baixo para o número de réplicas de leitura, a replicação será interrompida.

Para obter mais informações, consulte max_wal_senders na documentação do PostgreSQL.

> **Note**
>
> `max_wal_senders` é um parâmetro estático que requer a reinicialização da instância de banco de dados para que as alterações sejam aplicadas.

wal_keep_segments

O parâmetro `wal_keep_segments` especifica o número de arquivos de log de gravação antecipada (WAL) que a instância de banco de dados de origem mantém no diretório `pg_wal`. A configuração padrão é 32.

Se `wal_keep_segments` não estiver definido como um valor grande o suficiente para sua implantação, uma réplica de leitura poderá ficar tão distante que interromperá a replicação de streaming. Se isso acontecer, o Amazon RDS gerará um erro de replicação e iniciará a recuperação na réplica de leitura. Ele faz isso reproduzindo os dados de WAL arquivados da instância de banco de dados de origem no Amazon S3. Este processo de recuperação continua até que a réplica de leitura tenha alcançado o suficiente para continuar com a replicação de streaming. É possível ver esse processo em ação, conforme capturado pelo log do PostgreSQL em Exemplo: como uma réplica de leitura se recupera de interrupções de replicação.

> **Note**
>
> No PostgreSQL versão 13, o parâmetro `wal_keep_segments` é chamado de `wal_keep_size`. Ele atende ao mesmo propósito que `wal_keep_segments`, mas seu valor padrão é em megabytes (MB) (2048 MB) em vez do número de arquivos. Para obter mais informações, consulte wal_keep_segments e wal_keep_size na documentação do PostgreSQL.

max_slot_wal_keep_size

O parâmetro `max_slot_wal_keep_size` controla a quantidade de dados de WAL que a instância de banco de dados do RDS para PostgreSQL retém no diretório `pg_wal` para atender a slots. Esse parâmetro é usado para configurações que usam slots de replicação. O valor padrão desse parâmetro é `-1`, o que significa que não há limite para a quantidade de dados de WAL mantidos na instância de banco de dados de origem. Para obter informações sobre como monitorar os slots de replicação, consulte Monitorar os slots de replicação da instância de banco de dados do RDS para PostgreSQL.

Para obter mais informações sobre esse parâmetro, consulte max_slot_wal_keep_size na documentação do PostgreSQL.

Quando a transmissão que fornece dados de WAL a uma réplica de leitura é interrompida, o PostgreSQL muda para o modo de recuperação. Ele restaura a réplica de leitura usando os dados de WAL arquivados no Amazon S3 ou usando dados de WAL associados ao slot de replicação. Quando esse processo é concluído, o PostgreSQL restabelece a replicação de streaming.

Exemplo: como uma réplica de leitura se recupera de interrupções de replicação

No exemplo a seguir, você encontra os detalhes do log que demonstram o processo de recuperação de uma réplica de leitura. O exemplo é de uma instância de banco de dados do RDS para PostgreSQL que executa o PostgreSQL versão 12.9 na mesma Região da AWS que o banco de dados de origem, portanto, os slots de replicação não são usados. O processo de recuperação é o mesmo para outras instâncias de banco de dados do RDS para PostgreSQL que executam PostgreSQL anterior à versão 14.1 com réplicas de leitura na região.

Quando a réplica de leitura perde contato com a instância de banco de dados de origem, o Amazon RDS registra o problema no log como mensagem `FATAL: could not receive data from WAL stream`, junto com `ERROR: requested WAL segment ... has already been removed`. Conforme mostrado na linha em negrito, o Amazon RDS recupera a réplica reproduzindo um arquivo de WAL arquivado.

```
2014-11-07 19:01:10 UTC::@:[23180]:DEBUG:  switched WAL source from archive to stream
 after failure
2014-11-07 19:01:10 UTC::@:[11575]:LOG: started streaming WAL from primary at 1A/
D3000000 on timeline 1
2014-11-07 19:01:10 UTC::@:[11575]:FATAL: could not receive data from WAL stream:
ERROR:  requested WAL segment 000000010000001A000000D3 has already been removed
2014-11-07 19:01:10 UTC::@:[23180]:DEBUG: could not restore file "00000002.history"
 from archive: return code 0
2014-11-07 19:01:15 UTC::@:[23180]:DEBUG: switched WAL source from stream to archive
 after failure recovering 000000010000001A000000D3
2014-11-07 19:01:16 UTC::@:[23180]:LOG:  restored log file "000000010000001A000000D3"
 from archive
```

Quando o Amazon RDS reproduz arquivos de WAL arquivados suficientes na réplica para recuperar o atraso, o streaming para a réplica de leitura pode começar novamente. Quando o streaming é retomado, o Amazon RDS grava uma entrada no arquivo de log, semelhante à mostrada a seguir.

```
2014-11-07 19:41:36 UTC::@:[24714]:LOG:started streaming WAL from primary at 1B/
B6000000 on timeline 1
```

Definir os parâmetros que controlam a memória compartilhada

Os parâmetros que você define determinam o tamanho da memória compartilhada para rastrear IDs de transações, bloqueios e transações preparadas. A estrutura de memória compartilhada de uma instância em espera deve ser igual ou maior que a de uma instância primária. Isso garante que a primeira não fique sem memória compartilhada durante a recuperação. Se os valores dos parâmetros na réplica forem menores que os valores dos parâmetros na primária, o Amazon RDS ajustará automaticamente os parâmetros da réplica e reiniciará o mecanismo.

Os parâmetros afetados são:

- max_connections
- max_worker_processes
- max_wal_senders
- max_prepared_transactions
- max_locks_per_transaction

Para evitar que o RDS reinicialize réplicas em decorrência de memória insuficiente, recomendamos aplicar as alterações de parâmetro como uma reinicialização contínua de cada réplica. Você deve aplicar as seguintes regras ao definir os parâmetros:

- Aumentar os valores dos parâmetros:
  - Você deve sempre aumentar primeiro os valores dos parâmetros de todas as réplicas de leitura e realizar uma reinicialização contínua de todas as réplicas. Em seguida, aplique as alterações de parâmetro na instância primária e reinicialize.
- Diminuir os valores dos parâmetros:
  - Primeiro, você deve diminuir os valores dos parâmetros da instância primária e realizar uma reinicialização. Em seguida, aplique as alterações de parâmetro a todas as réplicas de leitura correspondentes e execute uma reinicialização contínua.

Monitoração e ajuste do processo de replicação

Convém monitorar rotineiramente a instância de banco de dados do RDS para PostgreSQL e as réplicas de leitura. Verifique se suas réplicas de leitura estão acompanhando as alterações na instância de banco de dados de origem. O Amazon RDS recupera suas réplicas de leitura de forma transparente quando ocorrem interrupções no processo de replicação. No entanto, é melhor evitar qualquer necessidade de recuperação. A recuperação que usa slots de replicação é mais rápida do que usar o arquivo do Amazon S3, mas qualquer processo de recuperação pode afetar a performance de leitura.

Para determinar como suas réplicas de leitura estão acompanhando a instância de banco de dados de origem, é possível fazer o seguinte:

- Verifique a quantidade de **`ReplicaLag`** entre a instância de banco de dados de origem e as réplicas. Atraso da réplica refere-se ao tempo em milissegundos durante o qual uma réplica de leitura fica atrasada em relação à instância de banco de dados de origem. Essa métrica retorna o resultado a seguir.

  ```
  SELECT extract(epoch from now() - pg_last_xact_replay_timestamp()) AS "ReplicaLag";
  ```

  O atraso de réplica é uma indicação de como uma réplica de leitura está acompanhando a instância de banco de dados de origem. É a quantidade de latência entre a instância de banco de dados de origem e uma instância de leitura específica. Um valor alto de atraso de réplica pode indicar uma incompatibilidade entre as classes ou os tipos de armazenamento (ou ambos) usados pela instância de banco de dados de origem e suas réplicas de leitura. A classe e os tipos de armazenamento da instância de banco de dados de origem e de todas as réplicas de leitura devem ser iguais.

  O atraso de réplica também pode ser o resultado de problemas de conexão intermitentes. Você pode monitorar o atraso da replicação no Amazon CloudWatch visualizando a métrica `ReplicaLag` do Amazon RDS. Para saber mais sobre `ReplicaLag` e outras métricas do Amazon RDS, consulte Métricas do Amazon CloudWatch para o Amazon RDS.
- Verifique o log do PostgreSQL para obter informações que podem ser usadas para ajustar suas configurações. Por exemplo, em cada ponto de verificação, o log do PostgreSQL captura o número de arquivos de log de transações reciclados, como mostrado no exemplo a seguir.

  ```
  2014-11-07 19:59:35 UTC::@:[26820]:LOG:  checkpoint complete: wrote 376 buffers
   (0.2%);
  ```

```
0 transaction log file(s) added, 0 removed, 1 recycled; write=35.681 s, sync=0.013 s,
 total=35.703 s;
sync files=10, longest=0.013 s, average=0.001 s
```

É possível usar essas informações para descobrir quantos arquivos de transação estão sendo reciclados em um determinado período de tempo. É possível alterar a configuração para `wal_keep_segments` se necessário. Por exemplo, suponha que o log do PostgreSQL em `checkpoint complete` exibe `35 recycled` por um intervalo de 5 minutos. Neste caso, o valor padrão de 32 de `wal_keep_segments` não é suficiente para acompanhar a atividade de streaming, e recomendamos aumentar o valor desse parâmetro.

- Use o Amazon CloudWatch para monitorar métricas que podem prever problemas de replicação. Em vez de analisar diretamente o log do PostgreSQL, é possível usar o Amazon CloudWatch para verificar as métricas coletadas. Por exemplo, é possível verificar o valor da métrica `TransactionLogsGeneration` para ver quantos dados de WAL estão sendo gerados pela instância de banco de dados de origem. Em alguns casos, a workload em sua instância de banco de dados pode gerar uma grande quantidade de dados de WAL. Nesse caso, talvez seja necessário alterar a classe da instância de banco de dados de origem e das réplicas de leitura. Usar uma classe de instância com alta performance de rede (10 Gbps) pode reduzir o atraso de réplica.

Monitorar os slots de replicação da instância de banco de dados do RDS para PostgreSQL

Todas as versões do RDS para PostgreSQL usam slots de replicação para réplicas de leitura entre regiões. O RDS para PostgreSQL 14.1 e versões superiores usam slots de replicação para réplicas de leitura na região. As réplicas de leitura na região também usam o Amazon S3 para arquivar dados de WAL. Em outras palavras, se sua instância de banco de dados e réplicas de leitura estiverem executando o PostgreSQL 14.1 ou superior, os slots de replicação e os arquivos do Amazon S3 estarão disponíveis para recuperar a réplica de leitura. Recuperar uma réplica de leitura usando seu slot de replicação é mais rápido do que recuperar do arquivo do Amazon S3. Portanto, convém monitorar os slots de replicação e as métricas relacionadas.

É possível visualizar os slots de replicação em suas instâncias de banco de dados do RDS para PostgreSQL consultando a visualização `pg_replication_slots` da forma a seguir.

```
postgres=> SELECT * FROM pg_replication_slots;
```

```
slot_name                  | plugin | slot_type | datoid | database | temporary |
 active | active_pid | xmin | catalog_xmin | restart_lsn | confirmed_flush_lsn |
 wal_status | safe_wal_size | two_phase
---------------------------+--------+-----------+--------+----------+-----------
+--------+------------+------+--------------+-------------+---------------------
+------------+--------------+-----------
rds_us_west_1_db_555555555 |        | physical  |        |          | f         | t
  |      13194 |      |              | 23/D8000060 |                     | reserved   |
               | f
(1 row)
```

O `wal_status` do valor `reserved` indica que a quantidade de dados de WAL mantidos pelo slot está dentro dos limites do parâmetro `max_wal_size`. Ou seja, o slot de replicação está dimensionado corretamente. Outros valores de status possíveis são os seguintes:

- `extended`: o slot excede a configuração de `max_wal_size`, mas os dados de WAL estão retidos.
- `unreserved`: o slot não tem mais todos os dados de WAL necessários. Alguns deles serão removidos no próximo ponto de verificação.
- `lost`: alguns dados de WAL necessários foram removidos. O slot não é mais utilizável.

Os estados `unreserved` e `lost` do `wal_status` são vistos somente quando `max_slot_wal_keep_size` não é negativo.

A visualização `pg_replication_slots` mostra o estado atual dos slots de replicação. Para avaliar a performance dos slots de replicação, é possível usar o Amazon CloudWatch e monitorar as seguintes métricas:

- **`OldestReplicationSlotLag`**: mostra a quantidade de dados do log de gravação antecipada (WAL) na origem que não foi consumida pela réplica mais atrasada.
- **`TransactionLogsDiskUsage`** – Mostra quanto armazenamento está sendo usado para dados WAL. Quando uma réplica de leitura atrasa significativamente, o valor dessa métrica pode aumentar substancialmente.

Para saber mais sobre o uso do Amazon CloudWatch e suas métricas para o RDS para PostgreSQL, consulte Monitorar métricas do Amazon RDS com o Amazon CloudWatch. Para obter mais informações sobre como monitorar a replicação de streaming em suas instâncias de bancos de dados do RDS para PostgreSQL, consulte Práticas recomendadas para replicação do Amazon RDS para PostgreSQL no Blog de banco de dados da AWS.

Solução de problemas da réplica de leitura do RDS para PostgreSQL

A seguir, é possível encontrar ideias de solução de alguns problemas comuns de réplicas de leitura do RDS para PostgreSQL.

Encerrar a consulta que causa o atraso da réplica de leitura

As transações ativas ou ociosas no estado de transação que estão sendo executadas por um longo tempo no banco de dados podem interferir no processo de replicação do WAL, aumentando assim o atraso na replicação. Portanto, monitore o tempo de execução dessas transações com a visualização `pg_stat_activity` do PostgreSQL.

Execute uma consulta na instância primária semelhante à seguinte para encontrar o ID do processo (PID) da consulta que está sendo executada por um longo tempo:

```
SELECT datname, pid,usename, client_addr, backend_start,
xact_start, current_timestamp - xact_start AS xact_runtime, state,
backend_xmin FROM pg_stat_activity WHERE state='active';
```

```
SELECT now() - state_change as idle_in_transaction_duration, now() - xact_start as
 xact_duration,*
FROM  pg_stat_activity
WHERE state  = 'idle in transaction'
AND   xact_start is not null
ORDER BY 1 DESC;
```

Depois de identificar o PID da consulta, você pode optar por encerrá-la.

Execute uma consulta na instância primária semelhante à seguinte para encerrar a consulta que está sendo executada por um longo tempo:

```
SELECT pg_terminate_backend(PID);
```

Melhorar a performance das consultas para o RDS para PostgreSQL com leituras otimizadas pelo Amazon RDS

É possível acelerar o processamento de consultas para o RDS para PostgreSQL com leituras otimizadas pelo Amazon RDS. Uma instância de banco de dados ou um cluster de banco de dados multi-AZ do RDS para PostgreSQL que utiliza o recurso Leituras otimizadas pelo RDS pode alcançar um processamento de consultas até 50% mais rápido em comparação com uma instância ou um cluster não usa o recurso.

Tópicos

- Visão geral das leituras otimizadas pelo RDS no PostgreSQL
- Casos de uso para leituras otimizadas do RDS
- Práticas recomendadas para leituras otimizadas do RDS
- Utilizar leituras otimizadas do RDS
- Monitorar instâncias de banco de dados que utilizam leituras otimizadas do RDS
- Limitações das leituras otimizadas pelo RDS no PostgreSQL

Visão geral das leituras otimizadas pelo RDS no PostgreSQL

As leituras otimizadas estão disponíveis por padrão no RDS para PostgreSQL versões 15.2 e posterior, 14.7 e posterior e 13.10 e posterior ao usar as classes de instância de banco de dados baseadas em NVMe. Consulte especificações de hardware que indicam quais instâncias usam o NVMe em Especificações de hardware para classes de instância de banco de dados .

Quando você usa uma instância de banco de dados ou um cluster de banco de dados multi-AZ do RDS para PostgreSQL com o recurso Leituras otimizadas pelo RDS ativado, a instância ou o cluster alcança uma performance de consulta até 50% mais rápida por meio do armazenamento em bloco de unidade de estado sólido (SSD) local baseado em memória não volátil (NVMe). Você pode acelerar o processamento de consultas colocando as tabelas temporárias geradas pelo PostgreSQL no armazenamento local, o que reduz o tráfego para o Elastic Block Storage (EBS) pela rede.

No PostgreSQL, objetos temporários são atribuídos a um namespace temporário que é descartado automaticamente no final da sessão. O namespace temporário durante a eliminação remove todos os objetos que dependem da sessão, como objetos qualificados pelo esquema, por exemplo, tabelas, funções, operadores ou até mesmo extensões.

No RDS para PostgreSQL, o parâmetro `temp_tablespaces` é configurado para essa área de trabalho temporária em que os objetos temporários são armazenados.

As consultas a seguir retornam o nome do espaço de tabela e sua localização.

```
postgres=> show temp_tablespaces;
temp_tablespaces
--------------------
rds_temp_tablespace
(1 row)
```

O `rds_temp_tablespace` é um espaço de tabela configurado pelo RDS que aponta para o armazenamento local NVMe. Você sempre pode voltar para o armazenamento do Amazon EBS modificando esse parâmetro no `Parameter group` usando o AWS Management Console para apontar para qualquer espaço de tabela diferente de `rds_temp_tablespace`. Para ter mais informações, consulte Modificar parâmetros em um grupo de parâmetros de banco de dados no Amazon RDS. Você também pode usar o comando SET para modificar o valor do parâmetro `temp_tablespaces` para `pg_default` no nível da sessão usando o comando SET. A modificação do parâmetro redireciona a área de trabalho temporária para o Amazon EBS. Voltar para o Amazon EBS ajuda quando o armazenamento local de sua instância ou cluster do RDS não é suficiente para realizar uma operação SQL específica.

```
postgres=> SET temp_tablespaces TO 'pg_default';
SET
```

```
postgres=> show temp_tablespaces;

 temp_tablespaces
------------------
 pg_default
```

Casos de uso para leituras otimizadas do RDS

Veja a seguir alguns casos de uso que podem se beneficiar das leituras otimizadas:

- Consultas analíticas que incluem expressões de tabela comuns (CTEs), tabelas derivadas e operações de agrupamento.
- Réplicas de leitura que lidam com as consultas não otimizadas de uma aplicação.

- Consultas de relatórios dinâmicos ou sob demanda com operações complexas, como GROUP BY e ORDER BY, que nem sempre podem usar índices apropriados.
- Outras workloads que usam tabelas temporárias internas.
- Operações `CREATE INDEX` ou `REINDEX` de classificação.

Práticas recomendadas para leituras otimizadas do RDS

Use as práticas recomendadas a seguir para leituras otimizadas do RDS:

- Adicione uma lógica de repetição para consultas somente leitura caso elas falhem devido ao armazenamento de instância estar cheio durante a execução.
- Monitore o espaço de armazenamento disponível no armazenamento de instâncias com a métrica do CloudWatch `FreeLocalStorage`. Se o armazenamento de instância estiver atingindo seu limite devido à workload na instância de banco de dados ou no cluster de banco de dados multi-AZ, modifique a instância ou o cluster para usar uma classe de instância de banco de dados maior.

Utilizar leituras otimizadas do RDS

Quando você provisiona uma instância de banco de dados do RDS para PostgreSQL com uma das classes de instância de banco de dados baseada em NVMe em uma implantação de instância de banco de dados single-AZ, uma implantação de instância de banco de dados multi-AZ ou uma implantação de cluster de banco de dados multi-AZ, a instância de banco de dados utiliza automaticamente o recurso Leituras otimizadas pelo RDS.

Consulte mais informações sobre a implantação multi-AZ em Configurar e gerenciar uma implantação multi-AZ para o Amazon RDS.

Para ativar as leituras otimizadas do RDS, execute um destes procedimentos:

- Crie uma instância de banco de dados ou um cluster de banco de dados multi-AZ do RDS para PostgreSQL utilizando uma das classes de instância de banco de dados NVMe. Para ter mais informações, consulte Criar uma instância de banco de dados do Amazon RDS.
- Modifique uma instância de banco de dados ou um cluster de banco de dados multi-AZ existente do RDS para PostgreSQL para utilizar uma das classes de instância de banco de dados baseada em NVMe. Para ter mais informações, consulte Modificar uma instância de banco de dados do Amazon RDS.

O recurso Leituras otimizadas pelo RDS está disponível em todas as Regiões da AWS onde há suporte para uma ou mais dessas classes de instância de banco de dados com SSD NVMe local. Para ter mais informações, consulte Classes de instâncias de banco de dados .

Para voltar para uma instância do RDS sem otimização de leituras, modifique a classe de instância de banco de dados de sua instância ou cluster do RDS para a classe de instância semelhante que só é compatível com o armazenamento do EBS para suas workloads de banco de dados. Por exemplo, se a classe de instância de banco de dados atual for db.r6gd.4xlarge, selecione db.r6g.4xlarge para voltar. Para ter mais informações, consulte Modificar uma instância de banco de dados do Amazon RDS.

Monitorar instâncias de banco de dados que utilizam leituras otimizadas do RDS

Você pode monitorar instâncias de banco de dados que usam leituras otimizadas pelo RDS com as seguintes métricas do CloudWatch:

- `FreeLocalStorage`
- `ReadIOPSLocalStorage`
- `ReadLatencyLocalStorage`
- `ReadThroughputLocalStorage`
- `WriteIOPSLocalStorage`
- `WriteLatencyLocalStorage`
- `WriteThroughputLocalStorage`

Essas métricas fornecem dados sobre armazenamento de instâncias, IOPS e throughput. Para ter mais informações sobre essas métricas, consulte Métricas específicas da instância do Amazon CloudWatch para Amazon RDS.

Para monitorar o uso atual do armazenamento local, faça login no banco de dados e execute a seguinte consulta:

```
SELECT
    spcname AS "Name",
    pg_catalog.pg_size_pretty(pg_catalog.pg_tablespace_size(oid)) AS "size"
FROM
    pg_catalog.pg_tablespace
```

```
WHERE
    spcname IN ('rds_temp_tablespace');
```

Para obter mais informações sobre os arquivos temporários e o uso, consulte Gerenciar arquivos temporários com o PostgreSQL.

Limitações das leituras otimizadas pelo RDS no PostgreSQL

A limitação a seguir se aplica ao recurso Leituras otimizadas pelo RDS no PostgreSQL:

- As transações podem falhar quando o armazenamento de instância está cheio.

Como importar dados no PostgreSQL do Amazon RDS

Vamos supor que você tenha uma implantação do PostgreSQL que deseja mover para o Amazon RDS. A complexidade da tarefa dependerá do tamanho do banco de dados e dos tipos de objetos de banco de dados a serem transferidos. Por exemplo, considere um banco de dados que contenha conjuntos de dados na ordem de gigabytes, juntamente com gatilhos e procedimentos armazenados. Esse banco de dados será mais complexo do que um banco de dados simples com apenas alguns megabytes de dados de teste e sem gatilhos e procedimentos armazenados.

Recomendamos que você use ferramentas nativas de migração do banco de dados PostgreSQL nas seguintes condições:

- Quando há uma migração homogênea, em que você está migrando de um banco de dados com o mesmo mecanismo de banco de dados que o de destino.
- Quando for migrar um banco de dados inteiro.
- As ferramentas nativas permitirem que você migre o sistema com um tempo mínimo de inatividade.

Na maioria dos outros casos, executar uma migração de banco de dados usando o AWS Database Migration Service (AWS DMS) é a melhor abordagem. AWS O DMS pode migrar bancos de dados sem o tempo de inatividade e, para muitos mecanismos de banco de dados, continua a replicação contínua até que você esteja pronto para mudar para o banco de dados de destino. Você pode migrar para o mesmo mecanismo de banco de dados ou um mecanismo de banco de dados diferente usando o AWS DMS. Se você estiver migrando para um mecanismo de banco de dados diferente do seu banco de dados de origem, poderá usar a AWS Schema Conversion Tool (AWS SCT). Use o AWS SCT para migrar os objetos de esquema que não são migrados pelo AWS DMS. Para obter mais informações sobre como usar o AWS DMS, consulte O que é o AWS Database Migration Service?.

Modifique o grupo de parâmetros do banco de dados para incluir as seguintes configurações somente na sua importação. Você deve testar as configurações dos parâmetros para encontrar as que forem mais eficientes para o tamanho da instância do banco de dados. Você também precisará reverter esses parâmetros aos seus valores de produção após a conclusão da importação.

Modifique as configurações da instância de banco de dados da seguinte forma:

- Desabilite os backups de instâncias de banco de dados (defina backup_retention como 0).
- Desabilite o multi-AZ.

Modifique o grupo de parâmetros de banco de dados para incluir as seguintes configurações. Você só deve usar essas configurações para importar dados. Você deve testar as configurações dos parâmetros para encontrar as que forem mais eficientes para o tamanho da instância do banco de dados. Você também precisará reverter esses parâmetros aos seus valores de produção após a conclusão da importação.

Parâmetro	Valor recomendado para importação	Descrição
`maintenance_work_mem`	524288, 1048576, 2097152 ou 4194304 (em KB). Essas configurações são comparáveis a 512 MB, 1 GB, 2 GB e 4 GB.	O valor desta configuração depende do tamanho do host. Este parâmetro é usado durante instruções CREATE INDEX e cada comando paralelo pode usar essa quantidade de memória. Calcule o valor mais apropriado para não ajustá-lo tão alto resultando na falta de memória.
`max_wal_size`	256 (para a versão 9.6), 4096 (para as versões 10 e posteriores)	Tamanho máximo para deixar o WAL aumentar durante os pontos de verificação automáticos. Aumentar esse parâmetro pode aumentar a quantidade de tempo necessária para a recuperação de falhas. Esse parâmetro substitui `checkpoint_segments` para o PostgreSQL 9.6 e posteriores. Para o PostgreSQL versão 9.6, esse valor está em unidades de 16 MB. Para versões posteriores, o valor está em unidades de 1 MB. Por exemplo, na versão 9.6, 128 significa 128 blocos, cada um com tamanho de 16 MB. Na versão 12.4, 2048 significa 2048 blocos, cada um com tamanho de 1 MB.
`checkpoint_timeout`	1800	O valor desta configuração permite um revezamento menos frequente de WAL.
`synchronous_commit`	Desativado	Desabilite essa configuração para acelerar a gravação. Desativar esse parâmetro pode

Parâmetro	Valor recomendado para importação	Descrição
		aumentar o risco de perda de dados em caso de uma falha no servidor (não desative o FSYNC)
`wal_buffers`	8192	Este valor está em unidades de 8 KB. Reforçando, isso ajuda na velocidade de geração WAL
`autovacuum`	0	Desative o parâmetro de autovacuum do PostgreSQL enquanto estiver carregando dados para que ele não use os recursos

Use os comandos `pg_dump -Fc` (compactado) ou `pg_restore -j` (paralelo) com essas configurações.

> **Note**
>
> O comando do PostgreSQL `pg_dumpall` requer permissões de super_user que não são concedidas ao criar uma instância de banco de dados, por isso ele não pode ser usado para importação de dados.

Tópicos

- Importar um banco de dados PostgreSQL de uma instância do Amazon EC2
- Uso do comando \copy para importar dados em uma tabela em uma instância de banco de dados PostgreSQL
- Importar dados do Amazon S3 para uma instância de banco de dados do RDS para PostgreSQL
- Transportar bancos de dados PostgreSQL entre instâncias de banco de dados

Importar um banco de dados PostgreSQL de uma instância do Amazon EC2

Se houver dados em um servidor do PostgreSQL em uma instância do Amazon EC2 e você quiser movê-los para uma instância de banco de dados do PostgreSQL, você poderá seguir este processo para migrar os dados.

1. Crie um arquivo usando o pg_dump que contenha os dados a serem carregados
2. Crie a instância de banco de dados de destino
3. Use o psql para criar o banco de dados na instância de banco de dados e carregá-los
4. Crie um snapshot de banco de dados da instância de banco de dados

As seções a seguir apresentam mais detalhes sobre cada etapa listada acima.

Etapa 1: Criar um arquivo usando o pg_dump que contenha os dados a serem carregados

O utilitário `pg_dump` usa o comando COPY para criar um esquema e uma despejo de dados de um banco de dados PostgreSQL. O script de despejo gerado pelo `pg_dump` carrega os dados em um banco de dados com o mesmo nome e recreia as tabelas, os índices e as chaves externas. Você pode usar o comando `pg_restore` e o parâmetro `-d` para recuperar os dados em um banco de dados com um nome diferente.

Antes de criar o despejo de dados, você deve consultar as tabelas a serem despejadas para obter uma contagem de linhas, assim você pode confirmar a contagem na instância de banco de dados de destino.

O comando a seguir cria um arquivo de despejo denominado mydb2dump.sql para um banco de dados denominado mydb2.

```
prompt>pg_dump dbname=mydb2 -f mydb2dump.sql
```

Etapa 2: crie a instância de banco de dados de destino

Crie a instância de banco de dados PostgreSQL de destino usando o console, a AWS CLI ou a API do Amazon RDS. Crie a instância com a configuração de retenção de backup definida como 0 e desative o multi-AZ. Isso permite uma importação de dados mais rápida. Você deve criar um banco

de dados na instância antes de poder despejar os dados. O banco de dados pode ter o mesmo nome que o banco de dados que contém os dados de despejo. Se preferir, você pode criar um banco de dados com um nome diferente. Nesse caso, use o comando `pg_restore` e o parâmetro `-d` para restaurar os dados para o banco de dados recém-nomeado.

Por exemplo, os seguintes comandos podem ser usados para despejar, restaurar e renomear um banco de dados.

```
pg_dump -Fc -v -h [endpoint of instance] -U [master username] [database]
 > [database].dump
createdb [new database name]
pg_restore -v -h [endpoint of instance] -U [master username] -d [new database
 name] [database].dump
```

Etapa 3: Usar o psql para criar o banco de dados na instância de banco de dados e carregá-los

É possível usar a mesma conexão que você usou para executar o comando pg_dump para se conectar à instância de banco de dados de destino e recriar o banco de dados. Com o psql, use o nome de usuário mestre e a senha mestre para criar o banco de dados na instância de banco de dados

O exemplo a seguir usa o psql e um arquivo de despejo denominado mydb2dump.sql para criar um banco de dados denominado mydb2 em uma instância de banco de dados PostgreSQL chamada mypginstance:

Para Linux, macOS ou Unix:

```
psql \
   -f mydb2dump.sql \
   --host mypginstance.555555555555.aws-region.rds.amazonaws.com \
   --port 8199 \
   --username myawsuser \
   --password password \
   --dbname mydb2
```

Para Windows:

```
psql ^
```

```
-f mydb2dump.sql ^
--host mypginstance.555555555555.aws-region.rds.amazonaws.com ^
--port 8199 ^
--username myawsuser ^
--password password ^
--dbname mydb2
```

> **Note**
>
> Especifique uma senha diferente do prompt mostrado aqui como prática recomendada de segurança.

Etapa 4: crie um DB snapshot da instância de banco de dados

Após confirmar que os dados foram carregados na instância de banco de dados, recomendamos que você crie um snapshot de banco de dados da instância de banco de dados PostgreSQL de destino. Os snapshot de banco de dadoss são backups completos de sua instância de banco de dados que podem ser usados para restaurá-la em um estado conhecido. Um snapshot de banco de dados feito imediatamente após a carga evita que você tenha que carregar os dados novamente caso ocorra um problema. Você também pode usar esse snapshot para propagar as novas instâncias de banco de dados. Para obter informações sobre como criar um snapshot de banco de dados, consulte Criar um snapshot de banco de dados para uma instância de banco de dados single-AZ para o Amazon RDS.

Uso do comando \copy para importar dados em uma tabela em uma instância de banco de dados PostgreSQL

O comando `\copy` do PostgreSQL é um meta-comando disponível a partir da ferramenta cliente interativa `psql`. Você pode usar `\copy` para importar dados para uma tabela na instância de banco de dados do RDS para PostgreSQL Para usar o comando `\copy`, você precisa primeiro criar a estrutura da tabela na instância de banco de dados de destino para que `\copy` tenha um destino para os dados que estão sendo copiados.

Você pode usar `\copy` para carregar dados de um arquivo de valores separados por vírgulas (CSV), como um que foi exportado e salvo na estação de trabalho do cliente.

Para importar os dados CSV para a instância de banco de dados do RDS para PostgreSQL de destino, primeiro conecte-se à instância de banco de dados de destino usando `psql`.

```
psql --host=db-instance.111122223333.aws-region.rds.amazonaws.com --port=5432 --
username=postgres --password --dbname=target-db
```

Você então executa `\copy` com os parâmetros a seguir para identificar o destino para os dados e seu formato.

- `target_table` – O nome da tabela que deve receber os dados que estão sendo copiados do arquivo CSV.
- `column_list` – Especificações da coluna para a tabela.
- `'filename'` – O caminho completo para o arquivo CSV na estação de trabalho local.

```
 \copy target_table from '/path/to/local/filename.csv' WITH DELIMITER ',' CSV;
```

Se o arquivo CSV tiver informações de cabeçalho de coluna, você poderá usar esta versão do comando e dos parâmetros.

```
\copy target_table (column-1, column-2, column-3, ...)
    from '/path/to/local/filename.csv' WITH DELIMITER ',' CSV HEADER;
```

Se o comando `\copy` falhar, o PostgreSQL gera mensagens de erro.

Criar uma instância de banco de dados no comando `psql` do Ambiente de Pré-visualização do Banco de Dados com o metacomando `\copy`, conforme mostrado nos exemplos a seguir. Este exemplo usa source-table como o nome da tabela de origem, source-table.csv como o arquivo .csv e target-db como o banco de dados de destino:

Para Linux, macOS ou Unix:

```
$psql target-db \
    -U <admin user> \
    -p <port> \
    -h <DB instance name> \
    -c "\copy source-table from 'source-table.csv' with DELIMITER ','"
```

Para Windows:

```
$psql target-db ^
    -U <admin user> ^
```

```
    -p <port> ^
    -h <DB instance name> ^
    -c "\copy source-table from 'source-table.csv' with DELIMITER ','"
```

Para obter detalhes completos sobre o comando `\copy`, consulte a página psql na documentação do PostgreSQL, na seção Meta-comandos.

Importar dados do Amazon S3 para uma instância de banco de dados do RDS para PostgreSQL

Você pode importar dados que foram armazenados usando o Amazon Simple Storage Service em uma tabela em uma instância de banco de dados do RDS para PostgreSQL. Para fazer isso, primeiro instale a extensão `aws_s3` do RDS para PostgreSQL. Essa extensão fornece as funções usadas para importar dados de um bucket do Amazon S3. Um bucket é um contêiner do Amazon S3 para objetos e arquivos. Os dados podem estar em um arquivo de valores separados por vírgula (CSV), em um arquivo de texto ou em um arquivo compactado (gzip). Veja a seguir como instalar a extensão e importar dados do Amazon S3 para uma tabela.

Seu banco de dados deve estar executando o PostgreSQL versão 10.7 ou superior para importar do Amazon S3 para o RDS for PostgreSQL.

Se você não tiver dados armazenados no Amazon S3, crie um bucket e armazene os dados. Para ter mais informações, consulte os tópicos a seguir no Guia do usuário do Amazon Simple Storage Service.

- Criar um bucket
- Adicionar um objeto a um bucket

A importação entre contas do Amazon S3 é compatível. Para ter mais informações, consulte Conceder permissões entre contas no Guia do usuário do Amazon Simple Storage Service.

Você pode usar a chave gerenciada pelo cliente para criptografia ao importar dados do S3. Para ter mais informações, consulte Chaves do KMS armazenadas no AWS KMS no Guia do usuário do Amazon Simple Storage Service.

Tópicos

- Instalar a extensão aws_s3
- Visão geral da importação de dados do Amazon S3

- Configurar o acesso a um bucket do Amazon S3
- Importar dados do Amazon S3 para uma instância de banco de dados do RDS para PostgreSQL
- Referência de funções

Instalar a extensão aws_s3

Antes de usar o Amazon S3 com a sua instância de banco de dados do RDS para PostgreSQL, você precisa instalar a extensão `aws_s3`. Essa extensão fornece as funções para importar dados do Amazon S3. Ela também fornece funções para exportar dados de uma instância de banco de dados do RDS para PostgreSQL para um bucket do Amazon S3. Para ter mais informações, consulte Exportar dados de uma instância de banco de dados do RDS para PostgreSQL para o Amazon S3. A extensão `aws_s3` depende de algumas das funções auxiliares da extensão `aws_commons`, que é instalada automaticamente quando necessária.

Como instalar a extensão **`aws_s3`**

1. Use psql (ou pgAdmin) para se conectar à instância de banco de dados do RDS para PostgreSQL como um usuário que tem privilégios `rds_superuser`. Se você manteve o nome padrão durante o processo de configuração, se conectará como `postgres`.

```
psql --host=111122223333.aws-region.rds.amazonaws.com --port=5432 --
username=postgres --password
```

2. Para instalar a extensão, execute o comando a seguir.

```
postgres=> CREATE EXTENSION aws_s3 CASCADE;
NOTICE: installing required extension "aws_commons"
CREATE EXTENSION
```

3. Para verificar se a extensão está instalada, você pode usar o metacomando psql `\dx`.

```
postgres=> \dx
       List of installed extensions
    Name     | Version |   Schema   |                Description
-------------+---------+------------+-------------------------------------------
 aws_commons | 1.2     | public     | Common data types across AWS services
 aws_s3      | 1.1     | public     | AWS S3 extension for importing data from S3
 plpgsql     | 1.0     | pg_catalog | PL/pgSQL procedural language
(3 rows)
```

As funções para importar dados do Amazon S3 e exportar dados para o Amazon S3 agora estão disponíveis para uso.

Visão geral da importação de dados do Amazon S3

Para importar dados do S3 para o Amazon RDS

Primeiro, reúna os detalhes que você precisa fornecer à função. Isso inclui o nome da tabela na sua instância de banco de dados do RDS para PostgreSQL, e o nome do bucket, o caminho do arquivo, o tipo de arquivo e a Região da AWS em que os dados do Amazon S3 estão armazenados. Para ter mais informações, consulte Visualizar um objeto no Guia do usuário do Amazon Simple Storage Service.

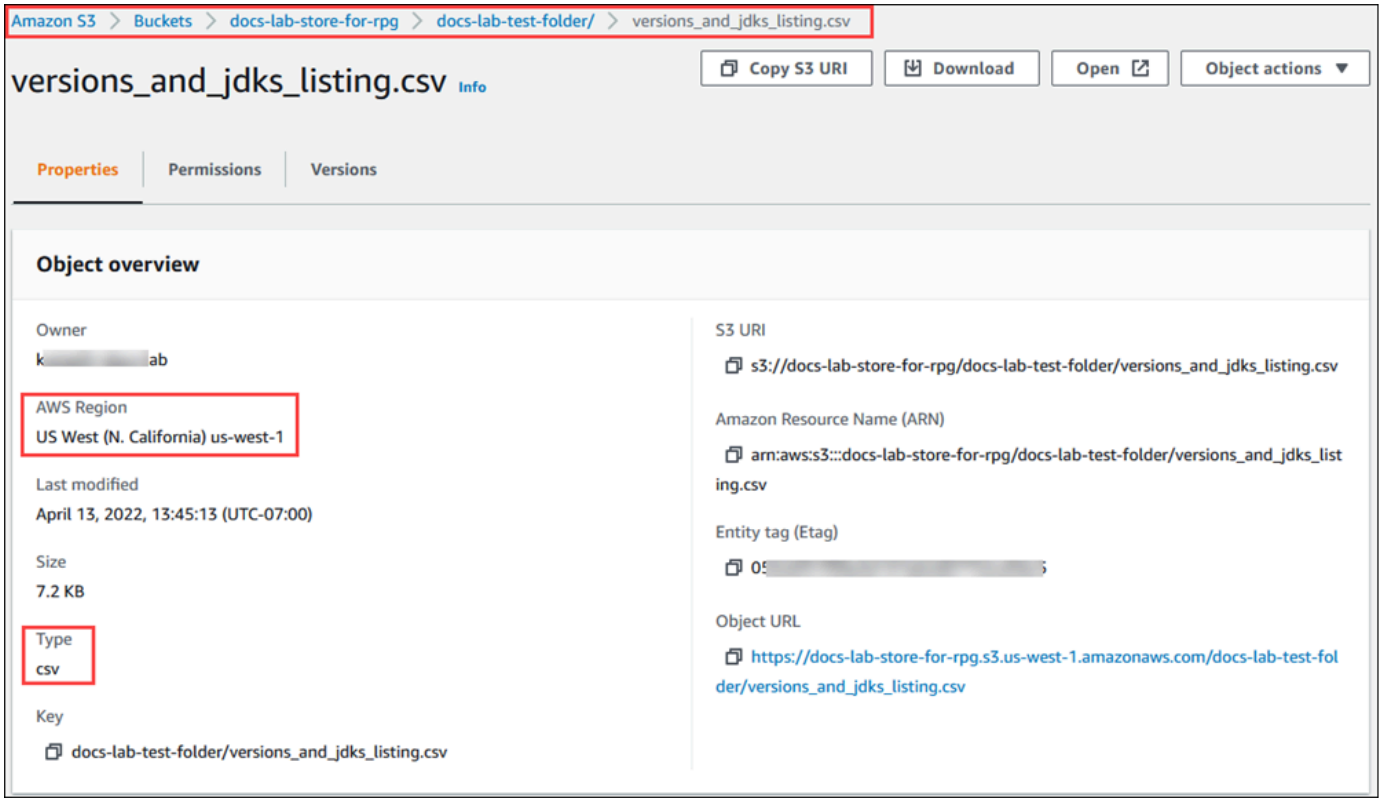
> **Note**
>
> A importação de dados multipart do Amazon S3 não é compatível atualmente.

1. Obtenha o nome da tabela para a qual a função `aws_s3.table_import_from_s3` deverá importar os dados. Por exemplo, o comando a seguir cria uma tabela `t1` que pode ser utilizada em etapas posteriores.

   ```
   postgres=> CREATE TABLE t1
       (col1 varchar(80),
       col2 varchar(80),
       col3 varchar(80));
   ```

2. Veja os detalhes sobre o bucket do Amazon S3 e os dados a importar. Para fazer isso, abra o console do Amazon S3 em https://console.aws.amazon.com/s3/ e escolha Buckets. Encontre o bucket que contém seus dados na lista. Escolha o bucket, abra a página Object overview (Visão geral do objeto) e escolha Properties (Propriedades).

   Anote o nome do bucket, o caminho, a Região da AWS e o tipo de arquivo. Posteriormente, você precisará do nome do recurso da Amazon (ARN) para configurar o acesso ao Amazon S3 por meio de um perfil do IAM. Para ter mais informações, consulte Configurar o acesso a um bucket do Amazon S3. A imagem a seguir mostra um exemplo.



3. Você pode verificar o caminho dos dados no bucket do Amazon S3 usando o comando `aws s3 cp` da AWS CLI. Se as informações estiverem corretas, esse comando baixará uma cópia do arquivo do Amazon S3.

```
aws s3 cp s3://amzn-s3-demo-bucket/sample_file_path ./
```

4. Configure permissões em sua instância de banco de dados do RDS para PostgreSQLpara permitir acesso ao arquivo no bucket do Amazon S3. Para fazer isso, use um perfil AWS Identity and Access Management (do IAM) ou credenciais de segurança. Para ter mais informações, consulte Configurar o acesso a um bucket do Amazon S3.

5. Forneça o caminho e outros detalhes do objeto do Amazon S3 coletados (consulte a etapa 2) para a função `create_s3_uri` a fim de construir um objeto URI do Amazon S3. Para saber mais sobre essa função, consulte aws_commons.create_s3_uri. Veja a seguir um exemplo de como construir esse objeto durante uma sessão psql.

```
postgres=> SELECT aws_commons.create_s3_uri(
   'docs-lab-store-for-rpg',
   'versions_and_jdks_listing.csv',
   'us-west-1'
) AS s3_uri \gset
```

Na próxima etapa, passe esse objeto (`aws_commons._s3_uri_1`) para a função `aws_s3.table_import_from_s3` a fim de importar os dados para a tabela.

6. Invoque a função `aws_s3.table_import_from_s3` para importar os dados do Amazon S3 para a sua tabela. Para obter informações de referência, consulte aws_s3.table_import_from_s3. Para obter exemplos, consulte Importar dados do Amazon S3 para uma instância de banco de dados do RDS para PostgreSQL.

Configurar o acesso a um bucket do Amazon S3

Para importar dados de um arquivo do Amazon S3, conceda à instância de banco de dados do RDS for PostgreSQL permissão para acessar o bucket do Amazon S3 no qual o arquivo se encontra. Conceda acesso a um bucket do Amazon S3 de uma das duas maneiras, conforme descrito nos tópicos a seguir.

Tópicos

- Usar uma função do IAM para acessar um bucket do Amazon S3
- Usar credenciais de segurança para acessar um bucket do Amazon S3
- Solução de problemas de acesso ao Amazon S3

Usar uma função do IAM para acessar um bucket do Amazon S3

Antes de carregar dados de um arquivo do Amazon S3, conceda à instância de banco de dados do RDS para PostgreSQL permissão para acessar o bucket do Amazon S3 no qual o arquivo está. Dessa forma, não é necessário gerenciar informações adicionais de credenciais ou fornecê-las na chamada da função aws_s3.table_import_from_s3.

Para fazer isso, crie uma política do IAM que conceda acesso ao bucket do Amazon S3. Crie uma função do IAM e associe a política a ela. Depois, atribua uma função do IAM à instância de banco de dados.

Para oferecer ao Simple Storage Service (Amazon S3) acesso a uma instância de banco de dados do RDS for PostgreSQL por meio de uma função do IAM

1. Crie uma política do IAM.

   Essa política concede ao bucket e ao objeto as permissões para que a instância de banco de dados do RDS para PostgreSQL acesse o Amazon S3.

Inclua na política as seguintes ações necessárias para permitir a transferência de arquivos de um bucket do Amazon S3 para o Amazon RDS do :

- `s3:GetObject`
- `s3:ListBucket`

Inclua na política os recursos a seguir para identificar o bucket e os objetos do Amazon S3 no bucket. Isso mostra o formato do nome de recurso da Amazon (ARN) para acessar o Amazon S3.

- arn:aws:s3:::*amzn-s3-demo-bucket*
- arn:aws:s3:::*amzn-s3-demo-bucket*/*

Para ter mais informações sobre como criar uma política do IAM para o RDS para PostgreSQL, consulte Criar e usar uma política do IAM para acesso do banco de dados do IAM. Consulte também Tutorial: Criar e anexar sua primeira política gerenciada pelo cliente no Guia do usuário do IAM.

O comando da AWS CLI a seguir cria uma política do IAM denominada `rds-s3-import-policy` com essas opções. Ele concede acesso a um bucket denominado *amzn-s3-demo-bucket*.

> **Note**
>
> Anote o nome do recurso da Amazon (ARN) da política que é retornado por esse comando. O ARN será necessário para uma etapa posterior, quando você anexar a política a um perfil do IAM.

Example

Para Linux, macOS ou Unix:

```
aws iam create-policy \
   --policy-name rds-s3-import-policy \
   --policy-document '{
     "Version": "2012-10-17",
```

```
    "Statement": [
      {
        "Sid": "s3import",
        "Action": [
          "s3:GetObject",
          "s3:ListBucket"
        ],
        "Effect": "Allow",
        "Resource": [
          "arn:aws:s3:::amzn-s3-demo-bucket",
          "arn:aws:s3:::amzn-s3-demo-bucket/*"
        ]
      }
    ]
  }'
```

Para Windows:

```
aws iam create-policy ^
   --policy-name rds-s3-import-policy ^
   --policy-document '{
     "Version": "2012-10-17",
     "Statement": [
       {
         "Sid": "s3import",
         "Action": [
           "s3:GetObject",
           "s3:ListBucket"
         ],
         "Effect": "Allow",
         "Resource": [
           "arn:aws:s3:::amzn-s3-demo-bucket",
           "arn:aws:s3:::amzn-s3-demo-bucket/*"
         ]
       }
     ]
   }'
```

2. Criar um perfil do IAM.

   Faça isso para que o Amazon RDS possa assumir essa função do IAM para acessar os buckets do Amazon S3. Para ter mais informações, consulte Criar um perfil para delegar permissões a um usuário do IAM no Guia do usuário do IAM.

Convém usar as chaves de contexto de condição global `aws:SourceArn` e `aws:SourceAccount` em políticas baseadas em recursos para limitar as permissões do serviço a um recurso específico. Essa é a maneira mais eficiente de se proteger contra o problema "confused deputy".

Se você utilizar ambas as chaves de contexto de condição global, e o valor `aws:SourceArn` contiver o ID da conta, o valor `aws:SourceAccount` e a conta no valor `aws:SourceArn` deverão utilizar o mesmo ID de conta quando utilizados na mesma instrução de política.

- Use `aws:SourceArn` se quiser acesso entre serviços para um único recurso.
- Use `aws:SourceAccount` se você quiser permitir que qualquer recurso nessa conta seja associado ao uso entre serviços.

Na política, certifique-se de usar a chave de contexto de condição global `aws:SourceArn` com o ARN completo do recurso. O exemplo a seguir mostra como fazer isso utilizando o comando da AWS CLI para criar uma função chamada `rds-s3-import-role`.

Example

Para Linux, macOS ou Unix:

```
aws iam create-role \
   --role-name rds-s3-import-role \
   --assume-role-policy-document '{
     "Version": "2012-10-17",
     "Statement": [
       {
         "Effect": "Allow",
         "Principal": {
            "Service": "rds.amazonaws.com"
          },
         "Action": "sts:AssumeRole",
         "Condition": {
             "StringEquals": {
                "aws:SourceAccount": "111122223333",
                "aws:SourceArn": "arn:aws:rds:us-east-1:111122223333:db:dbname"
                }
             }
       }
     ]
```

```
   }'
```

Para Windows:

```
aws iam create-role ^
   --role-name rds-s3-import-role ^
   --assume-role-policy-document '{
     "Version": "2012-10-17",
     "Statement": [
       {
         "Effect": "Allow",
         "Principal": {
            "Service": "rds.amazonaws.com"
          },
         "Action": "sts:AssumeRole",
         "Condition": {
             "StringEquals": {
                "aws:SourceAccount": "111122223333",
                "aws:SourceArn": "arn:aws:rds:us-east-1:111122223333:db:dbname"
                }
             }
       }
     ]
   }'
```

3. Anexe a política do IAM que você criou ao perfil do IAM que você criou.

   O comando da AWS CLI a seguir vincula a política criada na etapa anterior à função `rds-s3-import-role`. Substitua *`your-policy-arn`* pelo ARN da política que você anotou naquela ocasião.

   Example

   Para Linux, macOS ou Unix:

```
aws iam attach-role-policy \
   --policy-arn your-policy-arn \
   --role-name rds-s3-import-role
```

   Para Windows:

```
aws iam attach-role-policy ^
```

```
    --policy-arn your-policy-arn ^
    --role-name rds-s3-import-role
```

4. Adicione o perfil do IAM à instância de banco de dados.

   Faça isso usando o AWS Management Console ou a AWS CLI, conforme descrito a seguir.

Console

Para adicionar um perfil do IAM a uma instância de banco de dados do PostgreSQL usando o console

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. Escolha o nome da instância de de banco de dados do PostgreSQL para exibir os detalhes.
3. Na guia Connectivity & security (Conectividade e segurança), na seção Manage IAM roles (Gerenciar funções do IAM), escolha a função a ser adicionada sob Add IAM roles to this instance (Adicionar funções do IAM a este cluster/esta instância).
4. Em Feature (Recurso), escolha s3Import.
5. Escolha Add role (adicionar função).

AWS CLI

Para adicionar um perfil do IAM para uma instância de banco de dados do PostgreSQL usando a CLI

- Use o comando a seguir para adicionar a função à instância de banco de dados do PostgreSQL chamada `my-db-instance`. Substitua*your-role-arn* pelo ARN da função que você anotou em uma etapa anterior. Use `s3Import` para o valor da opção `--feature-name`.

  Example

  Para Linux, macOS ou Unix:

```
aws rds add-role-to-db-instance \
   --db-instance-identifier my-db-instance \
   --feature-name s3Import \
   --role-arn your-role-arn   \
   --region your-region
```

Para Windows:

```
aws rds add-role-to-db-instance ^
   --db-instance-identifier my-db-instance ^
   --feature-name s3Import ^
   --role-arn your-role-arn ^
   --region your-region
```

API do RDS

Para adicionar uma função do IAM a uma instância de banco de dados PostgreSQL utilizando a API do Amazon RDS, chame a operação AddRoleToDBInstance.

Usar credenciais de segurança para acessar um bucket do Amazon S3

Se preferir, você poderá usar credenciais de segurança para conceder acesso a um bucket do Amazon S3 em vez de conceder acesso com uma função do IAM. Faça isso especificando o parâmetro `credentials` na chamada da função aws_s3.table_import_from_s3.

O parâmetro `credentials` é uma estrutura do tipo `aws_commons._aws_credentials_1`, que contém credenciais da AWS. Use a função aws_commons.create_aws_credentials para definir a chave de acesso e a chave secreta em uma estrutura `aws_commons._aws_credentials_1`, conforme mostrado a seguir.

```
postgres=> SELECT aws_commons.create_aws_credentials(
   'sample_access_key', 'sample_secret_key', '')
AS creds \gset
```

Depois de criar a estrutura `aws_commons._aws_credentials_1` , use a função aws_s3.table_import_from_s3 com o parâmetro `credentials` para importar os dados, conforme mostrado a seguir.

```
postgres=> SELECT aws_s3.table_import_from_s3(
   't', '', '(format csv)',
   :'s3_uri',
   :'creds'
);
```

Outra opção é incluir a chamada de função aws_commons.create_aws_credentials em linha dentro da chamada de função `aws_s3.table_import_from_s3`.

```
postgres=> SELECT aws_s3.table_import_from_s3(
   't', '', '(format csv)',
   :'s3_uri',
   aws_commons.create_aws_credentials('sample_access_key', 'sample_secret_key', '')
);
```

Solução de problemas de acesso ao Amazon S3

Se você encontrar problemas de conexão ao tentar importar dados do Amazon S3, consulte o seguinte para obter recomendações:

- Solução de problemas de identidade e acesso do Amazon RDS
- Solução de problemas do Amazon S3 no Guia do usuário do Amazon Simple Storage Service
- Troubleshooting Amazon S3 and IAM (Solucionar problemas no Amazon S3 e no IAM) no Guia do usuário do IAM

Importar dados do Amazon S3 para uma instância de banco de dados do RDS para PostgreSQL

Importe dados do bucket do Amazon S3 usando a função `table_import_from_s3` da extensão aws_s3. Para obter informações de referência, consulte aws_s3.table_import_from_s3.

> **Note**
>
> Os exemplos a seguir usam o método de perfil do IAM para conceder acesso ao bucket do Amazon S3. Assim, as chamadas de função do `aws_s3.table_import_from_s3` não incluem parâmetros de credenciais.

Veja a seguir um exemplo típico.

```
postgres=> SELECT aws_s3.table_import_from_s3(
   't1',
   '',
   '(format csv)',
```

```
   :'s3_uri'
);
```

Os parâmetros são os seguintes:

- `t1` – o nome da tabela na instância de banco de dados do PostgreSQL na qual os dados devem ser copiados.
- `''` – uma lista opcional de colunas na tabela de banco de dados. É possível usar esse parâmetro para indicar quais colunas dos dados do S3 devem ser inseridos em quais colunas da tabela. Se nenhuma coluna for especificada, todas as colunas serão copiadas para a tabela. Para obter um exemplo de uso de uma lista de colunas, consulte Importar um arquivo do Amazon S3 que usa um delimitador personalizado.
- `(format csv)` – argumentos COPY do PostgreSQL. O processo de cópia usa os argumentos e o formato do comando COPY PostgreSQL para importar os dados. As opções de formato incluem valores separados por vírgula (CSV), conforme mostrado neste exemplo, texto e binário. O padrão é texto.
- `s3_uri` – uma estrutura que contém as informações que identificam o arquivo do Amazon S3. Para obter um exemplo de como usar a função aws_commons.create_s3_uri para criar uma estrutura `s3_uri`, consulte Visão geral da importação de dados do Amazon S3.

Para ter mais informações sobre essa função, consulte aws_s3.table_import_from_s3.

A função `aws_s3.table_import_from_s3` retorna texto. Para especificar outros tipos de arquivos para importação de um bucket do Amazon S3, veja um dos exemplos a seguir.

> **Note**
>
> Importar 0 byte causará um erro.

Tópicos

- Importar um arquivo do Amazon S3 que usa um delimitador personalizado
- Importar um arquivo do Amazon S3 compactado (gzip)
- Importar um arquivo do Amazon S3 codificado

Importar um arquivo do Amazon S3 que usa um delimitador personalizado

O exemplo a seguir mostra como importar um arquivo que usa um delimitador personalizado. Ele também mostra como controlar onde colocar os dados na tabela de banco de dados usando o parâmetro `column_list` da função aws_s3.table_import_from_s3.

Para este exemplo, suponha que as informações a seguir estejam organizadas em colunas delimitadas por barras verticais no arquivo do Amazon S3.

```
1|foo1|bar1|elephant1
2|foo2|bar2|elephant2
3|foo3|bar3|elephant3
4|foo4|bar4|elephant4
...
```

Como importar um arquivo que usa um delimitador personalizado

1. Crie uma tabela no banco de dados para os dados importados.

   ```
   postgres=> CREATE TABLE test (a text, b text, c text, d text, e text);
   ```

2. Use o seguinte formato da função aws_s3.table_import_from_s3 para importar dados do arquivo do Amazon S3.

   Você pode incluir a chamada de função aws_commons.create_s3_uri em linha dentro da chamada de função `aws_s3.table_import_from_s3` para especificar o arquivo.

   ```
   postgres=> SELECT aws_s3.table_import_from_s3(
      'test',
      'a,b,d,e',
      'DELIMITER ''|''',
      aws_commons.create_s3_uri('amzn-s3-demo-bucket', 'pipeDelimitedSampleFile', 'us-
   east-2')
   );
   ```

Os dados estão agora na tabela nas colunas a seguir.

```
postgres=> SELECT * FROM test;
a | b | c | d | e
---+------+---+---+------+-----------
```

```
1 | foo1 | | bar1 | elephant1
2 | foo2 | | bar2 | elephant2
3 | foo3 | | bar3 | elephant3
4 | foo4 | | bar4 | elephant4
```

Importar um arquivo do Amazon S3 compactado (gzip)

O exemplo a seguir mostra como importar um arquivo do Amazon S3 compactado com gzip. O arquivo que você importa precisa ter os seguintes metadados do Simple Storage Service (Amazon S3):

- Chave: `Content-Encoding`
- Valor: `gzip`

Se você carregar o arquivo usando o AWS Management Console, os metadados geralmente são aplicados pelo sistema. Para obter informações sobre o carregamento de arquivos para o Simple Storage Service (Amazon S3) usando o AWS Management Console, a AWS CLI ou a API, consulte Carregar objetos no Guia do usuário do Amazon Simple Storage Service.

Para ter mais informações sobre metadados do Simple Storage Service (Amazon S3) e detalhes sobre metadados fornecidos pelo sistema, consulte Editar metadados de objeto no console do Simple Storage Service (Amazon S3) no Guia do usuário do Amazon Simple Storage Service.

Importe o arquivo gzip para a sua instância de banco de dados do RDS for PostgreSQL do , conforme mostrado a seguir.

```
postgres=> CREATE TABLE test_gzip(id int, a text, b text, c text, d text);
postgres=> SELECT aws_s3.table_import_from_s3(
 'test_gzip', '', '(format csv)',
 'amzn-s3-demo-bucket', 'test-data.gz', 'us-east-2'
);
```

Importar um arquivo do Amazon S3 codificado

O exemplo a seguir mostra como importar um arquivo do Amazon S3 que tenha codificação Windows-1252.

```
postgres=> SELECT aws_s3.table_import_from_s3(
 'test_table', '', 'encoding ''WIN1252''',
```

```
 aws_commons.create_s3_uri('amzn-s3-demo-bucket', 'SampleFile', 'us-east-2')
);
```

Referência de funções

Funções

- aws_s3.table_import_from_s3
- aws_commons.create_s3_uri
- aws_commons.create_aws_credentials

aws_s3.table_import_from_s3

Importa dados do Amazon S3 para uma tabela doAmazon RDS do . A extensão `aws_s3` fornece a função `aws_s3.table_import_from_s3`. O valor de retorno é texto.

Sintaxe

Os parâmetros necessários são `table_name`, `column_list` e `options`. Eles identificam a tabela do banco de dados e especificam como os dados são copiados nessa tabela.

Você também pode usar os seguintes parâmetros:

- O parâmetro `s3_info` especifica o arquivo do Amazon S3 a ser importado. Quando você usa esse parâmetro, o acesso ao Amazon S3 é fornecido por uma função do IAM para a instância do de banco de dados do PostgreSQL.

```
aws_s3.table_import_from_s3 (
   table_name text,
   column_list text,
   options text,
   s3_info aws_commons._s3_uri_1
)
```

- O parâmetro `credentials` especifica as credenciais para acessar o Amazon S3. Ao usar esse parâmetro, você não usa uma função do IAM.

```
aws_s3.table_import_from_s3 (
   table_name text,
   column_list text,
```

```
   options text,
   s3_info aws_commons._s3_uri_1,
   credentials aws_commons._aws_credentials_1
)
```

Parâmetros

table_name

Uma string de texto necessária contendo o nome da tabela de banco de dados PostgreSQL para a qual importar os dados.

column_list

Uma string de texto necessária contendo uma lista opcional das colunas da tabela de banco de dados PostgreSQL para a qual copiar os dados. Se a string estiver vazia, todas as colunas da tabela serão usadas. Para ver um exemplo, consulte Importar um arquivo do Amazon S3 que usa um delimitador personalizado.

options

Uma string de texto necessária contendo argumentos para o comando PostgreSQL `COPY`. Esses argumentos especificam como os dados devem ser copiados para a tabela do PostgreSQL. Para obter mais detalhes, consulte Documentação de COPY do PostgreSQL.

s3_info

Um tipo composto `aws_commons._s3_uri_1` contendo as seguintes informações sobre o objeto do S3:

- `bucket` – O nome do Amazon S3 bucket que contém o arquivo.
- `file_path` – o nome do Amazon S3 arquivo, incluindo o caminho do arquivo.
- `region`: a região da AWS em que o arquivo se encontra. Para obter uma listagem de nomes de regiões da AWS e os valores associados, consulte Regiões, zonas de disponibilidade e Local Zones.

credenciais

Um tipo composto `aws_commons._aws_credentials_1` contendo as seguintes credenciais a serem usadas para a operação de importação:

- Chave de acesso

- Chave secreta
- Token de sessão

Para obter informações sobre como criar uma estrutura `aws_commons._aws_credentials_1` composta, consulte aws_commons.create_aws_credentials.

Sintaxe alternativa

Para ajudar nos testes, você pode usar um conjunto expandido de parâmetros em vez dos parâmetros `s3_info` e `credentials`. Veja a seguir as variações acionais da sintaxe da função `aws_s3.table_import_from_s3`.

- Em vez de usar o parâmetro `s3_info` para identificar um arquivo do Amazon S3, use a combinação dos parâmetros `bucket`, `file_path` e `region`. Com essa forma da função, o acesso ao Amazon S3 é fornecido por uma função do IAM na instância de banco de dados do PostgreSQL.

```
aws_s3.table_import_from_s3 (
   table_name text,
   column_list text,
   options text,
   bucket text,
   file_path text,
   region text
)
```

- Em vez de usar o parâmetro `credentials` para especificar o acesso ao Amazon S3, use a combinação dos parâmetros `access_key`, `session_key` e `session_token`.

```
aws_s3.table_import_from_s3 (
   table_name text,
   column_list text,
   options text,
   bucket text,
   file_path text,
   region text,
   access_key text,
   secret_key text,
   session_token text
)
```

Parâmetros alternativos

bucket

Uma string de texto contendo o nome do bucket do Amazon S3 que contém o arquivo.

file_path

Uma string de texto contendo o nome do arquivo do Amazon S3, incluindo o caminho do arquivo.

região

Uma string de texto que identifica a Região da AWS do arquivo. Para obter uma listagem de nomes de Região da AWS e os valores associados, consulte Regiões, zonas de disponibilidade e Local Zones.

access_key

Uma string de texto contendo a chave de acesso a ser usada para a operação de importação. O padrão é NULL.

secret_key

Uma string de texto contendo a chave secreta a ser usada para a operação de importação. O padrão é NULL.

session_token

(Opcional) Uma string de texto contendo a chave de sessão a ser usada para a operação de importação. O padrão é NULL.

aws_commons.create_s3_uri

Cria uma estrutura `aws_commons._s3_uri_1` para comportar informações do arquivo do Amazon S3. Use os resultados da função `aws_commons.create_s3_uri` no parâmetro `s3_info` da função aws_s3.table_import_from_s3.

Sintaxe

```
aws_commons.create_s3_uri(
   bucket text,
   file_path text,
   region text
)
```

Parâmetros

bucket

Uma string de texto necessária contendo o nome do bucket do Amazon S3 para o arquivo.

file_path

Uma string de texto necessária contendo o nome do arquivo do Amazon S3, incluindo o caminho do arquivo.

região

Uma string de texto necessária que contém a Região da AWS na qual o arquivo se encontra. Para obter uma listagem de nomes de Região da AWS e os valores associados, consulte Regiões, zonas de disponibilidade e Local Zones.

aws_commons.create_aws_credentials

Define uma chave de acesso e uma chave secreta em uma estrutura `aws_commons._aws_credentials_1`. Use os resultados da função `aws_commons.create_aws_credentials` no parâmetro `credentials` da função aws_s3.table_import_from_s3.

Sintaxe

```
aws_commons.create_aws_credentials(
   access_key text,
   secret_key text,
   session_token text
)
```

Parâmetros

access_key

Uma string de texto necessária contendo a chave de acesso a ser usada para importar um arquivo do Amazon S3. O padrão é NULL.

secret_key

Uma string de texto necessária contendo a chave secreta a ser usada para importar um arquivo do Amazon S3. O padrão é NULL.

session_token

Uma string de texto opcional contendo o token de sessão a ser usado para importar um arquivo do Amazon S3. O padrão é NULL. Se você fornecer um `session_token` opcional, poderá usar credenciais temporárias.

Transportar bancos de dados PostgreSQL entre instâncias de banco de dados

Usando os bancos de dados PostgreSQL transportáveis para Amazon RDS, você pode mover um banco de dados PostgreSQL entre duas instâncias de banco de dados. Essa é uma maneira muito rápida de migrar bancos de dados grandes entre diferentes instâncias de banco de dados. Para usar essa abordagem, suas instâncias de banco de dados devem executar a mesma versão principal do PostgreSQL.

Esse recurso requer que você instale a extensão `pg_transport` nas instâncias de banco de dados de origem e de destino. A extensão `pg_transport` fornece um mecanismo de transporte físico que move os arquivos de banco de dados com o mínimo de processamento. Esse mecanismo move os dados muito mais rapidamente que os processos tradicionais de despejo e carregamento, com menos tempo de inatividade.

> **Note**
>
> Os bancos de dados PostgreSQL transportáveis estão disponíveis no RDS para PostgreSQL versões 11.5 e posteriores e 10.10 e posteriores.

Para transportar uma instância de banco de dados PostgreSQL de uma instância de banco de dados do RDS for PostgreSQL para outra, você primeiro configura as instâncias de origem e de destino, conforme detalhado em Configurar uma instância de banco de dados para transporte. Em seguida, você pode transportar o banco de dados usando a função descrita em Como transportar um banco de dados PostgreSQL.

Tópicos

- O que acontece durante o transporte do banco de dados
- Limitações para o uso de bancos de dados PostgreSQL transportáveis
- Configurar o transporte de um banco de dados PostgreSQL

- Transportar um banco de dados PostgreSQL para o destino a partir da origem
- Referência de funções de bancos de dados transportáveis
- Referência de parâmetros de bancos de dados transportáveis

O que acontece durante o transporte do banco de dados

O recurso de bancos de dados PostgreSQL transportáveis usa um modelo pull para importar o banco de dados da instância de banco de dados de origem para a de destino. A função `transport.import_from_server` cria o banco de dados em trânsito na instância de banco de dados de destino. O banco de dados em trânsito está inacessível na instância de banco de dados de destino durante o transporte.

Quando o transporte começa, todas as sessões atuais no banco de dados de origem são encerradas. Quaisquer bancos de dados que não sejam o banco de dados de origem na instância de banco de dados de origem não são afetados pelo transporte.

O banco de dados de origem é colocado em um modo somente leitura especial. Enquanto estiver nesse modo, você pode se conectar ao banco de dados de origem e executar consultas somente leitura. No entanto, as consultas habilitadas para gravação e alguns outros tipos de comandos estão bloqueados. Somente o banco de dados de origem específico que está sendo transportado é afetado por essas restrições.

Durante o transporte, você não pode restaurar a instância do banco de dados de destino em um determinado momento. Isso ocorre porque o transporte não é transacional e não usa o log de gravação antecipada do PostgreSQL para registrar as alterações. Se a instância de banco de dados de destino tiver backups automáticos ativados, um backup será feito automaticamente após a conclusão do transporte. As restaurações em um ponto anterior no tempo ficam disponíveis por algumas horas após a conclusão do backup.

Se o transporte falhar, a extensão `pg_transport` tenta desfazer todas as alterações nas instâncias de banco de dados de origem e destino. Isso inclui a remoção do banco de dados parcialmente transportado do destino. Dependendo do tipo de falha, o banco de dados de origem pode continuar a rejeitar consultas habilitadas para gravação. Se isso acontecer, use o comando a seguir para permitir consultas habilitadas para gravação.

```
ALTER DATABASE db-name SET default_transaction_read_only = false;
```

Limitações para o uso de bancos de dados PostgreSQL transportáveis

Os bancos de dados transportáveis têm as seguintes limitações:

- Réplicas de leitura – não é possível usar bancos de dados transportáveis em réplicas de leitura nem em instâncias pai de réplicas de leitura.
- Tipos de coluna não compatíveis – não é possível usar os tipos de dados `reg` em nenhuma tabela de banco de dados que você planeja transportar com esse método. Esses tipos dependem dos IDs de objeto (OIDs) do catálogo do sistema, que geralmente são alterados durante o transporte.
- Espaços de tabela – todos os objetos do banco de dados de origem devem estar no espaço de tabela `pg_default` padrão.
- Compatibilidade – as instâncias de banco de dados de origem e destino devem executar a mesma versão principal do PostgreSQL.
- Extensões: a instância de banco de dados de origem pode ter apenas a extensão `pg_transport` instalada.
- Funções e ACLs – os privilégios de acesso e as informações de propriedade do banco de dados de origem não são transferidos para o banco de dados de destino. Todos os objetos de banco de dados são criados e pertencentes ao usuário de destino local do transporte.
- Transportes simultâneos: uma única instância de banco de dados pode aceitar até 32 transportes simultâneos, incluindo importações e exportações, se os processos do operador tiverem sido configurados corretamente.
- Somente para instâncias de banco de dados do RDS for PostgreSQL: os bancos de dados transportáveis do PostgreSQL são compatíveis apenas com instâncias de banco de dados do RDS for PostgreSQL. Não é possível utilizá-los com bancos de dados locais ou bancos de dados em execução no Amazon EC2.

Configurar o transporte de um banco de dados PostgreSQL

Antes de começar, certifique-se de que as instâncias de banco de dados do RDS for PostgreSQL atendam aos seguintes requisitos:

- As instâncias de banco de dados do RDS for PostgreSQL para a origem e o destino devem ser executadas na mesma versão do PostgreSQL.
- O banco de dados de destino não pode ter um banco de dados com o mesmo nome do banco de dados de origem que você deseja transportar.

- A conta que você usa para executar o transporte precisa dos privilégios `rds_superuser` nos bancos de dados de origem e de destino.
- O grupo de segurança da instância de banco de dados de origem deve permitir acesso de entrada da instância de banco de dados de destino. Isso pode já ser o caso se as instâncias de banco de dados de origem e de destino estiverem localizadas na VPC. Para obter mais informações sobre grupo de seguranças, consulte Controlar acesso com grupos de segurança.

O transporte de bancos de dados de uma instância de banco de dados de origem para uma instância de banco de dados de destino requer várias alterações no grupo de parâmetros de banco de dados associado a cada instância. Isso significa que você deve criar um grupo de parâmetros de banco de dados personalizado para a instância de banco de dados de origem e criar um grupo de parâmetros de banco de dados personalizado para a instância de banco de dados de destino.

> **Note**
>
> Se suas instâncias de banco de dados já estiverem configuradas usando grupos de parâmetros de banco de dados personalizados, você poderá começar com a etapa 2 no procedimento a seguir.

Para configurar os parâmetros de grupos de banco de dados personalizados para o transporte de bancos de dados

Para as etapas a seguir, use uma conta que tenha os privilégios `rds_superuser`.

1. Se as instâncias de banco de dados de origem e de destino usarem um grupo de parâmetros de banco de dados padrão, você precisará criar um grupo de parâmetros de banco de dados personalizado usando a versão apropriada para suas instâncias. Você faz isso para poder alterar valores para vários parâmetros. Para obter mais informações, consulte Grupos de parâmetros para Amazon RDS.
2. No grupo de parâmetros de banco de dados personalizado, altere os valores dos seguintes parâmetros:
   - `shared_preload_libraries`: adicionar `pg_transport` à lista de bibliotecas.
   - `pg_transport.num_workers`: o valor padrão é 3. Aumente ou reduza esse valor conforme necessário para o banco de dados. Para um banco de dados de 200 GB, recomendamos não

mais que 8. Tenha em mente que, se você aumentar o valor padrão desse parâmetro, você também deverá aumentar o valor de `max_worker_processes`.

- `pg_transport.work_mem`: o valor padrão é 128 MB ou 256 MB, dependendo da versão do PostgreSQL. A configuração padrão geralmente pode ser deixada inalterada.
- `max_worker_processes`: o valor desse parâmetro precisa ser definido usando o seguinte cálculo:

```
(3 * pg_transport.num_workers) + 9
```

Esse valor é obrigatório no destino para lidar com vários processos de operador em segundo plano envolvidos no transporte. Para saber mais sobre `max_worker_processes`, consulte Consumo de recursos na documentação do PostgreSQL.

Para obter mais informações sobre parâmetros do `pg_transport`, consulte Referência de parâmetros de bancos de dados transportáveis .

3. Reinicialize a instância de banco de dados do RDS for PostgreSQL de origem e a instância de destino para que as configurações dos parâmetros entrem em vigor.
4. Conecte-se à sua instância de banco de dados do RDS for PostgreSQL de origem.

```
psql --host=source-instance.111122223333.aws-region.rds.amazonaws.com --port=5432
 --username=postgres --password
```

5. Remova extensões estranhas do esquema público da instância de banco de dados. Somente a extensão `pg_transport` é permitida durante a operação de transporte real.
6. Instale a extensão `pg_transport` da seguinte forma:

```
postgres=> CREATE EXTENSION pg_transport;
CREATE EXTENSION
```

7. Conecte-se à sua instância de banco de dados do RDS for PostgreSQL de destino. Remova qualquer extensão estranha e, em seguida, instale a extensão `pg_transport`.

```
postgres=> CREATE EXTENSION pg_transport;
CREATE EXTENSION
```

Transportar um banco de dados PostgreSQL para o destino a partir da origem

Depois de concluir o processo descrito em Configurar o transporte de um banco de dados PostgreSQL, você pode iniciar o transporte. Para fazer isso, execute a função `transport.import_from_server` na instância de banco de dados de destino. Na sintaxe a seguir, você pode encontrar os parâmetros da função.

```
SELECT transport.import_from_server(
   'source-db-instance-endpoint',
    source-db-instance-port,
   'source-db-instance-user',
   'source-user-password',
   'source-database-name',
   'destination-user-password',
   false);
```

O valor `false` mostrado no exemplo diz à função que esta não é uma simulação. Para testar sua configuração de transporte, você pode especificar `true` para `dry_run` quando você chama a função, conforme mostrado a seguir:

```
postgres=> SELECT transport.import_from_server(
    'docs-lab-source-db.666666666666aws-region.rds.amazonaws.com', 5432,
    'postgres', '********', 'labdb', '******', true);
INFO:  Starting dry-run of import of database "labdb".
INFO:  Created connections to remote database        (took 0.03 seconds).
INFO:  Checked remote cluster compatibility          (took 0.05 seconds).
INFO:  Dry-run complete                          (took 0.08 seconds total).
 import_from_server
--------------------

(1 row)
```

As linhas INFO são emitidas porque o parâmetro `pg_transport.timing` está definido como seu valor padrão, `true`. Defina `dry_run` para `false` quando você executa o comando e o banco de dados de origem é importado para o destino, conforme mostrado a seguir:

```
INFO:  Starting import of database "labdb".
INFO:  Created connections to remote database        (took 0.02 seconds).
INFO:  Marked remote database as read only           (took 0.13 seconds).
INFO:  Checked remote cluster compatibility          (took 0.03 seconds).
INFO:  Signaled creation of PITR blackout window     (took 2.01 seconds).
```

```
INFO:  Applied remote database schema pre-data       (took 0.50 seconds).
INFO:  Created connections to local cluster          (took 0.01 seconds).
INFO:  Locked down destination database              (took 0.00 seconds).
INFO:  Completed transfer of database files          (took 0.24 seconds).
INFO:  Completed clean up                            (took 1.02 seconds).
INFO:  Physical transport complete              (took 3.97 seconds total).
import_from_server
--------------------
(1 row)
```

Esta função requer que você forneça senhas de usuário do banco de dados. Portanto, recomendamos que você altere as senhas das funções de usuário usadas após a conclusão do transporte. Ou você pode usar variáveis de ligação do SQL para criar funções de usuário temporárias. Use essas funções temporárias para o transporte e descarte as funções posteriormente.

Quando o transporte não for bem-sucedido, talvez você veja uma mensagem de erro semelhante à seguinte:

```
pg_transport.num_workers=8 25% of files transported failed to download file data
```

A mensagem de erro “falha ao baixar dados do arquivo” indica que o número de processos de trabalho não está definido corretamente para o tamanho do banco de dados. Talvez seja necessário aumentar ou diminuir o valor definido para `pg_transport.num_workers`. Cada falha informa a porcentagem de conclusão, para que você possa ver o impacto de suas alterações. Por exemplo, alterar a configuração de 8 para 4 em um caso resultou no seguinte:

```
pg_transport.num_workers=4 75% of files transported failed to download file data
```

Lembre-se de que o parâmetro `max_worker_processes` também é levado em consideração durante o processo de transporte. Em outras palavras, talvez seja necessário modificar `pg_transport.num_workers` e `max_worker_processes` para transportar o banco de dados com êxito. O exemplo mostrado finalmente funcionou quando `pg_transport.num_workers` foi definido como 2:

```
pg_transport.num_workers=2 100% of files transported
```

Para obter mais informações sobre a função `transport.import_from_server` seus respectivos parâmetros de configuração, consulte Referência de funções de bancos de dados transportáveis.

Referência de funções de bancos de dados transportáveis

A função `transport.import_from_server` transporta um banco de dados PostgreSQL importando-o de uma instância de banco de dados de origem para uma instância de banco de dados de destino. Isso é feito usando um mecanismo de transporte de conexão de banco de dados físico.

Antes de iniciar o transporte, essa função verifica se as instâncias de banco de dados de origem e de destino são da mesma versão e são compatíveis com a migração. Também confirma que a instância de banco de dados de destino tem espaço suficiente para a origem.

Sintaxe

```
transport.import_from_server(
   host text,
   port int,
   username text,
   password text,
   database text,
   local_password text,
   dry_run bool
)
```

Valor de retorno

Nenhum.

Parâmetros

Você pode encontrar descrições dos parâmetros da função `transport.import_from_server` na tabela a seguir.

Parâmetro	Descrição
`host`	O endpoint da instância de banco de dados de origem.
`port`	Um número inteiro que representa a porta da instância de banco de dados de origem. As instâncias de banco de dados do PostgreSQL costumam usar a porta 5432.

Parâmetro	Descrição
`username`	O usuário da instância de banco de dados de origem. Este usuário deve ser um membro da função `rds_superuser` .
`password`	A senha da instância de banco de dados de origem.
`database`	O nome do banco de dados na instância do banco de dados de origem a ser transportada.
`local_password`	A senha local do usuário atual para a instância de banco de dados de destino. Este usuário deve ser um membro da função `rds_superuser` .
`dry_run`	Um valor booleano opcional que especifica se é necessário executar uma simulação. O padrão é `false`, o que significa que o transporte continua. Para confirmar a compatibilidade entre as instâncias de banco de dados de origem e destino sem executar o transporte real, configure `dry_run` como `true`.

Exemplo

Para ver um exemplo, consulte Transportar um banco de dados PostgreSQL para o destino a partir da origem.

Referência de parâmetros de bancos de dados transportáveis

Vários parâmetros controlam o comportamento da extensão `pg_transport`. A seguir, você pode encontrar as descrições desses parâmetros.

`pg_transport.num_workers`

O número de operadores a serem usados para o processo de transporte. O padrão é 3. Os valores válidos são 1–32. Geralmente, mesmo os maiores transportes de banco de dados exigem menos de oito operadores. O valor dessa configuração na instância de banco de dados de destino é usada pelo destino e pela origem durante o transporte.

`pg_transport.timing`

Especifica se é necessário relatar informações de tempo durante o transporte. O padrão é `true`, o que significa que as informações de tempo são relatadas. Recomendamos deixar esse

parâmetro definido como `true` para que você possa monitorar o progresso. Veja um exemplo de resultado em Transportar um banco de dados PostgreSQL para o destino a partir da origem.

`pg_transport.work_mem`

A quantidade máxima de memória a ser alocada para cada operador. O padrão é 131.072 kilobytes (KB) ou 262.144 KB (256 MB), dependendo da versão do PostgreSQL. O valor mínimo é de 64 megabytes (65.536 KB). Os valores válidos estão em kilobytes (KBs) como unidades binárias de base 2, em que 1 KB = 1.024 bytes.

O transporte pode usar menos memória que o especificado neste parâmetro. Geralmente, mesmo transportes grandes de banco de dados exigem menos de 256 MB (262.144 KB) de memória por operador.

Exportar dados de uma instância de banco de dados do RDS para PostgreSQL para o Amazon S3

Consulte dados em um RDS para PostgreSQL instância de banco de dados e exporte-os diretamente para arquivos armazenados em um bucket do Amazon S3. Para fazer isso, primeiro instale a extensão `aws_s3` do RDS para PostgreSQL. Essa extensão fornece as funções usadas para exportar os resultados de consultas para o Amazon S3. A seguir, é possível descobrir como instalar a extensão e exportar dados para o Amazon S3.

> **Note**
>
> A exportação entre contas não é compatível com o Amazon S3.

Todas as versões do RDS para PostgreSQL disponíveis no momento são compatíveis com a exportação de dados do Amazon Simple Storage Service. Para obter informações detalhadas sobre as versões, consulte Atualizações do Amazon RDS para PostgreSQL nas Notas de lançamento do Amazon RDS para PostgreSQL.

Se você não tiver um bucket configurado para sua exportação, consulte os tópicos a seguir no Guia do usuário do Amazon Simple Storage Service.

- Configuração do Amazon S3
- Como criar um bucket

Por padrão, os dados exportados do RDS para PostgreSQL ao Amazon S3 usam criptografia do lado do servidor com uma Chave gerenciada pela AWS. Se estiver usando a criptografia do bucket, o bucket do Amazon S3 deverá ser criptografado com uma chave do AWS Key Management Service (AWS KMS) (SSE-KMS). No momento, os buckets criptografados com chaves gerenciadas pelo Amazon S3 (SSE-S3) não são compatíveis.

> **Note**
>
> É possível salvar os dados do snapshot do banco de dados no Amazon S3 usando o AWS Management Console, a AWS CLI ou a API do Amazon RDS. Para ter mais informações, consulte Exportação de dados de snapshot de banco de dados para o Amazon S3 para Amazon RDS.

Tópicos

- Instalar a extensão aws_s3
- Visão geral da exportação de dados para o Amazon S3
- Especificar o caminho do arquivo do Amazon S3 para o qual exportar
- Configurar o acesso a um bucket do Amazon S3
- Exportar dados de consulta usando a função aws_s3.query_export_to_s3
- Referência de funções
- Solução de problemas de acesso ao Amazon S3

Instalar a extensão aws_s3

Antes de usar o Amazon Simple Storage Service com a sua instância de banco de dados do RDS para PostgreSQL, você precisa instalar a extensão `aws_s3`. Essa extensão fornece funções para exportar dados de uma instância de banco de dados do RDS para PostgreSQL para um bucket do Amazon S3. Também fornece as funções para importar dados do Amazon S3. Para ter mais informações, consulte Importar dados do Amazon S3 para uma instância de banco de dados do RDS para PostgreSQL. A extensão `aws_s3` depende de algumas das funções auxiliares da extensão `aws_commons`, que é instalada automaticamente quando necessária.

Como instalar a extensão **`aws_s3`**

1. Use psql (ou pgAdmin) para se conectar à instância de banco de dados do RDS para PostgreSQL como um usuário que tem privilégios `rds_superuser`. Se você manteve o nome padrão durante o processo de configuração, se conectará como `postgres`.

   ```
   psql --host=111122223333.aws-region.rds.amazonaws.com --port=5432 --
   username=postgres --password
   ```

2. Para instalar a extensão, execute o comando a seguir.

   ```
   postgres=> CREATE EXTENSION aws_s3 CASCADE;
   NOTICE: installing required extension "aws_commons"
   CREATE EXTENSION
   ```

3. Para verificar se a extensão está instalada, você pode usar o metacomando psql `\dx`.

   ```
   postgres=> \dx
   ```

```
       List of installed extensions
    Name     | Version |   Schema   |                Description
-------------+---------+------------+-------------------------------------------
 aws_commons | 1.2     | public     | Common data types across AWS services
 aws_s3      | 1.1     | public     | AWS S3 extension for importing data from S3
 plpgsql     | 1.0     | pg_catalog | PL/pgSQL procedural language
(3 rows)
```

As funções para importar dados do Amazon S3 e exportar dados para o Amazon S3 agora estão disponíveis para uso.

Confira se a sua versão do RDS para PostgreSQL oferece suporte a exportações para o Amazon S3

Você pode verificar se sua versão do RDS para PostgreSQL oferece suporte à exportação para o Amazon S3 usando o comando `describe-db-engine-versions`. O exemplo a seguir verifica a compatibilidade com a versão 10.14.

```
aws rds describe-db-engine-versions --region us-east-1
--engine postgres --engine-version 10.14 | grep s3Export
```

Se a saída incluir a string `"s3Export"`, o mecanismo é compatível com exportações do Amazon S3. Caso contrário, não há compatibilidade.

Visão geral da exportação de dados para o Amazon S3

Para exportar dados armazenados em um banco de dados RDS para PostgreSQL para um bucket do Amazon S3, use o procedimento a seguir.

Para exportar dados do RDS para PostgreSQL para o S3

1. Identifique um caminho de arquivo do Amazon S3 a ser usado para exportar dados. Para obter detalhes sobre esse processo, consulte Especificar o caminho do arquivo do Amazon S3 para o qual exportar.
2. Conceda permissão para acessar o bucket do Amazon S3.

   Para exportar dados para um arquivo do Amazon S3, forneça à instância de banco de dados do RDS para PostgreSQL permissão para acessar o bucket do Amazon S3 a ser usado para armazenamento pela exportação. Isso inclui as seguintes etapas:

1. Crie uma política do IAM que forneça acesso a um bucket do Amazon S3 para o qual você deseja exportar.
2. Crie uma função do IAM.
3. Anexe a política criada à função criada.
4. Adicione esse perfil do IAM à instância de banco de dados.

   Para obter detalhes sobre esse processo, consulte Configurar o acesso a um bucket do Amazon S3.

3. Identifique uma consulta de banco de dados para obter os dados. Exporte os dados da consulta chamando a função `aws_s3.query_export_to_s3`.

   Após concluir as tarefas de preparação anteriores, use a função aws_s3.query_export_to_s3 para exportar os resultados da consulta para o Amazon S3. Para obter detalhes sobre esse processo, consulte Exportar dados de consulta usando a função aws_s3.query_export_to_s3.

Especificar o caminho do arquivo do Amazon S3 para o qual exportar

Especifique as seguintes informações para identificar o local no Amazon S3 para o qual deseja exportar dados:

- Nome do bucket – um bucket é um contêiner para objetos ou arquivos do Amazon S3.

  Para ter mais informações sobre como armazenar dados com o Amazon S3, consulte Criação de um bucket e Trabalhar com objetos no Guia do usuário do Amazon Simple Storage Service.

- Caminho do arquivo – o caminho do arquivo identifica onde a exportação é armazenada no bucket do Amazon S3. O caminho do arquivo consiste no seguinte:
  - Um prefixo de caminho opcional que identifica um caminho de pasta virtual.
  - Um prefixo de arquivo que identifica um ou mais arquivos a serem armazenados. Exportações maiores são armazenadas em vários arquivos, cada um com um tamanho máximo de aproximadamente 6 GB. Os nomes de arquivos adicionais têm o mesmo prefixo de arquivo, mas com o acréscimo de `_part`*XX*. O *XX* representa 2, depois 3 e assim por diante.

  Por exemplo, um caminho de arquivo com uma pasta `exports` e um prefixo de arquivo `query-1-export` é `/exports/query-1-export`.

- Região da AWS (opcional): a região da AWS onde o bucket do Amazon S3 está localizado. Se você não especificar um valor de região da AWS, o Amazon RDS salvará os arquivos no Amazon S3, na mesma região da AWS que o cluster de banco de dados de exportação.

  > **Note**
  >
  > Atualmente, a região AWS deve ser a mesma que a região da instância de Bancos de Dados de exportação .

  Para obter uma listagem de nomes de regiões da AWS e os valores associados, consulte Regiões, zonas de disponibilidade e Local Zones.

Para manter as informações do arquivo do Amazon S3 sobre onde a exportação deve ser armazenada, você pode usar a função aws_commons.create_s3_uri para criar uma estrutura `aws_commons._s3_uri_1` composta da seguinte forma.

```
psql=> SELECT aws_commons.create_s3_uri(
   'amzn-s3-demo-bucket',
   'sample-filepath',
   'us-west-2'
) AS s3_uri_1 \gset
```

Posteriormente, você fornece esse valor de `s3_uri_1` como um parâmetro na chamada para a função aws_s3.query_export_to_s3. Para ver exemplos, consulte Exportar dados de consulta usando a função aws_s3.query_export_to_s3.

Configurar o acesso a um bucket do Amazon S3

Para exportar dados ao Amazon S3, forneça à instância de banco de dados PostgreSQL permissão para acessar o bucket do Amazon S3 para o qual os arquivos serão enviados.

Para fazer isso, use o procedimento a seguir.

Como conceder acesso a uma instância de banco de dados PostgreSQL ao Amazon S3 por meio de um perfil do IAM

1. Crie uma política do IAM.

Essa política concede ao bucket e ao objeto as permissões para que a instância de banco de dados PostgreSQL acesse o Amazon S3.

Como parte da criação dessa política, execute as seguintes etapas:

a. Inclua na política as seguintes ações necessárias para permitir a transferência de arquivos da instância de banco de dados PostgreSQL para um bucket do Amazon S3:

   - `s3:PutObject`
   - `s3:AbortMultipartUpload`

b. Inclua o nome do recurso da Amazon (ARN) que identifica o bucket do Amazon S3 e os objetos no bucket. O formato do ARN para acessar o Amazon S3 é: `arn:aws:s3:::`*`amzn-s3-demo-bucket`*`/*`

Para obter mais informações sobre como criar uma política do IAM para o Amazon RDS para PostgreSQL, consulte Criar e usar uma política do IAM para acesso do banco de dados do IAM. Consulte também Tutorial: criar e anexar sua primeira política gerenciada pelo cliente no Guia do usuário do IAM.

O comando da AWS CLI a seguir cria uma política do IAM denominada `rds-s3-export-policy` com essas opções. Ele concede acesso a um bucket denominado *`amzn-s3-demo-bucket`*.

> **Warning**
>
> Recomendamos configurar o banco de dados em uma VPC privada que tenha políticas de endpoint configuradas para acessar buckets específicos. Para obter mais informações, consulte Usar políticas de endpoint para o Amazon S3 no Guia do usuário da Amazon VPC.
> É altamente recomendável não criar uma política com acesso a todos os recursos. Esse acesso pode representar uma ameaça à segurança dos dados. Se você criar uma política que forneça o acesso `S3:PutObject` a todos os recursos usando `"Resource":"*"`, um usuário com privilégios de exportação poderá exportar dados para todos os buckets em sua conta. Além disso, o usuário poderá exportar dados para qualquer bucket gravável publicamente dentro de sua região daAWS.

Depois de criar a política, observe o nome do recurso da Amazon (ARN) da política. O ARN será necessário para uma etapa posterior, quando você anexar a política a um perfil do IAM.

```
aws iam create-policy  --policy-name rds-s3-export-policy  --policy-document '{
     "Version": "2012-10-17",
     "Statement": [
       {
         "Sid": "s3export",
         "Action": [
           "s3:PutObject*",
           "s3:ListBucket",
           "s3:GetObject*",
           "s3:DeleteObject*",
           "s3:GetBucketLocation",
           "s3:AbortMultipartUpload"
         ],
         "Effect": "Allow",
         "Resource": [
           "arn:aws:s3:::amzn-s3-demo-bucket/*"
         ]
       }
     ]
   }'
```

2. Crie uma função do IAM.

   Faça isso para que o Amazon RDS do possa assumir esse perfil do IAM em seu nome para acessar os buckets do Amazon S3. Para ter mais informações, consulte Criar um perfil para delegar permissões a um usuário do IAM no Guia do usuário do IAM.

   Convém usar as chaves de contexto de condição global `aws:SourceArn` e `aws:SourceAccount` em políticas baseadas em recursos para limitar as permissões do serviço a um recurso específico. Essa é a maneira mais eficiente de se proteger contra o problema "confused deputy".

   Se você utilizar ambas as chaves de contexto de condição global, e o valor `aws:SourceArn` contiver o ID da conta, o valor `aws:SourceAccount` e a conta no valor `aws:SourceArn` deverão utilizar o mesmo ID de conta quando utilizados na mesma instrução de política.

   - Use `aws:SourceArn` se quiser acesso entre serviços para um único recurso.

- Use `aws:SourceAccount` se você quiser permitir que qualquer recurso nessa conta seja associado ao uso entre serviços.

Na política, certifique-se de usar a chave de contexto de condição global `aws:SourceArn` com o ARN completo do recurso. O exemplo a seguir mostra como fazer isso utilizando o comando da AWS CLI para criar uma função chamada `rds-s3-export-role`.

Example

Para Linux, macOS ou Unix:

```
aws iam create-role  \
    --role-name rds-s3-export-role  \
    --assume-role-policy-document '{
     "Version": "2012-10-17",
     "Statement": [
       {
         "Effect": "Allow",
         "Principal": {
            "Service": "rds.amazonaws.com"
          },
         "Action": "sts:AssumeRole",
         "Condition": {
             "StringEquals": {
                "aws:SourceAccount": "111122223333",
                "aws:SourceArn": "arn:aws:rds:us-east-1:111122223333:db:dbname"
                }
             }
       }
     ]
   }'
```

Para Windows:

```
aws iam create-role  ^
    --role-name rds-s3-export-role  ^
    --assume-role-policy-document '{
     "Version": "2012-10-17",
     "Statement": [
       {
         "Effect": "Allow",
```

```
        "Principal": {
           "Service": "rds.amazonaws.com"
         },
        "Action": "sts:AssumeRole",
        "Condition": {
            "StringEquals": {
               "aws:SourceAccount": "111122223333",
               "aws:SourceArn": "arn:aws:rds:us-east-1:111122223333:db:dbname"
               }
            }
      }
    ]
  }'
```

3. Anexe a política do IAM que você criou à função do IAM que você criou.

   O comando da AWS CLI a seguir anexa a política criada anteriormente à função chamada `rds-s3-export-role.`. Substitua *`your-policy-arn`* pelo ARN da política que você anotou na etapa anterior.

   ```
   aws iam attach-role-policy  --policy-arn your-policy-arn  --role-name rds-s3-export-role
   ```

4. Adicione o perfil do IAM à instância de banco de dados. Faça isso usando o AWS Management Console ou a AWS CLI, conforme descrito a seguir.

Console

Para adicionar um perfil do IAM a uma instância de banco de dados do PostgreSQL usando o console

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. Escolha o nome da instância de de banco de dados do PostgreSQL para exibir os detalhes.
3. Na guia Connectivity & security (Conectividade e segurança), na seção Manage IAM roles (Gerenciar perfis do IAM), escolha o perfil a ser adicionado em Add IAM roles to this instance (Adicionar perfis do IAM a essa instância).
4. Em Feature (Recurso), escolha s3Export.
5. Escolha Add role (adicionar função).

AWS CLI

Para adicionar um perfil do IAM para uma instância de banco de dados do PostgreSQL usando a CLI

- Use o comando a seguir para adicionar a função à instância de banco de dados do PostgreSQL chamada `my-db-instance`. Substitua *`your-role-arn`* pelo ARN da função que você anotou em uma etapa anterior. Use `s3Export` para o valor da opção `--feature-name`.

  Example

  Para Linux, macOS ou Unix:

  ```
  aws rds add-role-to-db-instance \
     --db-instance-identifier my-db-instance \
     --feature-name s3Export \
     --role-arn your-role-arn   \
     --region your-region
  ```

  Para Windows:

  ```
  aws rds add-role-to-db-instance ^
     --db-instance-identifier my-db-instance ^
     --feature-name s3Export ^
     --role-arn your-role-arn ^
     --region your-region
  ```

Exportar dados de consulta usando a função aws_s3.query_export_to_s3

Exporte os dados do PostgreSQL para o Amazon S3 chamando a função aws_s3.query_export_to_s3.

Tópicos

- Pré-requisitos
- Chamar aws_s3.query_export_to_s3
- Exportar para um arquivo CSV que usa um delimitador personalizado
- Exportar para um arquivo binário com codificação

Pré-requisitos

Antes de usar a função `aws_s3.query_export_to_s3`, verifique se você concluiu os seguintes pré-requisitos:

- Instalar as extensões do PostgreSQL necessárias, conforme descrito em Visão geral da exportação de dados para o Amazon S3.
- Determinar para onde exportar os dados para o Amazon S3, conforme descrito em Especificar o caminho do arquivo do Amazon S3 para o qual exportar.
- Verifique se a instância de banco de dados tem acesso ao Amazon S3 conforme descrito em Configurar o acesso a um bucket do Amazon S3.

Os exemplos a seguir usam uma tabela de banco de dados chamada `sample_table`. Esses exemplos exportam os dados para um bucket chamado *`amzn-s3-demo-bucket`*. A tabela e os dados de exemplo são criados com as seguintes instruções SQL no psql.

```
psql=> CREATE TABLE sample_table (bid bigint PRIMARY KEY, name varchar(80));
psql=> INSERT INTO sample_table (bid,name) VALUES (1, 'Monday'), (2,'Tuesday'), (3,
 'Wednesday');
```

Chamar aws_s3.query_export_to_s3

Veja a seguir as formas básicas de chamar a função aws_s3.query_export_to_s3.

Esses exemplos usam a variável `s3_uri_1` para identificar uma estrutura que contém as informações que identificam o arquivo do Amazon S3. Use a função aws_commons.create_s3_uri para criar a estrutura.

```
psql=> SELECT aws_commons.create_s3_uri(
   'amzn-s3-demo-bucket',
   'sample-filepath',
   'us-west-2'
) AS s3_uri_1 \gset
```

Embora os parâmetros variem para as duas chamadas de função `aws_s3.query_export_to_s3` a seguir, os resultados são os mesmos para esses exemplos. Todas as linhas da tabela `sample_table` são exportadas para um bucket chamado *`amzn-s3-demo-bucket`*.

```
psql=> SELECT * FROM aws_s3.query_export_to_s3('SELECT * FROM
 sample_table', :'s3_uri_1');

psql=> SELECT * FROM aws_s3.query_export_to_s3('SELECT * FROM
 sample_table', :'s3_uri_1', options :='format text');
```

Os parâmetros são descritos da seguinte forma:

- `'SELECT * FROM sample_table'` – o primeiro parâmetro é uma string de texto necessária que contém uma consulta SQL. O mecanismo PostgreSQL executa essa consulta. Os resultados da consulta são copiados no bucket do S3 identificado em outros parâmetros.
- `:'s3_uri_1'` – esse parâmetro é uma estrutura que identifica o arquivo do Amazon S3. Esse exemplo usa uma variável para identificar a estrutura criada anteriormente. Em vez disso, você pode criar a estrutura incluindo a chamada de função `aws_commons.create_s3_uri` em linha dentro da chamada de função `aws_s3.query_export_to_s3` da seguinte forma.

  ```
  SELECT * from aws_s3.query_export_to_s3('select * from sample_table',
     aws_commons.create_s3_uri('amzn-s3-demo-bucket', 'sample-filepath', 'us-west-2')
  );
  ```
- `options :='format text'` – o parâmetro `options` é uma string de texto opcional que contém argumentos `COPY` do PostgreSQL. O processo de cópia usa os argumentos e o formato do comando COPY PostgreSQL.

Se o arquivo especificado não existir no bucket do Amazon S3, ele será criado. Se o arquivo já existir, ele será substituído. A sintaxe para acessar os dados exportados no Amazon S3 é a seguinte.

```
s3-region://bucket-name[/path-prefix]/file-prefix
```

Exportações maiores são armazenadas em vários arquivos, cada um com um tamanho máximo de aproximadamente 6 GB. Os nomes de arquivos adicionais têm o mesmo prefixo de arquivo, mas com o acréscimo de `_part`*XX*. O *XX* representa 2, depois 3 e assim por diante. Por exemplo, suponha que você especifique o caminho onde armazena arquivos de dados como o seguinte.

```
s3-us-west-2://amzn-s3-demo-bucket/my-prefix
```

Se a exportação precisar criar três arquivos de dados, o bucket do Amazon S3 conterá os seguintes arquivos de dados.

```
s3-us-west-2://amzn-s3-demo-bucket/my-prefix
s3-us-west-2://amzn-s3-demo-bucket/my-prefix_part2
s3-us-west-2://amzn-s3-demo-bucket/my-prefix_part3
```

Para obter a referência completa para esta função e as formas adicionais de chamá-la, consulte aws_s3.query_export_to_s3. Para obter mais informações sobre como acessar arquivos no Amazon S3, consulte Visualização de um objeto no Guia do usuário do Amazon Simple Storage Service.

Exportar para um arquivo CSV que usa um delimitador personalizado

O exemplo a seguir mostra como chamar a função aws_s3.query_export_to_s3 para exportar dados para um arquivo que usa um delimitador personalizado. O exemplo usa argumentos do comando COPY do PostgreSQL para especificar o formato de valor separado por vírgula (CSV) e um delimitador de dois pontos (:).

```
SELECT * from aws_s3.query_export_to_s3('select * from basic_test', :'s3_uri_1',
 options :='format csv, delimiter $$:$$');
```

Exportar para um arquivo binário com codificação

O exemplo a seguir mostra como chamar a função aws_s3.query_export_to_s3 para exportar dados para um arquivo binário que tenha a codificação Windows-1253.

```
SELECT * from aws_s3.query_export_to_s3('select * from basic_test', :'s3_uri_1',
 options :='format binary, encoding WIN1253');
```

Referência de funções

Funções

- aws_s3.query_export_to_s3
- aws_commons.create_s3_uri

aws_s3.query_export_to_s3

Exporta um resultado de consulta do PostgreSQL para um bucket do Amazon S3. A extensão `aws_s3` fornece a função `aws_s3.query_export_to_s3`.

Os dois parâmetros necessários são `query` e `s3_info`. Eles definem a consulta a ser exportada e identificam o bucket do Amazon S3 para o qual exportar. Um parâmetro opcional chamado `options` fornece a definição de vários parâmetros de exportação. Para obter exemplos de como usar a função `aws_s3.query_export_to_s3`, consulte Exportar dados de consulta usando a função aws_s3.query_export_to_s3.

Sintaxe

```
aws_s3.query_export_to_s3(
    query text,
    s3_info aws_commons._s3_uri_1,
    options text,
    kms_key text
)
```

Parâmetros de entrada

query

Uma string de texto necessária que contém uma consulta SQL executada pelo mecanismo do PostgreSQL. Os resultados dessa consulta são copiados em um bucket do S3 identificado no parâmetro `s3_info`.

s3_info

Um tipo composto `aws_commons._s3_uri_1` contendo as seguintes informações sobre o objeto do S3:

- `bucket` – o nome do bucket do Amazon S3 que deve conter o arquivo.
- `file_path` – o nome e o caminho do arquivo do Amazon S3.
- `region`: a região da AWS na qual o bucket se encontra. Para obter uma listagem de nomes de regiões da AWS e os valores associados, consulte Regiões, zonas de disponibilidade e Local Zones.

  Atualmente, esse valor deve ser a mesma região da AWS que a instância de Bancos de Dados de exportação . O padrão é a região da AWS para exportação da instância de Bancos de Dados .

Para criar uma estrutura `aws_commons._s3_uri_1` composta, consulte a função aws_commons.create_s3_uri.

options

Uma string de texto opcional que contém argumentos para o comando `COPY` do PostgreSQL. Esses argumentos especificam como os dados devem ser copiados quando exportados. Para obter mais detalhes, consulte Documentação de COPY do PostgreSQL.

kms_key text

Uma string de texto opcional contendo a chave do KMS gerenciada pelo cliente do bucket do S3 para o qual exportar os dados.

Parâmetros de entrada alternativos

Para ajudar nos testes, você pode usar um conjunto expandido de parâmetros em vez do parâmetro `s3_info`. Veja a seguir as variações adicionais de sintaxe da função `aws_s3.query_export_to_s3`.

Em vez de usar o parâmetro `s3_info` para identificar um arquivo do Amazon S3, use a combinação dos parâmetros `bucket`, `file_path` e `region`.

```
aws_s3.query_export_to_s3(
    query text,
    bucket text,
    file_path text,
    region text,
    options text,
    kms_key text
)
```

query

Uma string de texto necessária que contém uma consulta SQL executada pelo mecanismo do PostgreSQL. Os resultados dessa consulta são copiados em um bucket do S3 identificado no parâmetro `s3_info`.

bucket

Uma string de texto necessária que contém o nome do bucket do Amazon S3 que contém o arquivo.

file_path

Uma string de texto necessária contendo o nome do arquivo do Amazon S3, incluindo o caminho do arquivo.

região

Uma string de texto opcional que contém a região da AWS na qual o bucket se contra. Para obter uma listagem de nomes de regiões da AWS e os valores associados, consulte Regiões, zonas de disponibilidade e Local Zones.

Atualmente, esse valor deve ser a mesma região da AWS que a instância de Bancos de Dados de exportação . O padrão é a região da AWS para exportação da instância de Bancos de Dados .

options

Uma string de texto opcional que contém argumentos para o comando `COPY` do PostgreSQL. Esses argumentos especificam como os dados devem ser copiados quando exportados. Para obter mais detalhes, consulte Documentação de COPY do PostgreSQL.

kms_key text

Uma string de texto opcional contendo a chave do KMS gerenciada pelo cliente do bucket do S3 para o qual exportar os dados.

Parâmetros de saída

```
aws_s3.query_export_to_s3(
    OUT rows_uploaded bigint,
    OUT files_uploaded bigint,
    OUT bytes_uploaded bigint
)
```

rows_uploaded

O número de linhas da tabela que foram carregadas com êxito no Amazon S3 para a determinada consulta.

files_uploaded

O número de arquivos carregados no Amazon S3. Os arquivos são criados em tamanhos de aproximadamente 6 GB. Cada arquivo adicional criado tem `_part`*XX* acrescentado ao nome. O *XX* representa 2, depois 3 e assim por diante, conforme necessário.

bytes_uploaded

O número total de bytes carregados no Amazon S3.

Exemplos

```
psql=> SELECT * from aws_s3.query_export_to_s3('select * from sample_table', 'amzn-s3-demo-bucket', 'sample-filepath');
psql=> SELECT * from aws_s3.query_export_to_s3('select * from sample_table', 'amzn-s3-demo-bucket', 'sample-filepath','us-west-2');
psql=> SELECT * from aws_s3.query_export_to_s3('select * from sample_table', 'amzn-s3-demo-bucket', 'sample-filepath','us-west-2','format text');
```

aws_commons.create_s3_uri

Cria uma estrutura `aws_commons._s3_uri_1` para comportar informações do arquivo do Amazon S3. Use os resultados da função `aws_commons.create_s3_uri` no parâmetro `s3_info` da função aws_s3.query_export_to_s3. Para obter um exemplo de uso da função `aws_commons.create_s3_uri`, consulte Especificar o caminho do arquivo do Amazon S3 para o qual exportar.

Sintaxe

```
aws_commons.create_s3_uri(
   bucket text,
   file_path text,
   region text
)
```

Parâmetros de entrada

bucket

Uma string de texto necessária contendo o nome do bucket do Amazon S3 para o arquivo.

file_path

Uma string de texto necessária contendo o nome do arquivo do Amazon S3, incluindo o caminho do arquivo.

região

Uma string de texto necessária que contém a região da AWS na qual o arquivo se encontra. Para obter uma listagem de nomes de regiões da AWS e os valores associados, consulte Regiões, zonas de disponibilidade e Local Zones.

Solução de problemas de acesso ao Amazon S3

Se você encontrar problemas de conexão ao tentar exportar dados para o Amazon S3, primeiro confirme se as regras de acesso de saída para o grupo de segurança da VPC associado à sua instância de banco de dados permitem conectividade de rede. Especificamente, o grupo de segurança deve ter uma regra que permita que a instância de banco de dados envie tráfego TCP para a porta 443 e para todos os endereços IPv4 (0.0.0.0/0). Para obter mais informações, consulte Fornecer acesso à instância de banco de dados na VPC criando um grupo de segurança.

Veja também as seguintes recomendações:

- Solução de problemas de identidade e acesso do Amazon RDS
- Solução de problemas do Amazon S3 no Guia do usuário do Amazon Simple Storage Service
- Troubleshooting Amazon S3 and IAM (Solucionar problemas no Amazon S3 e no IAM) no Guia do usuário do IAM

Invocar uma função do AWS Lambda de uma instância de banco de dados do RDS for PostgreSQL

O AWS Lambda é um serviço computacional orientado a eventos que permite executar código sem o provisionamento ou gerenciamento de servidores. Está disponível para uso com muitos serviços da AWS, incluindo o RDS for PostgreSQL. Por exemplo, você pode usar funções Lambda para processar notificações de eventos de um banco de dados ou para carregar dados de arquivos sempre que um novo arquivo é carregado para o Simple Storage Service (Amazon S3). Para saber mais sobre o Lambda, consulte O que é o AWS Lambda? no Guia do desenvolvedor do AWS Lambda.

> **Note**
>
> A chamada de uma função AWS Lambda é aceita nas seguintes versões do RDS for PostgreSQL:
>
> - Todas as versões 17 do PostgreSQL
> - Todas as versões 16 do PostgreSQL
> - Todas as versões do PostgreSQL 15
> - PostgreSQL 14.1 e versões secundárias posteriores
> - PostgreSQL 13.2 e versões secundárias posteriores
> - PostgreSQL 12.6 e versões secundárias posteriores

Configurar o RDS for PostgreSQL para trabalhar com funções Lambda é um processo de várias etapas envolvendo o AWS Lambda, o IAM, sua VPC e sua instância de banco de dados do RDS for PostgreSQL. A seguir, você pode encontrar resumos das etapas necessárias.

Para obter mais informações sobre como criar uma função Lambda, consulte Conceitos básicos do Lambda e Tópicos essenciais do AWS Lambda no Guia do desenvolvedor do AWS Lambda.

Tópicos

- Etapa 1: configurar a instância de banco de dados do RDS for PostgreSQL para conexões de saída para o AWS Lambda
- Etapa 2: configurar o IAM para a instância de banco de dados do RDS for PostgreSQL e AWS Lambda

- Etapa 3: instalar a extensão aws_lambda para uma instância de banco de dados do RDS for PostgreSQL
- Etapa 4: usar as funções Lambda auxiliares com a instância de banco de dados do RDS for PostgreSQL (opcional)
- Etapa 5: invocar uma função Lambda da sua instância de banco de dados do RDS for PostgreSQL
- Etapa 6: Conceder a outros usuários permissão para invocar funções do Lambda
- Exemplos: Invocar uma função do Lambda da instância de banco de dados do RDS for PostgreSQL
- Mensagens de erro da função Lambda
- Função do AWS Lambda e referência de parâmetros

Etapa 1: configurar a instância de banco de dados do RDS for PostgreSQL para conexões de saída para o AWS Lambda

As funções do Lambda sempre são executadas dentro de uma Amazon VPC de propriedade do serviço AWS Lambda. O Lambda aplica regras de segurança e acesso à rede a essa VPC e mantém e monitora a VPC automaticamente. Sua instância de banco de dados do RDS para PostgreSQL envia tráfego de rede para a VPC do serviço Lambda. Como você configura isso depende se sua instância de banco de dados primária do é pública ou privada.

- Instância de banco de dados pública do RDS para PostgreSQL: uma instância de banco de dados é pública se está localizada em uma sub-rede pública na VPC e se a propriedade "PubliclyAccessible" da instância é `true`. Para encontrar o valor dessa propriedade, você pode usar o comando da AWS CLI describe-db-instances. Ou, você pode usar o AWS Management Console para abrir a guia Connectivity & security (Conectividade e segurança) e verificar se Publicly accessible (Publicamente acessível) está definido como Yes (Sim). Para verificar se a instância está na sub-rede pública da VPC, use o AWS Management Console ou a AWS CLI.

  Para configurar o acesso ao Lambda, use o AWS Management Console ou a AWS CLI para criar uma regra de saída no grupo de segurança da VPC. A regra de saída especifica que o TCP pode usar a porta 443 para enviar pacotes para qualquer endereço IPv4 (0.0.0.0/0).
- Instância de banco de dados privada do RDS para PostgreSQL: nesse caso, a propriedade "PubliclyAccessible" da instância é `false` ou está em uma sub-rede privada. Para permitir que a instância funcione com o Lambda, use um gateway de conversão de endereços de rede (NAT). Para obter mais informações, consulte Gateways de NAT. Ou configure sua VPC com um endpoint

da VPC para o Lambda. Para obter mais informações, consulte Endpoints da VPC no Guia do usuário da Amazon VPC. O endpoint responde às chamadas feitas pela instância de banco de dados do RDS para PostgreSQL para suas funções do Lambda. O endpoint da VPC usa sua própria resolução DNS privada. O RDS for PostgreSQL não pode usar o endpoint da VPC do Lambda até que você altere o valor do `rds.custom_dns_resolution` de seu valor padrão 0 (não habilitado) para 1. Para fazer isso:

- Crie um grupo de parâmetros de banco de dados personalizado.
- Altere o valor do parâmetro `rds.custom_dns_resolution` de seu padrão `0` para `1`.
- Modifique sua instância de banco de dados para aplicar seu grupo de parâmetros de banco de dados personalizado.
- Reinicialize a instância de banco de dados para que a alteração do parâmetro tenha efeito.

Agora, sua VPC pode interagir com o VPC do AWS Lambda no nível da rede. Depois, configure as permissões usando o IAM.

Etapa 2: configurar o IAM para a instância de banco de dados do RDS for PostgreSQL e AWS Lambda

Invocar funções Lambda da sua instância de banco de dados do RDS for PostgreSQL requer certos privilégios. Para configurar os privilégios necessários, recomendamos que você crie uma política do IAM que permita chamar funções Lambda, atribuir essa política a uma função e, em seguida, aplicar a função ao à instância de banco de dados. Essa abordagem dá à instância de banco de dados privilégios para invocar a função Lambda especificada em seu nome. As etapas a seguir mostram como fazer isso usando a AWS CLI.

Para configurar permissões do IAM para usar sua instância do Amazon RDS com o Lambda

1. Use o comando da AWS CLI create-policy para criar uma política do IAM que permita que a sua instância de banco de dados do RDS for PostgreSQL invoque a função Lambda especificada. (O ID da instrução (Sid) é uma descrição opcional para sua instrução de política e não tem efeito sobre o uso.) Esta política fornece à sua instância de banco de dados as permissões mínimas necessárias para invocar a função Lambda especificada.

```
aws iam create-policy  --policy-name rds-lambda-policy --policy-document '{
    "Version": "2012-10-17",
    "Statement": [
        {
```

```
        "Sid": "AllowAccessToExampleFunction",
        "Effect": "Allow",
        "Action": "lambda:InvokeFunction",
        "Resource": "arn:aws:lambda:aws-region:444455556666:function:my-function"
        }
    ]
}'
```

Você também pode usar a política `AWSLambdaRole` predefinida que permite invocar qualquer uma das suas funções Lambda. Para obter mais informações, consulte Políticas do IAM baseadas em identidade para o Lambda.

2. Use o comando da AWS CLI create-role para criar uma função do IAM que a política possa assumir em tempo de execução.

```
aws iam create-role  --role-name rds-lambda-role --assume-role-policy-document '{
    "Version": "2012-10-17",
    "Statement": [
        {
        "Effect": "Allow",
        "Principal": {
            "Service": "rds.amazonaws.com"
        },
        "Action": "sts:AssumeRole"
        }
    ]
}'
```

3. Anexe a política à função usando o comando da AWS CLI attach-role-policy.

```
aws iam attach-role-policy \
    --policy-arn arn:aws:iam::444455556666:policy/rds-lambda-policy \
    --role-name rds-lambda-role --region aws-region
```

4. Aplique a função à sua instância de banco de dados do RDS for PostgreSQL usando o comando da AWS CLI add-role-to-db-instance Esta última etapa permite que seus usuários de banco de dados de instâncias de banco de dados invoquem funções Lambda.

```
aws rds add-role-to-db-instance \
       --db-instance-identifier my-instance-name \
       --feature-name Lambda \
       --role-arn  arn:aws:iam::444455556666:role/rds-lambda-role   \
```

```
        --region aws-region
```

Com a VPC e as configurações do IAM concluídas, agora você pode instalar a extensão `aws_lambda`. (Observe que você pode instalar a extensão a qualquer momento, mas até configurar o suporte à VPC e os privilégios do IAM corretos, a extensão `aws_lambda` não adiciona nada às capacidades da instância de banco de dados do RDS for PostgreSQL.)

Etapa 3: instalar a extensão **`aws_lambda`** para uma instância de banco de dados do RDS for PostgreSQL

Para usar o AWS Lambda com a instância de banco de dados do RDS para PostgreSQL, adicione a extensão `aws_lambda` do PostgreSQL à instância de banco de dados do RDS para PostgreSQL. Esta extensão fornece a instância de banco de dados do RDS for PostgreSQL com a capacidade de chamar funções Lambda do PostgreSQL.

Para instalar a extensão **`aws_lambda`** em sua instância de banco de dados do RDS for PostgreSQL

Use a linha de comando `psql` do PostgreSQL ou a ferramenta pgAdmin para se conectar à sua instância de banco de dados do RDS for PostgreSQL.

1. Conecte-se à sua instância de banco de dados do RDS for PostgreSQL como usuário com privilégios `rds_superuser`. O usuário padrão `postgres` é mostrado no exemplo.

   ```
   psql -h instance.444455556666.aws-region.rds.amazonaws.com -U postgres -p 5432
   ```

2. Instale a extensão `aws_lambda`. A extensão `aws_commons` também é necessária. Ela fornece funções auxiliares para `aws_lambda` e muitas outras extensões do Aurora para PostgreSQL. Se ainda não estiver na sua instância de banco de dados do RDS for PostgreSQL, ela é instalada com `aws_lambda` como mostrado a seguir.

   ```
   CREATE EXTENSION IF NOT EXISTS aws_lambda CASCADE;
   NOTICE:  installing required extension "aws_commons"
   CREATE EXTENSION
   ```

A extensão `aws_lambda` está instalada em sua instância de banco de dados primária do . Agora você pode criar estruturas de conveniência para chamar suas funções Lambda.

Etapa 4: usar as funções Lambda auxiliares com a instância de banco de dados do RDS for PostgreSQL (opcional)

Você pode usar as funções auxiliares na extensão `aws_commons` para preparar entidades que você pode invocar mais facilmente a partir do PostgreSQL. Para isso, você precisa ter as seguintes informações sobre suas funções Lambda:

- Nome da função – O nome, nome do recurso da Amazon (ARN), versão ou apelido da função Lambda. A política do IAM criada em Etapa 2: configurar o IAM para o cluster e o Lambda requer o ARN, portanto, recomendamos que você use o ARN da sua função.
- Região da AWS – (Opcional) A região da AWS onde a função Lambda está localizada se não estiver na mesma região que sua instância de banco de dados do RDS for PostgreSQL.

Para manter as informações do nome da função Lambda, você pode usar a função aws_commons.create_lambda_function_arn. Esta função auxiliar cria uma estrutura composta `aws_commons._lambda_function_arn_1` com os detalhes necessários para a função de invocação. A seguir, você pode encontrar três abordagens alternativas para configurar essa estrutura composta.

```
SELECT aws_commons.create_lambda_function_arn(
   'my-function',
   'aws-region'
) AS aws_lambda_arn_1 \gset
```

```
SELECT aws_commons.create_lambda_function_arn(
   '111122223333:function:my-function',
   'aws-region'
) AS lambda_partial_arn_1 \gset
```

```
SELECT aws_commons.create_lambda_function_arn(
   'arn:aws:lambda:aws-region:111122223333:function:my-function'
) AS lambda_arn_1 \gset
```

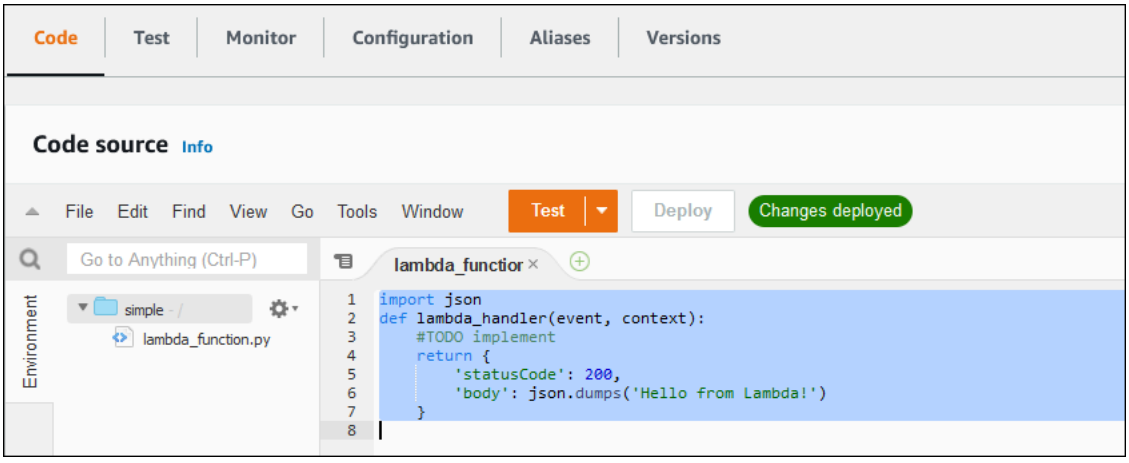
Qualquer um desses valores pode ser usado em chamadas para a função aws_lambda.invoke. Para obter exemplos, consulte Etapa 5: invocar uma função Lambda da sua instância de banco de dados do RDS for PostgreSQL.

Etapa 5: invocar uma função Lambda da sua instância de banco de dados do RDS for PostgreSQL

A função `aws_lambda.invoke` se comporta de forma síncrona ou assíncrona, dependendo do `invocation_type`. As duas alternativas para esse parâmetro são `RequestResponse` (o padrão) e `Event`, da seguinte forma:

- **`RequestResponse`** – Este tipo de invocação é síncrono. É o comportamento padrão quando a chamada é feita sem especificar um tipo de invocação. A carga útil da resposta inclui os resultados da função `aws_lambda.invoke`. Use esse tipo de invocação de quando seu fluxo de trabalho exigir o recebimento de resultados da função Lambda antes de continuar.
- **`Event`** – Este tipo de invocação é assíncrono. A resposta não inclui uma carga útil contendo resultados. Use esse tipo de invocação quando o fluxo de trabalho não precisar de um resultado da função Lambda para continuar o processamento.

Como um teste simples de sua configuração, você pode se conectar à sua instância de banco de dados usando `psql` e chamar uma função de exemplo a partir da linha de comando. Suponha que você tenha uma das funções básicas configuradas em seu serviço Lambda, como a função Python simples mostrada na captura de tela a seguir.



Para invocar uma função de exemplo

1. Conecte-se à sua instância de banco de dados usando `psql` ou pgAdmin.

   ```
   psql -h instance.444455556666.aws-region.rds.amazonaws.com -U postgres -p 5432
   ```

2. Invoque a função usando seu ARN.

```
SELECT * from
 aws_lambda.invoke(aws_commons.create_lambda_function_arn('arn:aws:lambda:aws-
region:444455556666:function:simple', 'us-west-1'), '{"body": "Hello from
 Postgres!"}'::json );
```

A resposta se parece com esta a seguir.

```
status_code |                        payload                         |
 executed_version | log_result
-------------+--------------------------------------------------------
+------------------+------------
         200 | {"statusCode": 200, "body": "\"Hello from Lambda!\""} | $LATEST
     |
(1 row)
```

Se sua tentativa de invocação não for bem-sucedida, consulte Mensagens de erro da função Lambda .

Etapa 6: Conceder a outros usuários permissão para invocar funções do Lambda

Neste ponto dos procedimentos, apenas você como `rds_superuser` pode invocar suas funções do Lambda. Para permitir que outros usuários invoquem quaisquer funções criadas por você, é necessário conceder permissão a eles.

Como conceder a outros permissão para invocar funções do Lambda

1. Conecte-se à sua instância de banco de dados usando `psql` ou pgAdmin.

```
psql -h instance.444455556666.aws-region.rds.amazonaws.com -U postgres -p 5432
```

2. Execute os seguintes comandos SQL:

```
postgres=>  GRANT USAGE ON SCHEMA aws_lambda TO db_username;
GRANT EXECUTE ON ALL FUNCTIONS IN SCHEMA aws_lambda TO db_username;
```

Exemplos: Invocar uma função do Lambda da instância de banco de dados do RDS for PostgreSQL

A seguir, você pode encontrar alguns exemplos de como chamar a função aws_lambda.invoke. A maioria dos exemplos usa a estrutura composta `aws_lambda_arn_1` que você cria em Etapa 4: usar as funções Lambda auxiliares com a instância de banco de dados do RDS for PostgreSQL (opcional) para simplificar a passagem dos detalhes da função. Para obter um exemplo de invocação assíncrona, consulte Exemplo: invocação assíncrona (evento) de funções Lambda. Todos os outros exemplos listados usam invocação síncrona.

Para saber mais sobre os tipos de invocação do Lambda, consulte Invocação de funções Lambda no Guia do desenvolvedor do AWS Lambda. Para obter mais informações sobre o `aws_lambda_arn_1`, consulte aws_commons.create_lambda_function_arn.

Lista de exemplos

- Exemplo: invocação síncrona (RequestResponse) de funções Lambda
- Exemplo: invocação assíncrona (evento) de funções Lambda
- Exemplo: capturar o log de execução do Lambda em uma resposta de função
- Exemplo: incluir o contexto do cliente em uma função Lambda
- Exemplo: invocar uma versão específica de uma função Lambda

Exemplo: invocação síncrona (RequestResponse) de funções Lambda

A seguir estão dois exemplos de uma invocação síncrona de função Lambda. Os resultados dessas chamadas de funções `aws_lambda.invoke` são os mesmos.

```
SELECT * FROM aws_lambda.invoke('aws_lambda_arn_1', '{"body": "Hello from
 Postgres!"}'::json);
```

```
SELECT * FROM aws_lambda.invoke('aws_lambda_arn_1', '{"body": "Hello from
 Postgres!"}'::json, 'RequestResponse');
```

Os parâmetros são descritos da seguinte forma:

- `:'aws_lambda_arn_1'` – Este parâmetro identifica a estrutura composta criada em Etapa 4: usar as funções Lambda auxiliares com a instância de banco de dados do RDS for PostgreSQL

(opcional), com a função auxiliar `aws_commons.create_lambda_function_arn`. Você também pode criar essa estrutura em linha dentro da sua chamada `aws_lambda.invoke` da seguinte forma:

```
SELECT * FROM aws_lambda.invoke(aws_commons.create_lambda_function_arn('my-function',
 'aws-region'),
'{"body": "Hello from Postgres!"}'::json
);
```

- `'{"body": "Hello from PostgreSQL!"}'::json` – A carga útil JSON para passar para a função Lambda.
- `'RequestResponse'` – O Lambda tipo de invocação.

Exemplo: invocação assíncrona (evento) de funções Lambda

Segue-se um exemplo de uma invocação de função Lambda assíncrona. O tipo de invocação `Event` agenda a invocação de função Lambda com a carga de entrada específica e retorna imediatamente. Use o tipo de invocação de `Event` em determinados fluxos de trabalho que não dependem dos resultados da função Lambda.

```
SELECT * FROM aws_lambda.invoke('aws_lambda_arn_1', '{"body": "Hello from
 Postgres!"}'::json, 'Event');
```

Exemplo: capturar o log de execução do Lambda em uma resposta de função

Você pode incluir os últimos 4 KB do log de execução na resposta da função usando o parâmetro `log_type` em sua chamada de função `aws_lambda.invoke`. Por padrão, esse parâmetro é definido como `None`, mas você pode especificar `Tail` para capturar os resultados do log de execução do Lambda na resposta, conforme mostrado a seguir.

```
SELECT *, select convert_from(decode(log_result, 'base64'), 'utf-8') as log FROM
 aws_lambda.invoke(:'aws_lambda_arn_1', '{"body": "Hello from Postgres!"}'::json,
 'RequestResponse', 'Tail');
```

Defina o parâmetro aws_lambda.invoke da função `log_type` para `Tail` incluir o log de execução na resposta. O valor padrão para o parâmetro `log_type` é `None`.

O `log_result` que é retornado é uma string codificada de `base64`. Você pode decodificar os conteúdos usando uma combinação das funções PostgreSQL `decode` e `convert_from`.

Para obter mais informações sobre o `log_type`, consulte aws_lambda.invoke.

Exemplo: incluir o contexto do cliente em uma função Lambda

A função `aws_lambda.invoke` tem um parâmetro `context` que você pode usar para passar informações separadas da carga útil, como mostrado a seguir.

```
SELECT *, convert_from(decode(log_result, 'base64'), 'utf-8') as log FROM
 aws_lambda.invoke(:'aws_lambda_arn_1', '{"body": "Hello from Postgres!"}'::json,
 'RequestResponse', 'Tail');
```

Para incluir o contexto do cliente, use um objeto JSON para o parâmetro aws_lambda.invoke da função`context`.

Para obter mais informações sobre os parâmetros do `context`, consulte a referência aws_lambda.invoke.

Exemplo: invocar uma versão específica de uma função Lambda

Você pode especificar uma determinada versão de uma função Lambda incluindo o parâmetro `qualifier` com a chamada `aws_lambda.invoke`. A seguir, você pode encontrar um exemplo que faz isso usando '*`custom_version`*' como um alias para a versão.

```
SELECT * FROM aws_lambda.invoke('aws_lambda_arn_1', '{"body": "Hello from
 Postgres!"}'::json, 'RequestResponse', 'None', NULL, 'custom_version');
```

Você também pode fornecer um qualificador de função Lambda com as informações de nome da função da forma a seguir.

```
SELECT * FROM aws_lambda.invoke(aws_commons.create_lambda_function_arn('my-
function:custom_version', 'us-west-2'),
'{"body": "Hello from Postgres!"}'::json);
```

Para obter mais informações sobre `qualifier` e outros parâmetros, consulte a referência aws_lambda.invoke.

Mensagens de erro da função Lambda

Na lista a seguir, você pode encontrar informações sobre mensagens de erro, com possíveis causas e soluções.

- Problemas de configuração da VPC

  Problemas de configuração da VPC podem gerar as seguintes mensagens de erro ao tentar se conectar:

  ```
  ERROR:  invoke API failed
  DETAIL: AWS Lambda client returned 'Unable to connect to endpoint'.
  CONTEXT:  SQL function "invoke" statement 1
  ```

  Uma causa comum para esse erro é o grupo de segurança da VPC configurado incorretamente. É necessário ter uma regra de saída para TCP aberta na porta 443 para que o grupo de segurança de sua VPC possa se conectar à VPC do Lambda.

  Se sua instância de banco de dados for privada, verifique a configuração DNS privada para sua VPC. Verifique se você definiu o parâmetro `rds.custom_dns_resolution` como 1 e configurou o AWSPrivateLink conforme descrito em Etapa 1: configurar a instância de banco de dados do RDS for PostgreSQL para conexões de saída para o AWS Lambda. Para obter mais informações, consulte Endpoints da VPC (AWS PrivateLink).

- Falta de permissões necessárias para invocar funções do Lambda

  Se você vir uma das seguintes mensagens de erro, isso significa que o usuário (função) que está invocando a função não tem as permissões adequadas.

  ```
  ERROR:  permission denied for schema aws_lambda
  ```

  ```
  ERROR:  permission denied for function invoke
  ```

  Um usuário (função) deve receber concessões específicas para invocar funções do Lambda. Para obter mais informações, consulte Etapa 6: Conceder a outros usuários permissão para invocar funções do Lambda.

- Tratamento inadequado de erros em suas funções do Lambda

  Se uma função Lambda lança uma exceção durante o processamento da solicitação, `aws_lambda.invoke` terá um erro do PostgreSQL, como o seguinte.

  ```
  SELECT * FROM aws_lambda.invoke('aws_lambda_arn_1', '{"body": "Hello from
   Postgres!"}'::json);
  ERROR:  lambda invocation failed
  ```

```
DETAIL:  "arn:aws:lambda:us-west-2:555555555555:function:my-function" returned error
 "Unhandled", details: "<Error details string>".
```

Certifique-se de lidar com erros nas funções Lambda ou na aplicação PostgreSQL.

Função do AWS Lambda e referência de parâmetros

Veja a seguir a referência para as funções e os parâmetros a serem usados para invocar o Lambda com o RDS para PostgreSQL.

Funções e parâmetros

- aws_lambda.invoke
- aws_commons.create_lambda_function_arn
- Parâmetros aws_lambda

aws_lambda.invoke

Executa uma função do Lambda para uma instância de banco de dados do RDS for PostgreSQL.

Para obter mais detalhes sobre invocar funções Lambda, consulte também Invocar no Guia do desenvolvedor do AWS Lambda.

Sintaxe

JSON

```
aws_lambda.invoke(
IN function_name TEXT,
IN payload JSON,
IN region TEXT DEFAULT NULL,
IN invocation_type TEXT DEFAULT 'RequestResponse',
IN log_type TEXT DEFAULT 'None',
IN context JSON DEFAULT NULL,
IN qualifier VARCHAR(128) DEFAULT NULL,
OUT status_code INT,
OUT payload JSON,
OUT executed_version TEXT,
OUT log_result TEXT)
```

```
aws_lambda.invoke(
IN function_name aws_commons._lambda_function_arn_1,
IN payload JSON,
IN invocation_type TEXT DEFAULT 'RequestResponse',
IN log_type TEXT DEFAULT 'None',
IN context JSON DEFAULT NULL,
IN qualifier VARCHAR(128) DEFAULT NULL,
OUT status_code INT,
OUT payload JSON,
OUT executed_version TEXT,
OUT log_result TEXT)
```

JSONB

```
aws_lambda.invoke(
IN function_name TEXT,
IN payload JSONB,
IN region TEXT DEFAULT NULL,
IN invocation_type TEXT DEFAULT 'RequestResponse',
IN log_type TEXT DEFAULT 'None',
IN context JSONB DEFAULT NULL,
IN qualifier VARCHAR(128) DEFAULT NULL,
OUT status_code INT,
OUT payload JSONB,
OUT executed_version TEXT,
OUT log_result TEXT)
```

```
aws_lambda.invoke(
IN function_name aws_commons._lambda_function_arn_1,
IN payload JSONB,
IN invocation_type TEXT DEFAULT 'RequestResponse',
IN log_type TEXT DEFAULT 'None',
IN context JSONB DEFAULT NULL,
IN qualifier VARCHAR(128) DEFAULT NULL,
OUT status_code INT,
OUT payload JSONB,
OUT executed_version TEXT,
OUT log_result TEXT
)
```

Parâmetros de entrada

function_name

O nome de identificação da função Lambda. O valor pode ser o nome da função, um ARN ou um ARN parcial. Para obter uma lista de formatos possíveis, consulte Formatos de nome da função Lambda no Guia do desenvolvedor do AWS Lambda.

payload

A entrada para a função Lambda. O formato pode ser JSON ou JSONB. Para obter mais informações, consulte Tipos JSON na documentação do PostgreSQL.

região

(Opcional) A Lambda Região da função. Por padrão, o RDS resolve a região da AWS a partir do ARN completo na `function_name` ou usa a região da instância de banco de dados RDS for PostgreSQL. Se este valor de região entrar em conflito com o fornecido no ARN de `function_name`, um erro será gerado.

invocation_type

O tipo de invocação da função do Lambda. O valor diferencia letras maiúsculas de minúsculas. Os valores possíveis incluem o seguinte:

- `RequestResponse` – O padrão. Esse tipo de invocação para uma função Lambda é síncrona e retorna uma carga útil de resposta no resultado. Use o tipo de invocação de `RequestResponse` quando seu fluxo de trabalho depende de receber o resultado da função Lambda imediatamente.
- `Event` – Esse tipo de invocação para uma função Lambda é assíncrono e retorna imediatamente sem uma carga devolvida. Use o tipo de invocação de`Event` quando não precisar de resultados da função Lambda antes de seu fluxo de trabalho prosseguir.
- `DryRun` – Este tipo de invocação testa o acesso sem executar a função Lambda.

log_type

O tipo de log Lambda a ser retornado no parâmetro de saída de `log_result`. O valor diferencia letras maiúsculas de minúsculas. Os valores possíveis incluem o seguinte:

- Cauda – O parâmetro de saída `log_result` devolvido incluirá os últimos 4 KB do log de execução.
- Nenhuma – Nenhuma informação de log Lambda é devolvida.

context

Contexto do cliente no formato JSON ou JSONB. Os campos a serem usados incluem que `custom` e `env`.

qualifier

Um qualificador que identifica a versão de uma função Lambda a ser invocada. Se esse valor entrar em conflito com um fornecido no ARN do `function_name`, gera um erro.

Parâmetros de saída

status_code

Um código de resposta de status HTTP. Para obter mais informações, consulte Elementos de resposta de invocação do Lambda no Guia do desenvolvedor do AWS Lambda.

payload

As informações devolvidas da função Lambda executada. O formato está em JSON ou JSONB.

executed_version

A versão da função Lambda executada.

log_result

As informações do log de execução devolvidas se o valor `log_type` é `Tail` quando a função Lambda for invocada. O resultado contém os últimos 4 KB do log de execução codificado em Base64.

aws_commons.create_lambda_function_arn

Cria uma estrutura `aws_commons._lambda_function_arn_1` para manter as informações do nome da função Lambda. Você pode usar os resultados da função `aws_commons.create_lambda_function_arn` no parâmetro `function_name` da função aws_lambda.invoke.

Sintaxe

```
aws_commons.create_lambda_function_arn(
    function_name TEXT,
    region TEXT DEFAULT NULL
```

```
    )
    RETURNS aws_commons._lambda_function_arn_1
```

Parâmetros de entrada

function_name

Uma string de texto necessária que contém o nome da função Lambda. O valor pode ser um nome de função, um ARN parcial ou um ARN completo.

região

Uma string de texto opcional que contém a região da AWS na qual a função Lambda está inserida. Para obter uma listagem de nomes de regiões da e os valores associados, consulte Regiões, zonas de disponibilidade e Local Zones.

Parâmetros aws_lambda

Na tabela, é possível encontrar parâmetros associados à função do `aws_lambda`.

Parâmetro	Descrição
`aws_lambda.connect_timeout_ms`	É um parâmetro dinâmico e define o tempo máximo de espera durante a conexão com o AWS Lambda. O valor padrão é `1000`. Os valores permitidos para esse parâmetro são de 1 a 900000.
`aws_lambda.request_timeout_ms`	É um parâmetro dinâmico e define o tempo máximo de espera enquanto aguarda a resposta do AWS Lambda. O valor padrão é `3000`. Os valores permitidos para esse parâmetro são de 1 a 900000.
`aws_lambda.endpoint_override`	Especifica o endpoint que pode ser usado para se conectar ao AWS Lambda. Uma string vazia seleciona o endpoint padrão do AWS Lambda para a região. É necessário reiniciar o banco de dados para esse parâmetro ter efeito.

Tarefas comuns de DBA do Amazon RDS para PostgreSQL

Os administradores de banco de dados (DBAs) executam várias tarefas ao administrar uma instância de banco de dados Amazon RDS para PostgreSQL. Se você já é um DBA familiarizado com o PostgreSQL, esteja ciente sobre algumas das diferenças importantes entre a execução do PostgreSQL em seu hardware e do RDS para PostgreSQL. Por exemplo, por ser um serviço gerenciado, o Amazon RDS não permite acesso ao shell às suas instâncias de banco de dados. Isso significa que você não tem acesso direto a `pg_hba.conf` e a outros arquivos de configuração. Para o RDS para PostgreSQL, as alterações que normalmente são feitas no arquivo de configuração do PostgreSQL de uma instância on-premises são feitas em um grupo de parâmetros de banco de dados personalizado associado à instância de banco de dados do RDS para PostgreSQL. Para ter mais informações, consulte Grupos de parâmetros para Amazon RDS.

Você também não pode acessar arquivos de log da mesma forma que faz com uma instância do PostgreSQL on-premises. Para saber mais sobre registro em log, consulte Arquivos de log do banco de dados do RDS para PostgreSQL.

Como outro exemplo, não é possível ter acesso à conta `superuser` do PostgreSQL. No RDS para PostgreSQL, o perfil `rds_superuser` é o perfil mais privilegiado, e é concedido ao `postgres` ao configurar. Se você estiver familiarizado com o uso do PostgreSQL on-premises ou for completamente novo no RDS para PostgreSQL, recomendamos que você entenda o perfil `rds_superuser` e como trabalhar com perfis, usuários, grupos e permissões. Para ter mais informações, consulteNoções básicas de perfis e permissões do PostgreSQL

Veja a seguir algumas tarefas comuns do DBA para o RDS para PostgreSQL.

Tópicos

- Agrupamentos compatíveis com RDS para PostgreSQL
- Noções básicas de perfis e permissões do PostgreSQL
- Tratamento de conexões inativas no PostgreSQL
- Trabalhar com o autovacuum do PostgreSQL no Amazon RDS para PostgreSQL
- Trabalhar com os mecanismos de registro em log compatíveis com o RDS para PostgreSQL
- Gerenciar arquivos temporários com o PostgreSQL
- Uso de pgBadger para análise de logs com o PostgreSQL
- Usar o PGSnapper para monitorar o PostgreSQL
- Trabalhar com parâmetros na instância de banco de dados do RDS para PostgreSQL

Agrupamentos compatíveis com RDS para PostgreSQL

Agrupamentos são conjuntos de regras que determinam como as strings de caracteres armazenadas no banco de dados são classificadas e comparadas. Eles desempenham um papel fundamental no sistema de computador e são incluídos como parte do sistema operacional. Os agrupamentos mudam com o tempo quando novos caracteres são adicionados aos idiomas ou quando ocorrem alterações nas regras de ordenação.

As bibliotecas de agrupamentos definem regras e algoritmos específicos para um agrupamento. As bibliotecas de agrupamentos mais populares usadas no PostgreSQL são GNU C (glibc) e Componentes Internacionais para Unicode (ICU). Por padrão, o RDS para PostgreSQL usa o agrupamento glibc que inclui ordens de classificação de caracteres Unicode para sequências de caracteres de vários bytes.

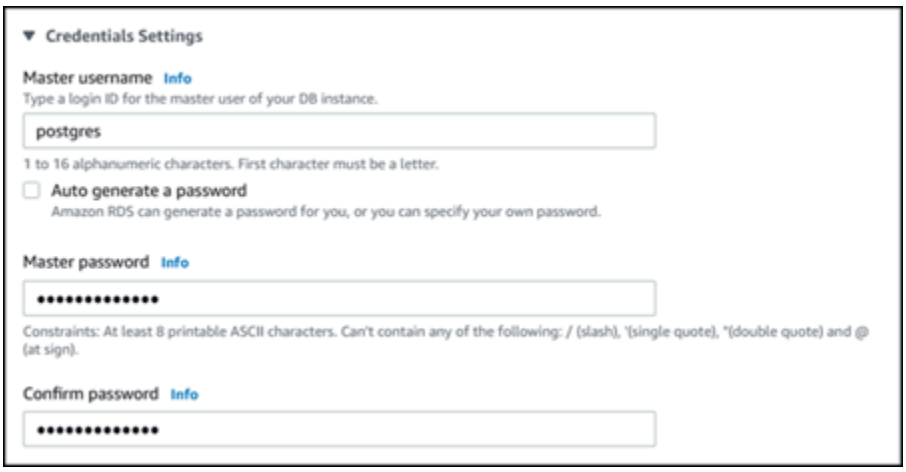
Quando você cria uma instância de banco de dados no RDS para PostgreSQL, ele verifica o agrupamento disponível no sistema operacional. Os parâmetros do PostgreSQL `LC_COLLATE` e `LC_CTYPE` do comando `CREATE DATABASE` são usados para especificar um agrupamento, que representa o agrupamento padrão nesse banco de dados. Como alternativa, você também pode usar o parâmetro `LOCALE` em `CREATE DATABASE` para definir esses parâmetros. Isso determina o agrupamento padrão para strings de caracteres no banco de dados e as regras para classificar caracteres como letras, números ou símbolos. Você também pode escolher um agrupamento para usar em uma coluna, um índice ou uma consulta.

O RDS para PostgreSQL depende da biblioteca glibc no sistema operacional para oferecer suporte a agrupamentos. A instância do RDS para PostgreSQL é atualizada periodicamente com as versões mais recentes do sistema operacional. Essas atualizações às vezes incluem uma versão mais recente da biblioteca glibc. Em situações raras, as versões mais recentes da glibc alteram a ordem de classificação ou o agrupamento de alguns caracteres, o que pode fazer com que os dados sejam classificados de forma diferente ou produzam entradas de índice inválidas. Se você descobrir problemas na ordem de classificação para agrupamento durante uma atualização, poderá ser necessário recompilar os índices.

Para reduzir os possíveis impactos das atualizações da glibc, o RDS para PostgreSQL agora inclui uma biblioteca de agrupamentos padrão independente. Essa biblioteca de agrupamentos está disponível no RDS para PostgreSQL 14.6, 13.9, 12.13, 11.18, 10.23 e versões secundárias mais recentes. É compatível com glibc 2.26-59.amzn2 e oferece estabilidade da ordem de classificação para evitar resultados de consulta incorretos.

Noções básicas de perfis e permissões do PostgreSQL

Ao criar uma instância de banco de dados do RDS para PostgreSQL usando o AWS Management Console, uma conta de administrador é criada ao mesmo tempo. Por padrão, o nome é `postgres`, conforme mostrado na captura de tela a seguir:



É possível escolher outro nome em vez de aceitar esse padrão (`postgres`). Se você fizer isso, o nome escolhido deverá começar com uma letra e ter 1 a 16 caracteres alfanuméricos. Para simplificar, nós nos referimos a essa conta de usuário principal pelo seu valor padrão (`postgres`) ao longo deste guia.

Ao usar `create-db-instance` na AWS CLI em vez de usar o AWS Management Console, você cria o nome ao passá-lo com o parâmetro `master-username` no comando. Para obter mais informações, consulte Criar uma instância de banco de dados do Amazon RDS.

Se você usar o AWS Management Console, a AWS CLI ou a API do Amazon RDS e se usar o nome `postgres` padrão ou escolher um nome diferente, essa primeira conta de usuário do banco de dados será membro do grupo `rds_superuser` e terá privilégios de `rds_superuser`.

Tópicos

- Noções básicas sobre o perfil rds_superuser
- Controlar o acesso de usuários ao banco de dados PostgreSQL
- Delegar e controlar o gerenciamento de senhas de usuários
- Usar criptografia de senha SCRAM para PostgreSQL

Noções básicas sobre o perfil rds_superuser

No PostgreSQL, um perfil pode definir um usuário, um grupo ou um conjunto de permissões específicas concedidas a um grupo ou usuário a vários objetos no banco de dados. Os comandos do PostgreSQL para `CREATE USER` e `CREATE GROUP` foram substituídos pelo comando mais geral `CREATE ROLE` com propriedades específicas para distinguir usuários de banco de dados. Um usuário de banco de dados pode ser considerado um perfil com o privilégio LOGIN.

> **Note**
>
> Os comandos `CREATE USER` e `CREATE GROUP` ainda podem ser usados. Para obter mais informações, consulte Database Roles (Perfis de banco de dados) na documentação do PostgreSQL.

O usuário `postgres` é o usuário de banco de dados mais privilegiado na instância de banco de dados do RDS para PostgreSQL. Ele tem as características definidas pela seguinte instrução `CREATE ROLE`.

```
CREATE ROLE postgres WITH LOGIN NOSUPERUSER INHERIT CREATEDB CREATEROLE NOREPLICATION
 VALID UNTIL 'infinity'
```

As propriedades `NOSUPERUSER`, `NOREPLICATION`, `INHERIT` e `VALID UNTIL 'infinity'` são as opções padrão da instrução CREATE ROLE, a menos que especificado de outra forma.

Por padrão, `postgres` tem privilégios concedidos à função `rds_superuser` e permissões para criar funções e bancos de dados. O perfil `rds_superuser` permite que o usuário `postgres` faça o seguinte:

- Adicione as extensões que estão disponíveis para o uso com o Amazon RDS. Para obter mais informações, consulte Trabalhar com recursos do PostgreSQL compatíveis com o Amazon RDS para PostgreSQL
- Crie funções para usuários e conceda privilégios aos usuários. Para obter mais informações, consulte CREATE ROLE e GRANT na documentação do PostgreSQL.
- Crie bancos de dados. Para obter mais informações, consulte CREATE DATABASE na documentação do PostgreSQL.
- Conceda privilégios de `rds_superuser` a outras funções de usuário que não têm esses privilégios e revogue esses privilégios conforme necessário. Recomendamos que você conceda

esse perfil somente aos usuários que executam tarefas de superusuário. Em outras palavras, você pode conceder esse perfil a administradores de banco de dados (DBAs) ou administradores de sistema.

- Conceda (e revogue) o perfil `rds_replication` a usuários de banco de dados que não têm o perfil `rds_superuser`.
- Conceda (e revogue) o perfil `rds_password` a usuários de banco de dados que não têm o perfil `rds_superuser`.
- Obtenha informações de status sobre todas as conexões de banco de dados usando a visualização `pg_stat_activity`. Quando necessário, `rds_superuser` pode interromper qualquer conexão usando `pg_terminate_backend` ou `pg_cancel_backend`.

Na instrução `CREATE ROLE postgres...`, é possível ver que o perfil do usuário `postgres` não autoriza especificamente as permissões de `superuser` do PostgreSQL. O RDS para PostgreSQL é um serviço gerenciado, portanto, você não pode acessar o sistema operacional host nem se conectar usando a conta `superuser` do PostgreSQL. Muitas das tarefas que exigem o acesso de `superuser` em um PostgreSQL autônomo são gerenciadas automaticamente pelo Amazon RDS.

Para obter mais informações sobre como conceder privilégios, consulte GRANT na documentação do PostgreSQL.

O perfil `rds_superuser` é um dos vários perfis predefinidos em um Instância de banco de dados do RDS para PostgreSQL.

> **Note**
>
> No PostgreSQL 13 e em versões anteriores, os perfis predefinidos são conhecidos como perfis padrão.

Na lista a seguir, você encontra alguns dos outros perfis predefinidos que são criados automaticamente para um novo Instância de banco de dados do RDS para PostgreSQL. Os perfis predefinidos e seus privilégios não podem ser alterados. Não é possível descartar, renomear ou modificar os privilégios desses perfis predefinidos. Qualquer tentativa de fazer isso gerará um erro.

- rds_password: um perfil que pode alterar senhas e configurar restrições de senha para usuários de bancos de dados. O perfil `rds_superuser` recebe esse perfil por padrão e pode concedê-lo

aos usuários do banco de dados. Para obter mais informações, consulte Controlar o acesso de usuários ao banco de dados PostgreSQL.

  - Para versões do RDS para PostgreSQL anteriores à 14, o perfil `rds_password` pode alterar senhas e configurar restrições de senha para usuários de bancos de dados e usuários com o perfil `rds_superuser`. Para versões do RDS para PostgreSQL 14 e posteriores, o perfil `rds_password` pode alterar senhas e configurar restrições de senha somente para usuários de banco de dados. Somente usuários com o perfil `rds_superuser` podem realizar essas ações em outros usuários com o perfil `rds_superuser`.
- rdsadmin: um perfil criado para lidar com muitas das tarefas de gerenciamento que o administrador com privilégios de `superuser` executaria em um banco de dados PostgreSQL autônomo. Esse perfil é usado internamente pelo RDS para PostgreSQL para várias tarefas de gerenciamento.
- rdstopmgr: um perfil usado internamente pelo Amazon RDS para comportar implantações multi-AZ.
- rds_reserved: função usada internamente pelo Amazon RDS para reservar conexões de banco de dados.

Visualizar perfis e os respectivos privilégios

Você pode visualizar perfis predefinidos e os respectivos privilégios na instância de banco de dados do RDS para PostgreSQL usando comandos diferentes, dependendo da versão do PostgreSQL.

Para o PostgreSQL 15 e versões anteriores

Conecte-se à instância de banco de dados do RDS para PostgreSQL e use o comando \du no psql:

```
postgres=> \du
                                                   List of roles
    Role name    |                         Attributes                         |
                 Member of
-----------------+------------------------------------------------------------
+-------------------------------------------------
 postgres        | Create role, Create DB                                    +|
 {rds_superuser}
                 | Password valid until infinity                              |
 rds_ad          | Cannot login                                               | {}
 rds_iam         | Cannot login                                               | {}
 rds_password    | Cannot login                                               | {}
 rds_replication | Cannot login                                               | {}
 rds_superuser   | Cannot login                                               |
 {pg_monitor,pg_signal_backend,rds_password,rds_replication}
```

```
 rdsadmin        | Superuser, Create role, Create DB, Replication, Bypass RLS+| {}
                 | Password valid until infinity                              |
```

Para o PostgreSQL 16 e versões posteriores

A partir do PostgreSQL 16, use o comando `\drg+` para visualizar informações detalhadas sobre associação a perfis:

```
postgres=> \drg+
                              List of role grants
   Role name   |          Member of          |       Options       | Grantor
---------------+-----------------------------+---------------------+----------
 postgres      | rds_superuser               | INHERIT, SET        | rdsadmin
 rds_superuser | pg_checkpoint               | ADMIN, INHERIT, SET | rdsadmin
 rds_superuser | pg_monitor                  | ADMIN, INHERIT, SET | rdsadmin
 rds_superuser | pg_signal_backend           | ADMIN, INHERIT, SET | rdsadmin
 rds_superuser | pg_use_reserved_connections | ADMIN, INHERIT, SET | rdsadmin
 rds_superuser | rds_password                | ADMIN, INHERIT, SET | rdsadmin
 rds_superuser | rds_replication             | ADMIN, INHERIT, SET | rdsadmin
```

Para verificar a associação a perfis, independentemente da versão do PostgreSQL, você pode usar a seguinte consulta SQL:

```
SELECT m.rolname AS "Role name", r.rolname AS "Member of"
FROM pg_catalog.pg_roles m
JOIN pg_catalog.pg_auth_members pam ON (pam.member = m.oid)
LEFT JOIN pg_catalog.pg_roles r ON (pam.roleid = r.oid)
LEFT JOIN pg_catalog.pg_roles g ON (pam.grantor = g.oid)
WHERE m.rolname !~ '^pg_'
ORDER BY 1, 2;
```

Na saída, é possível ver que `rds_superuser` não é um perfil de usuário de banco de dados (não pode fazer login), mas tem os privilégios de muitos outros perfis. Também é possível ver que esse usuário do banco de dados `postgres` é membro do perfil `rds_superuser`. Como mencionado anteriormente, `postgres` é o valor padrão na página Create database (Criar banco de dados) do console do Amazon RDS. Se você escolheu outro nome, esse nome será mostrado na lista de perfis.

Controlar o acesso de usuários ao banco de dados PostgreSQL

Os novos bancos de dados no PostgreSQL são sempre criados com um conjunto padrão de privilégios no esquema `public` do banco de dados que permite que todos os usuários e perfis do

banco de dados criem objetos. Esses privilégios permitem que os usuários do banco de dados se conectem ao banco de dados, por exemplo, e criem tabelas temporárias durante a conexão.

Para controlar melhor o acesso dos usuários às instâncias de bancos de dados que você cria na instância de banco de dados do RDS para PostgreSQL, recomendamos que você revogue esses privilégios de `public` padrão. Depois disso, conceda privilégios específicos aos usuários do banco de dados de forma mais granular, conforme mostrado no procedimento a seguir.

Como configurar perfis e privilégios para uma nova instância de banco de dados

Suponha que você esteja configurando um banco de dados em uma instância de banco de dados do RDS para PostgreSQL recém-criada para uso por vários pesquisadores que precisam de acesso de leitura-gravação ao banco de dados.

1. Use o `psql` (ou o pgAdmin) para se conectar à na instância de banco de dados do RDS para PostgreSQL:

   ```
   psql --host=your-db-instance.666666666666.aws-region.rds.amazonaws.com --port=5432
    --username=postgres --password
   ```

   Insira sua senha quando for solicitado. O cliente `psql` conecta-se e exibe o banco de dados de conexão administrativa padrão, `postgres=>`, como o prompt.

2. Para evitar que os usuários do banco de dados criem objetos no esquema `public`, faça o seguinte:

   ```
   postgres=> REVOKE CREATE ON SCHEMA public FROM PUBLIC;
   REVOKE
   ```

3. Em seguida, crie uma nova instância de banco de dados:

   ```
   postgres=> CREATE DATABASE lab_db;
   CREATE DATABASE
   ```

4. Revogue todos os privilégios do esquema `PUBLIC` nesse novo banco de dados.

   ```
   postgres=> REVOKE ALL ON DATABASE lab_db FROM public;
   REVOKE
   ```

5. Crie um perfil para os usuários do banco de dados.

```
postgres=> CREATE ROLE lab_tech;
CREATE ROLE
```

6. Permita que os usuários do banco de dados que têm esse perfil conectem-se ao banco de dados.

```
postgres=> GRANT CONNECT ON DATABASE lab_db TO lab_tech;
GRANT
```

7. Conceda a todos os usuários com o perfil `lab_tech` todos os privilégios nesse banco de dados.

```
postgres=> GRANT ALL PRIVILEGES ON DATABASE lab_db TO lab_tech;
GRANT
```

8. Crie os usuários do banco de dados, da seguinte forma:

```
postgres=> CREATE ROLE lab_user1 LOGIN PASSWORD 'change_me';
CREATE ROLE
postgres=> CREATE ROLE lab_user2 LOGIN PASSWORD 'change_me';
CREATE ROLE
```

9. Conceda a esses dois usuários os privilégios associados ao perfil lab_tech:

```
postgres=> GRANT lab_tech TO lab_user1;
GRANT ROLE
postgres=> GRANT lab_tech TO lab_user2;
GRANT ROLE
```

Neste ponto, o `lab_user1` e o `lab_user2` podem conectar-se ao banco de dados `lab_db`. Este exemplo não segue as práticas recomendadas para uso corporativo, o que pode incluir a criação de várias instâncias de banco de dados, esquemas diferentes e concessão de permissões limitadas. Para obter informações mais completas e cenários adicionais, consulte Managing PostgreSQL Users and Roles (Gerenciar usuários e perfis do PostgreSQL).

Para obter mais informações sobre privilégios nos bancos de dados PostgreSQL, consulte o comando GRANT na documentação do PostgreSQL.

Delegar e controlar o gerenciamento de senhas de usuários

Como DBA, talvez você queira delegar o gerenciamento de senhas de usuários. Ou talvez você queira impedir que os usuários do banco de dados alterem senhas ou reconfigurem restrições de senha, como o tempo de vida da senha. Para garantir que somente os usuários do banco de dados escolhidos possam alterar as configurações de senha, é possível ativar o recurso de gerenciamento restrito de senhas. Quando você ativa esse recurso, somente os usuários do banco de dados que receberam o perfil `rds_password` podem gerenciar senhas.

> **Note**
>
> Para usar o gerenciamento restrito de senhas, a instância de banco de dados do RDS para PostgreSQL deve estar executando o Amazon Aurora PostgreSQL 10.6 ou superior.

Por padrão, esse recurso está `off`, conforme mostrado a seguir:

```
postgres=> SHOW rds.restrict_password_commands;
  rds.restrict_password_commands
--------------------------------
 off
(1 row)
```

Para ativar esse recurso, use um grupo de parâmetros personalizado e altere a configuração de `rds.restrict_password_commands` para 1. Reinicialize a instância de banco de dados do RDS para PostgreSQL para que a configuração entre em vigor.

Com esse recurso ativo, os privilégios de `rds_password` são necessários para os seguintes comandos SQL:

```
CREATE ROLE myrole WITH PASSWORD 'mypassword';
CREATE ROLE myrole WITH PASSWORD 'mypassword' VALID UNTIL '2023-01-01';
ALTER ROLE myrole WITH PASSWORD 'mypassword' VALID UNTIL '2023-01-01';
ALTER ROLE myrole WITH PASSWORD 'mypassword';
ALTER ROLE myrole VALID UNTIL '2023-01-01';
ALTER ROLE myrole RENAME TO myrole2;
```

A renomeação de um perfil (`ALTER ROLE myrole RENAME TO newname`) também será restrita se a senha usar o algoritmo de hash MD5.

Com esse recurso ativo, qualquer tentativa de executar um desses comandos SQL sem as permissões do perfil `rds_password` gerará o seguinte erro:

```
ERROR: must be a member of rds_password to alter passwords
```

Recomendamos conceder a `rds_password` a apenas alguns perfis usados exclusivamente para o gerenciamento de senhas. Se você conceder privilégios de `rds_password` a usuários de banco de dados que não têm privilégios de `rds_superuser`, também precisará conceder o atributo `CREATEROLE` a eles.

Verifique os requisitos de senha, como expiração e complexidade necessária, no lado do cliente. Se você usar seu próprio utilitário do lado do cliente para alterações relacionadas a senhas, o utilitário precisará ser membro de `rds_password` e ter privilégios de `CREATE ROLE`.

Usar criptografia de senha SCRAM para PostgreSQL

O Salted Challenge Response Authentication Mechanism (SCRAM) é uma alternativa ao algoritmo de resumo de mensagens padrão (MD5) do PostgreSQL para senhas de criptografia. O mecanismo de autenticação SCRAM é considerado mais seguro que o MD5. Para saber mais sobre essas duas abordagens diferentes para proteger senhas, consulte Autorização com senha na documentação do PostgreSQL.

Recomendamos que você use o SCRAM em vez de o MD5 como o esquema de criptografia de senha para seu Instância de banco de dados do RDS para PostgreSQL. É um mecanismo criptográfico de resposta a desafios que usa o algoritmo scram-sha-256 para autenticação e criptografia de senha.

Convém atualizar as bibliotecas das aplicações cliente para oferecer compatibilidade com o SCRAM. Por exemplo, versões do JDBC anteriores à 42.2.0 não são compatíveis com o SCRAM. Para obter mais informações, consulte Driver JDBC do PostgreSQL na documentação do driver JDBC do PostgreSQL. Para obter uma lista de outros drivers do PostgreSQL e compatibilidade com o SCRAM, consulte Lista de drivers na documentação do PostgreSQL.

A versão 13.1 do RDS para PostgreSQL e versões posteriores são compatíveis com scram-sha-256. Essas versões também permitem configurar sua instância de banco de dados para exigir o SCRAM, conforme abordado nos procedimentos a seguir.

Configurar a instância de banco de dados do RDS para PostgreSQL para exigir o SCRAM

você pode exigir que a instância de banco de dados do RDS para PostgreSQL aceite apenas senhas que usem o algoritmo scram-sha-256.

> **Important**
>
> Para proxies RDS existentes com bancos de dados PostgreSQL, se você modificar a autenticação do banco de dados para usar somente `SCRAM`, o proxy ficará indisponível por até 60 segundos. Para evitar o problema, faça o seguinte:
>
> - O banco de dados deve permitir tanto a autenticação `SCRAM` quanto a `MD5`.
> - Para usar somente a autenticação `SCRAM`, crie um proxy, migre o tráfego da aplicação para o novo proxy e exclua o proxy anteriormente associado ao banco de dados.

Antes de fazer alterações em seu sistema, entenda o processo completo da seguinte forma:

- Obtenha informações sobre todos os perfis e criptografia de senha para todos os usuários do banco de dados.
- Confira novamente as configurações dos parâmetros da instância de banco de dados do RDS para PostgreSQL para verificar os parâmetros que controlam a criptografia de senha.
- Se sua instância de banco de dados do RDS para PostgreSQL usa um grupo de parâmetros padrão, você precisará criar um grupo de parâmetros de banco de dados personalizado e aplicá-lo à sua instância de banco de dados do RDS para PostgreSQL para que você possa modificar parâmetros quando necessário. Se sua instância de banco de dados do RDS para PostgreSQL usa um grupo de parâmetros personalizado, você poderá modificar os parâmetros necessários posteriormente no processo, conforme necessário.
- Altere o parâmetro `password_encryption` para `scram-sha-256`.
- Avise a todos os usuários do banco de dados que eles precisam atualizar as senhas. Faça o mesmo para a conta `postgres`. As novas senhas são criptografadas e armazenadas usando o algoritmo scram-sha-256.
- Verifique se todas as senhas são criptografadas usando o tipo de criptografia.
- Se todas as senhas utilizarem scram-sha-256, você poderá alterar o parâmetro de `rds.accepted_password_auth_method` para `md5+scram`.

> **Warning**
>
> Depois de alterar `rds.accepted_password_auth_method` para scram-sha-256, os usuários (perfis) com senhas criptografadas com `md5` não conseguirão se conectar.

Preparar-se para exigir o SCRAM para sua instância de banco de dados do RDS para PostgreSQL

Antes de fazer qualquer alteração em sua instância de banco de dados do RDS para PostgreSQL, confira todas as contas de usuário do banco de dados existentes. Além disso, verifique o tipo de criptografia usada para senhas. Você pode realizar essas tarefas usando a extensão `rds_tools`. Consulte quais versões do PostgreSQL são compatíveis com `rds_tools` em Extension versions for Amazon RDS for PostgreSQL.

Como obter uma lista de usuários (perfis) de banco de dados e métodos de criptografia de senha

1. Use o `psql` para conectar-se à instância de banco de dados do RDS para PostgreSQL conforme mostrado a seguir.

   ```
   psql --host=db-name.111122223333.aws-region.rds.amazonaws.com --port=5432 --
   username=postgres --password
   ```

2. Instale a extensão `rds_tools`.

   ```
   postgres=> CREATE EXTENSION rds_tools;
   CREATE EXTENSION
   ```

3. Obtenha uma lista de perfis e criptografias.

   ```
   postgres=> SELECT * FROM
         rds_tools.role_password_encryption_type();
   ```

   Você verá uma saída semelhante à seguinte.

   ```
          rolname        | encryption_type
   ----------------------+-----------------
    pg_monitor           |
    pg_read_all_settings |
    pg_read_all_stats    |
    pg_stat_scan_tables  |
   ```

```
 pg_signal_backend    |
 lab_tester           | md5
 user_465             | md5
 postgres             | md5
(8 rows)
```

Criar um grupo de parâmetros de banco de dados personalizado

> **Note**
>
> Se sua instância de banco de dados do RDS para PostgreSQL já usa um grupo de parâmetros personalizado, não é necessário criar outro.

Para obter uma visão geral dos grupos de parâmetros do Amazon RDS, consulte Trabalhar com parâmetros na instância de banco de dados do RDS para PostgreSQL.

O tipo de criptografia de senha usada para senhas é definido em um parâmetro, `password_encryption`. A criptografia permitida pela instância de banco de dados do RDS para PostgreSQL é definida em outro parâmetro, `rds.accepted_password_auth_method`. Alterar qualquer um desses valores padrão exige que você crie um grupo de parâmetros de banco de dados personalizado e aplique-o à sua instância.

Você também pode usar o AWS Management Console ou a API do RDS para criar um grupo de parâmetros de banco de dados personalizado. Consulte mais informações em .

Associe o grupo de parâmetros personalizado à sua instância de banco de dados.

Como criar um grupo de parâmetros de banco de dados personalizado

1. Use o comando `create-db-parameter-group` da CLI para criar o grupo de parâmetros de banco de dados personalizado. Este exemplo usa `postgres13` como a origem desse grupo de parâmetros personalizado.

   Para Linux, macOS ou Unix:

   ```
   aws rds create-db-parameter-group --db-parameter-group-name 'docs-lab-scram-passwords' \
     --db-parameter-group-family postgres13  --description 'Custom parameter group for SCRAM'
   ```

Para Windows:

```
aws rds create-db-parameter-group --db-parameter-group-name "docs-lab-scram-passwords" ^
  --db-parameter-group-family postgres13  --description "Custom DB parameter group for SCRAM"
```

2. Use o comando modify-db-instance da CLI para aplicar esse grupo de parâmetros personalizado ao seu cluster de banco de dados do RDS para PostgreSQL.

   Para Linux, macOS ou Unix:

   ```
   aws rds modify-db-instance --db-instance-identifier 'your-instance-name' \
           --db-parameter-group-name "docs-lab-scram-passwords
   ```

   Para Windows:

   ```
   aws rds modify-db-instance --db-instance-identifier "your-instance-name" ^
           --db-parameter-group-name "docs-lab-scram-passwords
   ```

   Para sincronizar novamente sua instância de banco de dados do RDS para PostgreSQL com seu grupo de parâmetros de cluster de banco de dados personalizado, você precisa reiniciar a instância primária e todas as outras instâncias do cluster. Programe para que isso ocorra durante sua janela de manutenção regular a fim de minimizar o impacto para os usuários.

Configurar a criptografia de senha para usar o SCRAM

O mecanismo de criptografia de senha usado por uma instância de banco de dados do RDS para PostgreSQL é definido no grupo de parâmetros de banco de dados no parâmetro `password_encryption`. Os valores permitidos não estão definidos, `md5` ou `scram-sha-256`. O valor padrão depende da versão do RDS para PostgreSQL da seguinte forma:

- RDS para PostgreSQL 14 e superior: o padrão é `scram-sha-256`
- RDS para PostgreSQL 13: o padrão é `md5`

Com um grupo de parâmetros de banco de dados personalizado anexado à sua instância de banco de dados do RDS para PostgreSQL, você pode modificar valores para o parâmetro de criptografia de senha.

Name	Values	Allowed values	Modifiable	Source	Apply type
password_encryption	md5	md5,scram-sha-256	true	system	dynamic
rds.accepted_password_auth_method	md5+scram	md5+scram, scram	true	system	dynamic

Como alterar a configuração de criptografia de senha para scram-sha-256

- Altere o valor da criptografia de senha para scram-sha-256, conforme mostrado a seguir. A alteração pode ser aplicada imediatamente porque o parâmetro é dinâmico, portanto, não é necessário reiniciar para que a alteração seja implementada.

  Para Linux, macOS ou Unix:

```
aws rds modify-db-parameter-group --db-parameter-group-name \
  'docs-lab-scram-passwords' --parameters
 'ParameterName=password_encryption,ParameterValue=scram-
sha-256,ApplyMethod=immediate'
```

  Para Windows:

```
aws rds modify-db-parameter-group --db-parameter-group-name ^
  "docs-lab-scram-passwords" --parameters
 "ParameterName=password_encryption,ParameterValue=scram-
sha-256,ApplyMethod=immediate"
```

Migrar senhas para funções de usuário para o SCRAM

Você pode migrar senhas de perfis de usuário para o SCRAM conforme descrito a seguir.

Como migrar senhas de usuário (função) do banco de dados do MD5 para o SCRAM

1. Faça login como usuário administrador (nome de usuário padrão, `postgres`) conforme mostrado a seguir.

```
psql --host=db-name.111122223333.aws-region.rds.amazonaws.com --port=5432 --
username=postgres --password
```

2. Confira a configuração do parâmetro `password_encryption` em sua instância de banco de dados do RDS para PostgreSQL usando o comando a seguir.

```
postgres=> SHOW password_encryption;
 password_encryption
---------------------
 md5
 (1 row)
```

3. Altere o valor desse parâmetro para scram-sha-256. Esse é um parâmetro dinâmico, portanto, você não precisa reinicializar a instância depois de fazer essa alteração. Confira o valor novamente para garantir que agora ele esteja definido como `scram-sha-256` da seguinte forma.

```
postgres=> SHOW password_encryption;
 password_encryption
---------------------
 scram-sha-256
 (1 row)
```

4. Avise a todos os usuários do banco de dados para alterar senhas. Altere também sua própria senha para a conta `postgres` (o usuário do banco de dados com privilégios de `rds_superuser`).

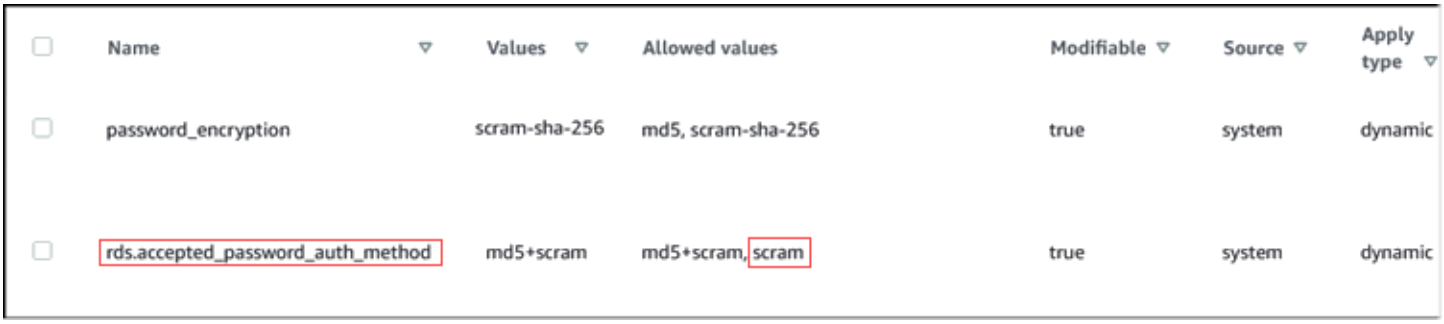
```
labdb=> ALTER ROLE postgres WITH LOGIN PASSWORD 'change_me';
ALTER ROLE
```

5. Repita o processo para todos os bancos de dados no Instância de banco de dados do RDS para PostgreSQL.

Alterar o parâmetro para exigir o SCRAM

Esta é a etapa final do processo. Depois de fazer a alteração no procedimento a seguir, as contas de usuário (perfis) que ainda usam a criptografia `md5` para senhas não poderão fazer login no Instância de banco de dados do RDS para PostgreSQL.

O `rds.accepted_password_auth_method` especifica o método de criptografia que a instância de banco de dados do RDS para PostgreSQL aceita para uma senha de usuário durante o processo de login. O valor padrão é `md5+scram`, o que significa que qualquer método é aceito. Na imagem a seguir, você pode encontrar a configuração padrão para esse parâmetro.



Name	Values	Allowed values	Modifiable	Source	Apply type
password_encryption	scram-sha-256	md5, scram-sha-256	true	system	dynamic
rds.accepted_password_auth_method	md5+scram	md5+scram, scram	true	system	dynamic

Os valores permitidos para esse parâmetro são `md5+scram` ou `scram`. Alterar esse valor de parâmetro para `scram` torna isso um requisito.

Como alterar o valor do parâmetro para exigir autenticação SCRAM para senhas

1. Verifique se todas as senhas de usuários para todos os bancos de dados na instância de banco de dados do RDS para PostgreSQL usam `scram-sha-256` para criptografia de senha. Para fazer isso, consulte `rds_tools` para obter o perfil (usuário) e o tipo de criptografia, da seguinte forma.

```
postgres=> SELECT * FROM rds_tools.role_password_encryption_type();
  rolname        | encryption_type
  ---------------------+-----------------
  pg_monitor           |
  pg_read_all_settings |
  pg_read_all_stats    |
  pg_stat_scan_tables  |
  pg_signal_backend    |
  lab_tester           | scram-sha-256
  user_465             | scram-sha-256
  postgres             | scram-sha-256
  ( rows)
```

2. Repita a consulta para todas as instâncias de banco de dados em seu Instância de banco de dados do RDS para PostgreSQL.

   Se todas as senhas usam scram-sha-256, você pode prosseguir.

3. Altere o valor da autenticação de senha aceita para scram-sha-256 da seguinte forma.

Para Linux, macOS ou Unix:

```
aws rds modify-db-parameter-group --db-parameter-group-name 'docs-lab-scram-
passwords' \
  --parameters
 'ParameterName=rds.accepted_password_auth_method,ParameterValue=scram,ApplyMethod=immediat
```

Para Windows:

```
aws rds modify-db-parameter-group --db-parameter-group-name "docs-lab-scram-
passwords" ^
  --parameters
 "ParameterName=rds.accepted_password_auth_method,ParameterValue=scram,ApplyMethod=immediat
```

Tratamento de conexões inativas no PostgreSQL

Conexões inativas ocorrem quando uma sessão de banco de dados permanece ativa no servidor mesmo que a aplicação cliente tenha sido abandonada ou encerrada de forma anormal. Essa situação geralmente ocorre quando os processos do cliente falham ou são encerrados inesperadamente sem fechar adequadamente as respectivas conexões de banco de dados ou cancelar solicitações em andamento.

O PostgreSQL identifica e limpa com eficiência as conexões inativas quando os processos do servidor estão ociosos ou tentam enviar dados aos clientes. No entanto, a detecção é um desafio para sessões que estão ociosas, aguardando a entrada do cliente ou executando consultas ativamente. Para lidar com esses cenários, o PostgreSQL oferece os parâmetros `tcp_keepalives_*`, `tcp_user_timeout` e `client_connection_check_interval`.

Tópicos

- Conceitos básicos sobre keepalive de TCP
- Principais parâmetros de keepalive de TCP no RDS para PostgreSQL
- Casos de uso para configurações de keepalive de TCP
- Práticas recomendadas

Conceitos básicos sobre keepalive de TCP

O keepalive de TCP é um mecanismo em nível de protocolo que ajuda a manter e verificar a integridade da conexão. Cada conexão TCP mantém configurações em nível de kernel que controlam o comportamento de keepalive. Quando o temporizador de keepalive expira, o sistema realiza o seguinte:

- Envia um pacote de sondagem sem dados e com o sinalizador ACK definido.
- Espera uma resposta do endpoint remoto de acordo com as especificações de TCP/IP.
- Gerencia o estado da conexão com base na resposta ou na falta dela.

Principais parâmetros de keepalive de TCP no RDS para PostgreSQL

Parameter	Descrição	Valores padrão
`tcp_keepalives_idle`	Specifies number of seconds of inactivity before sending keepalive message.	300
`tcp_keepalives_interval`	Specifies number of seconds between retransmissions of unacknowledged keepalive messages.	30
`tcp_keepalives_count`	Maximum lost keepalive messages before declaring connection dead	2
`tcp_user_timeout`	Specifies how long (in Milliseconds) unacknowledged data can remain before forcibly closing the connection.	0
`client_connection_check_interval`	Sets the interval (in Milliseconds) for checking client connection status during long-running queries. This ensures	0

Parameter	Descrição	Valores padrão
	quicker detection of closed connections.	

Casos de uso para configurações de keepalive de TCP

Manter as sessões ociosas ativas

Para evitar que conexões ociosas sejam encerradas por firewalls ou roteadores devido a inatividade:

- Configure `tcp_keepalives_idle` para enviar pacotes de keepalive em intervalos regulares.

Detectar conexões inativas

Para detectar conexões inativas imediatamente:

- Ajuste `tcp_keepalives_idle`, `tcp_keepalives_interval` e `tcp_keepalives_count`. Por exemplo, com os padrões do Aurora PostgreSQL, a detecção de uma conexão inativa leva cerca de 1 minuto (2 sondagens × 30 segundos). Reduzir esses valores pode acelerar a detecção.
- Use `tcp_user_timeout` para especificar o tempo máximo de espera para uma confirmação.

As configurações de keepalive de TCP ajudam o kernel a detectar conexões inativas, mas talvez o PostgreSQL não aja enquanto o soquete não for usado. Se uma sessão estiver executando uma consulta longa, as conexões inativas só poderão ser detectadas após a conclusão da consulta. No PostgreSQL 14 e versões posteriores, `client_connection_check_interval` pode acelerar a detecção de conexões inativas pesquisando periodicamente o soquete durante a execução da consulta.

Práticas recomendadas

- Defina intervalos razoáveis de keepalive: ajuste `tcp_user_timeout`, `tcp_keepalives_idle`, `tcp_keepalives_count` e `tcp_keepalives_interval` para contrabalançar a velocidade de detecção e o uso de recursos.
- Otimize para melhorar seu ambiente: alinhe as configurações ao comportamento da rede, às políticas de firewall e às necessidades da sessão.

- Aproveite os recursos do PostgreSQL: use `client_connection_check_interval` no PostgreSQL 14 e versões posteriores para realizar verificações de conexão eficientes.

Trabalhar com o autovacuum do PostgreSQL no Amazon RDS para PostgreSQL

É altamente recomendável que você use o recurso de autovacuum para bancos de dados PostgreSQL a fim de manter a integridade de sua instância de banco de dados PostgreSQL. O autovacuum automatiza a execução dos comandos VACUUM e ANALYZE. Ele verifica as tabelas com um grande número de tuplas inseridas, atualizadas ou excluídas. Após essa verificação, ele recupera o armazenamento removendo dados obsoletos ou tuplas do banco de dados PostgreSQL.

Por padrão, o autovacuum é ativado nas instâncias de banco de dados do RDS para PostgreSQL que você cria usando qualquer um dos grupos de parâmetros de banco de dados padrão do PostgreSQL. Outros parâmetros de configuração associados ao recurso autovacuum também são definidos por padrão. Como esses padrões são genéricos, você pode se beneficiar do ajuste de alguns dos parâmetros associados ao recurso autovacuum para seu workload específico.

A seguir, você pode encontrar mais informações sobre o autovacuum e como ajustar alguns dos respectivos parâmetros em sua instância de banco de dados do RDS para PostgreSQL. Para ter informações gerais, consulte Práticas recomendadas para trabalhar com PostgreSQL.

Tópicos

- Alocar memória para autovacuum
- Reduzir a probabilidade de conclusão de IDs de transação
- Determinar se as tabelas no seu banco de dados precisam de vacuum
- Determinar quais tabelas são atualmente elegíveis para autovacuum
- Determinar se o autovacuum está em execução e por quanto tempo
- Realização de um congelamento manual de vacuum
- Reindexação de uma tabela quando o autovacuum está em execução
- Gerenciar o autovacuum com grandes índices
- Outros parâmetros que afetam o autovacuum
- Definir parâmetros de autovacuum em nível de tabela
- Registrar atividades do autovacuum e do vacuum em log
- Noções básicas sobre o comportamento do autovacuum com bancos de dados inválidos

- Identificar e resolver bloqueadores de limpeza agressivos no RDS para PostgreSQL

Alocar memória para autovacuum

Um dos parâmetros mais importantes que influenciam a performance do autovacuum é o `autovacuum_work_mem`. No RDS para PostgreSQL versões 14 e anteriores, o parâmetro `autovacuum_work_mem` é definido como -1, indicando que a configuração de `maintenance_work_mem` é usada em vez disso. Para todas as outras versões, `autovacuum_work_mem` é determinado por GREATEST({DBInstanceClassMemory/32768}, 65536).

As operações vacuum manuais sempre usam a configuração `maintenance_work_mem`, com uma configuração padrão de GREATEST({DBInstanceClassMemory/63963136*1024}, 65536), e também é possível fazer o ajuste no nível da sessão usando o comando `SET` para operações `VACUUM` manuais mais específicas.

O `autovacuum_work_mem` determina a memória do autovacuum para armazenar identificadores de tuplas mortas (`pg_stat_all_tables.n_dead_tup`) para índices de vacuum.

Ao executar cálculos para determinar o valor do parâmetro `autovacuum_work_mem`, esteja ciente do seguinte:

- Se você definir o parâmetro com um valor muito baixo, o processo de vacuum talvez precise verificar a tabela várias vezes para concluir o trabalho. Essas várias verificações podem ter um impacto negativo sobre a performance. Para instâncias maiores, a configuração de `maintenance_work_mem` ou `autovacuum_work_mem` para pelo menos 1 GB pode melhorar o desempenho de vaccum nas tabelas com um grande número de tuplas mortas. No entanto, no PostgreSQL versões 16 e anteriores, o uso de memória do vacuum é limitado a 1 GB, o que é suficiente para processar aproximadamente 179 milhões de tuplas mortas em uma única passagem. Se uma tabela tiver mais tuplas mortas do que isso, o vacuum precisará fazer várias passagens pelos índices da tabela, aumentando significativamente o tempo necessário. A partir da versão 17 do PostgreSQL, não há um limite de 1 GB, e o autovacuum pode processar mais de 179 milhões de tuplas usando árvores radix.

  Um identificador de tupla tem 6 bytes de tamanho. Para estimar a memória necessária para limpar um índice de uma tabela, consulte `pg_stat_all_tables.n_dead_tup` para encontrar o número de tuplas mortas e multiplique esse número por seis para determinar a memória necessária para realizar vacuum do índice em uma única passagem. Você pode usar a seguinte consulta:

```
SELECT
    relname AS table_name,
    n_dead_tup,
    pg_size_pretty(n_dead_tup * 6) AS estimated_memory
FROM
    pg_stat_all_tables
WHERE
    relname = 'name_of_the_table';
```

- O parâmetro `autovacuum_work_mem` funciona em conjunto com o parâmetro `autovacuum_max_workers`. Cada operador entre `autovacuum_max_workers` pode usar a memória que você alocar. Se você tiver muitas tabelas pequenas, aloque mais `autovacuum_max_workers` e menos `autovacuum_work_mem`. Se você tiver tabelas grandes (com mais de 100 GB), aloque mais memória e menos processos de operadores. Você precisa ter memória suficiente alocada para ter sucesso na sua maior tabela. Portanto, garanta que a combinação de processos de operadores e memória seja igual à memória total que deseja alocar.

Reduzir a probabilidade de conclusão de IDs de transação

Em alguns casos, as configurações de grupos de parâmetros relacionadas ao autovacuum podem não ser agressivas o suficiente para evitar a conclusão de IDs de transação. Para resolver isso, o RDS para PostgreSQL oferece um mecanismo que adapta automaticamente os valores dos parâmetros de autovacuum. O autovacuum adaptável é um recurso do RDS para PostgreSQL. Uma explicação detalhada da conclusão de TransactionID encontra-se na documentação do PostgreSQL.

O autovacuum adaptável está ativado por padrão para instâncias do RDS para PostgreSQL com o parâmetro dinâmico `rds.adaptive_autovacuum` definido como ATIVADO. É altamente recomendável manter esse recurso ativado. No entanto, para desabilitar esse recurso, defina o parâmetro `rds.adaptive_autovacuum` como 0 ou OFF (desativado).

O wraparound de ID de transação ainda é possível mesmo quando o Amazon RDS ajusta os parâmetros de autovacuum. Nós encorajamos que você implemente um alarme do Amazon CloudWatch para a conclusão de IDs de transação. Para ter mais informações, consulte a postagem Implement an early warning system for transaction ID wraparound in RDS for PostgreSQL (Implementar um sistema de alertas antecipados para conclusão de IDs de transação no RDS for PostgreSQL) no Blog de banco de dados da AWS.

Com o ajuste de parâmetros de autovacuum adaptável ativado, o Amazon RDS começa a ajustar parâmetros de autovacuum quando a métrica do CloudWatch `MaximumUsedTransactionIDs` atingir o valor do parâmetro `autovacuum_freeze_max_age` ou 500.000.000, o que for maior.

O Amazon RDS continuará a ajustar os parâmetros para o autovacuum se uma tabela continuar a tendência para a conclusão de IDs de transação. Cada um desses ajustes dedica mais recursos ao autovacuum para evitar o envolvimento. O Amazon RDS atualiza os seguintes parâmetros relacionados ao autovacuum:

- autovacuum_vacuum_cost_delay
- autovacuum_vacuum_cost_limit
- `autovacuum_work_mem`
- autovacuum_naptime

O RDS modificará esses parâmetros somente se o novo valor tornar o autovacuum mais agressivo. Os parâmetros são modificados na memória na instância de banco de dados. Os valores no grupo de parâmetros não são alterados. Para visualizar as configurações atuais na memória, use o comando SQL SHOW PostgreSQL.

Quando o Amazon RDS modifica qualquer um desses parâmetros de autovacuum, ele gera um evento para a instância de banco de dados afetada. Esse evento é visível no AWS Management Console e por meio da API do Amazon RDS. Depois que a métrica `MaximumUsedTransactionIDs` do CloudWatch volta a ficar abaixo do limite, o Amazon RDS redefine os parâmetros relacionados a autovacuum na memória de volta aos valores especificados no grupo de parâmetros. Em seguida, ele gera outro evento correspondente a essa alteração.

Determinar se as tabelas no seu banco de dados precisam de vacuum

Você pode usar a consulta a seguir para mostrar o número de transações sem restrições em um banco de dados. A coluna `datfrozenxid` de uma linha `pg_database` do banco de dados é um limite inferior nos IDs de transação normais que aparecem nesse banco de dados. Essa coluna é o mínimo dos valores de `relfrozenxid` por tabela no banco de dados.

```
SELECT datname, age(datfrozenxid) FROM pg_database ORDER BY age(datfrozenxid) desc
 limit 20;
```

Por exemplo, os resultados da execução da consulta anterior podem ser os seguintes.

```
datname    | age
mydb       | 1771757888
template0  | 1721757888
template1  | 1721757888
rdsadmin   | 1694008527
postgres   | 1693881061
(5 rows)
```

Quando a idade de um banco de dados atingir 2 bilhões de IDs de transação, ocorrerá a conclusão de IDs de transação (XID), e o banco de dados se tornará somente leitura. Use essa consulta para produzir uma métrica e execute-a algumas vezes por dia. Por padrão, o autovacuum é definido para manter a idade das transações como um máximo de 200,000,000 (`autovacuum_freeze_max_age`).

Uma estratégia de monitoramento de exemplo pode ser assim:

- Defina o valor `autovacuum_freeze_max_age` como 200 milhões de transações.
- Se uma tabela atingir 500 milhões de transações sem restrições, um alarme de baixa gravidade será disparado. Este não é um valor não razoável, mas pode indicar que o autovacuum não está dando conta.
- Se uma tabela chegar a 1 bilhão, ela deverá ser tratada como um alarme que requer medidas. Em geral, convém manter as idades mais próximas de `autovacuum_freeze_max_age` por motivos de performance. Recomendamos que você investigue usando as recomendações a seguir.
- Se uma tabela atingir 1,5 bilhões de transações sem vacuum, um alarme de alta gravidade será disparado. Dependendo da rapidez com que seu banco de dados usa IDs de transação, esse alarme pode indicar que o sistema está ficando sem tempo para executar o autovacuum. Nesse caso, recomendamos que você resolva isso imediatamente.

Se uma tabela estiver ultrapassando constantemente esses limiares, modifique ainda mais seus parâmetros de autovacuum. Por padrão, usar VACUUM manualmente (que tem atrasos baseados em custos desabilitados) é mais agressivo do que usar o autovacuum padrão, mas também é mais intrusivo para o sistema como um todo.

Recomendamos o seguinte:

- Esteja ciente e ative um mecanismo de monitoramento para que você tenha conhecimento da idade das transações mais antigas.

Para obter informações sobre como criar um processo que avisa sobre a conclusão de IDs de transação, consulte a publicação do blog de banco de dados da AWS Implement an early warning system for transaction ID wraparound no Amazon RDS for PostgreSQL (Implementar um sistema de alerta antecipado para conclusão de IDs de transação no Amazon RDS for PostgreSQL).

- Para tabelas mais ocupadas, execute um congelamento de vacuum manual regularmente durante uma janela de manutenção além de depender do autovacuum. Para obter informações sobre a realização de um congelamento manual de vacuum, consulte Realização de um congelamento manual de vacuum.

Determinar quais tabelas são atualmente elegíveis para autovacuum

Muitas vezes, uma ou duas tabelas precisam de vacuum. Tabelas cujo valor de `relfrozenxid` é maior que o número de transações em `autovacuum_freeze_max_age` são sempre visadas pelo autovacuum. Caso contrário, se o número de tuplas obsoletas desde o último VACUUM exceder o limite de vacuum, um vacuum será realizado na tabela.

O limite de autovacuum é definido como:

```
Vacuum-threshold = vacuum-base-threshold + vacuum-scale-factor * number-of-tuples
```

em que o `vacuum base threshold` é `autovacuum_vacuum_threshold`, o `vacuum scale factor` é `autovacuum_vacuum_scale_factor` e o `number of tuples` é `pg_class.reltuples`.

Enquanto você estiver conectado ao seu banco de dados, execute a consulta a seguir para ver uma lista de tabelas que o autovacuum considera elegíveis para o vacuum.

```
WITH vbt AS (SELECT setting AS autovacuum_vacuum_threshold FROM
pg_settings WHERE name = 'autovacuum_vacuum_threshold'),
vsf AS (SELECT setting AS autovacuum_vacuum_scale_factor FROM
pg_settings WHERE name = 'autovacuum_vacuum_scale_factor'),
fma AS (SELECT setting AS autovacuum_freeze_max_age FROM pg_settings WHERE name =
 'autovacuum_freeze_max_age'),
sto AS (select opt_oid, split_part(setting, '=', 1) as param,
split_part(setting, '=', 2) as value from (select oid opt_oid, unnest(reloptions)
 setting from pg_class) opt)
SELECT '"'||ns.nspname||'"."'||c.relname||'"' as relation,
pg_size_pretty(pg_table_size(c.oid)) as table_size,
```

```
age(relfrozenxid) as xid_age,
coalesce(cfma.value::float, autovacuum_freeze_max_age::float)
 autovacuum_freeze_max_age,
(coalesce(cvbt.value::float, autovacuum_vacuum_threshold::float) +
coalesce(cvsf.value::float,autovacuum_vacuum_scale_factor::float) * c.reltuples)
AS autovacuum_vacuum_tuples, n_dead_tup as dead_tuples FROM
pg_class c join pg_namespace ns on ns.oid = c.relnamespace
join pg_stat_all_tables stat on stat.relid = c.oid join vbt on (1=1) join vsf on (1=1)
 join fma on (1=1)
left join sto cvbt on cvbt.param = 'autovacuum_vacuum_threshold' and c.oid =
 cvbt.opt_oid
left join sto cvsf on cvsf.param = 'autovacuum_vacuum_scale_factor' and c.oid =
 cvsf.opt_oid
left join sto cfma on cfma.param = 'autovacuum_freeze_max_age' and c.oid = cfma.opt_oid
WHERE c.relkind = 'r' and nspname <> 'pg_catalog'
AND (age(relfrozenxid) >= coalesce(cfma.value::float, autovacuum_freeze_max_age::float)
OR coalesce(cvbt.value::float, autovacuum_vacuum_threshold::float) +
coalesce(cvsf.value::float,autovacuum_vacuum_scale_factor::float) *
c.reltuples <= n_dead_tup)
ORDER BY age(relfrozenxid) DESC LIMIT 50;
```

Determinar se o autovacuum está em execução e por quanto tempo

Se você precisa realizar vacuum em uma tabela manualmente, determine se o autovacuum está sendo executado. Se estiver, talvez seja necessário ajustar parâmetros para fazê-lo funcionar mais eficientemente ou desativar o autovacuum temporariamente para poder executar VACUUM manualmente.

Use a seguinte consulta para determinar se o autovacuum está sendo executado e por quanto tempo, e se ele está esperando por outra sessão.

```
SELECT datname, usename, pid, state, wait_event, current_timestamp - xact_start AS
 xact_runtime, query
FROM pg_stat_activity
WHERE upper(query) LIKE '%VACUUM%'
ORDER BY xact_start;
```

Depois de executar a consulta, você deverá ver uma saída semelhante à seguinte:

```
 datname | usename  |  pid  | state  | wait_event |      xact_runtime       | query
```

```
 --------+----------+-------+--------+------------+------------------------
+-----------------------------------------------------------------------------------------------
 mydb    | rdsadmin | 16473 | active |            | 33 days 16:32:11.600656 |
 autovacuum: VACUUM ANALYZE public.mytable1 (to prevent wraparound)
 mydb    | rdsadmin | 22553 | active |            | 14 days 09:15:34.073141 |
 autovacuum: VACUUM ANALYZE public.mytable2 (to prevent wraparound)
 mydb    | rdsadmin | 41909 | active |            | 3 days 02:43:54.203349  |
 autovacuum: VACUUM ANALYZE public.mytable3
 mydb    | rdsadmin |   618 | active |            | 00:00:00                |
 SELECT datname, usename, pid, state, wait_event, current_timestamp - xact_start AS
 xact_runtime, query+
         |          |       |        |            |                         | FROM
 pg_stat_activity
           +
         |          |       |        |            |                         | WHERE
 query like '%VACUUM%'
          +
         |          |       |        |            |                         | ORDER BY
 xact_start;
       +
```

Vários problemas podem causar uma sessão de autovacuum de longa execução (ou seja, vários dias de duração). O problema mais comum é que o valor do seu parâmetro `maintenance_work_mem` está muito baixo para o tamanho da tabela ou a taxa de atualizações.

Recomendamos o uso da fórmula a seguir para definir o valor do parâmetro `maintenance_work_mem`.

```
GREATEST({DBInstanceClassMemory/63963136*1024},65536)
```

Sessões de autovacuum em execução por pouco tempo também podem indicar problemas:

- Isso pode indicar que não há `autovacuum_max_workers` suficientes para sua workload. Neste caso, você precisa indicar o número de operadores.
- Pode indicar que existe uma corrupção de índice (o autovacuum trava e reinicia na mesma relação, mas não faz nenhum progresso). Nesse caso, execute um `vacuum freeze verbose` *`table`* manual para ver a causa exata.

Realização de um congelamento manual de vacuum

Você pode querer realizar um vacuum manual em uma tabela que tenha um processo de vacuum já em execução. Isso é útil se você identificou uma tabela com uma idade XID próxima de 2 bilhões de transações (ou acima de qualquer limite que você esteja monitorando).

As etapas a seguir são uma orientação com diversas variações no processo. Por exemplo, durante testes, suponha que você perceba que o valor do parâmetro `maintenance_work_mem` está muito baixo e que precisa tomar medidas imediatas em uma tabela. No entanto, talvez você não queira devolver a instância no momento. Usando as consultas nas seções anteriores, você determina qual tabela é o problema e observa uma sessão de autovacuum de longa execução. Você sabe que precisa alterar a configuração do parâmetro `maintenance_work_mem`, mas também precisa tomar medidas imediatas e evacuar a tabela em questão. O procedimento a seguir mostra o que fazer nessa situação.

Para executar manualmente um congelamento de vacuum

1. Abra duas sessões no banco de dados que contém a tabela em que você deseja realizar vacuum. Para a segunda sessão, use "screen" ou outro utilitário que mantenha a sessão se a conexão for encerrada.
2. Na sessão um, obtenha o “Process ID” (PID – ID de processo) da sessão de autovacuum em execução na tabela.

   Execute a seguinte consulta para obter o PID da sessão de autovacuum.

   ```
   SELECT datname, usename, pid, current_timestamp - xact_start
   AS xact_runtime, query
   FROM pg_stat_activity WHERE upper(query) LIKE '%VACUUM%' ORDER BY
   xact_start;
   ```
3. Na sessão dois, calcule a quantidade de memória necessária para essa operação. Neste exemplo, determinamos que podemos usar até 2 GB de memória para essa operação e, portanto, definimos `maintenance_work_mem` da sessão atual como 2 GB.

   ```
   SET maintenance_work_mem='2 GB';
   SET
   ```
4. Na sessão dois, emita um comando `vacuum freeze verbose` para a tabela. A configuração detalhada é útil porque, embora não haja um relatório de andamento para isso no PostgreSQL, você pode ver as atividades.

```
\timing on
Timing is on.
vacuum freeze verbose pgbench_branches;
```

```
INFO:  vacuuming "public.pgbench_branches"
INFO:  index "pgbench_branches_pkey" now contains 50 row versions in 2 pages
DETAIL:  0 index row versions were removed.
0 index pages have been deleted, 0 are currently reusable.
CPU 0.00s/0.00u sec elapsed 0.00 sec.
INFO:  index "pgbench_branches_test_index" now contains 50 row versions in 2 pages
DETAIL:  0 index row versions were removed.
0 index pages have been deleted, 0 are currently reusable.
CPU 0.00s/0.00u sec elapsed 0.00 sec.
INFO:  "pgbench_branches": found 0 removable, 50 nonremovable row versions
     in 43 out of 43 pages
DETAIL:  0 dead row versions cannot be removed yet.
There were 9347 unused item pointers.
0 pages are entirely empty.
CPU 0.00s/0.00u sec elapsed 0.00 sec.
VACUUM
Time: 2.765 ms
```

5. Na sessão um, se o autovacuum estava bloqueando a sessão de vacuum, `pg_stat_activity` mostrará que a espera é T para a sessão de vacuum. Nesse caso, encerre o processo de autovacuum da maneira a seguir.

```
SELECT pg_terminate_backend('the_pid');
```

> **Note**
>
> Algumas versões anteriores do Amazon RDS não podem encerrar um processo de autovacuum usando o comando anterior e falham com o seguinte erro: `ERROR: 42501: must be a superuser to terminate superuser process LOCATION: pg_terminate_backend, signalfuncs.c:227`.

Nesse ponto, sua sessão começa. O autovacuum é reiniciado imediatamente, pois essa tabela é provavelmente a mais alta em sua lista de trabalho.

6. Inicie seu comando `vacuum freeze verbose` na sessão dois e depois termine o processo de autovacuum na sessão um.

Reindexação de uma tabela quando o autovacuum está em execução

Se um índice se tornar corrompido, o autovacuum continuará processando a tabela e falhará. Se você tentar um vacuum manual nessa situação, receberá uma mensagem de erro semelhante à mostrada a seguir.

```
postgres=>  vacuum freeze pgbench_branches;
ERROR: index "pgbench_branches_test_index" contains unexpected
   zero page at block 30521
HINT: Please REINDEX it.
```

Quando o índice está corrompido e o autovacuum está tentando ser executado na tabela, você lutará com uma sessão de autovacuum já em execução. Ao emitir um comando REINDEX, você remove um bloqueio exclusivo na tabela. As operações de gravação são bloqueadas, bem como as operações de leitura que usam esse índice específico.

Para reindexar uma tabela quando o autovacuum está em execução nela

1. Abra duas sessões no banco de dados que contém a tabela em que você deseja realizar vacuum. Para a segunda sessão, use "screen" ou outro utilitário que mantenha a sessão se a conexão for encerrada.
2. Na sessão um, obtenha o PID da sessão de autovacuum em execução na tabela.

   Execute a seguinte consulta para obter o PID da sessão de autovacuum.

   ```
   SELECT datname, usename, pid, current_timestamp - xact_start
   AS xact_runtime, query
   FROM pg_stat_activity WHERE upper(query) like '%VACUUM%' ORDER BY
   xact_start;
   ```

3. Na segunda sessão, emita o comando reindex.

   ```
   \timing on
   Timing is on.
   reindex index pgbench_branches_test_index;
   REINDEX
   ```

```
  Time: 9.966 ms
```

4. Na sessão um, se o autovacuum estava bloqueando o processo, você verá em `pg_stat_activity` que a espera é "T" para a sua sessão de vacuum. Nesse caso, você encerra o processo de autovacuum.

   ```
   SELECT pg_terminate_backend('the_pid');
   ```

   Nesse ponto, sua sessão começa. É importante observar que o autovacuum é reiniciado imediatamente, pois essa tabela é provavelmente a mais alta em sua lista de trabalho.
5. Inicie seu comando na sessão dois e depois termine o processo de autovacuum na sessão um.

Gerenciar o autovacuum com grandes índices

Como parte de sua operação, o autovacuum executa várias fases de aspiração ao trabalhar em uma tabela. Antes de a tabela ser limpa, todos os seus índices são aspirados primeiro. Ao remover vários índices grandes, essa fase consome uma quantidade significativa de tempo e recursos. Portanto, como prática recomendada, controle o número de índices em uma tabela e elimine os índices não usados.

Para esse processo, primeiro confira o tamanho geral do índice. Depois, determine se há índices possivelmente não usados que podem ser removidos conforme mostrado nos exemplos a seguir.

Como conferir o tamanho da tabela e os respectivos índices

```
postgres=> select pg_size_pretty(pg_relation_size('pgbench_accounts'));
pg_size_pretty
6404 MB
(1 row)
```

```
postgres=> select pg_size_pretty(pg_indexes_size('pgbench_accounts'));
pg_size_pretty
11 GB
(1 row)
```

Neste exemplo, o tamanho dos índices é maior do que a tabela. Essa diferença pode causar problemas de performance, pois os índices estão sobrecarregados ou não são usados, o que afeta as operações de autovacuum e de inserção.

Como conferir índices não usados

Usando a visualização pg_stat_user_indexes, você pode conferir com que frequência um índice é usado com a coluna idx_scan. No exemplo a seguir, os índices não usados têm o valor idx_scan de 0.

```
postgres=> select * from pg_stat_user_indexes where relname = 'pgbench_accounts' order
 by idx_scan desc;

relid  | indexrelid | schemaname | relname          | indexrelname          | idx_scan
 | idx_tup_read | idx_tup_fetch
-------+------------+------------+------------------+-----------------------+----------
+--------------+---------------
16433  | 16454      | public     | pgbench_accounts | index_f               | 6
 | 6            | 0
16433  | 16450      | public     | pgbench_accounts | index_b               | 3
 | 199999       | 0
16433  | 16447      | public     | pgbench_accounts | pgbench_accounts_pkey | 0
 | 0            | 0
16433  | 16452      | public     | pgbench_accounts | index_d               | 0
 | 0            | 0
16433  | 16453      | public     | pgbench_accounts | index_e               | 0
 | 0            | 0
16433  | 16451      | public     | pgbench_accounts | index_c               | 0
 | 0            | 0
16433  | 16449      | public     | pgbench_accounts | index_a               | 0
 | 0            | 0
(7 rows)
```

```
postgres=> select schemaname, relname, indexrelname, idx_scan from pg_stat_user_indexes
 where relname = 'pgbench_accounts' order by idx_scan desc;

schemaname  | relname          | indexrelname          | idx_scan
------------+------------------+-----------------------+----------
public      | pgbench_accounts | index_f               | 6
public      | pgbench_accounts | index_b               | 3
public      | pgbench_accounts | pgbench_accounts_pkey | 0
public      | pgbench_accounts | index_d               | 0
public      | pgbench_accounts | index_e               | 0
public      | pgbench_accounts | index_c               | 0
public      | pgbench_accounts | index_a               | 0
(7 rows)
```

> **Note**
>
> Essas estatísticas são incrementais a partir do momento em que as estatísticas são redefinidas. Suponha que você tenha um índice usado apenas no final de um trimestre comercial ou apenas para um relatório específico. É possível que esse índice não tenha sido usado desde que as estatísticas foram redefinidas. Para ter mais informações, consulte Funções de estatística. Os índices usados para impor a exclusividade não terão verificações realizadas e não devem ser identificados como índices não usados. Para identificar os índices não usados, você deve ter um conhecimento profundo da aplicação e das respectivas consultas.

Para conferir quando as estatísticas foram redefinidas pela última vez em um banco de dados, use `pg_stat_database`

```
postgres=> select datname, stats_reset from pg_stat_database where datname =
 'postgres';

datname   | stats_reset
----------+-------------------------------
postgres  | 2022-11-17 08:58:11.427224+00
(1 row)
```

Aspirar uma tabela o mais rápido possível

RDS para PostgreSQL 12 e posterior

Se você tiver muitos índices em uma tabela grande, a instância de banco de dados poderá estar se aproximando do encapsulamento de ID (XID), que é quando o contador XID chega a zero. Se não for conferida, essa situação poderá ocasionar perda de dados. No entanto, você pode aspirar rapidamente a tabela sem limpar os índices. No RDS para PostgreSQL 12 e posterior, você pode usar VACUUM com a cláusula `INDEX_CLEANUP`.

```
postgres=> VACUUM (INDEX_CLEANUP FALSE, VERBOSE TRUE) pgbench_accounts;

INFO: vacuuming "public.pgbench_accounts"
```

```
INFO: table "pgbench_accounts": found 0 removable, 8 nonremovable row versions in 1 out
 of 819673 pages
DETAIL: 0 dead row versions cannot be removed yet, oldest xmin: 7517
Skipped 0 pages due to buffer pins, 0 frozen pages.
CPU: user: 0.01 s, system: 0.00 s, elapsed: 0.01 s.
```

Se uma sessão de autovacuum já estiver em execução, você deverá encerrá-la para iniciar a aspiração manual. Para ter informações sobre a realização de um congelamento manual de aspiração, consulte Realização de um congelamento manual de vacuum.

> **Note**
>
> Ignorar a limpeza do índice regularmente causa sobrecarga no índice, o que diminui a performance geral da verificação. O índice retém as linhas inativas e a tabela retém os ponteiros das linhas inativas. Consequentemente, `pg_stat_all_tables.n_dead_tup` aumenta até que o autovacuum ou um VACUUM manual com limpeza de índice seja executado. Como prática recomendada, use esse procedimento somente para evitar o encapsulamento de ID.

RDS para PostgreSQL 11 e versões mais antigas

No entanto, no RDS para PostgreSQL 11 e versões anteriores, a única maneira de permitir que a aspiração seja concluída mais rapidamente é reduzir o número de índices em uma tabela. A eliminação de um índice pode afetar os planos de consulta. Recomendamos que você elimine primeiro os índices não usados e, depois, descarte os índices quando o encapsulamento XID estiver muito próximo. Depois que o processo de aspiração for concluído, você poderá recriar esses índices.

Outros parâmetros que afetam o autovacuum

A consulta a seguir mostra os valores de alguns dos parâmetros que afetam diretamente o autovacuum e seu comportamento. Os parâmetros de autovacuum são descritos inteiramente na documentação do PostgreSQL.

```
SELECT name, setting, unit, short_desc
FROM pg_settings
WHERE name IN (
'autovacuum_max_workers',
```

```
'autovacuum_analyze_scale_factor',
'autovacuum_naptime',
'autovacuum_analyze_threshold',
'autovacuum_analyze_scale_factor',
'autovacuum_vacuum_threshold',
'autovacuum_vacuum_scale_factor',
'autovacuum_vacuum_threshold',
'autovacuum_vacuum_cost_delay',
'autovacuum_vacuum_cost_limit',
'vacuum_cost_limit',
'autovacuum_freeze_max_age',
'maintenance_work_mem',
'vacuum_freeze_min_age');
```

Enquanto todos estes afetem o autovacuum, alguns dos mais importantes são:

- maintenance_work_mem
- autovacuum_freeze_max_age
- autovacuum_max_workers
- autovacuum_vacuum_cost_delay
- autovacuum_vacuum_cost_limit

Definir parâmetros de autovacuum em nível de tabela

Você pode definir parâmetros de armazenamento relacionados ao autovacuum em nível de tabela, o que pode ser melhor do que alterar o comportamento do banco de dados inteiro. Para tabelas grandes, talvez você precise definir configurações agressivas e talvez você não queira que o autovacuum se comporte dessa maneira para todas as tabelas.

A consulta a seguir mostra quais tabelas atualmente têm opções em nível de tabela.

```
SELECT relname, reloptions
FROM pg_class
WHERE reloptions IS NOT null;
```

Um exemplo em que isso pode ser útil é em tabelas que são muito maiores do que o resto das suas tabelas. Suponha que você tenha uma tabela de 300 GB e outras 30 tabelas com menos de 1 GB. Nesse caso, você pode definir alguns parâmetros específicos para a sua tabela grande, para não alterar o comportamento de todo o sistema.

```
ALTER TABLE mytable set (autovacuum_vacuum_cost_delay=0);
```

Fazer isso desativará o atraso de autovacuum baseado em custos para essa tabela, mas aumentará o uso de recursos no seu sistema. Normalmente, o autovacuum faz uma pausa durante `autovacuum_vacuum_cost_delay` sempre que `autovacuum_cost_limit` é alcançado. Para obtert mais detalhes, consulte a documentação do PostgreSQL sobre vacuum baseado no custo.

Registrar atividades do autovacuum e do vacuum em log

Informações sobre atividades de autovacuum são enviadas para `postgresql.log` com base no nível especificado no parâmetro `rds.force_autovacuum_logging_level`. A seguir estão os valores permitidos para esse parâmetro e as versões do PostgreSQL para as quais esse valor é a configuração padrão:

- `disabled` (PostgreSQL 10, PostgreSQL 9.6)
- `debug5`, `debug4`, `debug3`, `debug2`, `debug1`
- `info` (PostgreSQL 12, PostgreSQL 11)
- `notice`
- `warning` (PostgreSQL 13 e superior)
- `error`, log, `fatal`, `panic`

A configuração `rds.force_autovacuum_logging_level` funciona com o parâmetro `log_autovacuum_min_duration`. O valor do parâmetro `log_autovacuum_min_duration` é o limite (em milissegundos) acima do qual as ações autovacuum são registradas em log. Uma configuração `-1` não registra nada em log, enquanto uma configuração 0 registra todas as ações em log. Como ocorre com `rds.force_autovacuum_logging_level`, os valores padrão de `log_autovacuum_min_duration` são dependentes da versão, da seguinte maneira:

- `10000 ms`: PostgreSQL 14, PostgreSQL 13, PostgreSQL 12 e PostgreSQL 11
- `(empty)`: nenhum valor padrão para PostgreSQL 10 e PostgreSQL 9.6

Recomendamos que você defina `rds.force_autovacuum_logging_level` como `WARNING`. Também recomendamos definir `log_autovacuum_min_duration` para um valor entre 1000 e 5000. Uma configuração 5000 registra em log a atividade que leva mais de 5.000 milissegundos. Qualquer configuração diferente de -1 também registrará mensagens se a ação autovacuum for ignorada devido a um bloqueio conflitante ou relações descartadas simultaneamente. Para ter mais

informações, consulte Automatic Vacuuming (Aplicação automática do vacuum) na documentação do PostgreSQL.

Para solucionar problemas, você pode alterar o parâmetro `rds.force_autovacuum_logging_level` para um dos níveis de depuração, de `debug1` a `debug5`, para obter informações mais detalhadas. Recomendamos que você use as configurações de depuração por curtos períodos e apenas para fins de solução de problemas. Para saber mais, consulte When to log (Quando registrar em log) na documentação do PostgreSQL.

> **Note**
>
> O PostgreSQL permite que a conta `rds_superuser` visualize sessões de autovacuum em `pg_stat_activity`. Por exemplo, você pode identificar e encerrar uma sessão de autovacuum que esteja bloqueando a execução de um comando, ou executando de forma mais lenta do que um comando de vacuum emitido manualmente.

Noções básicas sobre o comportamento do autovacuum com bancos de dados inválidos

Um novo valor `-2` é introduzido na coluna `datconnlimit` do catálogo `pg_database` para indicar bancos de dados que foram interrompidos no meio da operação DROP DATABASE como inválidos.

Esse novo valor está disponível nas seguintes versões do RDS para PostgreSQL:

- 15.4 e todas as versões posteriores
- 14.9 e versões posteriores
- 13.12 e versões posteriores
- 12.16 e versões posteriores
- 11.21 e versões posteriores

Bancos de dados inválidos não afetam a capacidade do autovacuum de congelar a funcionalidade de bancos de dados válidos. O Autovacuum ignora bancos de dados inválidos. Consequentemente, as operações regulares de limpeza continuarão funcionando de forma adequada e eficiente em todos os bancos de dados válidos no ambiente do PostgreSQL.

Tópicos

- ID da transação de monitoramento
- Ajustar a consulta de monitoramento
- Resolver o problema de banco de dados inválido

ID da transação de monitoramento

A função `age(datfrozenxid)` geralmente é usada para monitorar a idade do ID de transação (XID) dos bancos de dados para evitar o encapsulamento de IDs de transação.

Como bancos de dados inválidos são excluídos do autovacuum, seu contador de ID de transação (XID) pode atingir o valor máximo de `2 billion`, ser encapsulado em `- 2 billion` e continuar esse ciclo indefinidamente. Uma consulta típica para monitorar o encapsulamento de IDs de transação pode ser:

```
SELECT max(age(datfrozenxid)) FROM pg_database;
```

No entanto, com a introdução do valor -2 para `datconnlimit`, bancos de dados inválidos podem distorcer os resultados dessa consulta. Como esses bancos de dados não são válidos e não devem fazer parte das verificações de manutenção regulares, eles podem causar falsos positivos, levando você a acreditar que o `age(datfrozenxid)` é maior do que realmente é.

Ajustar a consulta de monitoramento

Para garantir um monitoramento preciso, você deve ajustar sua consulta de monitoramento para excluir bancos de dados inválidos. Siga esta consulta recomendada:

```
SELECT
    max(age(datfrozenxid))
FROM
    pg_database
WHERE
    datconnlimit <> -2;
```

Essa consulta garante que somente bancos de dados válidos sejam considerados no cálculo `age(datfrozenxid)`, fornecendo um reflexo real da idade do ID da transação no ambiente do PostgreSQL.

Resolver o problema de banco de dados inválido

Ao tentar se conectar a um banco de dados inválido, talvez seja exibida uma mensagem de erro semelhante à seguinte:

```
postgres=> \c db1
connection to server at "mydb.xxxxxxxxxx.us-west-2.rds.amazonaws.com" (xx.xx.xx.xxx),
 port xxxx failed: FATAL:  cannot connect to invalid database "db1"
HINT:  Use DROP DATABASE to drop invalid databases.
Previous connection kept
```

Além disso, se o parâmetro `log_min_messages` estiver definido como `DEBUG2` ou posterior, você poderá observar as seguintes entradas de log indicando que o processo de autovacuum está ignorando o banco de dados inválido:

```
2024-07-30 05:59:00 UTC::@:[32000]:DEBUG:  autovacuum: skipping invalid database "db6"
2024-07-30 05:59:00 UTC::@:[32000]:DEBUG:  autovacuum: skipping invalid database "db1"
```

Para resolver o problema, siga a `HINT` fornecida durante a tentativa de conexão. Conecte-se a qualquer banco de dados válido usando sua conta principal do RDS ou uma conta de banco de dados com o perfil `rds_superuser` e elimine bancos de dados inválidos.

```
SELECT
    'DROP DATABASE ' || quote_ident(datname) || ';'
FROM
    pg_database
WHERE
    datconnlimit = -2 \gexec
```

Identificar e resolver bloqueadores de limpeza agressivos no RDS para PostgreSQL

No PostgreSQL, a limpeza é vital para garantir a integridade do banco de dados, pois recupera o armazenamento e evita problemas de conclusão do ID da transação. No entanto, há momentos em

que a limpeza pode ser impedida de operar conforme desejado, o que pode resultar em degradação do desempenho e sobrecarga do armazenamento e até mesmo afetar a disponibilidade da instância de banco de dados pelo wraparound de ID de transação. Portanto, identificar e resolver esses problemas é essencial para o desempenho e a disponibilidade ideais do banco de dados. Leia Understanding autovacuum in Amazon RDS for PostgreSQL environments para saber mais sobre o autovacuum.

A função `postgres_get_av_diag()` ajuda a identificar problemas que impedem ou atrasam o progresso da limpeza agressiva. São fornecidas sugestões, que podem incluir comandos para resolver o problema onde ele é identificável ou orientações para diagnósticos adicionais quando o problema não é identificável. Bloqueadores de limpeza agressiva são relatados quando a idade excede o limite de autovacuum adaptativo do RDS de 500 milhões de IDs de transação.

Qual é a idade do ID da transação?

A função `age()` para IDs de transação calcula o número de transações que ocorreram desde o ID de transação não congelado mais antigo de um banco de dados (`pg_database.datfrozenxid`) ou tabela (`pg_class.relfrozenxid`). Esse valor indica a atividade do banco de dados desde a última operação de limpeza agressiva e destaca a provável workload para os próximos processos de VACUUM.

O que é uma limpeza agressiva?

Uma operação VACUUM agressiva conduz uma varredura abrangente de todas as páginas em uma tabela, incluindo aquelas normalmente ignoradas durante operações VACUUM regulares. Essa verificação completa visa "congelar" os IDs de transação que se aproximam de sua idade máxima, evitando efetivamente uma situação conhecida como conclusão de ID de transação.

Para que `postgres_get_av_diag()` reporte bloqueadores, o bloqueador deve ter pelo menos 500 milhões de transações de idade.

Tópicos

- Instalar ferramentas de monitoramento e diagnóstico de autovacuum no RDS para PostgreSQL
- Funções de postgres_get_av_diag() no RDS para PostgreSQL
- Solucionar bloqueadores de limpeza identificáveis no RDS para PostgreSQL
- Solucionar bloqueadores de limpeza não identificáveis no RDS para PostgreSQL
- Resolver problemas de desempenho de limpeza no RDS para PostgreSQL
- Explicação das mensagens de AVISO no RDS para PostgreSQL

Instalar ferramentas de monitoramento e diagnóstico de autovacuum no RDS para PostgreSQL

No momento, a função `postgres_get_av_diag()` está disponível nas seguintes versões do RDS para PostgreSQL:

- 17.2 e versões 17 posteriores
- 16.7 e versões 16 posteriores
- 15.11 e versões 15 posteriores
- 14.16 e versões 14 posteriores
- 13.19 e versões 13 posteriores

Para usar `postgres_get_av_diag()`, crie a extensão `rds_tools`.

```
postgres=> CREATE EXTENSION rds_tools ;
CREATE EXTENSION
```

Para verificar se a extensão está instalada.

```
postgres=> \dx rds_tools
             List of installed extensions
   Name    | Version |  Schema   |                    Description
 ----------+---------+-----------
+-------------------------------------------------------------
 rds_tools |   1.8   | rds_tools | miscellaneous administrative functions for RDS
 PostgreSQL
 1 row
```

Verifique se a função foi criada.

```
postgres=> SELECT
    proname function_name,
    pronamespace::regnamespace function_schema,
    proowner::regrole function_owner
FROM
    pg_proc
WHERE
    proname = 'postgres_get_av_diag';
    function_name     | function_schema | function_owner
----------------------+-----------------+----------------
```

```
 postgres_get_av_diag | rds_tools       | rds_superuser
(1 row)
```

Funções de postgres_get_av_diag() no RDS para PostgreSQL

A função `postgres_get_av_diag()` recupera informações de diagnóstico sobre processos de autovacuum que estão bloqueando ou atrasando em um banco de dados do RDS para PostgreSQL. A consulta precisa ser executada no banco de dados com o ID de transação mais antigo para obter resultados precisos. Para obter mais informações sobre como usar o banco de dados com o ID de transação mais antigo, consulte Não conectado ao banco de dados com o ID de transação mais antigo

```
SELECT
    blocker,
    DATABASE,
    blocker_identifier,
    wait_event,
    TO_CHAR(autovacuum_lagging_by, 'FM9,999,999,999') AS autovacuum_lagging_by,
    suggestion,
    suggested_action
FROM (
    SELECT
        *
    FROM
        rds_tools.postgres_get_av_diag ()
    ORDER BY
        autovacuum_lagging_by DESC) q;
```

A função `postgres_get_av_diag()` retorna uma tabela com as informações a seguir:

blocker

Especifica a categoria da atividade do banco de dados que está bloqueando o vacuum.

- Instrução ativa
- Ocioso na transação
- Transação preparada
- Slot de replicação lógica
- Réplica de leitura com slot de replicação física
- Réplica de leitura com replicação de streaming

- Tabelas temporárias

banco de dados

Especifica o nome do banco de dados, quando aplicável e suportado. Este é o banco de dados no qual a atividade está em andamento e bloqueia ou bloqueará o autovacuum. Este é o banco de dados ao qual você precisa se conectar e tomar ação.

blocker_identifier

Especifica o identificador da atividade que está bloqueando ou bloqueará o autovacuum. O identificador pode ser um ID de processo junto com uma instrução SQL, uma transação preparada, um endereço IP de uma réplica de leitura e o nome do slot de replicação, seja lógico ou físico.

wait_event

Especifica o evento de espera da sessão de bloqueio e é aplicável para os seguintes bloqueadores:

- Instrução ativa
- Ocioso na transação

autovacuum_lagging_by

Especifica o número de transações que o autovacuum está atrasado no trabalho pendente por categoria.

suggestion

Especifica sugestões para resolver o bloqueador. Essas instruções incluem o nome do banco de dados no qual a atividade existe, quando aplicável, o ID do processo (PID) da sessão, quando aplicável, e a ação a ser tomada.

suggested_action

Sugere a ação que precisa ser tomada para resolver o bloqueador.

Solucionar bloqueadores de limpeza identificáveis no RDS para PostgreSQL

O Autovacuum realiza limpezas agressivas e reduz a idade dos IDs de transação para abaixo do limite especificado pelo parâmetro `autovacuum_freeze_max_age` de sua instância do RDS. É possível rastrear essa idade usando a métrica `MaximumUsedTransactionIDs` do Amazon CloudWatch.

Para encontrar a configuração de `autovacuum_freeze_max_age` (que tem um padrão de 200 milhões de IDs de transação) da sua instância do Amazon RDS, use a seguinte consulta:

```
SELECT
    TO_CHAR(setting::bigint, 'FM9,999,999,999') autovacuum_freeze_max_age
FROM
    pg_settings
WHERE
    name = 'autovacuum_freeze_max_age';
```

Observe que `postgres_get_av_diag()` só verifica se há bloqueadores de limpeza agressiva quando a idade excede o limite de 500 milhões de IDs de transação do autovacuum adaptativo do Amazon RDS. Para que `postgres_get_av_diag()` detecte bloqueadores, o bloqueador deve ter pelo menos 500 milhões de transações de idade.

A função `postgres_get_av_diag()` identifica os seguintes tipos de bloqueadores:

Tópicos

- Instrução ativa
- Ocioso na transação
- Transação preparada
- Slot de replicação lógica
- Réplicas de leitura
- Tabelas temporárias

Instrução ativa

No PostgreSQL, uma instrução ativa é uma instrução SQL que está sendo executada atualmente pelo banco de dados. Isso inclui consultas, transações ou quaisquer operações em andamento. Ao monitorar por meio do `pg_stat_activity`, a coluna de estado indica que o processo com o PID correspondente está ativo.

A função `postgres_get_av_diag()` exibe uma saída semelhante à saída a seguir quando identifica uma instrução que é uma instrução ativa.

```
blocker              | Active statement
database             | my_database
```

```
blocker_identifier    | SELECT pg_sleep(20000);
wait_event            | Timeout:PgSleep
autovacuum_lagging_by | 568,600,871
suggestion            | Connect to database "my_database", review carefully and you
 may consider terminating the process using suggested_action. For more information, see
 Working with PostgreSQL autovacuum in the Amazon RDS User Guide.
suggested_action      | {"SELECT pg_terminate_backend (29621);"}
```

Ação sugerida

Seguindo as orientações na coluna `suggestion`, o usuário pode se conectar ao banco de dados em que a instrução ativa está presente e, conforme especificado na coluna `suggested_action`, é recomendável analisar cuidadosamente a opção de encerrar a sessão. Se o encerramento for seguro, use a função `pg_terminate_backend()` para encerrar a sessão. Essa ação pode ser executada por um administrador (como a conta mestra do RDS) ou por um usuário com o privilégio `pg_terminate_backend()` necessário.

> **Warning**
>
> Uma sessão encerrada reverterá (`ROLLBACK`) as alterações feitas. Dependendo de seus requisitos, pode ser necessário executar a instrução novamente. No entanto, é recomendável fazer isso somente após o processo do autovacuum terminar a operação de limpeza agressiva.

Ocioso na transação

Uma instrução ociosa na transação se refere a qualquer sessão que abriu uma transação explícita (como emitir uma instrução `BEGIN`), realizou algum trabalho e agora está esperando que o cliente passe mais trabalho ou sinalize o final da transação emitindo um `COMMIT`, `ROLLBACK` ou `END` (o que resultaria em um `COMMIT` implícito).

A função `postgres_get_av_diag()` exibe uma saída semelhante à saída a seguir quando identifica uma instrução `idle in transaction` como um bloqueador.

```
blocker               | idle in transaction
database              | my_database
blocker_identifier    | INSERT INTO tt SELECT * FROM tt;
wait_event            | Client:ClientRead
autovacuum_lagging_by | 1,237,201,759
```

```
suggestion            | Connect to database "my_database", review carefully and you
 may consider terminating the process using suggested_action. For more information, see
 Working with PostgreSQL autovacuum in the Amazon RDS User Guide.
suggested_action      | {"SELECT pg_terminate_backend (28438);"}
```

Ação sugerida

Conforme indicado na coluna `suggestion`, conecte-se ao banco de dados em que a sessão ociosa na transação está presente e encerre a sessão usando a função `pg_terminate_backend()`. O usuário pode ser seu usuário administrador (conta mestra do RDS) ou um usuário com o privilégio `pg_terminate_backend()`.

> **Warning**
>
> Uma sessão encerrada reverterá (`ROLLBACK`) as alterações feitas. Dependendo de seus requisitos, pode ser necessário executar a instrução novamente. No entanto, é recomendável fazer isso somente após o processo do autovacuum terminar a operação de limpeza agressiva.

Transação preparada

O PostgreSQL permite transações que fazem parte de uma estratégia de confirmação bifásica chamada transações preparadas. Elas são habilitadas ao definir o parâmetro `max_prepared_transactions` como um valor diferente de zero. As transações preparadas são projetadas para garantir que uma transação seja durável e permaneça disponível mesmo após falhas no banco de dados, reinicializações ou desconexões do cliente. Como as transações regulares, elas recebem um ID de transação e podem afetar o autovacuum. Se deixado em um estado preparado, o autovacuum não realiza o congelamento e isso pode levar à conclusão do ID da transação.

Quando as transações são deixadas no estado de preparação indefinidamente, sem serem resolvidas por um gerenciador de transações, elas se tornam transações preparadas órfãs. A única maneira de corrigir isso é confirmar ou reverter a transação usando os comandos `COMMIT PREPARED` ou `ROLLBACK PREPARED`, respectivamente.

> **Note**
>
> Esteja ciente de que um backup feito durante uma transação preparada ainda conterá essa transação após a restauração. Consulte as informações a seguir sobre como localizar e fechar essas transações.

A função `postgres_get_av_diag()` exibe a saída a seguir quando identifica um bloqueador que é uma transação preparada.

```
blocker               | Prepared transaction
database              | my_database
blocker_identifier    | myptx
wait_event            | Not applicable
autovacuum_lagging_by | 1,805,802,632
suggestion            | Connect to database "my_database" and consider either COMMIT
 or ROLLBACK the prepared transaction using suggested_action. For more information, see
 Working with PostgreSQL autovacuum in the Amazon RDS User Guide.
suggested_action      | {"COMMIT PREPARED 'myptx';",[OR],"ROLLBACK PREPARED 'myptx';"}
```

Ação sugerida

Conforme mencionado na coluna de sugestões, conecte-se ao banco de dados em que a transação preparada está localizada. Com base na coluna `suggested_action`, analise cuidadosamente se deseja executar `COMMIT` ou `ROLLBACK`, e então execute a ação apropriada.

Para monitorar transações preparadas em geral, o PostgreSQL oferece uma visualização de catálogo chamada `pg_prepared_xacts`. Use a consulta a seguir para encontrar transações preparadas.

```
SELECT
    gid,
    prepared,
    owner,
    database,
    transaction AS oldest_xmin
FROM
    pg_prepared_xacts
ORDER BY
    age(transaction) DESC;
```

Slot de replicação lógica

O objetivo de um slot de replicação é manter as alterações não consumidas até que elas sejam replicadas em um servidor de destino. Consulte a página sobre replicação lógica do PostgreSQL para obter mais informações.

Existem dois tipos de slots de replicação lógica.

Slots de replicação lógica inativa

Quando a replicação é encerrada, os logs de transações não consumidos não podem ser removidos e o slot de replicação fica inativo. Embora um slot de replicação lógica inativo não seja usado atualmente por um assinante, ele permanece no servidor, levando à retenção de arquivos WAL e impedindo a remoção de logs de transações antigos. Isso pode aumentar o uso do disco e, especificamente, impedir que o autovacuum limpe as tabelas internas do catálogo, pois o sistema deve evitar que as informações do LSN sejam sobrescritas. Se não for resolvido, esse problema pode resultar em inchaço do catálogo, degradação do desempenho e aumento do risco de conclusão da limpeza, potencialmente causando tempo de inatividade das transações.

Slots de replicação lógica ativos, mas lentos

Às vezes, a remoção de tuplas inativas do catálogo é adiada devido à degradação do desempenho da replicação lógica. Esse atraso na replicação retarda a atualização do `catalog_xmin` e pode causar inchaço do catálogo e conclusão da limpeza.

A função `postgres_get_av_diag()` exibe uma saída semelhante à saída a seguir quando encontra um slot de replicação lógica como um bloqueador.

```
blocker               | Logical replication slot
database              | my_database
blocker_identifier    | slot1
wait_event            | Not applicable
autovacuum_lagging_by | 1,940,103,068
suggestion            | Ensure replication is active and resolve any lag for the slot
 if active. If inactive, consider dropping it using the command in suggested_action.
 For more information, see Working with PostgreSQL autovacuum in the Amazon RDS User
 Guide.
suggested_action      | {"SELECT pg_drop_replication_slot('slot1') FROM
 pg_replication_slots WHERE active = 'f';"}
```

Ação sugerida

Para resolver esse problema, verifique se há problemas com o esquema ou os dados de destino na configuração de replicação que possam estar encerrando o processo de aplicação. Os motivos mais comuns são:

- Colunas ausentes
- Tipos de dados incompatíveis
- Incompatibilidade de dados
- Tabela ausente

Se o problema estiver relacionado a questões de infraestrutura:

- Problemas de rede: Como resolver problemas com um banco de dados do Amazon RDS que está em um estado de rede incompatível?.
- O banco de dados ou a instância de banco de dados não está disponível devido aos seguintes motivos:
  - A instância da réplica está sem armazenamento: consulte Como resolver problemas que ocorrem quando as instâncias de banco de dados do Amazon RDS ficam sem armazenamento? para obter informações sobre como adicionar armazenamento.
  - Parâmetros incompatíveis: consulte Como corrijo uma instância de banco de dados do Amazon RDS que está presa no status de parâmetros incompatíveis? para obter mais informações sobre como resolver o problema.

Se sua instância estiver fora da rede da AWS ou no AWS EC2, consulte seu administrador sobre como resolver os problemas de disponibilidade ou problemas relacionados à infraestrutura.

Descartar o slot inativo

> **Warning**
>
> Cuidado: antes de descartar um slot de replicação, verifique cuidadosamente se ele não tem nenhuma replicação em andamento, está inativo e está em um estado irrecuperável. Descartar um slot prematuramente pode interromper a replicação ou causar perda de dados.

Depois de confirmar que o slot de replicação não é mais necessário, descarte-o para permitir que o autovacuum continue. A condição `active = 'f'` garante que somente um slot inativo seja descartado.

```
SELECT pg_drop_replication_slot('slot1') WHERE active ='f'
```

Réplicas de leitura

Quando a configuração `hot_standby_feedback` está habilitada para réplicas de leitura do Amazon RDS, ela impede que o autovacuum no banco de dados primário remova linhas inativas que ainda podem ser necessárias para consultas executadas na réplica de leitura. Isso afeta todos os tipos de réplicas de leitura físicas, incluindo as que são gerenciadas com ou sem slots de replicação. Esse comportamento é necessário porque as consultas executadas na réplica em espera exigem que essas linhas permaneçam disponíveis no primário, evitando conflitos de consultas e cancelamentos.

Réplica de leitura com slot de replicação física

As réplicas de leitura com slots de replicação física aumentam significativamente a confiabilidade e a estabilidade da replicação no RDS para PostgreSQL. Esses slots garantem que o banco de dados primário retenha os arquivos essenciais do log de gravação antecipada até que a réplica os processe, mantendo a consistência de dados mesmo durante interrupções na rede.

A partir do RDS para PostgreSQL versão 14, todas as réplicas utilizam slots de replicação. Nas versões anteriores, somente réplicas entre regiões usavam slots de replicação.

A função `postgres_get_av_diag()` exibe uma saída semelhante à saída a seguir quando encontra uma réplica de leitura com slot de replicação física como um bloqueador.

```
blocker               | Read replica with physical replication slot
database              |
blocker_identifier    | rds_us_west_2_db_xxxxxxxxxxxxxxxxxxxx
wait_event            | Not applicable
autovacuum_lagging_by | 554,080,689
suggestion            | Run the following query on the replica
 "rds_us_west_2_db_xxxxxxxxxxxxxxxxxxx" to find the long running query:

                      | SELECT * FROM pg_catalog.pg_stat_activity WHERE
 backend_xmin::text::bigint = 757989377;
```

```
                      | Review carefully and you may consdier terminating the query on
 read replica using suggested_action. For more information, see Working with PostgreSQL
 autovacuum in the Amazon RDS User Guide.                                          +
          |
suggested_action      | {"SELECT pg_terminate_backend(pid) FROM
 pg_catalog.pg_stat_activity WHERE backend_xmin::text::bigint = 757989377;","
                                                                                  +
                      | [OR]

                                                +
                      | ","Disable hot_standby_feedback","

                                                +
                      | [OR]

                                                +
                      | ","Delete the read replica if not needed"}
```

Réplica de leitura com replicação de streaming

O Amazon RDS permite configurar réplicas de leitura sem um slot de replicação física em versões mais antigas, até a versão 13. Essa abordagem reduz a sobrecarga ao permitir que o primário recicle arquivos WAL de forma mais agressiva, o que é vantajoso em ambientes com espaço em disco limitado e que podem tolerar ReplicaLag ocasional. No entanto, sem um slot, o standby deve permanecer sincronizado para evitar a perda de arquivos WAL. O Amazon RDS usa arquivos WAL arquivados para ajudar a réplica a se recuperar caso ela fique atrasada, mas esse processo exige monitoramento cuidadoso e pode ser lento.

A função `postgres_get_av_diag()` exibe uma saída semelhante à saída a seguir quando encontra uma réplica de leitura de streaming como um bloqueador.

```
blocker               | Read replica with streaming replication slot
database              | Not applicable
blocker_identifier    | xx.x.x.xxx/xx
wait_event            | Not applicable
autovacuum_lagging_by | 610,146,760
suggestion            | Run the following query on the replica "xx.x.x.xxx" to find the
 long running query:
                                                                                  +
                      | SELECT * FROM pg_catalog.pg_stat_activity WHERE
 backend_xmin::text::bigint = 348319343;
```

```
               +
                     | Review carefully and you may consdier terminating the query on
 read replica using suggested_action. For more information, see Working with PostgreSQL
 autovacuum in the Amazon RDS User Guide.                                          +
                     |
suggested_action     | {"SELECT pg_terminate_backend(pid) FROM
 pg_catalog.pg_stat_activity WHERE backend_xmin::text::bigint = 348319343;","

                      +
                     | [OR]

                                                                                   +
                     | ","Disable hot_standby_feedback","

                                                                                   +
                     | [OR]

                                                                                   +
                     | ","Delete the read replica if not needed"}
```

Ação sugerida

Conforme recomendado na coluna `suggested_action`, analise cuidadosamente essas opções para desbloquear o autovacuum.

- Encerrar a consulta: seguindo as orientações na coluna de sugestões, é possível se conectar à réplica de leitura, conforme especificado na coluna suggested_action. É recomendável analisar cuidadosamente a opção de encerrar a sessão. Se o encerramento for considerado seguro, use a função `pg_terminate_backend( )` para encerrar a sessão. Essa ação pode ser executada por um administrador (como a conta mestra do RDS) ou por um usuário com o privilégio pg_terminate_backend() necessário.

  Execute o comando SQL a seguir na réplica de leitura para encerrar a consulta que está impedindo a limpeza das linhas antigas no primário. O valor de `backend_xmin` é reportado na saída da função:

  ```
  SELECT
      pg_terminate_backend(pid)
  FROM
      pg_catalog.pg_stat_activity
  WHERE
  ```

```
    backend_xmin::text::bigint = backend_xmin;
```

- Desabilitar o feedback de standby a quente: considere desabilitar o parâmetro `hot_standby_feedback` se ele estiver causando atrasos significativos na limpeza.

  O parâmetro `hot_standby_feedback` permite que uma réplica de leitura informe o primário sobre sua atividade de consulta, impedindo que o primário faça a limpeza de tabelas ou linhas que estão em uso no standby. Embora isso garanta a estabilidade da consulta no standby, pode atrasar significativamente a limpeza no primário. Desabilitar esse recurso permite que o primário prossiga com a limpeza sem esperar que o standby se atualize. No entanto, isso pode levar a cancelamentos de consultas ou falhas no standby se ele tentar acessar linhas que foram removidas pela limpeza no primário.
- Excluir a réplica de leitura se não for necessária: se a réplica de leitura não for mais necessária, você poderá exclui-la. Isso removerá a sobrecarga de replicação associada e permitirá que o primário recicle os logs de transações sem ser retido pela réplica.

Tabelas temporárias

Tabelas temporárias, criadas usando a palavra-chave `TEMPORARY`, residem no esquema temporário, por exemplo pg_temp_xxx, e só podem ser acessadas pela sessão que as criou. As tabelas temporárias são removidas quando a sessão termina. No entanto, essas tabelas são invisíveis para o processo de autovacuum do PostgreSQL e devem ser limpas manualmente pela sessão que as criou. Tentar limpar a tabela temporária de outra sessão não tem efeito.

Em circunstâncias incomuns, uma tabela temporária pode existir sem uma sessão ativa que a possua. Se a sessão proprietária terminar inesperadamente devido a uma falha fatal, problemas de rede ou eventos do tipo, a tabela temporária pode não ser limpa, deixando-a como uma tabela "órfã". Quando o processo de autovacuum do PostgreSQL detecta uma tabela temporária órfã, ele registra em log a seguinte mensagem:

```
LOG: autovacuum: found orphan temp table \"%s\".\"%s\" in database \"%s\"
```

A função `postgres_get_av_diag()` exibe uma saída semelhante à saída a seguir quando identifica uma tabela temporária como um bloqueador. Para que a função exiba corretamente a saída relacionada às tabelas temporárias, ela precisa ser executada no mesmo banco de dados em que essas tabelas existem.

```
blocker               | Temporary table
```

```
database              | my_database
blocker_identifier    | pg_temp_14.ttemp
wait_event            | Not applicable
autovacuum_lagging_by | 1,805,802,632
suggestion            | Connect to database "my_database". Review carefully, you
 may consider dropping temporary table using command in suggested_action. For more
 information, see Working with PostgreSQL autovacuum in the Amazon RDS User Guide.
suggested_action      | {"DROP TABLE ttemp;"}
```

Ação sugerida

Siga as instruções fornecidas na coluna `suggestion` da saída para identificar e remover a tabela temporária que está impedindo a execução do autovacuum. Use o comando a seguir para eliminar a tabela temporária reportada por `postgres_get_av_diag()`. Substitua o nome da tabela com base na saída fornecida pela função `postgres_get_av_diag()`.

```
DROP TABLE my_temp_schema.my_temp_table;
```

A seguinte consulta pode ser usada para identificar tabelas temporárias:

```
SELECT
    oid,
    relname,
    relnamespace::regnamespace,
    age(relfrozenxid)
FROM
    pg_class
WHERE
relpersistence = 't'
ORDER BY
    age(relfrozenxid) DESC;
```

Solucionar bloqueadores de limpeza não identificáveis no RDS para PostgreSQL

Esta seção explora razões adicionais que podem impedir o progresso da limpeza. Esses problemas atualmente não são diretamente identificáveis pela função `postgres_get_av_diag()`.

Tópicos

- Páginas inválidas
- Inconsistência do índice

- Taxa de transação excepcionalmente alta

Páginas inválidas

Um erro de página inválida ocorre quando o PostgreSQL detecta uma incompatibilidade na soma de verificação de uma página ao acessá-la. O conteúdo é ilegível, impedindo que o autovacuum congele as tuplas. Isso efetivamente interrompe o processo de limpeza. O seguinte erro é registrado no log do PostgreSQL:

```
WARNING:  page verification failed, calculated checksum YYYYY but expected XXXX
ERROR:  invalid page in block ZZZZZ of relation base/XXXXX/XXXXX
CONTEXT:  automatic vacuum of table myschema.mytable
```

Determinar o tipo de objeto

```
ERROR: invalid page in block 4305910 of relation base/16403/186752608
WARNING: page verification failed, calculated checksum 50065 but expected 60033
```

A partir da mensagem de erro, o caminho `base/16403/186752608` fornece as seguintes informações:

- "base" é o nome do diretório sob o diretório de dados do PostgreSQL.
- "16403" é o OID do banco de dados, que você pode consultar no catálogo do sistema `pg_database`.
- "186752608" é o `relfilenode`, que você pode usar para pesquisar o esquema e o nome do objeto no catálogo do sistema `pg_class`.

Ao verificar a saída da consulta a seguir no banco de dados afetado, será possível determinar o tipo de objeto. A consulta a seguir recupera informações do objeto para oid: 186752608. Substitua o OID pelo relevante para o erro que você encontrou.

```
SELECT
    relname AS object_name,
    relkind AS object_type,
    nspname AS schema_name
FROM
    pg_class c
    JOIN pg_namespace n ON c.relnamespace = n.oid
WHERE
```

```
    c.oid = 186752608;
```

Para obter mais informações, consulte a documentação do PostgreSQL pg_class para ver todos os tipos de objetos compatíveis, indicados pela coluna relkind em pg_class.

Orientação

A solução mais eficaz para esse problema depende da configuração da instância específica do Amazon RDS e do tipo de dados afetados pela página inconsistente.

Se o tipo de objeto for um índice:

É recomendável reconstruir o índice.

- Usar a opção **CONCURRENTLY**: antes da versão 12 do PostgreSQL, a reconstrução de um índice exigia um bloqueio de tabela exclusivo, restringindo o acesso à tabela. Com a versão 12 e posteriores do PostgreSQL, a opção CONCURRENTLY permite o bloqueio em nível de linha, melhorando significativamente a disponibilidade da tabela. Este é o comando:

  ```
  REINDEX INDEX ix_name CONCURRENTLY;
  ```

  Embora CONCURRENTLY seja menos disruptivo, pode ser mais lento em tabelas acessadas com frequência. Considere construir o índice durante períodos de baixo tráfego, se possível.

  Para obter mais informações, consulte a documentação REINDEX do PostgreSQL.
- Usar a opção **INDEX_CLEANUP FALSE**: se os índices forem grandes e, segundo sua avaliação, exigirem muito tempo para serem concluídos, é possível desbloquear o autovacuum executando um VACUUM FREEZE manual ao excluir os índices. Essa funcionalidade está disponível na versão 12 e versões posteriores do PostgreSQL.

  Ignorar os índices permitirá que você pule o processo de limpeza do índice inconsistente e mitigue o problema de conclusão. No entanto, isso não resolverá o problema subjacente da página inválida. Para abordar e resolver totalmente o problema da página inválida, ainda será necessário reconstruir o índice.

Se o tipo de objeto for uma visão materializada:

Se ocorrer um erro de página inválida em uma visão materializada, faça login no banco de dados afetado e atualize-a para resolver a página inválida:

Atualize a visão materializada.

```
REFRESH MATERIALIZED VIEW schema_name.materialized_view_name;
```

Se a atualização falhar, tente recriar:

```
DROP MATERIALIZED VIEW schema_name.materialized_view_name;
CREATE MATERIALIZED VIEW schema_name.materialized_view_name AS query;
```

Atualizar ou recriar a visão materializada a restaura sem afetar os dados da tabela subjacente.

Para todos os outros tipos de objetos:

Para todos os outros tipos de objetos, entre em contato com o AWS Support.

Inconsistência do índice

Um índice logicamente inconsistente pode impedir que o autovacuum progrida. Os seguintes erros ou erros semelhantes são registrados em log durante a fase de limpeza do índice ou quando o índice é acessado por instruções SQL.

```
ERROR: right sibling's left-link doesn't match:block 5 links to 10 instead of expected
 2 in index ix_name
```

```
ERROR: failed to re-find parent key in index "XXXXXXXXXX" for deletion target page XXX
CONTEXT:  while vacuuming index index_name of relation schema.table
```

Orientação

Reconstrua o índice ou pule índices usando `INDEX_CLEANUP` no manual `VACUUM FREEZE`. Para obter informações sobre como reconstruir o índice, consulte Se o tipo de objeto for um índice.

- Usar a opção CONCURRENTLY: antes da versão 12 do PostgreSQL, a reconstrução de um índice exigia um bloqueio de tabela exclusivo, restringindo o acesso à tabela. Com a versão 12 e posteriores do PostgreSQL, a opção CONCURRENTLY permite o bloqueio em nível de linha, melhorando significativamente a disponibilidade da tabela. Este é o comando:

  ```
  REINDEX INDEX ix_name CONCURRENTLY;
  ```

Embora CONCURRENTLY seja menos disruptivo, pode ser mais lento em tabelas acessadas com frequência. Considere construir o índice durante períodos de baixo tráfego, se possível. Para ter mais informações, consulte a documentação REINDEX do PostgreSQL.

- Usar a opção INDEX_CLEANUP FALSE: se os índices forem grandes e, segundo sua avaliação, exigirem muito tempo para serem concluídos, é possível desbloquear o autovacuum executando um VACUUM FREEZE manual ao excluir os índices. Essa funcionalidade está disponível na versão 12 e versões posteriores do PostgreSQL.

  Ignorar os índices permitirá que você pule o processo de limpeza do índice inconsistente e mitigue o problema de conclusão. No entanto, isso não resolverá o problema subjacente da página inválida. Para abordar e resolver totalmente o problema da página inválida, ainda será necessário reconstruir o índice.

Taxa de transação excepcionalmente alta

No PostgreSQL, altas taxas de transação podem afetar significativamente o desempenho do autovacuum, levando a uma limpeza mais lenta das tuplas inativas e ao aumento do risco de conclusão do ID da transação. É possível monitorar a taxa de transação medindo a diferença em `max(age(datfrozenxid))` entre dois períodos, normalmente por segundo. Além disso, é possível usar as métricas de contador do Insights de Performance do RDS a seguir para medir a taxa de transação (a soma de xact_commit e xact_rollback), que é o número total de transações.

Contador	Type	Unidade	Métrica
xact_commit	Transações	Confirmações por segundo	db.Transactions.xact_commit
xact_rollback	Transações	Reversões por segundo	db.Transactions.xact_rollback

Um aumento rápido indica uma alta carga de transações, que pode sobrecarregar o autovacuum, causando sobrecarga, contenção de bloqueios e possíveis problemas de desempenho. Isso pode impactar negativamente o processo de autovacuum de duas maneiras:

- Atividade da tabela: a tabela específica que está sendo limpa pode estar passando por um alto volume de transações, causando atrasos.

- Recursos do sistema: o sistema geral pode estar sobrecarregado, dificultando que o autovacuum acesse os recursos necessários para funcionar com eficiência.

Considere as seguintes estratégias para permitir que o autovacuum opere com mais eficiência e acompanhe suas tarefas:

1. Reduza a taxa de transação, se possível. Considere agrupar transações semelhantes sempre que possível.
2. Defina tabelas atualizadas com frequência com a operação `VACUUM FREEZE` manual noturna, semanal ou quinzenal durante horários de baixo movimento.
3. Considere aumentar a escala verticalmente da classe de instância para alocar mais recursos do sistema para lidar com o alto volume de transações e o autovacuum.

Resolver problemas de desempenho de limpeza no RDS para PostgreSQL

Esta seção discute os fatores que geralmente contribuem para um desempenho mais lento de limpeza e como resolvê-los.

Tópicos

- Limpeza em índices grandes
- Muitas tabelas ou bancos de dados para limpar
- A limpeza agressiva (para evitar conclusão) está em execução

Limpeza em índices grandes

O VACUUM opera em fases sequenciais: inicialização, verificação do heap, índice e limpeza do heap, limpeza de índices, truncamento de heap e limpeza final. Durante a verificação do heap, o processo remove, desfragmenta e congela páginas. Depois que o heap é verificado, o VACUUM limpa os índices, devolve páginas vazias ao sistema operacional e realiza tarefas finais de limpeza, como limpar o mapa de espaço livre e atualizar as estatísticas.

A limpeza de índice pode exigir várias passagens quando `maintenance_work_mem` (ou `autovacuum_work_mem`) é insuficiente para processar o índice. No PostgreSQL 16 e anterior, um limite de memória de 1 GB para armazenar IDs de tuplas mortas geralmente forçava várias passagens em índices grandes. O PostgreSQL 17 introduz o `TidStore`, que aloca memória dinamicamente em vez de usar uma matriz de alocação única. Isso remove a restrição de 1 GB, usa a memória com maior eficiência e reduz a necessidade de várias verificações para cada índice.

Índices grandes ainda podem exigir várias passagens no PostgreSQL 17 se a memória disponível não for suficiente para o processamento completo do índice de uma só vez. Normalmente, os índices maiores contêm mais tuplas mortas que exigem várias passagens.

Detectar operações de limpeza lentas

A função `postgres_get_av_diag()` pode detectar quando a execução de operações de limpeza está lenta devido a memória insuficiente. Para ter mais informações sobre essa função, consulte Instalar ferramentas de monitoramento e diagnóstico de autovacuum no RDS para PostgreSQL.

A função `postgres_get_av_diag()` emite os avisos a seguir quando a memória disponível não é suficiente para concluir a limpeza do índice em uma única passagem.

`rds_tools` 1.8

```
NOTICE: Your database is currently running aggressive vacuum to prevent wraparound and
 it might be slow.
```

```
NOTICE: The current setting of autovacuum_work_mem is "XXX" and might not be
 sufficient. Consider increasing the setting, and if necessary, scaling up the Amazon
 RDS instance class for more memory.
       Additionally, review the possibility of manual vacuum with exclusion of indexes
 using (VACUUM (INDEX_CLEANUP FALSE, VERBOSE TRUE) table_name;).
```

`rds_tools` 1.9

```
NOTICE: Your database is currently running aggressive vacuum to prevent wraparound and
 it might be slow.
```

```
NOTICE: The current setting of autovacuum_work_mem is XX might not be sufficient.
 Consider increasing the setting to XXX, and if necessary, scaling up the RDS instance
 class for more
       memory. The suggested value is an estimate based on the current number of dead
 tuples for the table being vacuumed, which might not fully reflect the latest state.
 Additionally, review the possibility of manual
       vacuum with exclusion of indexes using (VACUUM (INDEX_CLEANUP FALSE, VERBOSE
 TRUE) table_name;). For more information, see
       Working with PostgreSQL autovacuum in the Amazon Amazon RDS User Guide
       .
```

> **Note**
>
> A função `postgres_get_av_diag()` depende do `pg_stat_all_tables.n_dead_tup` para estimar a quantidade de memória necessária para a limpeza do índice.

Quando a função `postgres_get_av_diag()` identificar uma operação de limpeza lenta que requer várias verificações do índice devido a `autovacuum_work_mem` insuficiente, ela gerará a seguinte mensagem:

```
NOTICE: Your vacuum is performing multiple index scans due to insufficient
 autovacuum_work_mem:XXX for index vacuuming.
       For more information, see Working with PostgreSQL autovacuum in the Amazon
 Amazon RDS User Guide.
```

Orientação

Você pode aplicar as soluções alternativas a seguir usando o `VACUUM FREEZE` manual para acelerar o congelamento da tabela.

Aumentar a memória para limpeza

Conforme sugerido pela função `postgres_get_av_diag()`, é recomendável aumentar o parâmetro `autovacuum_work_mem` para lidar com possíveis restrições de memória no nível da instância. Embora `autovacuum_work_mem` seja um parâmetro dinâmico, é importante observar que, para que a nova configuração de memória entre em vigor, o daemon autovacuum precisa reiniciar seus processos. Para fazer isso:

1. Confirme se a nova configuração está em vigor.
2. Encerre os processos que estão executando o autovacuum.

Essa abordagem garante que a alocação de memória ajustada seja aplicada às novas operações do autovacuum.

Para obter resultados mais imediatos, considere realizar manualmente uma operação `VACUUM FREEZE` com uma configuração `maintenance_work_mem` maior em sua sessão:

```
SET maintenance_work_mem TO '1GB';
```

```
VACUUM FREEZE VERBOSE table_name;
```

Se você estiver usando o Amazon RDS e notar que precisa de memória adicional para oferecer suporte a valores mais altos para `maintenance_work_mem` ou `autovacuum_work_mem`, considere fazer upgrade para uma classe de instância com mais memória. Isso pode fornecer os recursos necessários para aprimorar as operações de limpeza manuais e automáticas, levando a um melhor desempenho geral da limpeza e do banco de dados.

Desativar INDEX_CLEANUP

O `VACUUM` manual no PostgreSQL versão 12 e posteriores permite pular a fase de limpeza do índice, enquanto o autovacuum de emergência no PostgreSQL versão 14 e posteriores faz isso automaticamente com base no parâmetro `vacuum_failsafe_age`.

> **Warning**
>
> Pular a etapa de limpeza do índice pode causar inchaço no índice e afetar negativamente o desempenho da consulta. Para mitigar isso, considere reindexar ou fazer limpeza nos índices afetados durante uma janela de manutenção.

Para obter orientação adicional sobre como lidar com índices grandes, consulte a documentação em Gerenciar o autovacuum com grandes índices .

Limpeza paralelo de índices

Desde o PostgreSQL 13, os índices podem ser limpos em paralelo por padrão usando o `VACUUM` manual, com um processo de operação de limpeza atribuído a cada índice. No entanto, para que o PostgreSQL determine se uma operação de limpeza se qualifica para execução paralela, critérios específicos devem ser atendidos:

- Deve haver pelo menos dois índices.
- O parâmetro `max_parallel_maintenance_workers` deve ser definido como no mínimo 2.
- O tamanho do índice deve exceder o limite `min_parallel_index_scan_size`, que por padrão é 512 KB.

É possível ajustar a configuração `max_parallel_maintenance_workers` com base no número de vCPUs disponíveis na sua instância do Amazon RDS e no número de índices na tabela para otimizar o tempo de resposta da limpeza.

Para obter mais informações, consulte Parallel vacuuming in Amazon RDS for PostgreSQL and Amazon Aurora PostgreSQL.

Muitas tabelas ou bancos de dados para limpar

Conforme mencionado na documentação The Autovacuum Daemon do PostgreSQL, o daemon autovacuum é executado por meio de vários processos. Isso inclui um inicializador persistente de autovacuum, responsável por iniciar os processos de trabalho de autovacuum para cada banco de dados do sistema. O inicializador programa esses trabalhadores para iniciarem aproximadamente a cada `autovacuum_naptime` segundos por banco de dados.

Com 'N' bancos de dados, um novo trabalhador começa aproximadamente a cada [`autovacuum_naptime`/N segundos]. No entanto, o número total de trabalhadores simultâneos é limitado pela configuração `autovacuum_max_workers`. Se o número de bancos de dados ou tabelas que precisam ser limpos exceder esse limite, o próximo banco de dados ou tabela será processado assim que um trabalhador estiver disponível.

Quando muitas tabelas ou bancos de dados grandes precisam ser limpos simultaneamente, todos os trabalhadores de autovacuum disponíveis podem ficar ocupados por um longo período, atrasando a manutenção em outras tabelas e bancos de dados. Em ambientes com altas taxas de transação, esse gargalo pode aumentar rapidamente e potencialmente levar a problemas de conclusão de limpeza em sua instância do Amazon RDS.

Quando o `postgres_get_av_diag()` detecta um grande número de tabelas ou bancos de dados, ele fornece a seguinte recomendação:

```
NOTICE: Your database is currently running aggressive vacuum to prevent wraparound and
 it might be slow.
```

```
NOTICE: The current setting of autovacuum_max_workers:3 might not be sufficient.
 Consider increasing the setting and, if necessary, consider scaling up the Amazon RDS
 instance class for more workers.
```

Orientação

Aumentar autovacuum_max_workers

Para agilizar a limpeza, recomendamos ajustar o parâmetro `autovacuum_max_workers` para permitir mais trabalhadores simultâneos de autovacuum. Se os gargalos de desempenho persistirem,

considere aumentar a escala verticalmente de sua instância do Amazon RDS para uma classe com mais vCPUs, o que pode melhorar ainda mais as capacidades de processamento paralelo.

A limpeza agressiva (para evitar conclusão) está em execução

A idade do banco de dados (MaximumUsedTransactionIDs) no PostgreSQL só diminui quando uma limpeza agressiva (para evitar conclusão) é concluída com sucesso. Até que essa limpeza termine, a idade continuará aumentando dependendo da taxa de transação.

A função `postgres_get_av_diag()` gera o seguinte `NOTICE` quando detecta uma limpeza agressiva. No entanto, ela só aciona essa saída depois que a limpeza estiver ativa por pelo menos dois minutos.

```
NOTICE: Your database is currently running aggressive vacuum to prevent wraparound,
 monitor autovacuum performance.
```

Consulte When an aggressive vacuum is already running para obter mais informações sobre a limpeza agressiva.

Use a seguinte consulta para verificar se uma limpeza agressiva está em andamento:

```
SELECT
    a.xact_start AS start_time,
    v.datname "database",
    a.query,
    a.wait_event,
    v.pid,
    v.phase,
    v.relid::regclass,
    pg_size_pretty(pg_relation_size(v.relid)) AS heap_size,
    (
        SELECT
            string_agg(pg_size_pretty(pg_relation_size(i.indexrelid)) || ':' ||
 i.indexrelid::regclass || chr(10), ', ')
        FROM
            pg_index i
        WHERE
            i.indrelid = v.relid
    ) AS index_sizes,
    trunc(v.heap_blks_scanned * 100 / NULLIF(v.heap_blks_total, 0)) AS step1_scan_pct,
    v.index_vacuum_count || '/' || (
```

```
        SELECT
            count(*)
        FROM
            pg_index i
        WHERE
            i.indrelid = v.relid
    ) AS step2_vacuum_indexes,
    trunc(v.heap_blks_vacuumed * 100 / NULLIF(v.heap_blks_total, 0)) AS
 step3_vacuum_pct,
    age(CURRENT_TIMESTAMP, a.xact_start) AS total_time_spent_sofar
FROM
    pg_stat_activity a
    INNER JOIN pg_stat_progress_vacuum v ON v.pid = a.pid;
```

É possível determinar se é uma limpeza agressiva (para evitar conclusão) verificando a coluna de consulta na saída. A frase "para evitar conclusão" indica que se trata de uma limpeza agressiva.

```
query                 | autovacuum: VACUUM public.t3 (to prevent wraparound)
```

Por exemplo, suponha que você tenha um bloqueador com idade de transação de 1 bilhão e uma tabela exigindo uma limpeza agressiva para evitar conclusão na mesma idade da transação. Além disso, há outro bloqueador com idade de transação de 750 milhões. Depois de eliminar o bloqueador com idade de transação de 1 bilhão, a idade de transação não cairá imediatamente para 750 milhões. Ela permanecerá alta até que a tabela que precise da limpeza agressiva ou qualquer transação com idade superior a 750 milhões seja concluída. Durante esse período, a idade das transações do cluster do PostgreSQL continuará aumentando. Quando o processo de limpeza for concluído, a idade da transação cairá para 750 milhões, mas começará a aumentar novamente até que a limpeza adicional seja concluída. Esse ciclo continuará enquanto essas condições persistirem, até que a idade da transação caia para o nível configurado na instância do Amazon RDS, especificado por `autovacuum_freeze_max_age`.

Explicação das mensagens de AVISO no RDS para PostgreSQL

A função `postgres_get_av_diag()` fornece as seguintes mensagens de AVISO:

Quando a idade ainda não atingiu o limite de monitoramento

O limite de monitoramento de `postgres_get_av_diag()` para identificar bloqueadores é de 500 milhões de transações por padrão. Se `postgres_get_av_diag()` gerar o seguinte AVISO, isso indica que a idade da transação ainda não atingiu esse limite.

```
NOTICE: postgres_get_av_diag() checks for blockers that prevent aggressive vacuums
 only, it does so only after exceeding dvb_threshold which is 500,000,000 and age of
 this PostgreSQL cluster is currently at 2.
```

Não conectado ao banco de dados com o ID de transação mais antigo

A função `postgres_get_av_diag()` fornece a saída mais precisa quando conectada ao banco de dados com a idade de ID de transação mais antiga. O banco de dados com a idade de ID de transação mais antiga informada por `postgres_get_av_diag()` será diferente de “my_database” no seu caso. Se você não estiver conectado ao banco de dados correto, o seguinte AVISO será gerado:

```
NOTICE: You are not connected to the database with the age of oldest transaction
 ID. Connect to my_database database and run postgres_get_av_diag() for accurate
 reporting.
```

Conectar-se ao banco de dados com a idade de transação mais antiga é importante pelos seguintes motivos:

- Identificação de bloqueadores de tabelas temporárias: como os metadados das tabelas temporárias são específicos de cada banco de dados, eles geralmente são encontrados no banco de dados em que foram criados. No entanto, se uma tabela temporária for o principal bloqueador e residir no banco de dados com a transação mais antiga, isso pode ser enganoso. A conexão com o banco de dados correto garante a identificação precisa do bloqueador de tabelas temporárias.
- Diagnóstico de vacuums lentos: os metadados do índice e as informações de contagem de tabelas são específicos do banco de dados e necessários para diagnosticar problemas de vacuum lento.

O banco de dados com a transação mais antiga por idade está em um banco de dados rdsadmin ou template0

Em alguns casos, os bancos de dados `rdsadmin` ou `template0` podem ser identificados como o banco de dados com a idade de ID de transação mais antiga. Se isso acontecer, `postgres_get_av_diag()` emitirá o seguinte AVISO:

```
NOTICE: The database with the age of oldest transaction ID is rdsadmin or template0,
 reach out to support if the reported blocker is in rdsadmin or template0.
```

Verifique se o bloqueador listado não é originário de nenhum desses dois bancos de dados. Se for relatado que o bloqueador está presente em `rdsadmin` ou `template0`, entre em contato com o suporte, pois esses bancos de dados não são acessíveis ao usuário e exigem intervenção.

É altamente improvável que o banco de dados `rdsadmin` ou `template0` contenha um bloqueador principal.

Quando um vacuum agressivo já está em execução

A função `postgres_get_av_diag()` foi projetada para relatar quando um processo de vacuum agressivo está em execução, mas ela só vai acionar essa saída depois que o vacuum estiver ativo por pelo menos 1 minuto. Esse atraso intencional ajuda a reduzir as chances de falsos positivos. Ao esperar, a função garante que somente vacuums efetivos e significativos sejam relatados, levando a um monitoramento mais preciso e confiável da atividade do vacuum.

A função `postgres_get_av_diag()` gera o seguinte AVISO quando detecta um ou mais vacuums agressivos em andamento.

```
NOTICE: Your database is currently running aggressive vacuum to prevent wraparound,
 monitor autovacuum performance.
```

Conforme indicado no AVISO, continue monitorando o desempenho do vacuum. Para obter mais informações sobre o vacuum agressivo, consulte A limpeza agressiva (para evitar conclusão) está em execução

Quando o autovacuum está desativado

A função `postgres_get_av_diag()` vai gerar o seguinte AVISO se o autovacuum estiver desativado na instância de banco de dados:

```
NOTICE: Autovacuum is OFF, we strongly recommend to enable it, no restart is
 necessary.
```

O autovacuum é um recurso essencial da instância de banco de dados do RDS para PostgreSQL que garante uma operação tranquila do banco de dados. Ele remove automaticamente as versões antigas das linhas, recupera espaço de armazenamento e evita o inchaço das tabelas, ajudando a manter as tabelas e os índices eficientes para um desempenho ideal. Além disso, ele protege contra o wraparound de ID de transação, que pode interromper transações na instância do Amazon RDS. A desativação do autovacuum pode levar a quedas de longo prazo no desempenho e na estabilidade do banco de dados. Sugerimos que você o mantenha ativado

sempre. Para ter mais informações, consulte Understanding autovacuum in RDS for PostgreSQL environments.

> **Note**
>
> O desligamento do autovacuum não impede os vacuums agressivos. Estes ainda ocorrerão quando as tabelas atingirem o limite de `autovacuum_freeze_max_age`.

O número de transações restantes é criticamente baixo

A função `postgres_get_av_diag()` gera o seguinte AVISO quando um vacuum wraparound é iminente. Este AVISO é emitido quando a instância do Amazon RDS está a 100 milhões de transações de possivelmente rejeitar novas transações.

```
WARNING: Number of transactions remaining is critically low, resolve issues with
 autovacuum or perform manual VACUUM FREEZE before your instance stops accepting
 transactions.
```

Sua ação imediata é necessária para evitar tempo de inatividade do banco de dados. Você deve monitorar de perto suas operações de vacuum e considerar iniciar manualmente um `VACUUM FREEZE` no banco de dados afetado para evitar falhas nas transações.

Trabalhar com os mecanismos de registro em log compatíveis com o RDS para PostgreSQL

Existem vários parâmetros, extensões e outros itens configuráveis que você pode definir para registrar em log as atividades que ocorrem na sua instância de banco de dados PostgreSQL. Incluindo o seguinte:

- O parâmetro `log_statement` pode ser usado para registrar as atividades dos usuários no seu banco de dados PostgreSQL. Para saber mais sobre o registro em log do RDS para PostgreSQL e como monitorar os logs, consulte Arquivos de log do banco de dados do RDS para PostgreSQL.
- O parâmetro `rds.force_admin_logging_level` registra na instância de banco de dados as ações do usuário interno do Amazon RDS (rdsadmin) nos bancos de dados. Ele grava a saída no log de erros do PostgreSQL. Os valores permitidos são `disabled`, `debug5`, `debug4`, `debug3`, `debug2`, `debug1`, `info`, `notice`, `warning`, `error`, log,`fatal` e `panic`. O valor padrão é `disabled`.

- O parâmetro `rds.force_autovacuum_logging_level` pode ser definido para capturar várias operações de autovacuum no log de erros do PostgreSQL. Para ter mais informações, consulteRegistrar atividades do autovacuum e do vacuum em log
- A extensão do PostgreSQL Audit (pgAudit) pode ser instalada e configurada para capturar atividades no nível da sessão ou no nível do objeto. Para ter mais informações, consulteUsar pgAudit para registrar a atividade do banco de dados
- A extensão `log_fdw` possibilita que você acesse o log do mecanismo de banco de dados usando SQL. Para ter mais informações, consulte Usar a extensão log_fdw para acessar o log de banco de dados usando SQL.
- A biblioteca `pg_stat_statements` é especificada como padrão para o parâmetro `shared_preload_libraries` no RDS para PostgreSQL versão 10 e posteriores. É essa biblioteca que você pode usar para analisar consultas em execução. Certifique-se de que `pg_stat_statements` esteja definido no grupo de parâmetros de banco de dados. Para ter mais informações sobre o monitoramento da instância de banco de dados do RDS para PostgreSQL usando as informações fornecidas por essa biblioteca, consulte Estatísticas SQL do RDS PostgreSQL.
- O parâmetro `log_hostname` captura no log o nome do host de cada conexão do cliente. Para RDS para PostgreSQL versão 12 e versões superiores, esse parâmetro é definido como `off` por padrão. Se você ativá-lo, monitore os tempos de conexão da sessão. Quando ativado, o serviço usa a solicitação de pesquisa reversa do sistema de Nomes de Domínio (DNS) para obter o nome do host do cliente que está fazendo a conexão e adicioná-lo ao log do PostgreSQL. Isso tem um impacto perceptível durante a conexão da sessão. Recomendamos que você ative esse parâmetro apenas para solução de problemas.

Em termos gerais, o objetivo do registro em log é possibilitar que o DBA monitore, ajuste a performance e solucione problemas. Muitos dos logs são carregados automaticamente no Amazon CloudWatch ou no Performance Insights. Aqui, eles são classificados e agrupados para fornecer métricas completas para sua instância de banco de dados. Para saber mais sobre o monitoramento e as métricas do Amazon RDS, consulte Métricas de monitoramento em uma instância do Amazon RDS.

Gerenciar arquivos temporários com o PostgreSQL

No PostgreSQL, uma consulta complexa pode executar algumas operações de classificação ou de hash simultaneamente, com cada uma utilizando a memória da instância para armazenar resultados até o valor especificado no parâmetro `work_mem`. Quando a memória da instância não

é suficiente, arquivos temporários são criados para armazenar os resultados. Eles são gravados em disco para concluir a execução da consulta. Posteriormente, esses arquivos são removidos automaticamente após a conclusão da consulta. No RDS para PostgreSQL, esses arquivos são armazenados no Amazon EBS no volume de dados. Para obter mais informações, consulte Armazenamento de instâncias de banco de dados do Amazon RDS. Você pode monitorar a métrica `FreeStorageSpace` publicada no CloudWatch para garantir que a instância de banco de dados tenha espaço de armazenamento livre suficiente. Para obter mais informações, consulte `FreeStorageSpace`

Recomendamos o uso de instâncias do Amazon RDS com otimização de memória para workloads com várias consultas simultâneas que aumentam o uso de arquivos temporários. Essas instâncias usam o armazenamento ao nível do bloco de unidade de estado sólido (SSD) local baseado em memória expressa não volátil (NVMe) para guardar os arquivos temporários. Para ter mais informações, consulte Melhorar a performance das consultas para o RDS para PostgreSQL com leituras otimizadas pelo Amazon RDS.

Você pode usar os parâmetros e as funções a seguir para gerenciar os arquivos temporários em sua instância.

- **`temp_file_limit`**: esse parâmetro cancela qualquer consulta que exceda o tamanho de temp_files em KB. Esse limite impede que qualquer consulta seja executada indefinidamente e consuma espaço em disco com arquivos temporários. Você pode estimar o valor utilizando os resultados do parâmetro `log_temp_files`. Como prática recomendada, examine o comportamento da workload e defina o limite de acordo com a estimativa. O exemplo a seguir mostra como uma consulta é cancelada quando ela excede o limite.

  ```
  postgres=>select * from pgbench_accounts, pg_class, big_table;
  ```

  ```
  ERROR: temporary file size exceeds temp_file_limit (64kB)
  ```

- **`log_temp_files`**: esse parâmetro envia mensagens ao postgresql.log quando os arquivos temporários de uma sessão são removidos. Esse parâmetro produz logs após a conclusão bem-sucedida de uma consulta. Portanto, isso pode não ajudar na solução de problemas de consultas ativas e de longa duração.

  O exemplo a seguir mostra que, quando a consulta é concluída com êxito, as entradas são registradas no arquivo postgresql.log enquanto os arquivos temporários são limpos.

```
2023-02-06 23:48:35 UTC:205.251.233.182(12456):adminuser@postgres:[31236]:LOG:
 temporary file: path "base/pgsql_tmp/pgsql_tmp31236.5", size 140353536
2023-02-06 23:48:35 UTC:205.251.233.182(12456):adminuser@postgres:[31236]:STATEMENT:
  select a.aid from pgbench_accounts a, pgbench_accounts b where a.bid=b.bid order by
 a.bid limit 10;
2023-02-06 23:48:35 UTC:205.251.233.182(12456):adminuser@postgres:[31236]:LOG:
 temporary file: path "base/pgsql_tmp/pgsql_tmp31236.4", size 180428800
2023-02-06 23:48:35 UTC:205.251.233.182(12456):adminuser@postgres:[31236]:STATEMENT:
  select a.aid from pgbench_accounts a, pgbench_accounts b where a.bid=b.bid order by
 a.bid limit 10;
```

- **pg_ls_tmpdir**: essa função que está disponível no RDS para PostgreSQL 13 e versões posteriores oferece visibilidade sobre o uso atual de arquivos temporários. A consulta concluída não aparece nos resultados da função. No exemplo a seguir, você pode visualizar os resultados dessa função.

```
postgres=>select * from pg_ls_tmpdir();
```

```
      name       |    size    |      modification
-----------------+------------+------------------------
 pgsql_tmp8355.1 | 1072250880 | 2023-02-06 22:54:56+00
 pgsql_tmp8351.0 | 1072250880 | 2023-02-06 22:54:43+00
 pgsql_tmp8327.0 | 1072250880 | 2023-02-06 22:54:56+00
 pgsql_tmp8351.1 |  703168512 | 2023-02-06 22:54:56+00
 pgsql_tmp8355.0 | 1072250880 | 2023-02-06 22:54:00+00
 pgsql_tmp8328.1 |  835031040 | 2023-02-06 22:54:56+00
 pgsql_tmp8328.0 | 1072250880 | 2023-02-06 22:54:40+00
(7 rows)
```

```
postgres=>select query from pg_stat_activity where pid = 8355;

query
-------------------------------------------------------------------------------------
select a.aid from pgbench_accounts a, pgbench_accounts b where a.bid=b.bid order by
 a.bid
```

```
(1 row)
```

O nome do arquivo inclui o ID de processamento (PID) da sessão que gerou o arquivo temporário. Uma consulta mais avançada, como no exemplo a seguir, executa uma soma dos arquivos temporários para cada PID.

```
postgres=>select replace(left(name, strpos(name, '.')-1),'pgsql_tmp','') as pid,
 count(*), sum(size) from pg_ls_tmpdir() group by pid;
```

```
 pid  | count |   sum
------+-------------------
 8355 |     2 | 2144501760
 8351 |     2 | 2090770432
 8327 |     1 | 1072250880
 8328 |     2 | 2144501760
(4 rows)
```

- **pg_stat_statements**: se você ativar o parâmetro pg_stat_statements, poderá visualizar o uso médio de arquivos temporários por chamada. Você pode identificar o query_id da consulta e usá-lo para examinar o uso do arquivo temporário, conforme mostrado no exemplo a seguir.

```
postgres=>select queryid from pg_stat_statements where query like 'select a.aid from
 pgbench%';
```

```
       queryid
----------------------
 -7170349228837045701
(1 row)
```

```
postgres=>select queryid, substr(query,1,25), calls, temp_blks_read/calls
 temp_blks_read_per_call, temp_blks_written/calls temp_blks_written_per_call from
 pg_stat_statements where queryid = -7170349228837045701;
```

```
       queryid       |          substr           | calls | temp_blks_read_per_call |
 temp_blks_written_per_call
----------------------+---------------------------+-------+-------------------------
+---------------------------
 -7170349228837045701 | select a.aid from pgbench |    50 |                  239226 |
                      388678
(1 row)
```

- **Performance Insights**: no painel do Performance Insights, você pode visualizar o uso temporário de arquivos ativando as métricas temp_bytes e temp_files. Depois, você pode ver a média dessas duas métricas e como elas correspondem à workload da consulta. A exibição no Performance Insights não mostra especificamente as consultas que estão gerando os arquivos temporários. No entanto, ao combinar o Performance Insights com a consulta mostrada para `pg_ls_tmpdir`, você pode solucionar problemas, analisar e determinar as alterações em sua workload de consulta.

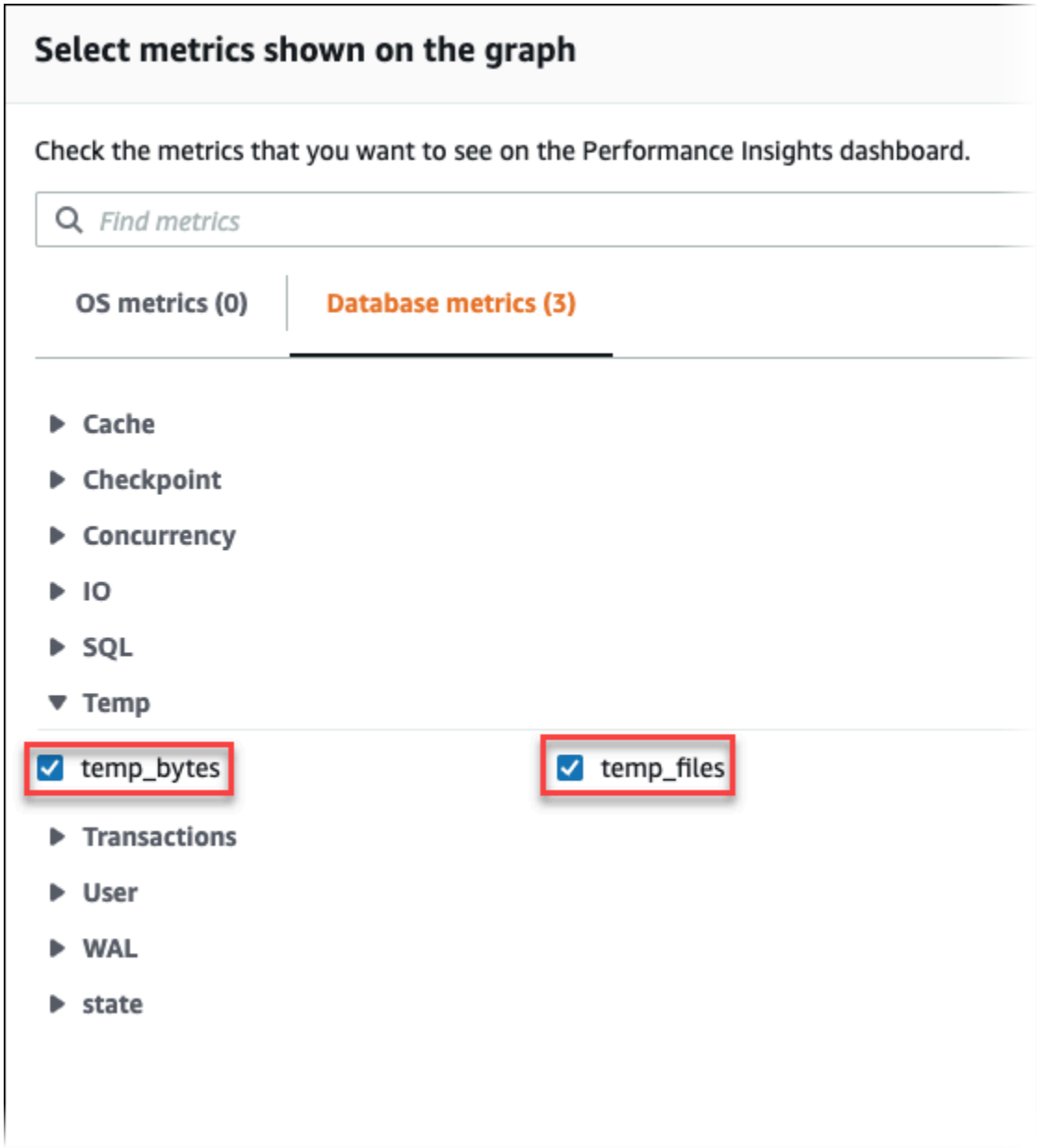
  Para ter mais informações sobre como analisar as métricas e as consultas com o Insights de Performance, consulte Análise de métricas usando o painel do Performance Insights.

  Para ver um exemplo de como visualizar o uso de arquivos temporários com o Insights de Performance, consulte Visualizar o uso de arquivos temporários com o Insights de Performance.

Visualizar o uso de arquivos temporários com o Insights de Performance

É possível usar o Insights de Performance para visualizar o uso de arquivos temporários ativando as métricas temp_bytes e temp_files. A visualização no Insights de Performance não mostra as consultas específicas que geram arquivos temporários, no entanto, ao combinar o Insights de Performance com a consulta mostrada para `pg_ls_tmpdir`, é possível solucionar problemas, analisar e determinar as alterações na workload de consulta.

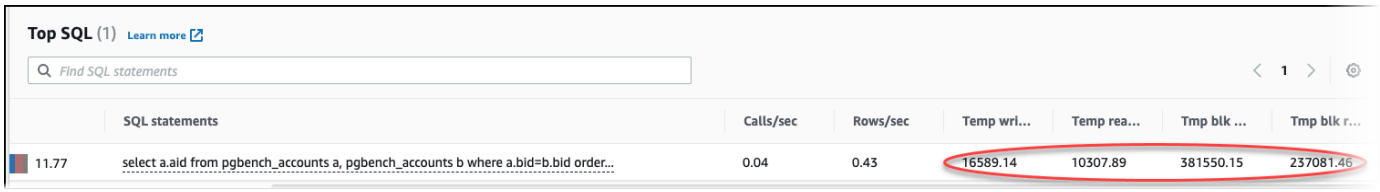
1. No painel do Performance Insights, selecione Gerenciar métricas.
2. Escolha Métricas de banco de dados e selecione as métricas temp_bytes e temp_files como mostrado na imagem a seguir.



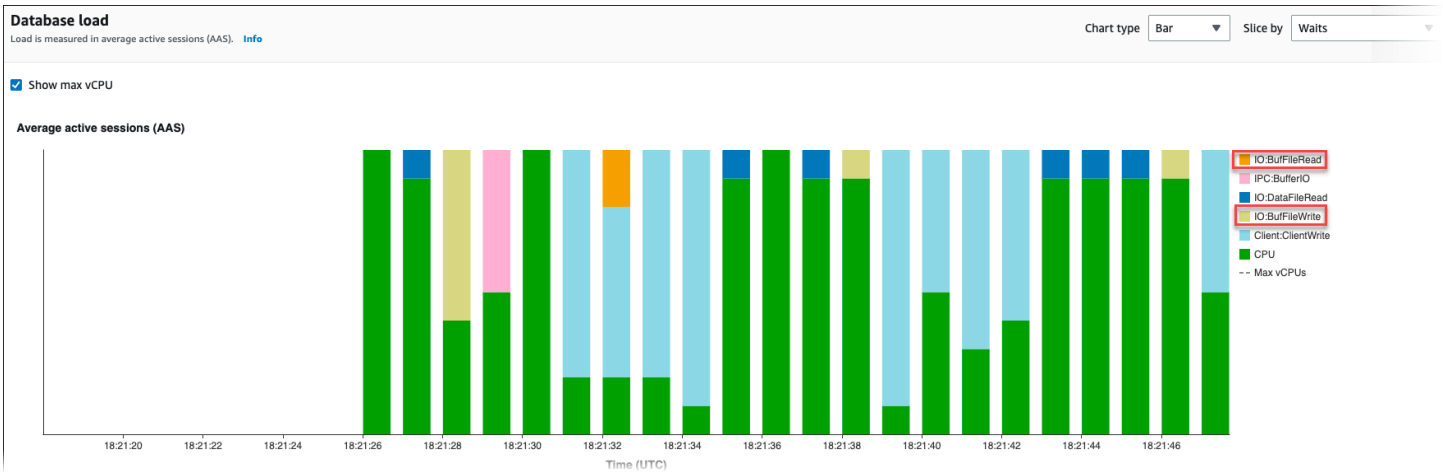
3. Na guia Top SQL, selecione o ícone Preferências.
4. Na janela Preferências, ative as estatísticas a seguir para serem exibidas na guia Top SQL e selecione Continuar.

   - Gravações temporárias/segundo
   - Leituras de temperatura/segundo
   - Gravação/chamada em bloco temporário
   - Leitura/chamada em bloco temporário

5. O arquivo temporário é dividido quando combinado com a consulta mostrada para `pg_ls_tmpdir`, conforme exibido no exemplo a seguir.



Os eventos `IO:BufFileRead` e `IO:BufFileWrite` ocorrem porque as principais consultas na workload geralmente criam arquivos temporários. Você pode usar o Insights de Performance para identificar as principais consultas que aguardam `IO:BufFileRead` e `IO:BufFileWrite` revisando "Média de sessões ativas (AAS)" nas seções "Carga do banco de dados" e "SQL principal".



Para obter mais informações sobre como usar o Insights de Performance para analisar as principais consultas e a carga por eventos de espera, consulte Visão geral da guia Top SQL (SQL principal) Você deve identificar e ajustar as consultas que aumentam o uso de arquivos temporários e os eventos de espera correspondentes. Para obter mais informações sobre esses eventos de espera e a correção, consulte IO:BufFileRead e IO:BufFileWrite.

> **Note**
>
> O parâmetro `work_mem` controla quando a operação de classificação fica sem memória e os resultados são gravados em arquivos temporários. Recomendamos que você não altere a configuração desse parâmetro acima do valor padrão, pois isso permitiria que cada sessão do banco de dados consumisse mais memória. Além disso, uma única sessão que executa junções e classificações complexas pode realizar operações paralelas nas quais cada operação consome memória.

Como prática recomendada, quando você tem um relatório grande com várias junções e classificações, defina esse parâmetro no nível da sessão usando o comando `SET work_mem`. Depois, a alteração é aplicada somente à sessão atual e não altera o valor globalmente.

Uso de pgBadger para análise de logs com o PostgreSQL

Você pode usar um analisador de log, como o pgBadger, para analisar logs do PostgreSQL. A documentação do pgBadger indica que o padrão %l (linha de log da sessão ou do processo) deve ser uma parte do prefixo. Contudo, se você fornecer o `log_line_prefix` atual do RDS como um parâmetro para pgBadger, isso ainda produzirá um relatório.

Por exemplo, o comando a seguir formata corretamente um arquivo de log do Amazon RDS para PostgreSQL com data de 04/02/2014 usando pgBadger.

```
./pgbadger -f stderr -p '%t:%r:%u@%d:[%p]:' postgresql.log.2014-02-04-00
```

Usar o PGSnapper para monitorar o PostgreSQL

Você pode usar o PGSnapper para auxiliar na coleta periódica de estatísticas e métricas relacionadas à performance do Amazon RDS para PostgreSQL. Para ter mais informações, consulte Monitorar a performance do Amazon RDS para PostgreSQL usando o PGSnapper.

Trabalhar com parâmetros na instância de banco de dados do RDS para PostgreSQL

Em alguns casos, você pode criar uma instância de banco de dados do RDS para PostgreSQL sem especificar um grupo de parâmetros personalizado. Se for o caso, sua instância de banco de dados será criada usando o grupo de parâmetros padrão para a versão do PostgreSQL que você escolher. Por exemplo, suponha que você crie uma instância de banco de dados do RDS para PostgreSQL usando o PostgreSQL 13.3. Nesse caso, a instância de banco de dados é criada usando os valores no grupo de parâmetros para versões do PostgreSQL 13, `default.postgres13`.

Você também pode criar seus próprios grupos de parâmetros de banco de dados personalizados. Você precisará fazer isso se quiser modificar qualquer configuração da instância de banco de dados do RDS para PostgreSQL de seus valores padrão. Para saber como, consulte Grupos de parâmetros para Amazon RDS.

Você pode rastrear as configurações em sua instância de banco de dados do RDS para PostgreSQL de várias maneiras diferentes. Você pode usar o AWS Management Console, a AWS CLI ou a API do Amazon RDS. Você também pode consultar os valores `pg_settings` da tabela PostgreSQL da instância, conforme mostrado a seguir.

```
SELECT name, setting, boot_val, reset_val, unit
 FROM pg_settings
 ORDER BY name;
```

Para saber mais sobre os valores retornados dessa consulta, consulte `pg_settings` na documentação do PostgreSQL.

Tenha cuidado especial ao alterar as configurações para `max_connections` e `shared_buffers` em sua instância de banco de dados do RDS para PostgreSQL. Por exemplo, suponha que você modifique as configurações de `max_connections` ou `shared_buffers` e use valores muito altos para sua workload real. Nesse caso, sua instância de banco de dados do RDS para PostgreSQL não será iniciada. Se isso acontecer, você verá o erro a seguir no `postgres.log`.

```
2018-09-18 21:13:15 UTC::@:[8097]:FATAL:  could not map anonymous shared memory: Cannot
 allocate memory
2018-09-18 21:13:15 UTC::@:[8097]:HINT:  This error usually means that PostgreSQL's
 request for a shared memory segment
exceeded available memory or swap space. To reduce the request size (currently
 3514134274048 bytes), reduce
PostgreSQL's shared memory usage, perhaps by reducing shared_buffers or
 max_connections.
```

No entanto, não é possível alterar nenhum valor das configurações contidas nos grupos de parâmetros de banco de dados padrão do RDS para PostgreSQL. Para alterar as configurações de qualquer parâmetro, primeiro crie um grupo de parâmetros de banco de dados personalizado. Em seguida, altere as configurações nesse grupo personalizado e, em seguida, aplique o grupo de parâmetros personalizado à sua instância de banco de dados do RDS para PostgreSQL. Para saber mais, consulte Grupos de parâmetros para Amazon RDS.

Existem dois tipos de parâmetro no RDS para PostgreSQL.

- Parâmetros estáticos – Os parâmetros estáticos exigem que a instância de banco de dados do RDS para PostgreSQL seja reinicializada após uma alteração para que o novo valor possa entrar em vigor.

- Parâmetros dinâmicos – Parâmetros dinâmicos não exigem uma reinicialização após alterar suas configurações.

> **Note**
>
> Se sua instância de banco de dados do RDS para PostgreSQL estiver usando seu próprio grupo de parâmetros de banco de dados personalizado, você poderá alterar os valores de parâmetros dinâmicos na instância de banco de dados em execução. Isso pode ser feito usando o AWS Management Console, a AWS CLI ou a API do Amazon RDS.

Você também poderá alterar valores de parâmetros se tiver os privilégios de segurança necessários para isso usando os comandos `ALTER DATABASE`, `ALTER ROLE` e `SET`.

Lista de parâmetros de instância de banco de dados do RDS para PostgreSQL

A tabela a seguir lista alguns dos parâmetros (mas nem todos) disponíveis em uma instância de banco de dados do RDS para PostgreSQL. Para visualizar todos os parâmetros disponíveis, use o comando describe-db-parameters da AWS CLI. Por exemplo, para obter a lista de todos os parâmetros disponíveis no grupo de parâmetros padrão do RDS para PostgreSQL versão 13, execute o seguinte.

```
aws rds describe-db-parameters --db-parameter-group-name default.postgres13
```

Você também pode usar o console. Selecione Parameter groups (Grupos de parâmetros) no menu do Amazon RDS e depois selecione o grupo de parâmetros dentre os disponíveis em sua Região da AWS.

Nome do parâmetro	Apply_Type	Descrição
application_name	Dinâmico	Define o nome da aplicação a ser informada em estatísticas e logs.
archive_command	Dinâmico	Define o comando shell que será chamado para arquivar um arquivo WAL.
array_nulls	Dinâmico	Permite entrada de elementos NULL em arrays.
authentication_timeout	Dinâmico	Define o tempo máximo permitido para concluir a autenticação de cliente.
autovacuum	Dinâmico	Inicia o subprocesso de autovacuum.
autovacuum_analyze_scale_factor	Dinâmico	Número de inserções, atualizações ou exclusões de tuplas anteriores à análise, como uma fração de reltuplas.
autovacuum_analyze_threshold	Dinâmico	Número mínimo de inserções, atualizações ou exclusões de tuplas antes da análise.

Nome do parâmetro	Apply_Type	Descrição
autovacuum_freeze_max_age	Estático	Idade na qual o autovacuum de uma tabela deve ocorrer para evitar a conclusão do ID de transação .
autovacuum_naptime	Dinâmico	Tempo de suspensão entre execuções de autovacuum.
autovacuum_max_workers	Estático	Define o número máximo de processos de trabalho de autovacuum em execução simultaneamente.
autovacuum_vacuum_cost_delay	Dinâmico	Atraso de custo de vacuum, em milissegundos, para autovacuum.
autovacuum_vacuum_cost_limit	Dinâmico	Valor do custo de vacuum disponível antes da suspensão, para autovacuum.
autovacuum_vacuum_scale_factor	Dinâmico	Número de atualizações ou exclusões de tuplas antes de vacuum, como uma fração de reltuplas.
autovacuum_vacuum_threshold	Dinâmico	Número mínimo de atualizações ou exclusões de tuplas antes de vacuum.
backslash_quote	Dinâmico	Define-se uma barra invertida (\) é permitida em literais de string.
bgwriter_delay	Dinâmico	Tempo de suspensão do gravador em segundo plano entre rodadas.
bgwriter_lru_maxpages	Dinâmico	Número máximo do gravador em segundo plano de páginas LRU para descarregamento por rodada.
bgwriter_lru_multiplier	Dinâmico	Múltiplo do uso médio do buffer para liberação por rodada.
bytea_output	Dinâmico	Define o formato da saída para bytes.

Nome do parâmetro	Apply_Type	Descrição
check_function_bodies	Dinâmico	Verifica corpos de funções durante CREATE FUNCTION.
checkpoint_completion_target	Dinâmico	Tempo gasto descarregando buffers sujos durante verificações, como uma fração do intervalo de verificação.
checkpoint_segments	Dinâmico	Define a distância máxima em segmentos de log entre verificações pontos de verificação WAL (write-ahead log)de log WAL automáticos.
checkpoint_timeout	Dinâmico	Define o tempo máximo entre verificações WAL automáticas.
checkpoint_warning	Dinâmico	Permite avisos se segmentos de verificação forem preenchidos mais frequentemente do que isso.
client_connection_check_interval	Dinâmico	Define o intervalo de tempo entre as verificações de desconexão durante a execução de consultas.
client_encoding	Dinâmico	Define a codificação do conjunto de caracteres do cliente.
client_min_messages	Dinâmico	Define os níveis de mensagem enviados ao cliente.
commit_delay	Dinâmico	Define o atraso em microssegundos entre a confirmação da transação e a liberação do WAL no disco.
commit_siblings	Dinâmico	Define o mínimo de transações simultâneas abertas antes de realizar commit_delay.
constraint_exclusion	Dinâmico	Permite que o planejador use restrições para otimizar consultas.

Nome do parâmetro	Apply_Type	Descrição
cpu_index_tuple_cost	Dinâmico	Define a estimativa do planejador sobre o custo do processamento de cada entrada de índice durante uma verificação de índice.
cpu_operator_cost	Dinâmico	Define a estimativa do planejador sobre o custo do processamento de cada operador ou chamada de função.
cpu_tuple_cost	Dinâmico	Define a estimativa do planejador sobre o custo do processamento de cada tupla (linha).
cursor_tuple_fraction	Dinâmico	Define a estimativa do planejador sobre a fração de linhas de um cursor que serão recuperadas.
datestyle	Dinâmico	Define o formato de exibição de valores de data e hora.
deadlock_timeout	Dinâmico	Define o tempo de espera em um bloqueio antes de verificar a existência de um deadlock.
debug_pretty_print	Dinâmico	Recua exibições de árvores de análise e plano.
debug_print_parse	Dinâmico	Registra a árvore de análise de cada consulta.
debug_print_plan	Dinâmico	Registra o plano de execução de cada consulta.
debug_print_rewritten	Dinâmico	Registra a árvore de análise regravada de cada consulta.
default_statistics_target	Dinâmico	Define o destino de estatísticas padrão.
default_tablespace	Dinâmico	Define o espaço de tabela padrão no qual criar tabelas e índices.
default_transaction_deferrable	Dinâmico	Define o status padrão postergável de novas transações.

Nome do parâmetro	Apply_Type	Descrição
default_transaction_isolation	Dinâmico	Define o nível de isolamento de transação de cada nova transação.
default_transaction_read_only	Dinâmico	Define o status padrão somente leitura de novas transações.
default_with_oids	Dinâmico	Cria tabelas com “Object IDs” (OIDs – IDs de objetos) por padrão.
effective_cache_size	Dinâmico	Define a pressuposição do planejador sobre o tamanho do cache de disco.
effective_io_concurrency	Dinâmico	Número de solicitações simultâneas que podem ser manipuladas de forma eficiente pelo subsistema de disco.
enable_bitmapscan	Dinâmico	Permite o uso do planejador de planos de verificação de bitmap.
enable_hashagg	Dinâmico	Permite o uso do planejador de planos de agregação em hash.
enable_hashjoin	Dinâmico	Permite o uso do planejador de planos de junção hash.
enable_indexscan	Dinâmico	Permite o uso do planejador de planos de verificação de índice.
enable_material	Dinâmico	Permite o uso do planejador da materialização.
enable_mergejoin	Dinâmico	Permite o uso do planejador de planos de junção de mesclagem.
enable_nestloop	Dinâmico	Permite o uso do planejador de planos de junção de loop aninhado.

Nome do parâmetro	Apply_Type	Descrição
enable_seqscan	Dinâmico	Permite o uso do planejador de planos de verificação sequencial.
enable_sort	Dinâmico	Permite o uso do planejador de etapas de classificação explícitas.
enable_tidscan	Dinâmico	Permite o uso do planejador de planos de verificação TID.
escape_string_warning	Dinâmico	Avisa sobre escapes de barra invertida (\) em literais de string comuns.
extra_float_digits	Dinâmico	Define o número de dígitos exibidos para valores de ponto flutuante.
from_collapse_limit	Dinâmico	Define o tamanho da lista FROM além do qual subconsultas não são recolhidas.
fsync	Dinâmico	Força a sincronização de atualizações no disco.
full_page_writes	Dinâmico	Grava páginas cheias no WAL ao serem modificadas pela primeira vez após uma verificação.
geqo	Dinâmico	Permite a otimização de consultas genéticas.
geqo_effort	Dinâmico	GEQO: esforço é usado para definir o padrão para outros parâmetros GEQO.
geqo_generations	Dinâmico	GEQO: número de iterações do algoritmo.
geqo_pool_size	Dinâmico	GEQO: número de indivíduos na população.
geqo_seed	Dinâmico	GEQO: propagação para seleção de caminho aleatório.
geqo_selection_bias	Dinâmico	GEQO: pressão seletiva dentro da população.

Nome do parâmetro	Apply_Type	Descrição
geqo_threshold	Dinâmico	Define o limite de itens FROM além do qual o GEQO é usado.
gin_fuzzy_search_limit	Dinâmico	Define o resultado máximo permitido para pesquisa exata por GIN.
hot_standby_feedback	Dinâmico	Determina se um standby a quente envia mensagens de comentários ao standby principal ou upstream.
intervalstyle	Dinâmico	Define o formato de exibição para valores de intervalo.
join_collapse_limit	Dinâmico	Define o tamanho da lista FROM além do qual constructos JOIN não são nivelados.
lc_messages	Dinâmico	Define o idioma em que as mensagens são exibidas.
lc_monetary	Dinâmico	Define a localidade para a formatação de valores monetários.
lc_numeric	Dinâmico	Define a localidade para a formatação de números.
lc_time	Dinâmico	Define a localidade para a formatação de valores de data e hora.
log_autovacuum_min_duration	Dinâmico	Define o tempo de execução mínimo acima do qual as ações de autovacuum serão registradas em log.
log_checkpoints	Dinâmico	Registra cada verificação.
log_connections	Dinâmico	Registra cada conexão bem-sucedida.

Nome do parâmetro	Apply_Type	Descrição
log_disconnections	Dinâmico	Registra o fim de uma sessão, incluindo a duração.
log_duration	Dinâmico	Registra a duração de cada instrução SQL concluída.
log_error_verbosity	Dinâmico	Define a verbosidade das mensagens registradas.
log_executor_stats	Dinâmico	Grava estatísticas de performance do executor no log do servidor.
log_filename	Dinâmico	Define o padrão de nome de arquivo para arquivos de log.
log_file_mode	Dinâmico	Define as permissões de arquivos de log. O valor padrão é 0644.
log_hostname	Dinâmico	Registra o nome do host nos logs de conexão. A partir do PostgreSQL 12 e versões posteriores, esse parâmetro está “desativado” por padrão. Quando ativada, a conexão usa a pesquisa reversa de DNS para obter o nome do host que é capturado nos logs de conexão. Se você ativar esse parâmetro, deverá monitorar o impacto que ele tem no tempo necessário para estabelecer conexões.
log_line_prefix	Dinâmico	Controla informações prefixadas para cada linha de log.
log_lock_waits	Dinâmico	Registra esperas de bloqueio longas.
log_min_duration_statement	Dinâmico	Define o tempo de execução mínimo acima do qual as instruções serão registradas em log.

Nome do parâmetro	Apply_Type	Descrição
log_min_error_statement	Dinâmico	Faz com que todas as instruções que geram um erro igual ou acima desse nível sejam registradas.
log_min_messages	Dinâmico	Define os níveis de mensagem registrados.
log_parser_stats	Dinâmico	Grava estatísticas de performance do analisador no log do servidor.
log_planner_stats	Dinâmico	Grava estatísticas de performance do planejador no log do servidor.
log_rotation_age	Dinâmico	A rotação automática de arquivos de log ocorrerá depois de N minutos.
log_rotation_size	Dinâmico	A rotação automática de arquivos de log ocorrerá depois de N kilobytes.
log_statement	Dinâmico	Define o tipo de instruções registradas.
log_statement_stats	Dinâmico	Grava estatísticas de performance cumulativas no log do servidor.
log_temp_files	Dinâmico	Registra o uso de arquivos temporários maiores do que esse número de kilobytes.
log_timezone	Dinâmico	Define o fuso horário a ser usado em mensagens de log.
log_truncate_on_rotation	Dinâmico	Trunca os arquivos de log existentes com o mesmo nome durante a alternância do log.
logging_collector	Estático	Inicia um subprocesso para capturar a saída stderr e/ou csvlogs em arquivos de log.
maintenance_work_mem	Dinâmico	Define a memória máxima a ser usada para operações de manutenção.

Nome do parâmetro	Apply_Type	Descrição
max_connections	Estático	Define o número máximo de conexões simultâneas.
max_files_per_process	Estático	Define o número máximo de arquivos abertos simultaneamente para cada processo do servidor.
max_locks_per_transaction	Estático	Define o número máximo de bloqueios por transação.
max_pred_locks_per_transaction	Estático	Define o número máximo de bloqueios de predicado por transação.
max_prepared_transactions	Estático	Define o número máximo de transações simultaneamente preparadas.
max_stack_depth	Dinâmico	Define a profundidade máxima da pilha, em kilobytes.
max_standby_archive_delay	Dinâmico	Define o atraso máximo antes de cancelar consultas quando um servidor em standby a quente está processando dados WAL arquivados.
max_standby_streaming_delay	Dinâmico	Define o atraso máximo antes de cancelar consultas quando um servidor em standby a quente está processando dados do WAL em stream.

Nome do parâmetro	Apply_Type	Descrição

`max_wal_size`	Dinâmico	Define o tamanho do WAL (MB) que aciona o ponto de verificação.
		• No RDS para PostgreSQL 15 e versões anteriores, o tamanho é 2 GB.
		• Com relação ao RDS para PostgreSQL 16 e versões posteriores, o RDS para PostgreSQL faz a configuração automaticamente com base no tamanho do armazenamento alocado da instância:
		  • 6 GB para instâncias com armazenamento alocado de 100 GB ou mais.
		  • 2 GB para instâncias com armazenamento alocado de 100 GB.
		Use o seguinte comando na instância de banco de dados do Amazon RDS para PostgreSQL para ver o valor atual:
		`SHOW max_wal_size;`

`min_wal_size`	Dinâmico	Define o tamanho mínimo até o qual diminuir o WAL. Para o PostgreSQL versão 9.6 e anterior, `min_wal_size` está em unidades de 16 MB. Para o PostgreSQL versão 10 e posterior, `min_wal_size` está em unidades de 1 MB.
`quote_all_identifiers`	Dinâmico	Adiciona aspas (") a todos os identificadores ao gerar fragmentos SQL.

Nome do parâmetro	Apply_Type	Descrição
random_page_cost	Dinâmico	Define a estimativa do planejador sobre o custo de uma página de disco não sequencialmente buscada. Esse parâmetro não tem valor, a menos que o gerenciamento do plano de consulta (QPM) esteja ativado. Quando o QPM está ativado, o valor padrão desse parâmetro é 4.
rds.adaptive_autovacuum	Dinâmico	Ajusta automaticamente os parâmetros de autovacuum sempre que os limites de ID de transação são excedidos.
rds.force_ssl	Dinâmico	Requer o uso de conexões SSL. O valor padrão é definido como 1 (ligado) para o RDS para PostgreSQL versão 15. Todas as outras versões principais do RDS para PostgreSQL 14 e anteriores têm o valor padrão definido como 0 (desativado).
rds.local_volume_spill_enabled	Estático	Permite gravar arquivos de despejo lógico no volume local.
rds.log_retention_period	Dinâmico	Define a retenção de logs para que o Amazon RDS exclua os logs do PostgreSQL com mais de n minutos.
rds.rds_superuser_reserved_connections	Estático	Define o número de slots de conexão reservados para rds_superusers. Esse parâmetro só está disponível nas versões 15 e posterior. Para ter mais informações, consulte reserved_connections na documentação do PostgreSQL.

Nome do parâmetro	Apply_Type	Descrição
`rds.replica_identity_full`	Dinâmico	Quando você define esse parâmetro como on, ele substitui a configuração de identidade da réplica por `FULL` em todas as tabelas do banco de dados. Isso significa que todos os valores das colunas são gravados no log de gravação antecipada (WAL), independentemente das configurações de `REPLICA IDENTITY FULL`. **Note** A ativação desse parâmetro pode aumentar as IOPS da instância de banco de dados devido ao registro em log adicional com gravação antecipada (WAL).
`rds.restrict_password_commands`	Estático	Restringe quem gerencia senhas para usuários com a função `rds_password` . Defina esse parâmetro como 1 para habilitar a restrição de senha. O padrão é 0.
`search_path`	Dinâmico	Define a ordem de pesquisa do esquema de nomes que não são qualificados para esquema.
`seq_page_cost`	Dinâmico	Define a estimativa do planejador sobre o custo de uma página de disco sequencialmente buscada.
`session_replication_role`	Dinâmico	Define o comportamento de sessões para disparadores e regras de regravação.
`shared_buffers`	Estático	Define o número de buffers de memória compartilhada usados pelo servidor.

Nome do parâmetro	Apply_Type	Descrição
shared_preload_libraries	Estático	Lista as bibliotecas compartilhadas a serem pré-carregadas na instância de banco de dados do RDS para PostgreSQL. Os valores compatíveis incluem auto_explain, orafce, pgaudit, pglogical, pg_bigm, pg_cron, pg_hint_plan, pg_prewarm, pg_similarity, pg_stat_statements, pg_tle, pg_transport, plprofiler e plrust.
ssl	Dinâmico	Habilita conexões SSL.
sql_inheritance	Dinâmico	Faz com que subtabelas sejam incluídas por padrão em vários comandos.
ssl_renegotiation_limit	Dinâmico	Define a quantidade de tráfego para envio e recebimento antes da renegociação das chaves de criptografia.
standard_conforming_strings	Dinâmico	Faz com que strings ... tratem barras invertidas literalmente.
statement_timeout	Dinâmico	Define a duração máxima permitida de qualquer instrução.
synchronize_seqscans	Dinâmico	Permite varreduras sequenciais sincronizadas.
synchronous_commit	Dinâmico	Define o nível de sincronização de transações atual.
tcp_keepalives_count	Dinâmico	Número máximo de retransmissões de keepalives TCP.
tcp_keepalives_idle	Dinâmico	Tempo entre a emissão de keepalives TCP.
tcp_keepalives_interval	Dinâmico	Tempo entre retransmissões de keepalives TCP.

Nome do parâmetro	Apply_Type	Descrição
`temp_buffers`	Dinâmico	Define o número máximo de buffers temporários usado por cada sessão.
`temp_file_limit`	Dinâmico	Define o tamanho máximo em KB até o qual os arquivos temporários podem aumentar.
`temp_tablespaces`	Dinâmico	Define os espaços de tabela a serem usados para tabelas temporárias e arquivos de classificação.
`timezone`	Dinâmico	Define o fuso horário para exibir e interpretar carimbos de data/hora. A Internet Assigned Numbers Authority (IANA) publica novos fusos horários várias vezes por ano em https://www.iana.org/time-zones. Toda vez que o RDS lança uma nova versão de manutenção secundária do PostgreSQL, ela vem com os dados de fuso horário mais recentes no momento do lançamento. Quando você usa as versões mais recentes do RDS para PostgreSQL, você tem dados de fuso horário recentes do RDS. Para garantir que sua instância de banco de dados tenha dados de fuso horário recentes, recomendamos atualizar para uma versão superior do mecanismo de banco de dados. Não é possível modificar manualmente as tabelas de fuso horário nas instâncias de banco de dados PostgreSQL. O RDS não modifica nem redefine os dados de fuso horário das instâncias de banco de dados em execução. Os novos dados de fuso horário são instalados somente quando você executa uma atualização da versão do mecanismo de banco de dados.

Nome do parâmetro	Apply_Type	Descrição
track_activities	Dinâmico	Coleta informações sobre a execução de comandos.
track_activity_query_size	Estático	Define o tamanho reservado para pg_stat_activity.current_query, em bytes.
track_counts	Dinâmico	Coleta estatísticas sobre a atividade do banco de dados.
track_functions	Dinâmico	Coleta estatísticas em nível de função sobre a atividade do banco de dados.
track_io_timing	Dinâmico	Coleta estatísticas de tempo sobre atividades de E/S do banco de dados.
transaction_deferrable	Dinâmico	Indica se uma transação serializável somente de leitura deve ser adiada até que ela possa ser iniciada sem falhas de serialização possíveis.
transaction_isolation	Dinâmico	Define o nível atual de isolamento de transações.
transaction_read_only	Dinâmico	Define o status somente leitura das transações atuais.
transform_null_equals	Dinâmico	Trata expr=NULL como expr IS NULL.
update_process_title	Dinâmico	Atualiza o título do processo para mostrar o comando SQL ativo.
vacuum_cost_delay	Dinâmico	Atraso de custo de vacuum, em milissegundos.
vacuum_cost_limit	Dinâmico	Valor do custo de vacuum disponível antes da suspensão.

Nome do parâmetro	Apply_Type	Descrição
vacuum_cost_page_dirty	Dinâmico	Custo de vacuum para uma página suja por vacuum.
vacuum_cost_page_hit	Dinâmico	Custo de vacuum para uma página encontrada no cache do buffer.
vacuum_cost_page_miss	Dinâmico	Custo de vacuum para uma página não encontrada no cache do buffer.
vacuum_defer_cleanup_age	Dinâmico	Número de transações pelas quais a limpeza vacuum e hot deve ser adiada, se houver.
vacuum_freeze_min_age	Dinâmico	Idade mínima na qual o vacuum deve congelar uma linha de tabela.
vacuum_freeze_table_age	Dinâmico	Idade na qual o vacuum deve varrer uma tabela inteira para congelar tuplas.
wal_buffers	Estático	Define o número de buffers da página de disco na memória compartilhada para WAL.
wal_writer_delay	Dinâmico	Tempo de suspensão do gravador WAL entre liberações do WAL.
work_mem	Dinâmico	Define o máximo de memória a ser usado para espaços de trabalho de consulta.
xmlbinary	Dinâmico	Define como valores binários devem ser codificados em XML.
xmloption	Dinâmico	Define se dados XML em operações de análise e serialização implícitas são considerados documentos ou fragmentos de conteúdo.

O Amazon RDS usa as unidades padrão do PostgreSQL para todos os parâmetros. A tabela a seguir mostra a unidade padrão de cada parâmetro do PostgreSQL.

Nome do parâmetro	Unidade
archive_timeout	s
authentication_timeout	s
autovacuum_naptime	s
autovacuum_vacuum_cost_delay	ms
bgwriter_delay	ms
checkpoint_timeout	s
checkpoint_warning	s
deadlock_timeout	ms
effective_cache_size	8 KB
lock_timeout	ms
log_autovacuum_min_duration	ms
log_min_duration_statement	ms
log_rotation_age	minutos
log_rotation_size	KB
log_temp_files	KB
maintenance_work_mem	KB
max_stack_depth	KB
max_standby_archive_delay	ms
max_standby_streaming_delay	ms
post_auth_delay	s

Nome do parâmetro	Unidade
pre_auth_delay	s
segment_size	8 KB
shared_buffers	8 KB
statement_timeout	ms
ssl_renegotiation_limit	KB
tcp_keepalives_idle	s
tcp_keepalives_interval	s
temp_file_limit	KB
work_mem	KB
temp_buffers	8 KB
vacuum_cost_delay	ms
wal_buffers	8 KB
wal_receiver_timeout	ms
wal_segment_size	B
wal_sender_timeout	ms
wal_writer_delay	ms
wal_receiver_status_interval	s

Ajustar com eventos de espera do RDS para PostgreSQL

Eventos de espera são uma ferramenta de ajuste importante do RDS para PostgreSQL. Se você puder descobrir por que as sessões estão aguardando recursos e o que elas estão fazendo, poderá reduzir melhor os gargalos. Use as informações nesta seção para encontrar possíveis causas e ações corretivas. Esta seção também aborda conceitos básicos de ajuste do PostgreSQL.

Os eventos de espera nesta seção são específicos do RDS para PostgreSQL.

Tópicos

- Conceitos essenciais para o ajuste do RDS para PostgreSQL
- Eventos de espera do RDS para PostgreSQL
- Client:ClientRead
- Client:ClientWrite
- CPU
- IO:BufFileRead and IO:BufFileWrite
- IO:DataFileRead
- IO:WALWrite
- IPC: eventos de espera paralelos
- Lock:advisory
- Lock:extend
- Lock:Relation
- Lock:transactionid
- Lock:tuple
- LWLock:BufferMapping (LWLock:buffer_mapping)
- LWLock:BufferIO (IPC:BufferIO)
- LWLock:buffer_content (BufferContent)
- LWLock:lock_manager (LWLock:lockmanager)
- LWLock:pg_stat_statements
- Tempo limite:PgSleep
- Timeout:VacuumDelay

Conceitos essenciais para o ajuste do RDS para PostgreSQL

Antes de ajustar seu banco de dados do RDS para PostgreSQL, aprenda o que são eventos de espera e por que eles ocorrem. Reveja também a arquitetura básica de memória e disco do RDS para PostgreSQL. Para obter um diagrama de arquitetura útil, consulte o wikibook PostgreSQL.

Tópicos

- Eventos de espera do RDS para PostgreSQL
- Memória do RDS para PostgreSQL
- Processo do RDS para PostgreSQL

Eventos de espera do RDS para PostgreSQL

Um evento de espera é uma indicação de que a sessão está aguardando um recurso. Por exemplo, o evento de espera `Client:ClientRead` ocorre quando o RDS para PostgreSQL está aguardando para receber dados do cliente. Normalmente, as sessões aguardam por recursos como os seguintes.

- Acesso com thread único a um buffer, por exemplo, quando uma sessão está tentando modificar um buffer
- Uma linha que está bloqueada por outra sessão
- Uma leitura de arquivo de dados
- Uma gravação em arquivo de log

Por exemplo, para satisfazer uma consulta, a sessão pode realizar uma varredura de tabela completa. Se esses dados ainda não estiverem na memória, a sessão aguardará a conclusão da E/S do disco. Quando os buffers são lidos na memória, talvez a sessão precise aguardar, pois outras sessões estão acessando os mesmos buffers. O banco de dados registra as esperas utilizando um evento de espera predefinido. Esses eventos estão agrupados em categorias.

Por si só, um único evento de espera não indica um problema de performance. Por exemplo, se os dados solicitados não estão na memória, é necessário ler dados do disco. Se uma sessão bloquear uma linha para uma atualização, outra sessão aguardará que essa linha seja desbloqueada para poder atualizá-la. Uma confirmação exige a conclusão da gravação em um arquivo de log. Esperas são componentes integrais do funcionamento normal de um banco de dados.

Por outro lado, uma série de eventos de espera geralmente mostra um problema de performance. Nesses casos, é possível utilizar os dados dos eventos de espera para determinar onde as sessões

estão perdendo tempo. Por exemplo, se um relatório que é normalmente executado em minutos agora demora várias horas, é possível identificar os eventos de espera que mais contribuem para o tempo de espera total. Se você puder determinar as causas dos principais eventos de espera, às vezes pode aplicar alterações que melhoram a performance. Por exemplo, se a sua sessão está aguardando uma linha que foi bloqueada por outra sessão, é possível encerrar a sessão responsável pelo bloqueio.

Memória do RDS para PostgreSQL

A memória do RDS para PostgreSQL está dividida em compartilhada e local.

Tópicos

- Memória compartilhada no RDS para PostgreSQL
- Memória local no RDS para PostgreSQL

Memória compartilhada no RDS para PostgreSQL

O RDS para PostgreSQL aloca memória compartilhada quando a instância é iniciada. A memória compartilhada está dividida em várias subáreas. As seções a seguir fornecem descrições das mais importantes.

Tópicos

- Buffers compartilhados
- Buffers de log de gravação antecipada (WAL)

Buffers compartilhados

O grupo de buffer compartilhado é uma área de memória do RDS para PostgreSQL que contém todas as páginas que estão ou estavam sendo utilizadas por conexões de aplicações. Uma página é a versão de memória de um bloco de disco. O grupo de buffer compartilhado armazena em cache os blocos de dados lidos do disco. O grupo reduz a necessidade de reler dados do disco, fazendo com que o banco de dados opere de maneira mais eficiente.

Cada tabela e índice são armazenados como uma matriz de páginas com tamanho fixo. Cada bloco contém várias tuplas, que correspondem a linhas. Uma tupla pode ser armazenada em qualquer página.

O grupo de buffer compartilhado possui memória finita. Se uma nova solicitação exigir uma página que não esteja na memória e não houver mais memória, o RDS para PostgreSQL removerá uma página utilizada com menos frequência para acomodar essa solicitação. A política de despejo é implementada por um algoritmo de varredura de relógio.

O parâmetro `shared_buffers` determina a quantidade de memória que o servidor dedica ao armazenamento em cache de dados. O valor padrão é definido como `{DBInstanceClassMemory/32768}` bytes, com base na memória disponível para a instância de banco de dados.

Buffers de log de gravação antecipada (WAL)

Um buffer de log de gravação antecipada (WAL) mantém dados de transação que o RDS para PostgreSQL grava posteriormente no armazenamento persistente. Utilizando o mecanismo WAL, o RDS para PostgreSQL pode fazer o seguinte:

- Recupere dados após uma falha
- Reduzir a E/S de disco, evitando gravações frequentes em disco

Quando um cliente altera dados, o RDS para PostgreSQL grava as alterações no buffer de WAL. Quando o cliente emite um `COMMIT`, o processo gravador WAL grava dados de transação no arquivo de WAL.

O parâmetro `wal_level` determina quantas informações são gravadas no WAL, com valores possíveis como `minimal`, `replica` e `logical`.

Memória local no RDS para PostgreSQL

Todo processo de backend aloca memória local para processamento de consultas.

Tópicos

- Área de memória de trabalho
- Área de memória de trabalho para manutenção
- Área de buffer temporária

Área de memória de trabalho

A área de memória de trabalhocontém dados temporários para consultas que executam classificações e hashes. Por exemplo, uma consulta com uma cláusula `ORDER BY` executa uma classificação. Consultas usam tabelas de hash em agregações e junções de hash.

O parâmetro `work_mem` é a quantidade de memória a ser utilizada por operações de classificação internas e tabelas de hash antes da gravação em arquivos de disco temporários, o que é medido em megabytes. O valor padrão é 4 MB. Várias sessões podem ser executadas simultaneamente, e cada uma pode executar operações de manutenção em paralelo. Por esse motivo, a memória de trabalho total utilizada pode ser múltiplos da configuração `work_mem`.

Área de memória de trabalho para manutenção

A área de memória de trabalho para manutenção armazena dados em cache para operações de manutenção. Essas operações incluem aspiração, criação de índices e adição de chaves externas.

O parâmetro `maintenance_work_mem` especifica a quantidade máxima de memória a ser utilizada por operações de manutenção, o que é medido em megabytes. O valor padrão é 64 MB. Uma sessão de banco de dados apenas pode executar uma operação de manutenção de cada vez.

Área de buffer temporária

A área de buffer temporária armazena tabelas temporárias em cache para cada sessão de banco de dados.

Cada sessão aloca buffers temporários conforme necessário até o limite especificado. Quando a sessão termina, o servidor limpa os buffers.

O parâmetro `temp_buffers` define o número máximo de buffers temporários utilizados por cada sessão, o que é medido em megabytes. O valor padrão é de 8 MB. Antes do primeiro uso de tabelas temporárias em uma sessão, é possível alterar o valor de `temp_buffers`.

Processo do RDS para PostgreSQL

O RDS para PostgreSQL utiliza vários processos.

Tópicos

- Processo Postmaster
- Processos de backend

- Processos em segundo plano

Processo Postmaster

O processo de postmaster é o primeiro a ser iniciado quando você inicia o RDS para PostgreSQL. Ele tem as seguintes responsabilidades principais:

- Bifurcar e monitorar processos em segundo plano
- Receba solicitações de autenticação dos processos do cliente e autentique-as antes de permitir que o banco de dados atenda às solicitações

Processos de backend

Se o postmaster autenticar uma solicitação de cliente, o postmaster bifurcará um novo processo de backend, também chamado de processo postgres. Um processo de cliente conecta-se exatamente a um processo de backend. O processo de cliente e o processo de backend se comunicam diretamente sem a intervenção do processo postmaster.

Processos em segundo plano

O processo postmaster bifurca vários processos que realizam diferentes tarefas de backend. Alguns dos mais importantes incluem:

- Gravador WAL

  O RDS para PostgreSQL grava dados no buffer de WAL (gravação antecipada) nos arquivos de log. O princípio do registro em log de gravação antecipada determina que o banco de dados não pode gravar alterações nos arquivos de dados até que o banco de dados grave registros de log descrevendo essas alterações no disco. O mecanismo WAL reduz a E/S do disco e permite que o RDS para PostgreSQL utilize os logs para recuperar o banco de dados após uma falha.
- Gravador em segundo plano

  Esse processo grava periodicamente páginas sujas (modificadas) dos buffers de memória nos arquivos de dados. Uma página fica suja quando um processo de backend a modifica na memória.
- Daemon autovacuum

  O daemon consiste no seguinte:

  - O launcher de autovacuum

- Os processos de operador de autovacuum

Quando o autovacuum está ativado, ele procura tabelas que tiveram um grande número de tuplas inseridas, atualizadas ou excluídas. Esse daemon tem as seguintes responsabilidades:

- Recuperar ou reutilizar o espaço em disco ocupado por linhas atualizadas ou excluídas
- Atualizar estatísticas utilizadas pelo planejador
- Proteger contra a perda de dados antigos devido à recorrência de IDs de transação

O recurso e autovacuum automatiza a execução de comandos `VACUUM` e `ANALYZE`. `VACUUM` tem as seguintes variantes: padrão e completo. O vacuum padrão é executado em paralelo com outras operações de banco de dados. `VACUUM FULL` requer um bloqueio exclusivo na tabela em que está trabalhando. Portanto, ele não pode ser executado em paralelo com operações que acessam a mesma tabela. `VACUUM` cria uma quantidade substancial de tráfego de E/S, podendo piorar a performance para outras sessões ativas.

Eventos de espera do RDS para PostgreSQL

A tabela a seguir lista os eventos de espera do RDS para PostgreSQL que indicam mais comumente problemas de performance e resume as causas e as medidas corretivas mais comuns.

Eventos de espera	Definição
Client:ClientRead	Esse evento ocorre quando o RDS para PostgreSQL está aguardando para receber dados do cliente.
Client:ClientWrite	Esse evento ocorre quando o RDS para PostgreSQL está aguardando para gravar dados no cliente.
CPU	Ocorre quando um thread está ativo na CPU ou está aguardando a CPU.
IO:BufFileRead and IO:BufFileWrite	Esses eventos ocorrem quando o RDS para PostgreSQL cria arquivos temporários.
IO:DataFileRead	Esse evento ocorre quando uma conexão aguarda em um processo de backend para ler uma página

Eventos de espera	Definição
	necessária do armazenamento porque essa página não está disponível na memória compartilhada.
IO:WALWrite	Esse evento ocorre quando o RDS para PostgreSQL está aguardando os buffers do log de gravação antecipada (WAL) serem gravados em um arquivo WAL.
Lock:advisory	Esse evento ocorre quando uma aplicação PostgreSQL utiliza um bloqueio para coordenar as atividades em várias sessões.
Lock:extend	Esse evento ocorre quando um processo de backend está aguardando para bloquear uma relação com o objetivo de a estender, enquanto outro processo tem um bloqueio nessa relação para o mesmo objetivo.
Lock:Relation	Esse evento ocorre quando uma consulta está aguardando para adquirir um bloqueio em uma tabela ou visualização que está atualmente bloqueada por outra transação.
Lock:transactionid	Esse evento ocorre quando uma transação está aguardando um bloqueio em nível de linha.
Lock:tuple	Esse evento ocorre quando um processo de backend está aguardando para adquirir um bloqueio em uma tupla.
LWLock:BufferMapping (LWLock:buffer_mapping)	Esse evento ocorre quando uma sessão está aguardando para associar um bloco de dados a um buffer no grupo de buffer compartilhado.

Eventos de espera	Definição
LWLock:BufferIO (IPC:BufferIO)	Esse evento ocorre quando o RDS para PostgreSQL está aguardando que outros processos terminem suas operações de entrada/saída (E/S) ao tentarem acessar simultaneamente uma página.
LWLock:buffer_content (BufferContent)	Esse evento ocorre quando uma sessão aguarda para ler ou gravar uma página de dados na memória enquanto outra sessão fica com a página bloqueada para gravação.
LWLock:lock_manager (LWLock:lockmanager)	Esse evento ocorre quando o mecanismo do RDS para PostgreSQL mantém a área de memória do bloqueio compartilhado para alocar, conferir e desalocar um bloqueio nos casos em que um bloqueio de caminho rápido não é possível.
Tempo limite:PgSleep	Esse evento ocorre quando um processo do servidor chama a função `pg_sleep` e está aguardando o tempo limite de suspensão expirar.
Timeout:VacuumDelay	Esse evento indica que o processo de vácuo está inativo porque o limite de custo estimado foi atingido.

Client:ClientRead

O evento `Client:ClientRead` ocorre quando o RDS para PostgreSQL está aguardando receber dados do cliente.

Tópicos

- Versões compatíveis do mecanismo
- Contexto
- Possíveis causas do maior número de esperas
- Ações

Versões compatíveis do mecanismo

Essas informações sobre eventos de espera são compatíveis com o RDS para PostgreSQL versão 10 e superiores.

Contexto

Uma instância de banco de dados do RDS para PostgreSQL está aguardando para receber dados do cliente. A instância de banco de dados do RDS para PostgreSQL deve receber os dados do cliente para poder enviar mais dados ao cliente. O tempo de espera da instância antes de receber dados do cliente é um evento `Client:ClientRead`.

Possíveis causas do maior número de esperas

Causas comuns do surgimento do evento `Client:ClientRead` nas principais esperas incluem:

Maior latência de rede

Pode haver maior latência de rede entre a instância de banco de dados do RDS para PostgreSQL e o cliente. A latência de rede mais alta aumenta o tempo necessário para a instância de banco de dados receber dados do cliente.

Maior carga no cliente

Pode haver pressão da CPU ou saturação da rede no lado do cliente. Um aumento na carga no cliente pode atrasar a transmissão de dados do cliente para a instância de banco de dados do RDS para PostgreSQL.

Excesso de viagens de ida e volta na rede

Um número elevado de viagens de ida e volta na rede entre a instância de banco de dados do RDS para PostgreSQL e o cliente pode atrasar a transmissão de dados do cliente para a instância do RDS para PostgreSQL.

Operação de cópia extensa

Durante uma operação de cópia, os dados são transferidos do sistema de arquivos do cliente para a instância de banco de dados do RDS para PostgreSQL. O envio de uma muitos dados para a instância de banco de dados pode atrasar a transmissão de dados do cliente para a instância de banco de dados.

Desconexão de um cliente inativo

Quando um cliente se conecta à instância de banco de dados do RDS para PostgreSQL em um estado `idle in transaction`, a instância de banco de dados pode esperar que o cliente envie mais dados ou emitir um comando. Uma conexão nesse estado pode resultar no aumento de eventos `Client:ClientRead`.

PgBouncer utilizado para agrupamento de conexões

PgBouncer tem uma configuração de rede de baixo nível chamada `pkt_buf` e que está definida como 4.096 por padrão. Se a workload estiver enviando pacotes de consulta maiores que 4.096 bytes por meio de PgBouncer, convém aumentar a configuração `pkt_buf` para 8.192. Se a nova configuração não diminuir o número de eventos `Client:ClientRead`, convém aumentar a configuração `pkt_buf` para valores maiores, como 16.384 ou 32.768. Se o texto da consulta for grande, a configuração maior pode ser particularmente útil.

Ações

Recomenda-se ações distintas, dependendo dos motivos do evento de espera.

Tópicos

- Colocar os clientes na mesma zona de disponibilidade e sub-rede da VPC que a instância
- Escalar seu cliente
- Utilizar instâncias da geração atual
- Aumentar a largura de banda da rede
- Monitorar máximos de performance da rede
- Monitorar transações no estado de “inatividade em transação”

Colocar os clientes na mesma zona de disponibilidade e sub-rede da VPC que a instância

Para reduzir a latência e aumentar o throughput da rede, coloque clientes na mesma zona de disponibilidade e na mesma sub-rede de nuvem privada virtual (VPC) que a instância de banco de dados do RDS para PostgreSQL. Os clientes devem estar o mais próximos possível da instância de banco de dados.

Escalar seu cliente

Utilizando o Amazon CloudWatch ou outras métricas de host, determine se o cliente está atualmente restrito pela CPU ou pela largura de banda da rede, ou por ambas. Se o cliente estiver restrito, escale-o de acordo.

Utilizar instâncias da geração atual

Em alguns casos, talvez você não esteja utilizando uma classe de instância de banco de dados que ofereça suporte a quadros jumbo. Se estiver executando sua aplicação no Amazon EC2, considere utilizar uma instância de geração atual para o cliente. Além disso, configure a MTU (unidade de transmissão máxima) no sistema operacional cliente. Essa técnica pode reduzir o número de idas e voltas pela rede e aumentar a taxa de transferência da rede. Consulte mais informações em Frames jumbo (9001 MTU) no Guia do usuário do Amazon EC2.

Para obter informações sobre classes de instância de banco de dados, consulte Classes de instâncias de banco de dados . Para determinar a classe de instância de banco de dados equivalente a um tipo de instância do Amazon EC2, coloque `db.` antes do nome do tipo de instância do Amazon EC2. Por exemplo, a instância `r5.8xlarge` do Amazon EC2 é equivalente à classe de instância de banco de dados `db.r5.8xlarge`.

Aumentar a largura de banda da rede

Utilize as métricas `NetworkReceiveThroughput` e `NetworkTransmitThroughput` do Amazon CloudWatch para monitorar o tráfego de rede de entrada e saída na instância de banco de dados. Essas métricas podem ajudar você a determinar se a largura de banda da rede é suficiente para a sua workload.

Se a largura de banda da rede não suficiente, aumente-a. Se o cliente AWS ou sua instância de banco de dados estiver atingindo os limites de largura de banda da rede, a única maneira de aumentar a largura de banda será ampliar o tamanho da instância de banco de dados. Para ter mais informações, consulte Tipos de classe de instância de banco de dados.

Para ter mais informações sobre métricas do CloudWatch, consulte Métricas do Amazon CloudWatch para o Amazon RDS.

Monitorar máximos de performance da rede

Se você utiliza clientes do Amazon EC2, o Amazon EC2 fornece máximos para métricas de performance da rede, incluindo largura de banda de rede agregada de entrada e saída. Ele também fornece rastreamento de conexões para garantir que os pacotes sejam retornados conforme

esperado e vinculem localmente o acesso para serviços como o Sistema de Nomes de Domínio (DNS). Para monitorar esses máximos, utilize um driver de rede avançado atual e monitore a performance de rede do seu cliente.

Consulte mais informações em Monitorar a performance de rede de sua instância do EC2 no Guia do usuário do Amazon EC2.

Monitorar transações no estado de “inatividade em transação”

Verifique se você tem um número cada vez maior de conexões `idle in transaction`. Para isso, monitore a coluna `state` na tabela `pg_stat_activity`. Talvez seja possível identificar a origem da conexão executando uma consulta semelhante à seguinte.

```
select client_addr, state, count(1) from pg_stat_activity
where state like 'idle in transaction%'
group by 1,2
order by 3 desc
```

Client:ClientWrite

O evento `Client:ClientWrite` ocorre quando o RDS para PostgreSQL está aguardando para gravar dados no cliente.

Tópicos

- Versões compatíveis do mecanismo
- Contexto
- Possíveis causas do maior número de esperas
- Ações

Versões compatíveis do mecanismo

Essas informações sobre eventos de espera são compatíveis com o RDS para PostgreSQL versão 10 e superiores.

Contexto

Um processo de cliente deve ler todos os dados recebidos de um cluster de banco de dados do RDS para PostgreSQL para que o cluster possa enviar mais dados. O tempo de espera do cluster antes de enviar mais dados ao cliente é um evento `Client:ClientWrite`.

O throughput reduzido da rede entre a instância de banco de dados do RDS para PostgreSQL e o cliente pode causar esse evento. A pressão da CPU e a saturação da rede no cliente também podem causar esse evento. Pressão da CPU é quando a CPU está totalmente utilizada e existem tarefas aguardando o tempo da CPU. Saturação de rede é quando a rede entre o banco de dados e o cliente está transportando mais dados do que ela pode manipular.

Possíveis causas do maior número de esperas

Causas comuns do surgimento do evento `Client:ClientWrite` nas principais esperas incluem:

Maior latência de rede

Pode haver maior latência de rede entre a instância de banco de dados do RDS para PostgreSQL e o cliente. A latência de rede mais alta aumenta o tempo necessário para o cliente receber os dados.

Maior carga no cliente

Pode haver pressão da CPU ou saturação da rede no lado do cliente. Um aumento na carga do cliente atrasa o recebimento de dados da instância de banco de dados do RDS para PostgreSQL.

Grande volume de dados enviados ao cliente

A instância de banco de dados do RDS para PostgreSQL pode estar enviando uma grande quantidade de dados ao cliente. É possível que um cliente não consiga receber os dados tão rápido quanto o cluster os está enviando. Atividades como cópias de tabelas grandes podem resultar no aumento de eventos `Client:ClientWrite`.

Ações

Recomenda-se ações distintas, dependendo dos motivos do evento de espera.

Tópicos

- Colocar os clientes na mesma zona de disponibilidade e sub-rede VPC que o cluster
- Utilizar instâncias da geração atual
- Reduzir a quantidade de dados enviados ao cliente
- Escalar seu cliente

Colocar os clientes na mesma zona de disponibilidade e sub-rede VPC que o cluster

Para reduzir a latência e aumentar o throughput da rede, coloque clientes na mesma zona de disponibilidade e na mesma sub-rede de nuvem privada virtual (VPC) que a instância de banco de dados do RDS para PostgreSQL.

Utilizar instâncias da geração atual

Em alguns casos, talvez você não esteja utilizando uma classe de instância de banco de dados que ofereça suporte a quadros jumbo. Se estiver executando sua aplicação no Amazon EC2, considere utilizar uma instância de geração atual para o cliente. Além disso, configure a MTU (unidade de transmissão máxima) no sistema operacional cliente. Essa técnica pode reduzir o número de idas e voltas pela rede e aumentar a taxa de transferência da rede. Consulte mais informações em Frames jumbo (9001 MTU) no Guia do usuário do Amazon EC2.

Para obter informações sobre classes de instância de banco de dados, consulte Classes de instâncias de banco de dados . Para determinar a classe de instância de banco de dados equivalente a um tipo de instância do Amazon EC2, coloque `db.` antes do nome do tipo de instância do Amazon EC2. Por exemplo, a instância `r5.8xlarge` do Amazon EC2 é equivalente à classe de instância de banco de dados `db.r5.8xlarge`.

Reduzir a quantidade de dados enviados ao cliente

Quando possível, ajuste a aplicação para reduzir a quantidade de dados que a instância de banco de dados do RDS para PostgreSQL envia ao cliente. Fazer esses ajustes alivia a contenção da CPU e da rede no lado do cliente.

Escalar seu cliente

Utilizando o Amazon CloudWatch ou outras métricas de host, determine se o cliente está atualmente restrito pela CPU ou pela largura de banda da rede, ou por ambas. Se o cliente estiver restrito, escale-o de acordo.

CPU

Ocorre quando um thread está ativo na CPU ou está aguardando a CPU.

Tópicos

- Versões compatíveis do mecanismo
- Contexto

- Possíveis causas do maior número de esperas
- Ações

Versões compatíveis do mecanismo

Essas informações de evento de espera são relevantes para todas as versões do RDS para PostgreSQL.

Contexto

A unidade de processamento central (CPU) é o componente de um computador que executa instruções. Por exemplo, instruções de CPU realizam operações aritméticas e trocam dados na memória. Se uma consulta aumentar o número de instruções que ela executa por meio do mecanismo de banco de dados, o tempo gasto na execução dessa consulta aumentará. Programação da CPU refere-se a alocar tempo de CPU a um processo. A programação é orquestrada pelo kernel do sistema operacional.

Tópicos

- Como saber quando essa espera ocorre
- Métrica DBLoadCPU
- Métricas os.cpuUtilization
- Provável causa da programação da CPU

Como saber quando essa espera ocorre

Esse evento de espera `CPU` indica que um processo de backend está ativo na CPU ou aguardando a CPU. É possível determinar que isso está ocorrendo quando uma consulta mostra as seguintes informações:

- A coluna `pg_stat_activity.state` tem o valor `active`.
- As colunas `wait_event_type` e `wait_event` em `pg_stat_activity` são ambas `null`.

Para ver os processos de backend que estão utilizando ou aguardando CPU, execute a seguinte consulta.

```
SELECT *
```

```
FROM   pg_stat_activity
WHERE  state = 'active'
AND    wait_event_type IS NULL
AND    wait_event IS NULL;
```

Métrica DBLoadCPU

A métrica do Performance Insights para CPU é `DBLoadCPU`. O valor de `DBLoadCPU` pode diferir do valor da métrica `CPUUtilization` do Amazon CloudWatch. A última métrica é coletada do HyperVisor para uma instância de banco de dados.

Métricas os.cpuUtilization

As métricas do Performance Insights para o sistema operacional fornecem informações detalhadas sobre a utilização da CPU. Por exemplo, é possível exibir as seguintes métricas:

- `os.cpuUtilization.nice.avg`
- `os.cpuUtilization.total.avg`
- `os.cpuUtilization.wait.avg`
- `os.cpuUtilization.idle.avg`

O Performance Insights relata o uso da CPU pelo mecanismo de banco de dados como `os.cpuUtilization.nice.avg`.

Provável causa da programação da CPU

O kernel do sistema operacional (SO) lida com agendamento para a CPU. Quando a CPU está ativa, talvez um processo precise esperar para ser agendado. A CPU está ativa enquanto executa cálculos. Também está ativa enquanto está com um thread ocioso que não está em execução, ou seja, um thread ocioso que está aguardando a E/S de memória. Esse tipo de E/S domina a workload típica de banco de dados.

É provável que os processos aguardem para serem programados em uma CPU quando as seguintes condições forem atendidas:

- A métrica `CPUUtilization` do CloudWatch está próxima dos 100%.
- A carga média é maior que o número de vCPUs, indicando uma carga pesada. Você pode encontrar a métrica `loadAverageMinute` na seção de métricas do sistema operacional do Performance Insights.

Possíveis causas do maior número de esperas

Quando o evento de espera de CPU ocorre mais que o normal, possivelmente indicando um problema de performance, as causas típicas incluem:

Tópicos

- Possíveis causas de picos súbitos
- Possíveis causas de alta frequência a longo prazo
- Casos excepcionais

Possíveis causas de picos súbitos

As causas mais prováveis de picos súbitos são as seguintes:

- Sua aplicação abriu muitas conexões simultâneas com o banco de dados. Esse cenário é conhecido como “tempestade de conexões”.
- A workload da sua aplicação foi alterada de qualquer uma das seguintes maneiras:
  - Novas consultas
  - Um aumento no tamanho do conjunto de dados
  - Manutenção ou criação de índices
  - Novas funções
  - Novos operadores
  - Um aumento na execução de consultas paralelas
- Seus planos de execução de consultas foram modificados. Em certos casos, uma alteração pode causar um aumento nos buffers. Por exemplo, a consulta agora está utilizando uma varredura sequencial quando utilizava anteriormente um índice. Nesse caso, ela precisa de mais CPU para atingir o mesmo objetivo.

Possíveis causas de alta frequência a longo prazo

As causas mais prováveis de eventos que se repetem por um longo período são:

- Muitos processos de backend estão em execução simultaneamente na CPU. Esses processos podem ser operadores paralelos.
- Consultas estão sendo executadas com performance abaixo do ideal porque precisam de um grande número de buffers.

Casos excepcionais

Se nenhuma das causas prováveis revelarem ser causas reais, as seguintes situações podem estar ocorrendo:

- A CPU está alternando processos para dentro e para fora.
- A CPU poderá estar gerenciando as entradas da tabela de páginas se o recurso de páginas enormes estiver desativado. Esse recurso de gerenciamento de memória é ativado por padrão para todas as classes de instância de banco de dados que não sejam classes de instância de banco de dados micro, pequenas e médias. Para obter mais informações, consulte Páginas grandes para RDS para PostgreSQL .

Ações

Se o evento de espera `CPU` domina a atividade do banco de dados, isso não indica necessariamente um problema de performance. Responda a esse evento somente quando a performance diminuir.

Tópicos

- Investigar se o banco de dados está causando o aumento da CPU
- Determinar se o número de conexões aumentou
- Responder a alterações de workload

Investigar se o banco de dados está causando o aumento da CPU

Examine a métrica `os.cpuUtilization.nice.avg` no Performance Insights. Se esse valor for muito menor que o uso da CPU, processos que não são do banco de dados são os principais colaboradores para a CPU.

Determinar se o número de conexões aumentou

Examine a métrica `DatabaseConnections` no Amazon CloudWatch. Sua ação depende se o número aumentou ou diminuiu durante o período de maior número de eventos de espera de CPU.

As conexões aumentaram

Se o número de conexões aumentou, compare o número de processos de backend que consumem CPU com o número de vCPUs. Os cenários a seguir são possíveis:

- O número de processos de backend que consomem CPU é menor que o número de vCPUs.

Nesse caso, o número de conexões não é um problema. Porém, você ainda pode tentar reduzir a utilização da CPU.

- O número de processos de backend que consomem CPU é maior que o número de vCPUs.

  Nesse caso, considere as opções a seguir:

  - Diminua o número de processos de backend conectados ao banco de dados. Por exemplo, implemente uma solução de agrupamento de conexões, como o RDS Proxy. Para saber mais, consulte Amazon RDS Proxy.
  - Atualize o tamanho da sua instância para obter um número maior de vCPUs.
  - Se aplicável, redirecione algumas workloads somente leitura para nós de leitor.

As conexões não aumentaram

Examine as métricas `blks_hit` no Performance Insights. Procure uma correlação entre um aumento em `blks_hit` e o uso da CPU. Os cenários a seguir são possíveis:

- O uso da CPU e `blks_hit` estão correlacionados.

  Nesse caso, encontre as principais instruções SQL vinculadas ao uso da CPU e procure modificações no plano. Você pode usar uma das seguintes técnicas:

  - Explicar os planos manualmente e compare-os com o plano de execução esperado.
  - Procurar um aumento nos acertos de bloco por segundo e nos acertos de blocos locais por segundo. Na seção Top SQL (SQL principal) do painel do Performance Insights, escolha Preferences (Preferências).
- O uso da CPU e `blks_hit` não estão correlacionados.

  Nesse caso, determine se alguma das seguintes situações ocorre:

  - A aplicação está se conectando e se desconectando rapidamente ao/do banco de dados.

    Diagnostique esse comportamento ativando `log_connections` e `log_disconnections` e analisando os logs do PostgreSQL. Considere utilizar o analisador de logs `pgbadger`. Para obter mais informações, consulte https://github.com/darold/pgbadger.
  - O sistema operacional está sobrecarregado.

    Nesse caso, o Performance Insights mostra que processos de backend estão consumindo CPU por mais tempo que o normal. Procure evidências nas métricas `os.cpuUtilization`

do Performance Insights ou na métrica `CPUUtilization` do CloudWatch. Se o sistema operacional estiver sobrecarregado, consulte as métricas de monitoramento avançado para aprofundar o diagnóstico. Especificamente, observe a lista de processos e a porcentagem de CPU consumida por cada um.

- As principais instruções SQL estão consumindo muita CPU.

  Examine instruções que estão vinculadas ao uso da CPU para verificar se elas podem utilizar menos CPU. Execute um comando `EXPLAIN` e concentre-se nos nós do plano que têm o maior impacto. Considere utilizar um visualizador de plano de execução do PostgreSQL. Para experimentar essa ferramenta, consulte http://explain.dalibo.com/.

Responder a alterações de workload

Se a sua workload mudou, procure os seguintes tipos de alterações:

Novas consultas

Verifique se novas consultas são esperadas. Em caso positivo, verifique se os planos de execução dessas consultas e o número de execuções por segundo são os esperados.

Um aumento no tamanho do conjunto de dados

Determine se o particionamento, caso ainda não esteja implementado, pode ajudar. Essa estratégia é capaz de reduzir o número de páginas que uma consulta precisa recuperar.

Manutenção ou criação de índices

Verifique se a programação de manutenção é a esperada. Uma prática recomendada é agendar atividades de manutenção fora das atividades de pico.

Novas funções

Confirme se essas funções são executadas conforme o esperado durante o teste. Especificamente, confira se o número de execuções por segundo é o esperado.

Novos operadores

Verifique se eles funcionam conforme o esperado durante os testes.

Um aumento na execução de consultas paralelas

Determine se alguma das situações a seguir ocorreu:

- As relações ou os índices envolvidos cresceram de repente a ponto de diferirem significativamente de `min_parallel_table_scan_size` ou `min_parallel_index_scan_size`.
- Alterações recentes foram feitas em `parallel_setup_cost` ou `parallel_tuple_cost`.
- Alterações recentes foram feitas em `max_parallel_workers` ou `max_parallel_workers_per_gather`.

IO:BufFileRead and IO:BufFileWrite

Os eventos `IO:BufFileRead` e `IO:BufFileWrite` ocorrem quando o RDS para PostgreSQL cria arquivos temporários. Quando as operações requerem mais memória do que os parâmetros de memória de trabalho definidos atualmente, elas gravam dados temporários no armazenamento persistente. Essa operação é chamada às vezes de “derramamento no disco”.

Tópicos

- Versões compatíveis do mecanismo
- Contexto
- Possíveis causas do maior número de esperas
- Ações

Versões compatíveis do mecanismo

Essas informações de eventos de espera são compatíveis com todas as versões do RDS para PostgreSQL.

Contexto

`IO:BufFileRead` e `IO:BufFileWrite` estão relacionados à área de memória de trabalho e a área de memória de trabalho de manutenção. Para obter mais informações sobre essas áreas de memória local, consulte Resource Consumption (Consumo de recursos) na documentação do PostgreSQL.

O valor padrão para `work_mem` é 4 MB. Se uma sessão executar operações em paralelo, cada operador que lidar com o paralelismo usará 4 MB de memória. Por essa razão, defina `work_mem` com cautela. Se você aumentar demais esse valor, um banco de dados que execute muitas sessões poderá consumir muita memória. Se você definir um valor muito baixo, o RDS para PostgreSQL

criará arquivos temporários no armazenamento local. A E/S de disco desses arquivos temporários pode reduzir a performance.

Se você observar a seguinte sequência de eventos, é possível que seu banco de dados esteja gerando arquivos temporários:

1. Redução súbita e acentuada na disponibilidade
2. Recuperação rápida para o espaço livre

Você também pode observar um padrão de “motosserra”. Esse padrão pode indicar que o banco de dados está criando arquivos pequenos constantemente.

Possíveis causas do maior número de esperas

Em geral, esses eventos de espera são causados por operações que consomem mais memória do que é alocado pelos parâmetros `work_mem` ou `maintenance_work_mem`. Para compensar isso, as operações gravam em arquivos temporários. Causas comuns dos eventos `IO:BufFileRead` e `IO:BufFileWrite` incluem:

Consultas que necessitam de mais memória do que existe na área de memória de trabalho

Consultas com as seguintes características utilizam a área de memória de trabalho:

- Junções de hash
- `ORDER BY`Cláusula
- `GROUP BY`Cláusula
- `DISTINCT`
- Funções de janela
- `CREATE TABLE AS SELECT`
- Atualização de visualizações materializadas

Instruções que necessitam de mais memória do que existe na área de memória do trabalho de manutenção

As seguintes instruções usam a área de memória do trabalho de manutenção:

- `CREATE INDEX`

- `CLUSTER`

Ações

Recomenda-se ações distintas, dependendo dos motivos do evento de espera.

Tópicos

- Identificar o problema
- Examinar suas consultas de junção
- Examinar suas consultas ORDER BY e GROUP BY
- Evite utilizar a operação DISTINCT
- Considere utilizar funções de janela em vez de funções GROUP BY
- Investigar visualizações materializadas e instruções CTAS
- Utilizar pg_repack ao recriar índices
- Aumentar maintenance_work_mem ao agrupar tabelas
- Ajustar a memória para evitar IO:BufFileRead e IO:BufFileWrite

Identificar o problema

Imagine uma situação na qual o Performance Insights não está ativado e você suspeita de que `IO:BufFileRead` e `IO:BufFileWrite` estejam ocorrendo com mais frequência que o normal. Para identificar a origem do problema, você pode definir o parâmetro `log_temp_files` para registrar todas as consultas que geram mais do que o limite especificado de KB de arquivos temporários. Por padrão,`log_temp_files` está definido como `-1`, o que desativa esse recurso de registro em log. Se você definir esse parâmetro como `0`, o RDS para PostgreSQL registrará todos os arquivos temporários. Se você defini-lo como `1024`, o RDS para PostgreSQL registrará todas as consultas que geram arquivos temporários maiores que 1 MB. Para obter mais informações sobre `log_temp_files`, consulte o tópico sobre Relatórios de erros e registro em log, na documentação do PostgreSQL.

Examinar suas consultas de junção

É provável que sua consulta use junções. Por exemplo, a consulta a seguir une quatro tabelas.

```
SELECT *
```

```
       FROM "order"
 INNER JOIN order_item
       ON (order.id = order_item.order_id)
 INNER JOIN customer
       ON (customer.id = order.customer_id)
 INNER JOIN customer_address
       ON (customer_address.customer_id = customer.id AND
           order.customer_address_id = customer_address.id)
 WHERE customer.id = 1234567890;
```

Uma causa possível dos picos no uso temporário de arquivos é um problema na própria consulta. Por exemplo, uma cláusula quebrada talvez não esteja filtrando as junções corretamente. Considere a segunda junção interna no exemplo a seguir.

```
SELECT *
       FROM "order"
 INNER JOIN order_item
       ON (order.id = order_item.order_id)
 INNER JOIN customer
       ON (customer.id = customer.id)
 INNER JOIN customer_address
       ON (customer_address.customer_id = customer.id AND
           order.customer_address_id = customer_address.id)
 WHERE customer.id = 1234567890;
```

A consulta anterior junta `customer.id` com `customer.id` por engano, gerando um produto cartesiano entre cada cliente e cada pedido. Esse tipo de junção acidental gera arquivos temporários grandes. Dependendo do tamanho das tabelas, uma consulta cartesiana pode até mesmo lotar o armazenamento. Sua aplicação pode ter junções cartesianas quando as seguintes condições são atendidas:

- Você percebe reduções grandes e acentuadas na disponibilidade do armazenamento, seguidas de uma rápida recuperação.
- Nenhum índice está sendo criado.
- Nenhuma instrução `CREATE TABLE FROM SELECT` está sendo emitida.
- Nenhuma visualização materializada está sendo atualizada.

Para verificar se as tabelas estão sendo unidas utilizando as chaves apropriadas, inspecione suas diretivas de mapeamento de consultas e objetos relacionais. Lembre-se de que certas

consultas da sua aplicação não são chamadas o tempo todo e que algumas consultas são geradas dinamicamente.

Examinar suas consultas ORDER BY e GROUP BY

Em alguns casos, uma cláusula `ORDER BY` pode resultar no excesso de arquivos temporários. Considere as seguintes diretrizes:

- Inclua somente colunas em uma cláusula `ORDER BY` quando elas precisarem ser ordenadas. Essa orientação é especialmente importante para consultas que retornam milhares de linhas e especificam muitas colunas na cláusula `ORDER BY`.
- Considere criar índices para acelerar cláusulas `ORDER BY` quando elas correspondem a colunas que tenham a mesma ordem crescente ou decrescente. Índices parciais são preferíveis, pois são menores. Índices menores são lidos e percorridos com mais rapidez.
- Se você criar índices para colunas que podem aceitar valores nulos, considere se deseja que esses valores nulos sejam armazenados no final ou no início dos índices.

  Se possível, reduza o número de linhas que precisam ser ordenadas, filtrando o conjunto de resultados. Se você usar instruções de cláusula `WITH` ou subconsultas, lembre-se de que uma consulta interna gera um conjunto de resultados e o transmite à consulta externa. Quanto mais linhas uma consulta puder remover, menos ordenação ela precisará fazer.
- Se não precisar obter o conjunto completo de resultados, utilize a cláusula `LIMIT`. Por exemplo, se quiser apenas as cinco principais linhas, uma consulta utilizando a cláusula `LIMIT` não continuará gerando resultados. Dessa forma, essa consulta requer menos memória e arquivos temporários.

Uma consulta que usa uma cláusula `GROUP BY` também pode exigir arquivos temporários. Consultas `GROUP BY` resumem valores utilizando funções como as seguintes:

- `COUNT`
- `AVG`
- `MIN`
- `MAX`
- `SUM`
- `STDDEV`

Para ajustar consultas `GROUP BY`, siga as recomendações para consultas `ORDER BY`.

Evite utilizar a operação DISTINCT

Se possível, evite utilizar a operação `DISTINCT` para remover linhas duplicadas. Quanto mais linhas desnecessárias e duplicadas sua consulta retornar, mais cara a operação `DISTINCT` se tornará. Se possível, adicione filtros à cláusula `WHERE` mesmo que você utilize os mesmos filtros para tabelas diferentes. Filtrar a consulta e a junção corretamente melhora a performance e reduz o uso de recursos. Isso também evita relatórios e resultados incorretos.

Se precisar usar `DISTINCT` para várias linhas de uma mesma tabela, considere criar um índice composto. O agrupamento de várias colunas em um índice pode melhorar o tempo para avaliar linhas distintas. Além disso, se utilizar o RDS para PostgreSQL versão 10 ou superior, você poderá correlacionar estatísticas entre várias colunas utilizando o comando `CREATE STATISTICS`.

Considere utilizar funções de janela em vez de funções GROUP BY

Usando `GROUP BY`, você altera o conjunto de resultados e, em seguida, recupera o resultado agregado. Usando funções de janela, você agrega dados sem modificar o conjunto de resultados. Uma função de janela usa a cláusula `OVER` para fazer cálculos entre os conjuntos definidos pela consulta, correlacionando uma linha com outra. Você pode utilizar todas as funções `GROUP BY` em funções de janela, mas também utilizar funções como as seguintes:

- `RANK`
- `ARRAY_AGG`
- `ROW_NUMBER`
- `LAG`
- `LEAD`

Para minimizar o número de arquivos temporários gerados por uma função de janela, remova duplicatas do mesmo conjunto de resultados quando precisar de duas agregações distintas. Considere a seguinte consulta.

```
SELECT sum(salary) OVER (PARTITION BY dept ORDER BY salary DESC) as sum_salary
     , avg(salary) OVER (PARTITION BY dept ORDER BY salary ASC) as avg_salary
  FROM empsalary;
```

Você pode reescrever essa consulta com a cláusula `WINDOW` da seguinte maneira.

```
SELECT sum(salary) OVER w as sum_salary
```

```
        , avg(salary) OVER w as_avg_salary
    FROM empsalary
WINDOW w AS (PARTITION BY dept ORDER BY salary DESC);
```

Por padrão, o planejador de execução do RDS para PostgreSQL consolida nós semelhantes para que ele não duplique operações. No entanto, utilizando uma declaração explícita para o bloco de janelas, é possível manter a consulta com mais facilidade. Também é possível melhorar a performance ao evitar a duplicação.

Investigar visualizações materializadas e instruções CTAS

Quando uma visualização materializada é atualizada, ela executa uma consulta. Essa consulta pode conter uma operação como `GROUP BY`, `ORDER BY` ou `DISTINCT`. Durante uma atualização, é possível observar um grande número de arquivos temporários e os eventos de espera `IO:BufFileWrite` e `IO:BufFileRead`. Da mesma forma, quando você cria uma tabela com base em uma instrução `SELECT`, a instrução `CREATE TABLE` executa uma consulta. Para reduzir os arquivos temporários necessários, otimize a consulta.

Utilizar pg_repack ao recriar índices

Quando você cria um índice, o mecanismo ordena o conjunto de resultados. À medida que o tamanho das tabelas aumenta e à medida que os valores na coluna indexada se tornam mais diversificados, os arquivos temporários exigem mais espaço. Na maioria dos casos, não é possível impedir a criação de arquivos temporários para tabelas grandes sem modificar a área de memória do trabalho de manutenção. Para obter mais informações sobre `maintenance_work_mem`, consulte https://www.postgresql.org/docs/current/runtime-config-resource.html na documentação do PostgreSQL.

Uma possível solução alternativa ao recriar um índice grande é utilizar a extensão pg_repack. Para obter mais informações, consulte o tópico sobre como Reorganizar tabelas em bancos de dados PostgreSQL com bloqueios mínimos, na documentação de pg_repack. Para obter informações sobre como configurar a extensão em sua instância de banco de dados do RDS para PostgreSQL, consulte Reduzir o inchaço em tabelas e índices com a extensão pg_repack.

Aumentar maintenance_work_mem ao agrupar tabelas

O comando `CLUSTER` agrupa a tabela especificada por table_name com base em um índice existente especificado por index_name. O RDS para PostgreSQL recria fisicamente a tabela para corresponder à ordem de um determinado índice.

Quando o armazenamento magnético era predominante, o agrupamento era comum, pois a taxa de transferência de armazenamento era limitada. Agora que o armazenamento baseado em SSD é comum, o agrupamento tornou-se menos popular. No entanto, se você agrupar tabelas, ainda poderá aumentar a performance ligeiramente, dependendo do tamanho da tabela, do índice, da consulta e assim por diante.

Se você executar o comando `CLUSTER` e observar os eventos de espera `IO:BufFileWrite` e `IO:BufFileRead`, ajuste `maintenance_work_mem`. Aumente o tamanho da memória para uma quantidade relativamente grande. Um valor alto significa que o mecanismo pode utilizar mais memória para a operação de agrupamento.

Ajustar a memória para evitar IO:BufFileRead e IO:BufFileWrite

Em algumas situações, você precisa ajustar a memória. Seu objetivo é equilibrar a memória nas seguintes áreas de consumo usando os parâmetros apropriados, da forma a seguir.

- O valor `work_mem`
- A memória restante após descontar o valor `shared_buffers`
- As conexões máximas abertas e em uso, o que é limitado por `max_connections`

Para obter mais informações sobre ajuste da memória, consulte Resource Consumption (Consumo de recursos) na documentação do PostgreSQL.

Aumentar o tamanho da área de memória de trabalho

Em algumas situações, a única opção é aumentar a memória utilizada pela sessão. Se as consultas estiverem gravadas corretamente e utilizando as chaves corretas para junções, considere aumentar o valor de `work_mem`.

Para descobrir quantos arquivos temporários são gerados por uma consulta, defina `log_temp_files` como `0`. Se você aumentar o valor de `work_mem` para o valor máximo identificado nos logs, impedirá que a consulta gere arquivos temporários. No entanto, `work_mem` define o máximo por nó de plano para cada conexão ou operador paralelo. Se o banco de dados tiver 5.000 conexões e cada uma utilizar 256 MiB de memória, o mecanismo precisará de 1,2 TiB de RAM. Portanto, sua instância pode ficar sem memória.

Reservar memória suficiente para o grupo de buffer compartilhado

Seu banco de dados usa áreas de memória, como o grupo de buffer compartilhado, e não apenas a área de memória de trabalho. Considere os requisitos dessas áreas de memória adicionais antes de aumentar `work_mem`.

Por exemplo, suponha que sua classe de instância do RDS para PostgreSQL seja db.r5.2xlarge. Essa classe tem 64 GiB de memória. Por padrão, 25% da memória são reservados para o grupo de buffer compartilhado. Depois de subtrair a quantidade alocada à área de memória compartilhada, permanecem 16.384 MB. Não aloque a memória restante exclusivamente à área de memória de trabalho, pois o sistema operacional e o mecanismo também precisam de memória.

A memória que é possível alocar a `work_mem` depende da classe da instância. Se você utilizar uma classe de instância maior, mais memória estará disponível. No entanto, no exemplo anterior, não é possível utilizar mais de 16 GiB. Caso contrário, sua instância estará indisponível quando ficar sem memória. Para recuperar a instância e retirá-la do estado indisponível, os serviços de automação do RDS para PostgreSQL são reiniciados automaticamente.

Gerenciar o número de conexões

Imagine que a sua instância de banco de dados tenha 5.000 conexões simultâneas. Cada conexão usa pelo menos 4 MiB de `work_mem`. O alto consumo de memória das conexões provavelmente diminuirá a performance. Em resposta, existem as seguintes opções:

- Faça upgrade para uma classe de instância maior.
- Diminua o número de conexões de banco de dados simultâneas utilizando um proxy de conexão ou pooler.

Para proxies, considere o Amazon RDS Proxy, o pgBouncer ou um pooler de conexão baseado na sua aplicação. Essa solução alivia a carga da CPU. Ela também reduz o risco quando todas as conexões exigem a área de memória de trabalho. Quando há menos conexões de banco de dados, é possível aumentar o valor de `work_mem`. Dessa forma, você reduz a ocorrência dos eventos de espera `IO:BufFileRead` e `IO:BufFileWrite`. Além disso, as consultas que aguardam a área de memória de trabalho são aceleradas significativamente.

IO:DataFileRead

O evento `IO:DataFileRead` ocorre quando uma conexão aguarda em um processo de backend para ler uma página necessária do armazenamento porque essa página não está disponível na memória compartilhada.

Tópicos

- Versões compatíveis do mecanismo
- Contexto
- Possíveis causas do maior número de esperas
- Ações

Versões compatíveis do mecanismo

Essas informações de eventos de espera são compatíveis com todas as versões do RDS para PostgreSQL.

Contexto

Todas as consultas e operações de manipulação de dados (DML) acessam páginas no grupo de buffer. Instruções que podem induzir leituras incluem `SELECT`, `UPDATE` e `DELETE`. Por exemplo, um `UPDATE` pode ler páginas de tabelas ou índices. Se a página que está sendo solicitada ou atualizada não estiver no grupo de buffer compartilhado, essa leitura poderá gerar o evento `IO:DataFileRead`.

Como o grupo de buffer compartilhado é finito, ele pode ficar lotado. Nesse caso, solicitações de páginas que não estão na memória forçam o banco de dados a ler blocos do disco. Se o evento `IO:DataFileRead` ocorre com frequência, o grupo de buffer compartilhado pode ser pequeno demais para acomodar sua workload. Esse problema é grave para consultas `SELECT` que fazem a leitura de um grande número de linhas que não cabem no grupo de buffer. Para obter mais informações sobre o grupo de buffer, consulte Resource Consumption (Consumo de recursos) na documentação do PostgreSQL.

Possíveis causas do maior número de esperas

As causas comuns do evento `IO:DataFileRead` incluem:

Picos de conexão

Você pode encontrar várias conexões gerando o mesmo número de eventos de espera IO:DatafileRead. Nesse caso, pode ocorrer um pico (aumento súbito e grande) em eventos `IO:DataFileRead`.

Instruções SELECT e DML realizando varreduras sequenciais

Sua aplicação pode estar executando uma nova operação. Ou uma operação existente pode mudar por conta de um novo plano de execução. Nesses casos, procure tabelas (particularmente grandes) que tenham um valor de `seq_scan` maior. Encontre-as consultando `pg_stat_user_tables`. Para rastrear consultas que estão gerando mais operações de leitura, utilize a extensão `pg_stat_statements`.

CTAS e CREATE INDEX para conjuntos de dados grandes

Um CTAS é uma instrução `CREATE TABLE AS SELECT`. Se você executar um CTAS utilizando um conjunto de dados grande como fonte ou criar um índice em uma tabela grande, o evento `IO:DataFileRead` poderá ocorrer. Quando você cria um índice, talvez o banco de dados precise ler o objeto inteiro utilizando uma varredura sequencial. Um CTAS gera leituras de `IO:DataFile` quando as páginas não estão na memória.

Vários operadores de vacuum em execução ao mesmo tempo

Operadores de vacuum podem ser acionados manual ou automaticamente. Convém adotar uma estratégia de vacuum agressiva. No entanto, quando uma tabela possui muitas linhas atualizadas ou excluídas, as esperas de `IO:DataFileRead` aumentam. Depois que o espaço é recuperado, o tempo de vacuum gasto em `IO:DataFileRead` diminui.

Ingestão de grandes quantidades de dados

Quando a aplicação ingere grandes quantidades de dados, operações `ANALYZE` podem ocorrer com mais frequência. O processo `ANALYZE` pode ser acionado por um launcher de autovacuum ou chamado manualmente.

A operação `ANALYZE` lê um subconjunto da tabela. O número de páginas que devem ser varridas é calculado multiplicando-se 30 pelo valor de `default_statistics_target`. Para obter mais informações, consulte a Documentação do PostgreSQL. O parâmetro `default_statistics_target` aceita valores entre 1 e 10.000, em que o padrão é 100.

Inanição de recursos

Se a largura de banda ou a CPU da rede da instância forem consumidas, o evento `IO:DataFileRead` poderá ocorrer com mais frequência.

Ações

Recomenda-se ações distintas, dependendo dos motivos do evento de espera.

Tópicos

- Verificar filtros de predicados em busca de consultas que geram esperas
- Minimizar o efeito de operações de manutenção
- Responder a um alto número de conexões

Verificar filtros de predicados em busca de consultas que geram esperas

Suponha que você identifique consultas específicas que estão gerando eventos de espera `IO:DataFileRead`. É possível identificá-los utilizando as seguintes técnicas:

- Insights de Performance
- Visualizações de catálogo, como a fornecida pela extensão `pg_stat_statements`
- A visualização de catálogo `pg_stat_all_tables`, se ela mostrar periodicamente um número mais alto de leituras físicas
- A visualização `pg_statio_all_tables`, se ela mostrar que os contadores de `_read` estão aumentando

Convém determinar quais filtros são utilizados no predicado (cláusula `WHERE`) dessas consultas. Siga estas diretrizes:

- Execute o comando `EXPLAIN`. Na saída, identifique quais tipos de varreduras são utilizados. Uma varredura sequencial não indica necessariamente um problema. Consultas que utilizam varreduras sequenciais naturalmente produzem mais eventos `IO:DataFileRead` quando comparados a consultas que utilizam filtros.

  Descubra se a coluna listada na cláusula `WHERE` é indexada. Caso contrário, considere criar um índice para essa coluna. Essa abordagem evita varreduras sequenciais e reduz eventos `IO:DataFileRead`. Se uma consulta tiver filtros restritivos e ainda produzir varreduras sequenciais, avalie se os índices adequados estão sendo utilizados.
- Descubra se a consulta está acessando uma tabela muito grande. Em alguns casos, o particionamento de uma tabela pode melhorar a performance, permitindo que a consulta leia apenas as partições necessárias.

- Observe a cardinalidade (número total de linhas) das suas operações de junção. Observe o quão restritivos são os valores que você está transmitindo nos filtros para a sua cláusula `WHERE`. Se possível, ajuste sua consulta para reduzir o número de linhas que são transmitidas em cada etapa do plano.

Minimizar o efeito de operações de manutenção

Operações de manutenção, como `VACUUM` e `ANALYZE`, são importantes. Recomendamos que você não as desative ao encontrar eventos de espera `IO:DataFileRead` relacionados a essas operações de manutenção. As abordagens a seguir podem minimizar o efeito dessas operações:

- Execute operações de manutenção manualmente fora do horário de pico. Essa técnica impede que o banco de dados atinja o limite para operações automáticas.
- Considere particionar tabelas muito grandes. Essa técnica reduz a sobrecarga das operações de manutenção. O banco de dados somente acessa as partições que exigem manutenção.
- Ao ingerir grandes quantidades de dados, considere desativar o recurso de análise automática.

O recurso de autovacuum é acionado automaticamente para uma tabela quando a seguinte fórmula é verdadeira.

```
pg_stat_user_tables.n_dead_tup > (pg_class.reltuples x autovacuum_vacuum_scale_factor)
 + autovacuum_vacuum_threshold
```

A visualização `pg_stat_user_tables` e o catálogo `pg_class` têm várias linhas. Uma linha pode corresponder a uma linha na sua tabela. Essa fórmula pressupõe que os `reltuples` sejam para uma tabela específica. Os parâmetros `autovacuum_vacuum_scale_factor` (0,20 por padrão) e `autovacuum_vacuum_threshold` (50 tuplas por padrão) geralmente são definidos globalmente para toda a instância. Porém, é possível definir valores diferentes para uma tabela específica.

Tópicos

- Localizar tabelas que consomem espaço desnecessariamente
- Localizar índices que consomem espaço desnecessariamente
- Localizar tabelas elegíveis qualificadas para receber autovacuum

Localizar tabelas que consomem espaço desnecessariamente

Para encontrar tabelas que consomem espaço desnecessariamente, você pode usar funções da extensão `pgstattuple` do PostgreSQL. Essa extensão (módulo) está disponível por padrão em todas as instâncias de banco de dados do RDS para PostgreSQL e pode ser instanciada na instância com o comando a seguir.

```
CREATE EXTENSION pgstattuple;
```

Para obter mais informações sobre essa extensão, consulte pgstattuple na documentação do PostgreSQL.

É possível verificar se há sobrecarga na tabela e no índice em sua aplicação. Para obter mais informações, consulte Diagnosticar a sobrecarga na tabela e no índice.

Localizar índices que consomem espaço desnecessariamente

Para encontrar índices inchados e estimar a quantidade de espaço consumida desnecessariamente nas tabelas para as quais você tem privilégios de leitura, é possível executar a consulta a seguir.

```
-- WARNING: rows with is_na = 't' are known to have bad statistics ("name" type is not
 supported).
-- This query is compatible with PostgreSQL 8.2 and later.

SELECT current_database(), nspname AS schemaname, tblname, idxname,
 bs*(relpages)::bigint AS real_size,
  bs*(relpages-est_pages)::bigint AS extra_size,
  100 * (relpages-est_pages)::float / relpages AS extra_ratio,
  fillfactor, bs*(relpages-est_pages_ff) AS bloat_size,
  100 * (relpages-est_pages_ff)::float / relpages AS bloat_ratio,
  is_na
  -- , 100-(sub.pst).avg_leaf_density, est_pages, index_tuple_hdr_bm,
  -- maxalign, pagehdr, nulldatawidth, nulldatahdrwidth, sub.reltuples, sub.relpages
  -- (DEBUG INFO)
FROM (
  SELECT coalesce(1 +
       ceil(reltuples/floor((bs-pageopqdata-pagehdr)/(4+nulldatahdrwidth)::float)), 0
       -- ItemIdData size + computed avg size of a tuple (nulldatahdrwidth)
    ) AS est_pages,
    coalesce(1 +
       ceil(reltuples/floor((bs-pageopqdata-pagehdr)*fillfactor/
(100*(4+nulldatahdrwidth)::float))), 0
    ) AS est_pages_ff,
```

```
   bs, nspname, table_oid, tblname, idxname, relpages, fillfactor, is_na
   -- , stattuple.pgstatindex(quote_ident(nspname)||'.'||quote_ident(idxname)) AS
pst,
   -- index_tuple_hdr_bm, maxalign, pagehdr, nulldatawidth, nulldatahdrwidth,
reltuples
   -- (DEBUG INFO)
 FROM (
   SELECT maxalign, bs, nspname, tblname, idxname, reltuples, relpages, relam,
table_oid, fillfactor,
     ( index_tuple_hdr_bm +
         maxalign - CASE -- Add padding to the index tuple header to align on MAXALIGN
           WHEN index_tuple_hdr_bm%maxalign = 0 THEN maxalign
           ELSE index_tuple_hdr_bm%maxalign
         END
       + nulldatawidth + maxalign - CASE -- Add padding to the data to align on
MAXALIGN
           WHEN nulldatawidth = 0 THEN 0
           WHEN nulldatawidth::integer%maxalign = 0 THEN maxalign
           ELSE nulldatawidth::integer%maxalign
         END
     )::numeric AS nulldatahdrwidth, pagehdr, pageopqdata, is_na
     -- , index_tuple_hdr_bm, nulldatawidth -- (DEBUG INFO)
   FROM (
     SELECT
       i.nspname, i.tblname, i.idxname, i.reltuples, i.relpages, i.relam, a.attrelid
AS table_oid,
       current_setting('block_size')::numeric AS bs, fillfactor,
       CASE -- MAXALIGN: 4 on 32bits, 8 on 64bits (and mingw32 ?)
         WHEN version() ~ 'mingw32' OR version() ~ '64-bit|x86_64|ppc64|ia64|amd64'
THEN 8
         ELSE 4
       END AS maxalign,
       /* per page header, fixed size: 20 for 7.X, 24 for others */
       24 AS pagehdr,
       /* per page btree opaque data */
       16 AS pageopqdata,
       /* per tuple header: add IndexAttributeBitMapData if some cols are null-able */
       CASE WHEN max(coalesce(s.null_frac,0)) = 0
         THEN 2 -- IndexTupleData size
         ELSE 2 + (( 32 + 8 - 1 ) / 8)
         -- IndexTupleData size + IndexAttributeBitMapData size ( max num filed per
index + 8 - 1 /8)
       END AS index_tuple_hdr_bm,
```

```
       /* data len: we remove null values save space using it fractionnal part from
 stats */
       sum( (1-coalesce(s.null_frac, 0)) * coalesce(s.avg_width, 1024)) AS
 nulldatawidth,
       max( CASE WHEN a.atttypid = 'pg_catalog.name'::regtype THEN 1 ELSE 0 END ) > 0
 AS is_na
     FROM pg_attribute AS a
       JOIN (
         SELECT nspname, tbl.relname AS tblname, idx.relname AS idxname,
           idx.reltuples, idx.relpages, idx.relam,
           indrelid, indexrelid, indkey::smallint[] AS attnum,
           coalesce(substring(
             array_to_string(idx.reloptions, ' ')
              from 'fillfactor=([0-9]+)')::smallint, 90) AS fillfactor
         FROM pg_index
           JOIN pg_class idx ON idx.oid=pg_index.indexrelid
           JOIN pg_class tbl ON tbl.oid=pg_index.indrelid
           JOIN pg_namespace ON pg_namespace.oid = idx.relnamespace
         WHERE pg_index.indisvalid AND tbl.relkind = 'r' AND idx.relpages > 0
       ) AS i ON a.attrelid = i.indexrelid
       JOIN pg_stats AS s ON s.schemaname = i.nspname
         AND ((s.tablename = i.tblname AND s.attname =
 pg_catalog.pg_get_indexdef(a.attrelid, a.attnum, TRUE))
         -- stats from tbl
         OR  (s.tablename = i.idxname AND s.attname = a.attname))
         -- stats from functional cols
       JOIN pg_type AS t ON a.atttypid = t.oid
     WHERE a.attnum > 0
     GROUP BY 1, 2, 3, 4, 5, 6, 7, 8, 9
    ) AS s1
  ) AS s2
    JOIN pg_am am ON s2.relam = am.oid WHERE am.amname = 'btree'
) AS sub
-- WHERE NOT is_na
ORDER BY 2,3,4;
```

Localizar tabelas elegíveis qualificadas para receber autovacuum

Para localizar tabelas qualificadas para receber autovacuum, execute a seguinte consulta.

```
--This query shows tables that need vacuuming and are eligible candidates.
--The following query lists all tables that are due to be processed by autovacuum.
-- During normal operation, this query should return very little.
WITH  vbt AS (SELECT setting AS autovacuum_vacuum_threshold
```

```
               FROM pg_settings WHERE name = 'autovacuum_vacuum_threshold')
    , vsf AS (SELECT setting AS autovacuum_vacuum_scale_factor
               FROM pg_settings WHERE name = 'autovacuum_vacuum_scale_factor')
    , fma AS (SELECT setting AS autovacuum_freeze_max_age
               FROM pg_settings WHERE name = 'autovacuum_freeze_max_age')
    , sto AS (SELECT opt_oid, split_part(setting, '=', 1) as param,
                 split_part(setting, '=', 2) as value
               FROM (SELECT oid opt_oid, unnest(reloptions) setting FROM pg_class) opt)
SELECT
    '"'||ns.nspname||'"."'||c.relname||'"' as relation
    , pg_size_pretty(pg_table_size(c.oid)) as table_size
    , age(relfrozenxid) as xid_age
    , coalesce(cfma.value::float, autovacuum_freeze_max_age::float)
 autovacuum_freeze_max_age
    , (coalesce(cvbt.value::float, autovacuum_vacuum_threshold::float) +
         coalesce(cvsf.value::float,autovacuum_vacuum_scale_factor::float) *
 c.reltuples)
         as autovacuum_vacuum_tuples
    , n_dead_tup as dead_tuples
FROM pg_class c
JOIN pg_namespace ns ON ns.oid = c.relnamespace
JOIN pg_stat_all_tables stat ON stat.relid = c.oid
JOIN vbt on (1=1)
JOIN vsf ON (1=1)
JOIN fma on (1=1)
LEFT JOIN sto cvbt ON cvbt.param = 'autovacuum_vacuum_threshold' AND c.oid =
 cvbt.opt_oid
LEFT JOIN sto cvsf ON cvsf.param = 'autovacuum_vacuum_scale_factor' AND c.oid =
 cvsf.opt_oid
LEFT JOIN sto cfma ON cfma.param = 'autovacuum_freeze_max_age' AND c.oid = cfma.opt_oid
WHERE c.relkind = 'r'
AND nspname <> 'pg_catalog'
AND (
    age(relfrozenxid) >= coalesce(cfma.value::float, autovacuum_freeze_max_age::float)
    or
    coalesce(cvbt.value::float, autovacuum_vacuum_threshold::float) +
      coalesce(cvsf.value::float,autovacuum_vacuum_scale_factor::float) * c.reltuples
 <= n_dead_tup
    -- or 1 = 1
)
ORDER BY age(relfrozenxid) DESC;
```

Responder a um alto número de conexões

Ao monitorar o Amazon CloudWatch, você pode descobrir que a métrica `DatabaseConnections` atinge picos. Esse aumento indica um número maior de conexões com o seu banco de dados. Recomendamos a seguinte abordagem:

- Limite o número de conexões que a aplicação pode abrir com cada instância. Se a aplicação tiver um recurso de grupo de conexões incorporado, defina um número razoável de conexões. Baseie o número no que as vCPUs na instância podem paralelizar de maneira eficiente.

  Se a aplicação não utilizar um recurso de grupo de conexões, considere utilizar o Amazon RDS Proxy ou uma alternativa. Essa abordagem permite que a aplicação abra várias conexões com o balanceador de carga. O balanceador pode então abrir um número restrito de conexões com o banco de dados. À medida que menos conexões são executadas em paralelo, sua instância de banco de dados realiza menos alternâncias de contexto no kernel. As consultas devem progredir com mais rapidez, resultando em menos eventos de espera. Para ter mais informações, consulte Amazon RDS Proxy.
- Sempre que possível, aproveite réplicas de leitura para o RDS para PostgreSQL. Quando sua aplicação executar uma operação somente leitura, envie essas solicitações às réplicas de leitura. Essa técnica reduz a pressão de E/S no nó primário (gravador).
- Considere aumentar a escala vertical da sua instância de banco de dados. Uma classe de instância com maior capacidade fornece mais memória, o que dá ao RDS para PostgreSQL um grupo de buffer compartilhado maior para conter páginas. O tamanho maior também dá à instância de banco de dados mais vCPUs para lidar com conexões. Mais vCPUs são particularmente úteis quando as operações que estão gerando eventos de espera `IO:DataFileRead` são gravações.

IO:WALWrite

Tópicos

- Versões compatíveis do mecanismo
- Contexto
- Possíveis causas do maior número de esperas
- Ações

Versões compatíveis do mecanismo

Essas informações de eventos de espera são compatíveis com todas as versões do RDS para PostgreSQL 10 e posteriores.

Contexto

A atividade no banco de dados que está gerando dados de log de gravação antecipada preenche primeiro os buffers do WAL e depois grava no disco, de forma assíncrona. O evento de espera `IO:WALWrite` é gerado quando a sessão SQL aguarda a conclusão da gravação dos dados do WAL no disco para que ela possa liberar a chamada COMMIT da transação.

Possíveis causas do maior número de esperas

Se esse evento de espera ocorrer com frequência, você deve revisar sua workload, o tipo de atualização que ela executa e sua frequência. Especificamente, procure os tipos de atividade a seguir.

Atividade intensa de DML

A alteração de dados nas tabelas do banco de dados não acontece instantaneamente. Uma inserção em uma tabela pode precisar aguardar uma inserção ou uma atualização para a mesma tabela de outro cliente. As instruções da linguagem de manipulação de dados (DML) para alterar valores de dados (INSERT, UPDATE, DELETE, COMMIT, ROLLBACK TRANSACTION) podem ocasionar contenções que fazem com que o arquivo de registro de gravação antecipada aguarde a liberação dos buffers. Essa situação é capturada nas métricas a seguir do Insights de Performance do Amazon RDS, que indicam atividade intensa de DML.

- `tup_inserted`
- `tup_updated`
- `tup_deleted`
- `xact_rollback`
- `xact_commit`

Para ter mais informações sobre essas métricas, consulte Contadores do Performance Insights para o Amazon RDS para PostgreSQL.

Atividade frequente no ponto de verificação

Pontos de verificação frequentes contribuem para um maior número de arquivos WAL. No RDS para PostgreSQL, as gravações de página inteira estão sempre “ativadas”. As gravações de página inteira ajudam a proteger contra a perda de dados. No entanto, quando a verificação ocorre com muita frequência, o sistema pode sofrer problemas gerais de performance. Isso é especialmente verdadeiro em sistemas com intensa atividade de DML. Em alguns casos, você pode encontrar mensagens de erro em seu `postgresql.log` afirmando que “os pontos de verificação estão ocorrendo com muita frequência”.

Recomendamos que, ao ajustar os pontos de verificação, você equilibre cuidadosamente a performance com o tempo esperado de recuperação no caso de um desligamento anormal.

Ações

Recomendamos as ações a seguir para reduzir os números desse evento de espera.

Tópicos

- Reduza o número de confirmações
- Monitorar os pontos de verificação
- Aumentar a escala de E/S verticalmente
- Volume de log dedicado (DLV)

Reduza o número de confirmações

Para reduzir o número de confirmações, você pode combinar instruções em blocos de transações. Utilize o Insights de Performance do Amazon RDS para examinar o tipo de consulta que está sendo executada. Você também pode transferir grandes operações de manutenção para o horário de pico. Por exemplo, crie índices ou use operações `pg_repack` fora do horário de produção.

Monitorar os pontos de verificação

Há dois parâmetros que você pode monitorar para ver com que frequência sua instância de banco de dados do RDS para PostgreSQL está gravando no arquivo WAL para pontos de verificação.

- `log_checkpoints`: por padrão, esse parâmetro está ativado. Isso faz com que uma mensagem seja enviada ao log do PostgreSQL para cada ponto de verificação. Essas mensagens de log

incluem o número de buffers gravados, o tempo gasto para gravá-los e o número de arquivos WAL adicionados, removidos ou reciclados para determinado ponto de verificação.

Para obter mais informações sobre esse parâmetro, consulte Error Reporting and Logging (Relatórios de erros e registro em log) na documentação do PostgreSQL.

- `checkpoint_warning`: esse parâmetro define um valor limite (em segundos) para a frequência do ponto de verificação acima da qual um aviso é gerado. Por padrão, esse parâmetro não é definido no RDS para PostgreSQL. Você pode definir o valor desse parâmetro para receber um aviso quando as alterações do banco de dados em sua instância de banco de dados do RDS para PostgreSQL forem gravadas a uma taxa para a qual os arquivos WAL não estão dimensionados para serem manipulados. Por exemplo, digamos que você definiu esse parâmetro como 30. Se sua instância do RDS para PostgreSQL precisar gravar alterações com maior frequência do que a cada 30 segundos, o aviso de que “pontos de verificação estão ocorrendo com muita frequência” será enviado ao log do PostgreSQL. Isso pode indicar que seu valor `max_wal_size` deve ser aumentado.

  Para obter mais informações, consulte Write Ahead Log (Log de gravação antecipada) na documentação do PostgreSQL.

Aumentar a escala de E/S verticalmente

Esse tipo de evento de espera de entrada/saída (IO) pode ser corrigido escalando as operações de entrada e saída por segundo (IOPs) para fornecer uma E/S mais rápida. Escalar a E/S é preferível a escalar a CPU, porque escalar a CPU pode ocasionar ainda mais contenção de E/S, pois a CPU aumentada pode lidar com mais trabalho e, assim, piorar ainda mais o gargalo de E/S. Em geral, recomendamos que você considere ajustar sua workload antes de realizar operações de escalabilidade.

Volume de log dedicado (DLV)

Use um volume de log dedicado (DLV) para uma instância de banco de dados que usa o armazenamento de IOPS provisionadas (PIOPS) utilizando o console do Amazon RDS, AWS CLI a ou a API do Amazon RDS. Um DLV move os logs de transações do banco de dados do PostgreSQL para um volume de armazenamento separado do volume que contém as tabelas do banco de dados. Para ter mais informações, consulte Volume de log dedicado (DLV).

IPC: eventos de espera paralelos

Os seguintes `IPC:parallel wait events` indicam que uma sessão está aguardando a comunicação entre processos relacionada às operações de execução paralela de consultas.

- `IPC:BgWorkerStartup` - um processo está aguardando um processo de operador concluir sua sequência de inicialização. Isso acontece ao inicializar operadores para execução paralela de consultas.
- `IPC:BgWorkerShutdown` - um processo está aguardando um processo de operador concluir sua sequência de shutdown. Isso ocorre durante a fase de limpeza da execução paralela da consulta.
- `IPC:ExecuteGather` - um processo está aguardando para receber dados de processos de operador paralelos durante a execução da consulta. Isso ocorre quando o processo líder precisa coletar resultados de seus operadores.
- `IPC:ParallelFinish` - um processo está aguardando operadores paralelos terminarem sua execução e relatarem seus resultados finais. Isso acontece durante a fase de conclusão da execução paralela da consulta.

Tópicos

- Versões compatíveis do mecanismo
- Contexto
- Possíveis causas do maior número de esperas
- Ações

Versões compatíveis do mecanismo

Essas informações de eventos de espera têm suporte para todas as versões do Aurora PostgreSQL.

Contexto

A execução paralela de consultas no PostgreSQL envolve vários processos trabalhando juntos para processar uma única consulta. Quando uma consulta é determinada como adequada para paralelização, um processo líder se coordena com um ou mais processos de operador paralelos com base na configuração de parâmetros `max_parallel_workers_per_gather`. O processo líder divide o trabalho entre os operadores, cada um processa sua parte dos dados de forma independente e os resultados são reunidos de volta ao processo líder.

> **Note**
>
> Cada operador paralelo opera como um processo separado com requisitos de recursos semelhantes a uma sessão completa de usuário. Isso significa que uma consulta paralela com 4 operadores pode consumir até 5 vezes mais recursos (CPU, memória, largura de banda de E/S) em comparação com uma consulta não paralela, pois tanto o processo líder quanto cada processo de operador mantêm suas próprias alocações de recursos. Por exemplo, configurações como `work_mem` são aplicadas individualmente a cada operador, potencialmente multiplicando o uso total de memória em todos os processos.

A arquitetura de consulta paralela consiste em três componentes principais:

- Processo líder: o processo principal que inicia a operação paralela, divide a workload e coordena com os processos de operador.
- Processos de operador: processos em segundo plano que executam partes da consulta em paralelo.
- Fusão Gather/Gather: operações que combinam resultados de vários processos de operador de volta ao líder

Durante a execução paralela, os processos precisam se comunicar entre si por meio de mecanismos de comunicação entre processos (IPC). Esses eventos de espera do IPC ocorrem durante diferentes fases:

- Inicialização do operador: quando operadores paralelos estão sendo inicializados
- Troca de dados: quando os operadores estão processando dados e enviando resultados para o líder
- Encerramento do operador: quando a execução paralela é concluída e os operadores estão sendo encerrados
- Pontos de sincronização: quando os processos precisam se coordenar ou esperar que outros processos concluam suas tarefas

Compreender esses eventos de espera é crucial para diagnosticar problemas de desempenho relacionados à execução paralela de consultas, especialmente em ambientes de alta concorrência, nos quais várias consultas paralelas podem ser executadas simultaneamente.

Possíveis causas do maior número de esperas

Vários fatores podem contribuir para um aumento nos eventos de espera do IPC relacionados ao paralelismo:

Alta concorrência de consultas paralelas

Quando muitas consultas paralelas são executadas simultaneamente, isso pode levar à contenção de recursos e ao aumento do tempo de espera para operações de IPC. Isso é particularmente comum em sistemas com altos volumes de transações ou workloads analíticas.

Planos de consulta paralela subótimos

Se o planejador de consultas escolher planos paralelos ineficientes, isso poderá resultar em paralelização desnecessária ou má distribuição do trabalho entre os operadores. Isso pode levar ao aumento das esperas de IPC, especialmente para eventos `IPC:ExecuteGather` e `IPC:ParallelFinish`. Esses problemas de planejamento geralmente resultam de estatísticas desatualizadas e do inchaço da tabela/índice.

Inicialização e encerramento frequentes de operadores paralelos

Consultas de curta duração que frequentemente iniciam e encerram operadores paralelos podem causar um aumento nos eventos `IPC:BgWorkerStartup` e `IPC:BgWorkerShutdown`. Isso geralmente é visto em workloads OLTP com muitas consultas pequenas e paralelizáveis.

Restrições de recursos

A capacidade limitada de CPU, memória ou E/S pode causar gargalos na execução paralela, levando ao aumento do tempo de espera em todos os eventos de IPC. Por exemplo, se a CPU estiver saturada, os processos de operador podem levar mais tempo para iniciar ou processar sua parte do trabalho.

Estruturas complexas de consulta

Consultas com vários níveis de paralelismo (por exemplo, junções paralelas seguidas por agregações paralelas) podem levar a padrões de IPC mais complexos e a tempos de espera potencialmente maiores, especialmente para eventos `IPC:ExecuteGather`.

Conjuntos de resultados grandes

Consultas que produzem grandes conjuntos de resultados podem causar maiores tempos de espera `IPC:ExecuteGather`, pois o processo líder passa mais tempo coletando e processando resultados dos processos de operador.

Entender esses fatores pode ajudar a diagnosticar e resolver problemas de desempenho relacionados à execução paralela de consultas no Aurora PostgreSQL.

Ações

Quando você vê esperas relacionadas à consulta paralela, isso normalmente significa que um processo de backend está coordenando ou aguardando processos de operador paralelos. Essas esperas são comuns durante a execução de planos paralelos. Você pode investigar e mitigar o impacto dessas esperas monitorando o uso de operadores paralelos, revisando as configurações de parâmetros e ajustando a execução da consulta e a alocação de recursos.

Tópicos

- Analisar planos de consulta para paralelismo ineficiente
- Monitorar o uso de consultas paralelas
- Revisar e ajustar as configurações de consulta paralela
- Otimizar a alocação de recursos
- Investigar o gerenciamento de conexão
- Revisar e otimizar as operações de manutenção

Analisar planos de consulta para paralelismo ineficiente

A execução paralela de consultas pode frequentemente levar à instabilidade do sistema, picos de CPU e variações imprevisíveis no desempenho da consulta. É crucial analisar minuciosamente se o paralelismo realmente melhora sua workload específica. Use EXPLAIN ANALYZE para revisar os planos de execução de consultas paralelas.

Desative temporariamente o paralelismo no nível da sessão para comparar a eficiência do plano:

```
SET max_parallel_workers_per_gather = 0;
EXPLAIN ANALYZE <your_query>;
```

Reative o paralelismo e compare:

```
RESET max_parallel_workers_per_gather;
EXPLAIN ANALYZE <your_query>;
```

Se a desativação do paralelismo produzir resultados melhores ou mais consistentes, desative-o para consultas específicas no nível da sessão usando os comandos SET. Para um impacto mais amplo, você pode querer desabilitar o paralelismo no nível da instância ajustando os parâmetros relevantes em seu grupo de parâmetros do banco de dados. Para obter mais informações, consulte Modificar parâmetros em um grupo de parâmetros de banco de dados no Amazon RDS.

Monitorar o uso de consultas paralelas

Use as consultas a seguir para obter visibilidade da atividade e da capacidade de consultas paralelas:

Verifique os processos de operador paralelos ativos:

```
SELECT
    COUNT(*)
FROM
    pg_stat_activity
WHERE
    backend_type = 'parallel worker';
```

Esta consulta mostra o número de processos de operador paralelos ativos. Um valor alto pode indicar que seu `max_parallel_workers` está configurado com um valor alto e você pode querer reduzi-lo.

Verifique as consultas paralelas simultâneas:

```
SELECT
    COUNT(DISTINCT leader_pid)
FROM
    pg_stat_activity
WHERE
    leader_pid IS NOT NULL;
```

Essa consulta retorna o número de processos líderes distintos que iniciaram consultas paralelas. Um número alto aqui indica que várias sessões estão executando consultas paralelas simultaneamente, o que pode aumentar a demanda de CPU e memória.

Revisar e ajustar as configurações de consulta paralela

Analise os seguintes parâmetros para garantir que estejam alinhados à sua workload:

- `max_parallel_workers`: número total de workers paralelos em todas as sessões.
- `max_parallel_workers_per_gather`: máximo de workers por consulta.

Para workloads OLAP, aumentar esses valores pode melhorar o desempenho. Para workloads OLTP, valores mais baixos geralmente são preferidos.

```
SHOW max_parallel_workers;
SHOW max_parallel_workers_per_gather;
```

Otimizar a alocação de recursos

Monitore a utilização da CPU e ajuste o número de vCPUs se for consistentemente alto e se seu aplicativo se beneficiar de consultas paralelas. Verifique se há memória adequada disponível para operações paralelas.

- Use as métricas do Performance Insights para determinar se o sistema está limitado pela CPU.
- Cada operador paralelo usa o seu próprio `work_mem`. Verifique se o uso total da memória está dentro dos limites da instância.

As consultas paralelas podem consumir substancialmente mais recursos do que as consultas não paralelas, porque cada processo de operador é completamente separado que tem aproximadamente o mesmo impacto no sistema que uma sessão de usuário adicional. Isso deve ser levado em consideração ao escolher um valor para essa configuração, bem como ao definir outras configurações que controlam a utilização de recursos, como `work_mem`. Para obter mais informações, consulte a Documentação do PostgreSQL. Limites de recursos, como `work_mem`, são aplicados individualmente a cada operador, o que significa que a utilização total pode ser muito maior em todos os processos do que normalmente seria em qualquer processo único.

Aumente as vCPUs ou ajuste os parâmetros de memória se sua workload estiver altamente paralelizada.

Investigar o gerenciamento de conexão

Se estiver experimentando esgotamento de conexões, revise as estratégias de pool de conexões do aplicativo. Implemente o pool de conexões no nível do aplicativo, se ainda não estiver em uso.

Revisar e otimizar as operações de manutenção

Coordene a criação de índices e outras tarefas de manutenção para evitar a contenção de recursos. Agende essas operações fora do horário de pico. Evite agendar manutenções pesadas (por exemplo, construções paralelas de índices) durante períodos de alta carga de consultas do usuário. Essas operações podem consumir workers paralelos e impactar o desempenho de consultas regulares.

Lock:advisory

O evento `Lock:advisory` ocorre quando uma aplicação PostgreSQL utiliza um bloqueio para coordenar as atividades em várias sessões.

Tópicos

- Versões de mecanismos relevantes
- Contexto
- Causas
- Ações

Versões de mecanismos relevantes

As informações sobre eventos de espera são relevantes para o RDS para PostgreSQL versão 9.6 e versões superiores.

Contexto

Bloqueios consultivos do PostgreSQL são bloqueios cooperativos em nível de aplicação, feitos explicitamente e desfeitos pelo código da aplicação do usuário. Uma aplicação pode utilizar bloqueios consultivos do PostgreSQL para coordenar atividades em várias sessões. Ao contrário

de bloqueios regulares, ou em nível de objeto ou linha, a aplicação tem controle total ao longo da vida útil do bloqueio. Para obter mais informações, consulte o tópico sobre Bloqueios consultivos na documentação do PostgreSQL.

Bloqueios consultivos podem ser liberados antes que uma transação termine ou podem ser mantidos por uma sessão em todas as transações. Isso não é válido para bloqueios implícitos aplicados pelo sistema, como um bloqueio exclusivo de acesso em uma tabela adquirida por uma instrução `CREATE INDEX`.

Para obter uma descrição das funções utilizadas para adquirir (bloquear) e liberar (desbloquear) bloqueios consultivos, consulte o tópico sobre Funções de bloqueios consultivos, na documentação do PostgreSQL.

Bloqueios consultivos são implementados sobre o sistema de bloqueio regular do PostgreSQL e ficam visíveis na visualização do sistema `pg_locks`.

Causas

Esse tipo de bloqueio é controlado exclusivamente por uma aplicação que o utiliza explicitamente. Bloqueios consultivos que são adquiridos para cada linha como parte de uma consulta podem causar um aumento nos bloqueios ou um acúmulo a longo prazo.

Esses efeitos acontecem quando a consulta é executada de uma maneira que adquire bloqueios em mais linhas do que as retornadas pela consulta. A aplicação deve eventualmente liberar todos os bloqueios, mas, se eles forem adquiridos em linhas que não são retornadas, a aplicação não poderá localizar todos os bloqueios.

O exemplo a seguir foi extraído do tópico Bloqueios consultivos na documentação do PostgreSQL.

```
SELECT pg_advisory_lock(id) FROM foo WHERE id > 12345 LIMIT 100;
```

Nesse exemplo, a cláusula `LIMIT` apenas pode interromper a saída da consulta depois que as linhas já foram selecionadas internamente e seus valores de ID estão bloqueados. Isso pode acontecer repentinamente quando um volume de dados crescente faz com que o planejador escolha um plano de execução diferente que não foi testado durante o desenvolvimento. O acúmulo nesse caso acontece porque a aplicação chama explicitamente `pg_advisory_unlock` para cada valor de ID bloqueado. No entanto, nesse caso, não é possível encontrar o conjunto de bloqueios adquiridos em linhas que não foram retornadas. Como os bloqueios são adquiridos em nível de sessão, eles não são liberados automaticamente no final da transação.

Outra possível causa para picos em tentativas de bloqueio bloqueadas são conflitos não intencionais. Nesses conflitos, partes não relacionadas da aplicação compartilham o mesmo espaço de ID de bloqueio por engano.

Ações

Revise o uso da aplicação de bloqueios consultivos e detalhe onde e quando no fluxo de aplicação cada tipo de bloqueio consultivo é adquirido e liberado.

Determine se uma sessão está adquirindo muitos bloqueios ou se uma sessão de longa execução não está liberando bloqueios cedo o suficiente, resultando em um acúmulo lento de bloqueios. Você pode corrigir um acúmulo lento de bloqueios em nível de sessão encerrando a sessão com `pg_terminate_backend(pid)`.

Um cliente que aguarda um bloqueio de consultoria aparece em `pg_stat_activity` com `wait_event_type=Lock` e `wait_event=advisory`. É possível obter valores de bloqueio específicos consultando a visualização do sistema `pg_locks` em busca do mesmo `pid`, procurando `locktype=advisory` e `granted=f`.

Em seguida, identifique a sessão de bloqueio consultando `pg_locks` em busca do mesmo bloqueio consultivo que possui `granted=t`, conforme mostrado no exemplo a seguir.

```
SELECT blocked_locks.pid AS blocked_pid,
         blocking_locks.pid AS blocking_pid,
         blocked_activity.usename AS blocked_user,
         blocking_activity.usename AS blocking_user,
         now() - blocked_activity.xact_start AS blocked_transaction_duration,
         now() - blocking_activity.xact_start AS blocking_transaction_duration,
         concat(blocked_activity.wait_event_type,':',blocked_activity.wait_event) AS
 blocked_wait_event,
         concat(blocking_activity.wait_event_type,':',blocking_activity.wait_event) AS
 blocking_wait_event,
         blocked_activity.state AS blocked_state,
         blocking_activity.state AS blocking_state,
         blocked_locks.locktype AS blocked_locktype,
         blocking_locks.locktype AS blocking_locktype,
         blocked_activity.query AS blocked_statement,
         blocking_activity.query AS blocking_statement
    FROM pg_catalog.pg_locks blocked_locks
    JOIN pg_catalog.pg_stat_activity blocked_activity ON blocked_activity.pid =
 blocked_locks.pid
```

```
    JOIN pg_catalog.pg_locks blocking_locks
        ON blocking_locks.locktype = blocked_locks.locktype
        AND blocking_locks.DATABASE IS NOT DISTINCT FROM blocked_locks.DATABASE
        AND blocking_locks.relation IS NOT DISTINCT FROM blocked_locks.relation
        AND blocking_locks.page IS NOT DISTINCT FROM blocked_locks.page
        AND blocking_locks.tuple IS NOT DISTINCT FROM blocked_locks.tuple
        AND blocking_locks.virtualxid IS NOT DISTINCT FROM blocked_locks.virtualxid
        AND blocking_locks.transactionid IS NOT DISTINCT FROM
blocked_locks.transactionid
        AND blocking_locks.classid IS NOT DISTINCT FROM blocked_locks.classid
        AND blocking_locks.objid IS NOT DISTINCT FROM blocked_locks.objid
        AND blocking_locks.objsubid IS NOT DISTINCT FROM blocked_locks.objsubid
        AND blocking_locks.pid != blocked_locks.pid
    JOIN pg_catalog.pg_stat_activity blocking_activity ON blocking_activity.pid =
blocking_locks.pid
    WHERE NOT blocked_locks.GRANTED;
```

Todas as funções de API de bloqueio consultivo têm dois conjuntos de argumentos, um argumento `bigint` ou dois argumentos `integer`:

- Para as funções de API com um único argumento `bigint`, os 32 bits superiores estão em `pg_locks.classid` e os 32 bits inferiores estão em `pg_locks.objid`.
- Para as funções da API com dois argumentos `integer`, o primeiro argumento é `pg_locks.classid` e o segundo é `pg_locks.objid`.

O valor `pg_locks.objsubid` indica qual formato de API foi utilizado: 1 significa um argumento `bigint`; 2significa dois argumentos `integer`.

Lock:extend

O evento `Lock:extend` ocorre quando um processo de backend está aguardando para bloquear uma relação com o objetivo de a estender, enquanto outro processo tem um bloqueio nessa relação para o mesmo objetivo.

Tópicos

- Versões compatíveis do mecanismo
- Contexto
- Possíveis causas do maior número de esperas
- Ações

Versões compatíveis do mecanismo

Essas informações de eventos de espera são compatíveis com todas as versões do RDS para PostgreSQL.

Contexto

O evento `Lock:extend` indica que um processo de backend está aguardando para estender uma relação na qual outro processo de backend mantém um bloqueio enquanto estende essa relação. Como apenas um processo por vez pode estender uma relação, o sistema gera um evento de espera `Lock:extend`. Operações `INSERT`, `COPY` e `UPDATE` podem gerar esse evento.

Possíveis causas do maior número de esperas

Quando o evento `Lock:extend` aparece mais que o normal, possivelmente indicando um problema de performance, as causas típicas incluem:

Surto de inserções simultâneas ou atualizações na mesma tabela

Pode haver um aumento no número de sessões simultâneas com consultas que inserem ou atualizam a mesma tabela.

Largura de banda da rede insuficiente

É possível que a largura de banda da rede na instância de banco de dados para as necessidades de comunicação de armazenamento da workload atual. Isso pode contribuir para a latência de armazenamento que causa um aumento em eventos `Lock:extend`.

Ações

Recomenda-se ações distintas, dependendo dos motivos do evento de espera.

Tópicos

- Reduzir inserções e atualizações simultâneas para a mesma relação
- Aumentar a largura de banda da rede

Reduzir inserções e atualizações simultâneas para a mesma relação

Primeiro, determine se há um aumento nas métricas `tup_inserted` e `tup_updated` e um aumento acompanhante nesse evento de espera. Em caso afirmativo, verifique quais relações

estão em alta disputa por operações de inserção e atualização. Para determinar isso, consulte a visualização `pg_stat_all_tables` para os valores nos campos `n_tup_ins` e `n_tup_upd`. Para obter mais informações sobre a visualização `pg_stat_all_tables`, consulte pg_stat_all_tables na documentação do PostgreSQL.

Para obter mais informações sobre bloqueio e consultas bloqueadas, consulte `pg_stat_activity` como no exemplo a seguir:

```
SELECT
    blocked.pid,
    blocked.usename,
    blocked.query,
    blocking.pid AS blocking_id,
    blocking.query AS blocking_query,
    blocking.wait_event AS blocking_wait_event,
    blocking.wait_event_type AS blocking_wait_event_type
FROM pg_stat_activity AS blocked
JOIN pg_stat_activity AS blocking ON blocking.pid = ANY(pg_blocking_pids(blocked.pid))
where
blocked.wait_event = 'extend'
and blocked.wait_event_type = 'Lock';

   pid  | usename  |            query             | blocking_id |
            blocking_query                          | blocking_wait_event |
 blocking_wait_event_type
  ------+----------+------------------------------+-------------
+-------------------------------------------------------------------
+---------------------+--------------------------
   7143 |  myuser  | insert into tab1 values (1); |        4600 | INSERT INTO tab1 (a)
 SELECT s FROM generate_series(1,1000000) s; | DataFileExtend      | IO
```

Depois de identificar relações que contribuem para o aumento dos eventos `Lock:extend`, utilize as seguintes técnicas para reduzir a contenção:

- Descubra se é possível utilizar o particionamento para reduzir a contenção para a mesma tabela. Separar tuplas inseridas ou atualizadas em partições diferentes pode reduzir a contenção. Para obter informações sobre particionamento, consulte Gerenciar partições do PostgreSQL com a extensão pg_partman.
- Se o evento de espera for principalmente devido a atividades de atualização, reduza o valor do fator de preenchimento da relação. Isso pode reduzir as solicitações de novos bloqueios durante a atualização. O fator de preenchimento é um parâmetro de armazenamento de uma tabela que

determina o espaço máximo para empacotar uma página de tabela. Ele é expresso como uma porcentagem do espaço total de uma página. Para obter mais informações sobre o parâmetro de fator de preenchimento, consulte CREATE TABLE na documentação do PostgreSQL.

> **Important**
>
> É altamente recomendável testar seu sistema se você alterar o fator de preenchimento, pois isso pode afetar negativamente a performance dependendo da workload.

Aumentar a largura de banda da rede

Para ver se há um aumento na latência de gravação, verifique a métrica `WriteLatency` no CloudWatch. Se houver, use as métricas `WriteThroughput` e `ReadThroughput` do Amazon CloudWatch para monitorar o tráfego relacionado ao armazenamento na instância de banco de dados. Essas métricas podem ajudar a determinar se a largura de banda da rede é suficiente para a atividade de armazenamento da sua workload.

Se a largura de banda da rede não suficiente, aumente-a. Se a sua instância de banco de dados estiver atingindo os limites de largura de banda da rede, a única maneira de aumentar a largura de banda será ampliar o tamanho da instância de banco de dados.

Para obter mais informações sobre métricas do CloudWatch, consulte Métricas específicas da instância do Amazon CloudWatch para Amazon RDS. Para obter informações sobre a performance de rede de cada classe de instância de banco de dados, consulte Métricas específicas da instância do Amazon CloudWatch para Amazon RDS.

Lock:Relation

O evento `Lock:Relation` ocorre quando uma consulta está aguardando para adquirir um bloqueio em uma tabela ou visualização (relação) que está atualmente bloqueada por outra transação.

Tópicos

- Versões compatíveis do mecanismo
- Contexto
- Possíveis causas do maior número de esperas
- Ações

Versões compatíveis do mecanismo

Essas informações de eventos de espera são compatíveis com todas as versões do RDS para PostgreSQL.

Contexto

A maioria dos comandos PostgreSQL utiliza bloqueios implicitamente para controlar o acesso simultâneo aos dados em tabelas. Também é possível utilizar esses bloqueios explicitamente no código da aplicação com o comando `LOCK`. Vários modos de bloqueio não são compatíveis entre si e podem bloquear transações quando tentam acessar o mesmo objeto. Quando isso acontece, o RDS para PostgreSQL gera um evento `Lock:Relation`. Veja a seguir alguns exemplos comuns:

- Bloqueios exclusivos, como `ACCESS EXCLUSIVE`, podem bloquear todo o acesso simultâneo. Operações de linguagem de definição de dados (DDL), como `DROP TABLE`, `TRUNCATE`, `VACUUM FULL` e `CLUSTER`, adquirem bloqueios `ACCESS EXCLUSIVE` implicitamente. `ACCESS EXCLUSIVE` também é o modo de bloqueio padrão para instruções `LOCK TABLE` que não especificam um modo explicitamente.
- Usar `CREATE INDEX (without CONCURRENT)` em uma tabela gera conflito com instruções de linguagem de manipulação de dados (DML) `UPDATE`, `DELETE` e `INSERT`, que adquirem bloqueios `ROW EXCLUSIVE`.

Para obter mais informações sobre bloqueios em nível da tabela e modos de bloqueio conflitantes, consulte o tópico sobre Bloqueio explícito na documentação do PostgreSQL.

Normalmente, o bloqueio de consultas e transações é liberado de uma das seguintes maneiras:

- Consulta de bloqueio: a aplicação pode cancelar a consulta ou o usuário pode encerrar o processo. O mecanismo também pode forçar a finalização da consulta devido a um tempo limite de declaração de uma sessão ou um mecanismo de detecção de deadlock.
- Transação de bloqueio: uma transação para de bloquear quando executa uma instrução `ROLLBACK` ou `COMMIT`. Reversões também acontecem automaticamente quando as sessões são desconectadas por um cliente ou por problemas de rede, ou quando são encerradas. As sessões podem ser encerradas quando o mecanismo de banco de dados é desligado, quando o sistema está sem memória e assim por diante.

Possíveis causas do maior número de esperas

Quando o evento `Lock:Relation` ocorre com maior frequência do que o normal, pode indicar um problema de performance. As causas típicas incluem:

Maior número de sessões simultâneas com bloqueios de tabela conflitantes

Pode haver um aumento no número de sessões simultâneas com consultas que bloqueiam a mesma tabela com modos de bloqueio conflitantes.

Operações de manutenção

Operações de manutenção de integridade, como `VACUUM` e `ANALYZE`, podem aumentar significativamente o número de bloqueios conflitantes. `VACUUM FULL` adquire um bloqueio `ACCESS EXCLUSIVE` e `ANALYSE` adquire um bloqueio `SHARE UPDATE EXCLUSIVE`. Ambos os tipos de bloqueios podem causar um evento de espera `Lock:Relation`. Operações de manutenção de dados de aplicações, como atualizar uma visualização materializada, também podem aumentar as consultas e transações bloqueadas.

Bloqueios em instâncias de leitor

Pode haver um conflito entre os bloqueios de relação mantidos pelo gravador e os leitores. No momento, só bloqueios de relação do `ACCESS EXCLUSIVE` são replicados para instâncias do leitor. No entanto, o bloqueio de relação do `ACCESS EXCLUSIVE` entrará em conflito com qualquer bloqueio de relação do `ACCESS SHARE` mantido pelo leitor. Isso pode causar um aumento nos eventos de espera de relação de bloqueio no leitor.

Ações

Recomenda-se ações distintas, dependendo dos motivos do evento de espera.

Tópicos

- Reduzir o impacto do bloqueio de instruções SQL
- Minimizar o efeito de operações de manutenção

Reduzir o impacto do bloqueio de instruções SQL

Para reduzir o impacto do bloqueio de instruções SQL, modifique o código da aplicação sempre que possível. Veja a seguir duas técnicas comuns para reduzir bloqueios:

- Usar a opção `NOWAIT`: alguns comandos SQL, como instruções `SELECT` e `LOCK`, oferecem suporte a essa opção. A diretiva `NOWAIT` cancela a consulta de solicitação de bloqueio quando o bloqueio não pode ser adquirido imediatamente. Essa técnica pode evitar que uma sessão de bloqueio cause um empilhamento de sessões bloqueadas por detrás dela.

  Por exemplo: suponha que a transação A esteja aguardando um bloqueio mantido pela transação B. Se B solicitar um bloqueio em uma tabela bloqueada pela transação C, a transação A poderá ser bloqueada até a conclusão da transação C. Porém, se a transação B utilizar um `NOWAIT` quando solicitar o bloqueio em C, ela pode falhar rapidamente e garantir que a transação A não precise aguardar indefinidamente.
- Usar `SET lock_timeout`: defina um valor de `lock_timeout` para limitar o tempo de espera de uma instrução SQL para adquirir um bloqueio em uma relação. Se o bloqueio não for adquirido dentro do tempo limite definido, a transação que o está solicitando será cancelada. Defina o valor no nível da sessão.

Minimizar o efeito de operações de manutenção

Operações de manutenção, como `VACUUM` e `ANALYZE`, são importantes. Recomendamos que você não as desative ao encontrar eventos de espera `Lock:Relation` relacionados a essas operações de manutenção. As abordagens a seguir podem minimizar o efeito dessas operações:

- Execute operações de manutenção manualmente fora do horário de pico.
- Para reduzir esperas `Lock:Relation` causadas por tarefas de autovacuum, realize qualquer ajuste de autovacuum necessário. Para obter mais informações sobre ajuste de autovacuum, consulte Trabalhar com o autovacuum do PostgreSQL no Amazon RDS, no Guia do usuário do Amazon RDS.

Lock:transactionid

O evento `Lock:transactionid` ocorre quando uma transação está aguardando um bloqueio em nível de linha.

Tópicos

- Versões compatíveis do mecanismo
- Contexto
- Possíveis causas do maior número de esperas

- Ações

Versões compatíveis do mecanismo

Essas informações de eventos de espera são compatíveis com todas as versões do RDS para PostgreSQL.

Contexto

O evento `Lock:transactionid` ocorre quando uma transação está tentando adquirir um bloqueio em nível de linha já concedido a uma transação que está sendo executada simultaneamente. A sessão que mostra o evento de espera `Lock:transactionid` está bloqueada devido a esse bloqueio. Depois que a transação de bloqueio terminar em uma instrução `COMMIT` ou `ROLLBACK`, a transação bloqueada pode continuar.

A semântica de controle de simultaneidade de várias versões do RDS para PostgreSQL garante que os leitores não bloqueiem gravadores e que os gravadores não bloqueiem os leitores. Para que ocorram conflitos em nível de linha, transações de bloqueio e bloqueadas devem emitir instruções conflitantes dos seguintes tipos:

- `UPDATE`
- `SELECT … FOR UPDATE`
- `SELECT … FOR KEY SHARE`

A instrução `SELECT … FOR KEY SHARE` é um caso especial. O banco de dados usa a cláusula `FOR KEY SHARE` para otimizar a performance da integridade referencial. Um bloqueio em nível de linha em uma linha pode bloquear comandos `INSERT`, `UPDATE` e `DELETE` em outras tabelas que fazem referência à linha.

Possíveis causas do maior número de esperas

Quando esse evento aparece mais do que o normal, a causa geralmente envolve instruções `UPDATE`, `SELECT … FOR UPDATE` ou `SELECT … FOR KEY SHARE` combinadas com as seguintes condições.

Tópicos

- Alta simultaneidade
- Ocioso na transação

- Transações de longa execução

Alta simultaneidade

O RDS para PostgreSQL pode utilizar semântica de bloqueio granular em nível de linha. A probabilidade de conflitos em nível de linha aumenta quando as condições a seguir são atendidas:

- Uma workload altamente simultânea controla as mesmas linhas.
- A simultaneidade aumenta.

Ocioso na transação

Às vezes, a coluna `pg_stat_activity.state` mostra o valor `idle in transaction`. Esse valor aparece para sessões que iniciaram uma transação, mas ainda não emitiram um `COMMIT` ou `ROLLBACK`. Se o valor de `pg_stat_activity.state` não for `active`, a consulta mostrada em `pg_stat_activity` será a mais recente a terminar a execução. A sessão de bloqueio não está processando uma consulta ativamente porque uma transação aberta está mantendo um bloqueio.

Se uma transação ociosa adquiriu um bloqueio em nível de linha, talvez ela esteja impedindo que outras sessões o adquiram. Essa condição leva à ocorrência frequente do evento de espera `Lock:transactionid`. Para diagnosticar o problema, examine a saída proveniente de `pg_stat_activity` e `pg_locks`.

Transações de longa execução

Transações executadas por um longo tempo recebem bloqueios por um longo tempo. Esses bloqueios de longa duração podem impedir que outras transações sejam executadas.

Ações

O bloqueio de linhas é um conflito entre instruções `UPDATE`, `SELECT … FOR UPDATE` ou `SELECT … FOR KEY SHARE`. Antes de tentar uma solução, descubra quando essas instruções estão sendo executadas na mesma linha. Use essas informações para escolher uma estratégia descrita nas seguintes seções.

Tópicos

- Responder a alta simultaneidade
- Responder a transações ociosas
- Responder a transações de longa duração

Responder a alta simultaneidade

Se a simultaneidade for o problema, tente uma das seguintes técnicas:

- Reduza a simultaneidade na aplicação. Por exemplo, reduza o número de sessões ativas.
- Implemente um pool de conexões. Para saber como agrupar conexões com o RDS Proxy, consulte Amazon RDS Proxy.
- Projete a aplicação ou o modelo de dados para evitar a contenção de instruções `UPDATE` e `SELECT … FOR UPDATE`. Também é possível diminuir o número de chaves estrangeiras acessadas por instruções `SELECT … FOR KEY SHARE`.

Responder a transações ociosas

Se `pg_stat_activity.state` mostrar `idle in transaction`, utilize as seguintes estratégias:

- Ative a confirmação automática sempre que possível. Essa abordagem impede que transações bloqueiem outras transações enquanto aguardam `COMMIT` ou `ROLLBACK`.
- Procure caminhos de código que não contenham `COMMIT`, `ROLLBACK` ou `END`.
- Assegure-se de que a lógica de tratamento de exceções na seu aplicação sempre tenha um caminho para um `end of transaction` válido.
- Assegure-se de que a sua aplicação processe os resultados da consulta depois de encerrar a transação com `COMMIT` ou `ROLLBACK`.

Responder a transações de longa duração

Se transações de longa duração estiverem causando a ocorrência frequente de `Lock:transactionid`, tente as estratégias a seguir:

- Mantenha os bloqueios de linha fora de transações de longa execução.
- Limite o comprimento das consultas implementando a confirmação automática sempre que possível.

Lock:tuple

O evento `Lock:tuple` ocorre quando um processo de backend está aguardando para adquirir um bloqueio em uma tupla.

Tópicos

- Versões compatíveis do mecanismo
- Contexto
- Possíveis causas do maior número de esperas
- Ações

Versões compatíveis do mecanismo

Essas informações de eventos de espera são compatíveis com todas as versões do RDS para PostgreSQL.

Contexto

O evento `Lock:tuple` indica que um backend está aguardando para adquirir um bloqueio em uma tupla enquanto outro backend mantém um bloqueio conflitante na mesma tupla. A tabela a seguir ilustra um cenário em que as sessões geram o evento `Lock:tuple`.

Temp	Sessão 1	Sessão 2	Sessão 3
t1	Inicia uma transação.		
t2	Atualiza a linha 1.		
t3		Atualiza a linha 1. A sessão adquire um bloqueio exclusivo na tupla e aguarda a sessão 1 liberar esse bloqueio via confirmação ou reversão.	
t4			Atualiza a linha 1. A sessão aguarda a sessão 2 liberar o bloqueio exclusivo na tupla.

Outra alternativa é simular esse evento de espera utilizando a ferramenta de benchmarking `pgbench`. Configure um número elevado de sessões simultâneas para atualizar a mesma linha em uma tabela com um arquivo SQL personalizado.

Para saber mais sobre modos de bloqueio conflitantes, consulte o tópico sobre Bloqueio explícito, na documentação do PostgreSQL. Para saber mais sobre `pgbench`, consulte pgbench na documentação do PostgreSQL.

Possíveis causas do maior número de esperas

Quando esse evento aparece mais que o normal, possivelmente indicando um problema de performance, as causas típicas incluem:

- Um número elevado de sessões simultâneas está tentando adquirir um bloqueio conflitante para a mesma tupla executando instruções `UPDATE` ou `DELETE`.
- Sessões altamente simultâneas estão executando uma instrução `SELECT` utilizando os modos de bloqueio `FOR UPDATE` ou `FOR NO KEY UPDATE`.
- Diversos fatores fazem com que grupos de aplicações ou conexões abram mais sessões para executar as mesmas operações. À medida que novas sessões estão tentando modificar as mesmas linhas, a carga de banco de dados pode atingir picos e `Lock:tuple` pode surgir.

Para obter mais informações, consulte o tópico sobre Bloqueios em nível de linha, na documentação do PostgreSQL.

Ações

Recomenda-se ações distintas, dependendo dos motivos do evento de espera.

Tópicos

- Investigar a lógica da sua aplicação
- Localizar a sessão bloqueadora
- Reduzir a simultaneidade quando ela estiver elevada
- Solucionar problemas de gargalos

Investigar a lógica da sua aplicação

Descubra se uma sessão bloqueadora está no estado `idle in transaction` por muito tempo. Em caso positivo, considere encerrar a sessão bloqueadora como uma solução de curto prazo. Também é possível utilizar a função `pg_terminate_backend`. Para obter mais informações sobre essa função, consulte Funções de sinalização de servidor, na documentação do PostgreSQL.

Para uma solução de longo prazo, faça o seguinte:

- Ajuste a lógica da aplicação.
- Use o parâmetro `idle_in_transaction_session_timeout`. Esse parâmetro encerra qualquer sessão com uma transação aberta que tenha ficado ociosa por um tempo maior que o especificado. Para obter mais informações, consulte o tópico sobre Padrões de conexão de clientes, na documentação do PostgreSQL.
- Use a confirmação automática o máximo possível. Para obter mais informações, consulte SET AUTOCOMMIT, na documentação do PostgreSQL.

Localizar a sessão bloqueadora

Enquanto o evento de espera `Lock:tuple` está ocorrendo, identifique a sessão bloqueadora e a sessão bloqueada descobrindo quais bloqueios dependem um do outro. Para obter mais informações, consulte o tópico sobre Informações de dependências de bloqueios, na wiki do PostgreSQL.

O exemplo a seguir consulta todas as sessões, filtrando em `tuple` e ordenando por `wait_time`.

```
SELECT blocked_locks.pid AS blocked_pid,
         blocking_locks.pid AS blocking_pid,
         blocked_activity.usename AS blocked_user,
         blocking_activity.usename AS blocking_user,
         now() - blocked_activity.xact_start AS blocked_transaction_duration,
         now() - blocking_activity.xact_start AS blocking_transaction_duration,
         concat(blocked_activity.wait_event_type,':',blocked_activity.wait_event) AS
 blocked_wait_event,
         concat(blocking_activity.wait_event_type,':',blocking_activity.wait_event) AS
 blocking_wait_event,
         blocked_activity.state AS blocked_state,
         blocking_activity.state AS blocking_state,
         blocked_locks.locktype AS blocked_locktype,
         blocking_locks.locktype AS blocking_locktype,
         blocked_activity.query AS blocked_statement,
         blocking_activity.query AS blocking_statement
    FROM pg_catalog.pg_locks blocked_locks
    JOIN pg_catalog.pg_stat_activity blocked_activity ON blocked_activity.pid =
 blocked_locks.pid
    JOIN pg_catalog.pg_locks blocking_locks
        ON blocking_locks.locktype = blocked_locks.locktype
        AND blocking_locks.DATABASE IS NOT DISTINCT FROM blocked_locks.DATABASE
```

```
        AND blocking_locks.relation IS NOT DISTINCT FROM blocked_locks.relation
        AND blocking_locks.page IS NOT DISTINCT FROM blocked_locks.page
        AND blocking_locks.tuple IS NOT DISTINCT FROM blocked_locks.tuple
        AND blocking_locks.virtualxid IS NOT DISTINCT FROM blocked_locks.virtualxid
        AND blocking_locks.transactionid IS NOT DISTINCT FROM
blocked_locks.transactionid
        AND blocking_locks.classid IS NOT DISTINCT FROM blocked_locks.classid
        AND blocking_locks.objid IS NOT DISTINCT FROM blocked_locks.objid
        AND blocking_locks.objsubid IS NOT DISTINCT FROM blocked_locks.objsubid
        AND blocking_locks.pid != blocked_locks.pid
    JOIN pg_catalog.pg_stat_activity blocking_activity ON blocking_activity.pid =
blocking_locks.pid
    WHERE NOT blocked_locks.GRANTED;
```

Reduzir a simultaneidade quando ela estiver elevada

O evento `Lock:tuple` pode ocorrer constantemente, especialmente em um tempo de workload ocupado. Nessa situação, considere reduzir a simultaneidade elevada para linhas muito ocupadas. Várias vezes, apenas algumas linhas controlam uma fila ou a lógica booleana, o que torna essas linhas muito ocupadas.

Você pode reduzir a simultaneidade utilizando diferentes abordagens com base no seu requisito de negócios, na lógica da aplicação e no tipo de workload. Por exemplo, você pode fazer o seguinte:

- Reformule sua tabela e a lógica de dados para reduzir a simultaneidade elevada.
- Altere a lógica da aplicação para reduzir a simultaneidade elevada no nível da linha.
- Aproveite e reformule consultas com bloqueios em nível de linha.
- Use a cláusula `NOWAIT` com operações de novas tentativas.
- Considere utilizar o controle de simultaneidade lógico otimista e bloqueio híbrido.
- Considere modificar o nível de isolamento do banco de dados.

Solucionar problemas de gargalos

O `Lock:tuple` pode ocorrer com gargalos, como falta de CPU ou uso máximo da largura de banda do Amazon EBS. Para diminuir gargalos, considere as seguintes abordagens:

- Aumente a escala do seu tipo de classe de instância na vertical.
- Otimize consultas que consomem muitos recursos.
- Altere a lógica da aplicação.

- Arquive dados acessados raramente.

LWLock:BufferMapping (LWLock:buffer_mapping)

Esse evento ocorre quando uma sessão está aguardando para associar um bloco de dados a um buffer no grupo de buffer compartilhado.

> **Note**
>
> Esse evento é denominado `LWLock:BufferMapping` para o RDS para PostgreSQL versão 13 e posteriores. No caso do RDS para PostgreSQL versão 12 e posteriores, esse evento é denominado `LWLock:buffer_mapping`.

Tópicos

- Versões compatíveis do mecanismo
- Contexto
- Causas
- Ações

Versões compatíveis do mecanismo

As informações de eventos de espera são relevantes para o RDS para PostgreSQL versão 9.6 e posteriores.

Contexto

O grupo de buffer compartilhado é uma área de memória do PostgreSQL que contém todas as páginas que estão ou estavam sendo utilizadas por processos. Quando um processo precisa de uma página, ele lê a página no grupo de buffer compartilhado. O parâmetro `shared_buffers` define o tamanho do buffer compartilhado e reserva uma área de memória para armazenar a tabela e as páginas de índice. Se você alterar esse parâmetro, reinicie o banco de dados.

O evento de espera `LWLock:buffer_mapping` ocorre nos seguintes cenários:

- Um processo pesquisa a tabela de buffer em busca de uma página e adquire um bloqueio de mapeamento de buffer compartilhado.

- Um processo carrega uma página no grupo de buffer e adquire um bloqueio exclusivo de mapeamento de buffer.
- Um processo remove uma página do grupo e adquire um bloqueio exclusivo de mapeamento de buffer.

Causas

Quando esse evento aparece mais do que o normal, possivelmente indicando um problema de performance, o banco de dados está paginando dentro e fora do grupo de buffer compartilhado. As causas típicas incluem:

- Consultas grandes
- Índices e tabelas inchados
- Varreduras completas de tabelas
- Um tamanho de grupo compartilhado menor que o conjunto de trabalho

Ações

Recomenda-se ações distintas, dependendo dos motivos do evento de espera.

Tópicos

- Monitorar métricas relacionadas ao buffer
- Avaliar sua estratégia de indexação
- Diminuir o número de buffers que devem ser alocados rapidamente

Monitorar métricas relacionadas ao buffer

Quando as esperas `LWLock:buffer_mapping` atingirem picos, investigue a taxa de acertos de buffer. É possível utilizar essas métricas para entender melhor o que está acontecendo no cache de buffer. Examine as métricas a seguir:

`blks_hit`

Essa métrica de contador do Performance Insights indica o número de bloqueios que foram recuperados do grupo de buffer compartilhado. Após o surgimento do evento de espera `LWLock:buffer_mapping`, é possível observar um pico em `blks_hit`.

`blks_read`

Essa métrica de contador do Performance Insights indica o número de bloqueios que exigiram a leitura da E/S no grupo de buffer compartilhado. Você pode observar um pico em `blks_read` em preparação para o evento de espera `LWLock:buffer_mapping`.

Avaliar sua estratégia de indexação

Para confirmar que sua estratégia de indexação não está diminuindo a performance, verifique o seguinte:

Inchaço de índice

Assegure-se de que o índice e o inchaço da tabela não estejam fazendo com que páginas desnecessárias sejam lidas no buffer compartilhado. Se as suas tabelas contiverem linhas não utilizadas, considere arquivar os dados e remover as linhas das tabelas. Em seguida, você poderá reconstruir os índices para as tabelas redimensionadas.

Índices para consultas utilizadas com frequência

Para determinar se você tem os índices ideais, monitore as métricas do mecanismo de banco de dados no Performance Insights. A métrica `tup_returned` mostra o número de linhas lidas. A métrica `tup_fetched` mostra o número de linhas retornadas ao cliente. Se `tup_returned` for significativamente maior que `tup_fetched`, talvez os dados não estejam devidamente indexados. Além disso, as estatísticas da tabela podem não estar atualizadas.

Diminuir o número de buffers que devem ser alocados rapidamente

Para diminuir os eventos de espera `LWLock:buffer_mapping`, tente reduzir o número de buffers que devem ser alocados rapidamente. Uma estratégia é realizar operações em lote menores. Talvez seja possível obter lotes menores particionando tabelas.

LWLock:BufferIO (IPC:BufferIO)

O evento `LWLock:BufferIO` ocorre quando o RDS para PostgreSQL aguarda outros processos terminarem suas operações de entrada/saída (E/S) ao tentarem acessar simultaneamente uma página. Sua finalidade é fazer com que a mesma página seja lida no buffer compartilhado.

Tópicos

- Versões de mecanismos relevantes
- Contexto
- Causas
- Ações

Versões de mecanismos relevantes

Essas informações de evento de espera são relevantes para todas as versões do RDS para PostgreSQL. Para o RDS para PostgreSQL 12 e versões anteriores, esse evento de espera é denominado lwlock:buffer_io, ao passo que, no RDS para PostgreSQL 13, ele é denominado lwlock:bufferio. A partir do RDS para PostgreSQL 14, o evento de espera BufferIO mudou do tipo de evento de espera `LWLock` para `IPC` (IPC:BufferIO).

Contexto

Cada buffer compartilhado tem um bloqueio de E/S associado ao evento de espera `LWLock:BufferIO`, todas as vezes que um bloqueio (ou uma página) precisa ser recuperado fora do grupo de buffer compartilhado.

Esse bloqueio é utilizado para lidar com várias sessões que requerem acesso ao mesmo bloco. Esse bloco precisa ser lido de fora do grupo de buffer compartilhado, definido pelo parâmetro `shared_buffers`.

Assim que a página for lida dentro do grupo de buffer compartilhado, o bloqueio de `LWLock:BufferIO` será liberado.

> **Note**
>
> O evento de espera `LWLock:BufferIO` precede o evento de espera IO:DataFileRead. O evento de espera `IO:DataFileRead` ocorre enquanto os dados estão sendo lidos do armazenamento.

Para obter mais informações sobre bloqueios leves, consulte Visão geral de bloqueios.

Causas

Causas comuns do surgimento do evento `LWLock:BufferIO` nas principais esperas incluem:

- Vários backends ou conexões tentando acessar a mesma página para a qual também há uma operação de E/S pendente
- A proporção entre o tamanho do grupo de buffer compartilhado (definido pelo parâmetro `shared_buffers`) e o número de buffers necessários para a workload atual
- O tamanho do grupo de buffer compartilhado não está bem equilibrado com o número de páginas que estão sendo despejadas por outras operações
- Índices grandes ou inchados que exigem que o mecanismo leia mais páginas que o necessário no grupo de buffer compartilhado
- Ausência índices, o que força o mecanismo de banco de dados a ler mais páginas das tabelas que o necessário
- Pontos de verificação ocorrendo com muita frequência ou que precisam liberar muitas páginas modificadas
- Picos repentinos de conexões de banco de dados tentando realizar operações na mesma página

Ações

Recomenda-se ações distintas, dependendo dos motivos do evento de espera:

- Observe as métricas do Amazon CloudWatch para encontrar uma correlação entre reduções acentuadas nos eventos de espera `BufferCacheHitRatio` e `LWLock:BufferIO`. Esse efeito pode significar que existe uma pequena configuração de buffers compartilhados. Talvez seja necessário aumentá-lo ou aumentar a escala da classe de instância de banco de dados na vertical. Você pode dividir sua workload em mais nós de leitura.
- Ajuste `max_wal_size` e `checkpoint_timeout` com base no horário de pico da workload, se você vir `LWLock:BufferIO` correspondendo a quedas da métrica `BufferCacheHitRatio`. Em seguida, identifique qual consulta pode estar causando isso.
- Verifique se existem índices não utilizados e remova-os.
- Use tabelas particionadas (que também tenham índices particionados). Fazer isso ajuda a manter a reordenação do índice baixo e reduz seu impacto.
- Evite indexar colunas desnecessariamente.
- Evite picos repentinos de conexão de banco de dados utilizando um grupo de conexões.
- Restrinja o número máximo de conexões com o banco de dados como prática recomendada.

LWLock:buffer_content (BufferContent)

O evento `LWLock:buffer_content` ocorre quando uma sessão aguarda para ler ou gravar uma página de dados na memória enquanto outra sessão fica com a página bloqueada para gravação. No RDS para PostgreSQL 13 e versões superiores, esse evento de espera é chamado de `BufferContent`.

Tópicos

- Versões compatíveis do mecanismo
- Contexto
- Possíveis causas do maior número de esperas
- Ações

Versões compatíveis do mecanismo

Essas informações de eventos de espera são compatíveis com todas as versões do RDS para PostgreSQL.

Contexto

Para ler ou manipular dados, o PostgreSQL os acessa por meio de buffers de memória compartilhada. Para ler a partir do buffer, um processo obtém um bloqueio leve (LWLock) no conteúdo do buffer no modo compartilhado. Para gravar no buffer, ele adquire esse bloqueio no modo exclusivo. Bloqueios compartilhados permitem que outros processos adquiram bloqueios compartilhados simultaneamente nesse conteúdo. Bloqueios exclusivos impedem que outros processos obtenham qualquer tipo de bloqueio nele.

O evento `LWLock:buffer_content` (`BufferContent`) indica que vários processos estão tentando obter um bloqueio no conteúdo de um buffer específico.

Possíveis causas do maior número de esperas

Quando o evento `LWLock:buffer_content` (`BufferContent`) aparece mais que o normal, possivelmente indicando um problema de performance, as causas típicas incluem:

Maior número de atualizações simultâneas para os mesmos dados

Pode haver um aumento no número de sessões simultâneas com consultas que atualizam o mesmo conteúdo do buffer. Essa contenção pode ser mais evidente em tabelas com vários índices.

Os dados da workload não estão na memória

Quando os dados que a workload ativa está processando não estão na memória, esses eventos de espera podem aumentar. Esse efeito ocorre porque os processos que mantêm bloqueios podem fazer isso por mais tempo enquanto executam operações de E/S de disco.

Uso excessivo de restrições de chaves externas

Restrições de chave externas podem aumentar o tempo durante o qual um processo mantém um bloqueio de conteúdo de buffer. Esse efeito ocorre porque operações de leitura exigem um bloqueio de conteúdo de buffer compartilhado na chave referenciada enquanto esta está sendo atualizada.

Ações

Recomenda-se ações distintas, dependendo dos motivos do evento de espera. Você pode identificar eventos `LWLock:buffer_content` (`BufferContent`) utilizando o Amazon RDS Performance Insights ou consultando a visualização `pg_stat_activity`.

Tópicos

- Melhorar a eficiência na memória
- Reduzir o uso de restrições de chaves externas
- Remover índices não utilizados
- Aumentar o tamanho do cache ao usar sequências

Melhorar a eficiência na memória

Para aumentar as chances de que os dados da workload ativa estejam na memória, particione tabelas ou aumente a escala da sua classe de instância na vertical. Para obter informações sobre classes de instância de banco de dados, consulte Classes de instâncias de banco de dados .

Reduzir o uso de restrições de chaves externas

Investigue workloads com um número elevado de eventos de espera `LWLock:buffer_content` (`BufferContent`) quanto ao uso de restrições de chaves externas. Remova restrições desnecessárias de chaves externas.

Remover índices não utilizados

Para workloads com um número elevado de eventos de espera `LWLock:buffer_content` (`BufferContent`), identifique índices não utilizados e remova-os.

Aumentar o tamanho do cache ao usar sequências

Se suas tabelas usarem sequências, aumente o tamanho do cache para remover a contenção nas páginas de sequência e nas páginas de índice. Cada sequência é uma única página na memória compartilhada. O cache predefinido é de acordo com a conexão. Isso pode não ser suficiente para lidar com a workload quando muitas sessões simultâneas estão recebendo um valor de sequência.

LWLock:lock_manager (LWLock:lockmanager)

Esse evento ocorre quando o mecanismo do RDS para PostgreSQL mantém a área de memória do bloqueio compartilhado para alocar, conferir e desalocar um bloqueio nos casos em que um bloqueio de caminho rápido não é possível.

Tópicos

- Versões compatíveis do mecanismo
- Contexto
- Possíveis causas do maior número de esperas
- Ações

Versões compatíveis do mecanismo

As informações de eventos de espera são relevantes para o RDS para PostgreSQL versão 9.6 e posteriores. Para versões do RDS para PostgreSQL anteriores à versão 13, o nome desse evento de espera é `LWLock:lock_manager`. Para o RDS para PostgreSQL versão 13 e posteriores, o nome desse evento de espera é `LWLock:lockmanager`.

Contexto

Quando você emite uma instrução SQL, o RDS para PostgreSQL registra bloqueios para proteger a estrutura, os dados e a integridade do banco de dados durante operações simultâneas. O mecanismo pode atingir esse objetivo utilizando um bloqueio de caminho rápido ou um bloqueio de caminho que não é rápido. Um bloqueio de caminho que não é rápido é mais caro e gera mais sobrecarga do que um bloqueio de caminho rápido.

Bloqueio de caminho rápido

Para reduzir a sobrecarga de bloqueios que são retirados e liberados com frequência, mas que raramente entram em conflito, os processos de backend podem utilizar o bloqueio de caminho rápido. O banco de dados usa esse mecanismo para bloqueios que atendem aos seguintes critérios:

- Usam o método de bloqueio DEFAULT.
- Representam um bloqueio em uma relação de banco de dados em vez de uma relação compartilhada.
- São bloqueios fracos que provavelmente não entrarão em conflito.
- O mecanismo é capaz de verificar rapidamente se nenhum bloqueio conflitante pode existir.

O mecanismo não pode utilizar o bloqueio rápido de caminho quando uma das seguintes condições é verdadeira:

- O bloqueio não atende aos critérios anteriores.
- Não há mais slots disponíveis para o processo de backend.

Para ajustar as consultas para bloqueio rápido, é possível usar a consulta a seguir.

```
SELECT count(*), pid, mode, fastpath
  FROM pg_locks
 WHERE fastpath IS NOT NULL
 GROUP BY 4,3,2
 ORDER BY pid, mode;
 count | pid  |      mode       | fastpath
-------+------+-----------------+----------
16 | 9185 | AccessShareLock | t
336 | 9185 | AccessShareLock | f
1 | 9185 | ExclusiveLock   | t
```

A consulta a seguir mostra somente o total em todo o banco de dados.

```
SELECT count(*), mode, fastpath
  FROM pg_locks
 WHERE fastpath IS NOT NULL
 GROUP BY 3,2
 ORDER BY mode,1;
count |      mode       | fastpath
-------+-----------------+----------
16 | AccessShareLock | t
337 | AccessShareLock | f
1 | ExclusiveLock   | t
(3 rows)
```

Para obter mais informações sobre o bloqueio de caminho rápido, consulte o tópico sobre caminho rápido, no README do gerenciador de bloqueios do PostgreSQL epg-locks na documentação do PostgreSQL.

Exemplo de um problema de escalabilidade para o gerenciador de bloqueios

Neste exemplo, uma tabela chamada `purchases` armazena cinco anos de dados, particionados por dia. Cada partição possui dois índices. A seguinte sequência de eventos ocorre:

1. Você consulta muitos dias de dados, o que exige que o banco de dados leia várias partições.
2. O banco de dados cria uma entrada de bloqueio para cada partição. Se os índices de partição fizerem parte do caminho de acesso do otimizador, o banco de dados também criará uma entrada de bloqueio para eles.
3. Quando o número de entradas de bloqueios solicitadas para o mesmo processo de backend for maior que 16, que é o valor de `FP_LOCK_SLOTS_PER_BACKEND`, o gerenciador de bloqueio usará o método de bloqueio de caminho não rápido.

Aplicações modernas podem ter centenas de sessões. Se sessões simultâneas estiverem consultando o pai sem a devida limpeza da partição, o banco de dados poderá criar centenas ou até milhares de bloqueios de caminho não rápidos. Em geral, quando essa simultaneidade é maior que o número de vCPUs, o evento de espera `LWLock:lock_manager` é exibido.

> **Note**
>
> O evento de espera `LWLock:lock_manager` não está relacionado ao número de partições ou índices em um esquema de banco de dados e sim ao número de bloqueios de caminho não rápidos que o banco de dados deve controlar.

Possíveis causas do maior número de esperas

Quando o evento de espera `LWLock:lock_manager` ocorre mais do que o normal, possivelmente indicando um problema de performance, as causas mais prováveis de picos súbitos são:

- Sessões ativas simultâneas estão executando consultas que não utilizam bloqueios de caminho rápido. Essas sessões também excedem a vCPU máxima.
- Um número elevado de sessões ativas simultâneas está acessando uma tabela fortemente particionada. Cada partição possui vários índices.
- O banco de dados está passando por uma tempestade de conexões. Por padrão, algumas aplicações e softwares de grupo de conexões criam mais conexões quando o banco de dados está lento. Essa prática piora o problema. Ajuste o software do grupo de conexões para que tempestades de conexões não ocorram.
- Um número elevado de sessões consulta uma tabela pai sem separar partições.
- Um comando de linguagem de definição de dados (DDL), linguagem de manipulação de dados (DML) ou manutenção bloqueia exclusivamente uma relação ocupada ou tuplas que são frequentemente acessadas ou modificadas.

Ações

Se o evento de espera `CPU` ocorrer, isso não indica necessariamente um problema de performance. Responda a esse evento somente quando a performance piorar e esse evento de espera estiver dominando a carga do banco de dados.

Tópicos

- Usar a limpeza de partição
- Remover índices desnecessários
- Ajustar suas consultas para bloqueio de caminho rápido
- Fazer ajustes com base em eventos de espera

- Reduzir gargalos de hardware
- Usar um agrupador de conexões
- Fazer upgrade da versão do RDS para PostgreSQL

Usar a limpeza de partição

Limpeza de partição é uma estratégia de otimização de consultas para tabelas declarativamente particionadas que exclui partições desnecessárias das verificações de tabelas, melhorando assim a performance. Por padrão, a remoção de partição está ativada. Se ela estiver desativada, ative-a da seguinte maneira.

```
SET enable_partition_pruning = on;
```

As consultas podem tirar proveito da limpeza de partição quando suas cláusula `WHERE` contêm a coluna utilizada para o particionamento. Para obter mais informações, consulte o tópico sobre Limpeza de partição, na documentação do PostgreSQL.

Remover índices desnecessários

Seu banco de dados pode conter índices não utilizados ou raramente utilizados. Se esse for o caso, considere excluí-los. Realize um dos procedimentos a seguir:

- Saiba mais sobre como encontrar índices desnecessários lendo o tópico sobre Índices não utilizados na wiki do PostgreSQL.
- Execute o PG Collector. Esse script SQL reúne informações do banco de dados e as apresenta em um relatório HTML consolidado. Confira a seção “Índices não utilizados”. Para obter mais informações, consulte pg-collector no Repositório AWS Labs GitHub.

Ajustar suas consultas para bloqueio de caminho rápido

Para descobrir se as suas consultas usam o bloqueio de caminho rápido, consulte a coluna `fastpath` na tabela `pg_locks`. Se as suas consultas não estiverem utilizando o bloqueio de caminho rápido, tente reduzir o número de relações por consulta para menos de 16.

Fazer ajustes com base em eventos de espera

Se `LWLock:lock_manager` for o primeiro ou o segundo na lista de principais esperas, verifique se os seguintes eventos de espera também aparecem na lista:

- `Lock:Relation`
- `Lock:transactionid`
- `Lock:tuple`

Se os eventos anteriores aparecerem em posição elevada na lista, considere ajustar esses eventos de espera primeiro. Esses eventos podem ser um fator impulsionador para `LWLock:lock_manager`.

Reduzir gargalos de hardware

É possível que haja um gargalo de hardware, como falta de CPU ou uso máximo da largura de banda do Amazon EBS. Nesses casos, considere reduzir gargalos de hardware. Considere as ações a seguir:

- Aumente a escala da sua classe de instância na vertical.
- Otimize consultas que consomem grandes quantidades de CPU e memória.
- Altere a lógica da aplicação.
- Arquive os dados.

Para obter mais informações sobre CPU, memória e largura de banda da rede do EBS, consulte Tipos de instâncias do Amazon RDS.

Usar um agrupador de conexões

Se o número total de conexões ativas exceder o máximo de vCPU, mais processos do SO exigirão CPU do que o tipo de instância pode suportar. Nesse caso, considere utilizar ou ajustar um grupo de conexões. Para obter mais informações sobre as vCPUs para o seu tipo de instância, consulte o tópico sobre Tipos de instância do Amazon RDS.

Para obter mais informações sobre agrupamento de conexões, consulte os seguintes recursos:

- Amazon RDS Proxy
- pgbouncer
- Grupos conexões e fontes de dados, na Documentação do PostgreSQL

Fazer upgrade da versão do RDS para PostgreSQL

Se a versão atual do RDS para PostgreSQL for menor que 12, atualize para a versão 12 ou posterior. As versões 12 e posteriores do PostgreSQL têm um mecanismo de partição aprimorado. Para obter mais informações sobre a versão 12, consulte Notas de release do PostgreSQL 12.0. Para obter mais informações sobre atualização do RDS para PostgreSQL, consulte Atualizações do mecanismo de banco de dados do RDS para PostgreSQL.

LWLock:pg_stat_statements

O evento de espera LWLock:pg_stat_statements ocorre quando a extensão `pg_stat_statements` obtém um bloqueio exclusivo na tabela de hash que rastreia instruções SQL. Isso acontece nos seguintes cenários:

- Quando o número de instruções rastreadas atinge o valor configurado do parâmetro `pg_stat_statements.max` e é necessário abrir espaço para mais entradas, a extensão executa uma classificação no número de chamadas, remove 5% das instruções menos executadas e preenche novamente o hash com as entradas restantes.
- Quando `pg_stat_statements` executa uma operação `garbage collection` no arquivo `pgss_query_texts.stat` no disco e reescreve o arquivo.

Tópicos

- Versões compatíveis do mecanismo
- Contexto
- Possíveis causas do maior número de esperas
- Ações

Versões compatíveis do mecanismo

É possível obter informações sobre eventos de espera em todas as versões do RDS para PostgreSQL.

Contexto

Conceitos básicos sobre a extensão pg_stat_statements: essa extensão rastreia as estatísticas de execução de instruções SQL em uma tabela de hash. Ela rastreia as instruções SQL até o limite

definido pelo parâmetro `pg_stat_statements.max`. Esse parâmetro determina o número máximo de instruções que podem ser rastreadas, o qual corresponde ao número máximo de linhas na visualização pg_stat_statements.

Persistência das estatísticas de instrução: a extensão mantém as estatísticas da instrução em todas as reinicializações da instância:

- Gravando dados em um arquivo chamado pg_stat_statements.stat.
- Usando o parâmetro pg_stat_statements.save para controlar o comportamento de persistência.

Quando pg_stat_statements.save é definido como:

- ativado (padrão): as estatísticas são salvas no desligamento e recarregadas na inicialização do servidor.
- desativado: as estatísticas não são salvas no desligamento nem recarregadas na inicialização do servidor.

Armazenamento de texto de consulta: a extensão armazena o texto das consultas rastreadas em um arquivo chamado `pgss_query_texts.stat`. Esse arquivo pode aumentar até o dobro do tamanho médio de todas as instruções SQL rastreadas antes que a coleta de resíduos ocorra. A extensão requer um bloqueio exclusivo na tabela de hash durante as operações de limpeza e regravação do arquivo `pgss_query_texts.stat`.

Processo de desalocação de instruções: quando o número de instruções rastreadas atinge o limite `pg_stat_statements.max` e novas instruções precisam ser rastreadas, a extensão:

- Obtém um bloqueio exclusivo (LWLock:pg_stat_statements) na tabela de hash.
- Carrega os dados existentes na memória local.
- Executa uma classificação rápida com base no número de chamadas.
- Remove as instruções menos chamadas (5% inferiores).
- Preenche novamente a tabela de hash com as entradas restantes.

Monitoramento da desalocação de instruções: no PostgreSQL 14 e versões posteriores, você pode monitorar a desalocação de instruções usando a visualização pg_stat_statements_info. Essa visualização inclui uma coluna de desalocação que mostra quantas vezes as instruções foram desalocadas para abrir espaço para novas.

Se a desalocação de instruções ocorrer com frequência, haverá uma coleta de resíduos mais frequente do arquivo `pgss_query_texts.stat` no disco.

Possíveis causas do maior número de esperas

As causas típicas do aumento de esperas `LWLock:pg_stat_statements` incluem:

- Um aumento no número de consultas exclusivas usadas pela aplicação.
- O valor do parâmetro `pg_stat_statements.max` é pequeno em comparação ao número de consultas exclusivas que estão sendo usadas.

Ações

Recomenda-se ações distintas, dependendo dos motivos do evento de espera. Você pode identificar eventos `LWLock:pg_stat_statements` utilizando o Insights de Performance do Amazon RDS ou consultando a visualização `pg_stat_activity`.

Ajuste os parâmetros `pg_stat_statements` a seguir para controlar o comportamento de rastreamento e reduzir os eventos de espera das instruções LWLock:pg_stat_.

Tópicos

- Desabilitar o parâmetro pg_stat_statements.track
- Aumentar o parâmetro pg_stat_statements.max
- Desabilitar o parâmetro pg_stat_statements.track_utility

Desabilitar o parâmetro pg_stat_statements.track

Se o evento de espera LWLock:pg_stat_statements estiver afetando negativamente o desempenho do banco de dados e for necessária uma solução rápida antes de uma análise mais aprofundada da visualização `pg_stat_statements` para identificar a causa-raiz, o parâmetro `pg_stat_statements.track` poderá ser desabilitado configurando-o como `none`. Isso desabilitará a coleta de estatísticas de instrução.

Aumentar o parâmetro pg_stat_statements.max

Para reduzir a desalocação e minimizar a coleta de resíduos do arquivo `pgss_query_texts.stat` no disco, aumente o valor do parâmetro `pg_stat_statements.max`. O valor padrão é `5,000`.

> **Note**
>
> O parâmetro `pg_stat_statements.max` é estático. É necessário reinicializar a instância de banco de dados para aplicar qualquer alteração nesse parâmetro.

Desabilitar o parâmetro pg_stat_statements.track_utility

Você pode analisar a visualização pg_stat_statements para determinar quais comandos do utilitário estão consumindo mais recursos rastreados por `pg_stat_statements`.

O parâmetro `pg_stat_statements.track_utility` controla se o módulo rastreia os comandos do utilitário, que incluem todos os comandos, exceto SELECT, INSERT, UPDATE, DELETE e MERGE. Esse parâmetro é definido como on por padrão.

Por exemplo, quando sua aplicação usa muitas consultas de ponto de salvamento, que são inerentemente exclusivas, ela pode aumentar a desalocação de instruções. Para resolver isso, você pode desabilitar o parâmetro `pg_stat_statements.track_utility` para impedir que `pg_stat_statements` rastreie consultas de pontos de salvamento.

> **Note**
>
> O parâmetro `pg_stat_statements.track_utility` é dinâmico. Você pode alterar o respectivo valor sem reiniciar a instância de banco de dados.

Example Exemplo de consultas exclusivas de ponto de salvamento em pg_stat_statements

```
                    query                         |       queryid
--------------------------------------------------+---------------------
 SAVEPOINT JDBC_SAVEPOINT_495701                  | -7249565344517699703
 SAVEPOINT JDBC_SAVEPOINT_1320                    | -1572997038849006629
 SAVEPOINT JDBC_SAVEPOINT_26739                   |  54791337410474486
 SAVEPOINT JDBC_SAVEPOINT_1294466                 |  8170064357463507593
 ROLLBACK TO SAVEPOINT JDBC_SAVEPOINT_65016       | -33608214779996400
 SAVEPOINT JDBC_SAVEPOINT_14185                   | -2175035613806809562
 SAVEPOINT JDBC_SAVEPOINT_45837                   | -6201592986750645383
 SAVEPOINT JDBC_SAVEPOINT_1324                    |  6388797791882029332
```

O PostgreSQL 17 inclui vários aprimoramentos para o rastreamento de comandos de utilitário:

- Os nomes de ponto de salvamento agora são exibidos como constantes.
- Os IDs de transação global (GIDs) dos comandos de confirmação em duas fases agora são exibidos como constantes.
- Os nomes das instruções DEALLOCATE são mostrados como constantes.
- Os parâmetros CALL agora são exibidos como constantes.

Tempo limite:PgSleep

O evento `Timeout:PgSleep` ocorre quando um processo do servidor chama a função `pg_sleep` e está aguardando o tempo limite de suspensão expirar.

Tópicos

- Versões compatíveis do mecanismo
- Possíveis causas do maior número de esperas
- Ações

Versões compatíveis do mecanismo

Essas informações de eventos de espera são compatíveis com todas as versões do RDS para PostgreSQL.

Possíveis causas do maior número de esperas

Esse evento de espera ocorre quando uma aplicação, uma função armazenada ou um usuário emite uma instrução SQL que chama uma das seguintes funções:

- `pg_sleep`
- `pg_sleep_for`
- `pg_sleep_until`

As funções anteriores atrasarão a execução até que o número especificado de segundos tenha decorrido. Por exemplo, `SELECT pg_sleep(1)` pausa por 1 segundo. Para obter mais informações, consulte Atrasar a execução na documentação do PostgreSQL.

Ações

Identifique a instrução que estava executando a função `pg_sleep`. Determine se o uso da função é apropriado.

Timeout:VacuumDelay

O evento `Timeout:VacuumDelay` indica que o limite de custo para E/S de vácuo foi excedido e que o processo de vácuo foi suspenso. As operações de vácuo são interrompidas pelo período especificado no respectivo parâmetro de atraso de custo e, depois, ele retoma seu trabalho. Para o comando de vácuo manual, o atraso é especificado no parâmetro `vacuum_cost_delay`. Para o daemon de vácuo automático, o atraso é especificado no `autovacuum_vacuum_cost_delay parameter.`

Tópicos

- Versões compatíveis do mecanismo
- Contexto
- Possíveis causas do maior número de esperas
- Ações

Versões compatíveis do mecanismo

Essas informações de eventos de espera são compatíveis com todas as versões do RDS para PostgreSQL.

Contexto

O PostgreSQL tem um daemon de vácuo automático e um comando de vácuo manual. O processo de vácuo automático está “ativado” por padrão para instâncias de banco de dados do RDS para PostgreSQL. O comando de vácuo manual é utilizado conforme a necessidade, por exemplo, para limpar tabelas de tuplas inoperantes ou gerar novas estatísticas.

Quando a aspiração está em andamento, o PostgreSQL usa um contador interno para acompanhar os custos estimados à medida que o sistema executa várias operações de E/S. Quando o contador atinge o valor especificado pelo parâmetro de limite de custo, o processo que executa a operação permanece inativo pelo breve período especificado no parâmetro de atraso de custo. Depois, ele redefine o contador e continua as operações.

O processo de vácuo tem parâmetros que podem ser usados para regular o consumo de recursos. O vácuo automático e o comando de vácuo manual têm seus próprios parâmetros para definir o valor limite de custo. Eles também têm seus próprios parâmetros para especificar um atraso de custo, uma quantidade de tempo para colocar o vácuo em repouso quando o limite é atingido. Dessa forma, o parâmetro de atraso de custo funciona como um controle de utilização do consumo de recursos. Nas listas a seguir, você pode encontrar a descrição desses parâmetros.

Parâmetros que afetam o controle de utilização do daemon de vácuo automático

- `autovacuum_vacuum_cost_limit`: especifica o valor limite de custo a ser utilizado em operações automáticas de vácuo. Aumentar a configuração desse parâmetro permite que o processo de vácuo use mais recursos e diminua o evento de espera `Timeout:VacuumDelay`.
- `autovacuum_vacuum_cost_delay`: especifica o valor de atraso de custo a ser utilizado em operações automáticas de vácuo. O valor padrão é 2 milissegundos. Definir o parâmetro de atraso como 0 desativa o controle de utilização e, portanto, o evento de espera `Timeout:VacuumDelay` não será exibido.

Para ter mais informações, consulte Automatic Vacuuming (Aplicação automática do vacuum) na documentação do PostgreSQL.

Parâmetros que afetam o controle de utilização do processo de vácuo manual

- `vacuum_cost_limit`: o limite no qual o processo de vácuo é suspenso. Por padrão, o limite é 200. Esse número representa as estimativas de custo acumuladas para E/S extras necessárias para vários recursos. Aumentar esse valor reduz o número do evento de espera `Timeout:VacuumDelay`.
- `vacuum_cost_delay`: a quantidade de tempo que o processo de vácuo permanece inativo quando o limite de custo do vácuo é atingido. A configuração padrão é 0, o que significa que esse recurso está desativado. Você pode definir isso como um valor inteiro para especificar o número de milissegundos para ativar esse recurso, mas recomendamos que você o deixe na configuração padrão.

Para obter mais informações sobre o parâmetro `vacuum_cost_delay`, consulte Resource Consumption (Consumo de recursos) na documentação do PostgreSQL.

Para saber mais sobre como configurar e usar o vácuo automático com o RDS for PostgreSQL, consulte Trabalhar com o autovacuum do PostgreSQL no Amazon RDS para PostgreSQL.

Possíveis causas do maior número de esperas

O `Timeout:VacuumDelay` é afetado pelo equilíbrio entre as configurações dos parâmetros de limite de custo (`vacuum_cost_limit`, `autovacuum_vacuum_cost_limit`) e os parâmetros de atraso de custo (`vacuum_cost_delay`, `autovacuum_vacuum_cost_delay`) que controlam a duração da suspensão do vácuo. Aumentar o valor de um parâmetro de limite de custo permite que mais recursos sejam utilizados pelo vácuo antes de serem colocados em repouso. Isso ocasiona menos eventos de espera `Timeout:VacuumDelay`. O aumento de qual quer um dos parâmetros de atraso faz com que o evento de espera `Timeout:VacuumDelay` ocorra com maior frequência e por longos períodos.

A configuração do parâmetro `autovacuum_max_workers` também pode aumentar o número de `Timeout:VacuumDelay`. Cada processo adicional de processamento do vácuo automático contribui para o mecanismo interno do contador e, portanto, o limite pode ser atingido mais rapidamente do que com um único processo de vácuo automático. À medida que o limite de custo é atingido mais rapidamente, o atraso de custo é aplicado com maior frequência, ocasionando mais eventos de espera `Timeout:VacuumDelay`. Para obter mais informações, consulte autovacuum_max_workers na documentação do PostgreSQL.

Objetos grandes, como 500 GB ou maiores, também aumentam esse evento de espera porque pode levar algum tempo para que o vácuo conclua o processamento de objetos grandes.

Ações

Se as operações de vácuo forem concluídas conforme o esperado, nenhuma correção será necessária. Em outras palavras, esse evento de espera não indica necessariamente um problema. Isso indica que o vácuo está sendo colocado em repouso pelo período especificado no parâmetro de atraso para que os recursos possam ser aplicados a outros processos que precisam ser concluídos.

Se quiser que as operações de vácuo sejam concluídas mais rapidamente, você poderá reduzir os parâmetros de atraso. Isso reduz o tempo em que o vácuo permanece suspenso.

Ajustar o RDS para PostgreSQL com insights proativos do Amazon DevOps Guru

Os insights proativos do DevOps Guru detectam condições em instâncias de banco de dados do RDS para PostgreSQL que podem causar problemas, permitindo que você as conheça antes que ocorram. Os insights proativos podem alertar quando ocorre um período longo de inatividade na conexão da transação. Para ter mais informações sobre como solucionar períodos longos de inatividade nas conexões das transações, consulte O banco de dados está inativo há muito tempo na conexão da transação.

O DevOps Guru pode fazer o seguinte:

- Evitar muitos problemas comuns de bancos de dados ao comparar a configuração do seu banco de dados com as configurações comuns recomendadas.
- Alertar sobre problemas críticos em sua frota que, se não forem controlados, poderão causar problemas maiores no futuro.
- Alertar sobre problemas recém-descobertos.

Cada insight proativo contém uma análise da causa do problema e recomendações de ações corretivas.

Para ter informações sobre o Amazon DevOps Guru para Amazon RDS, consulte Analisar anomalias de performance com o DevOps Guru para Amazon RDS.

O banco de dados está inativo há muito tempo na conexão da transação

A conexão com o banco de dados está no estado `idle in transaction` há mais de 1.800 segundos.

Tópicos

- Versões compatíveis do mecanismo
- Contexto
- Causas prováveis desse problema
- Ações
- Métricas relevantes

Versões compatíveis do mecanismo

Essas informações de insights são compatíveis com todas as versões do RDS para PostgreSQL.

Contexto

Uma transação no estado `idle in transaction` pode conter bloqueios que impedem outras consultas. Também pode impedir que `VACUUM` (incluindo o autovacuum) limpe linhas mortas, causando inchaço no índice ou na tabela ou recorrência de IDs de transação.

Causas prováveis desse problema

Uma transação iniciada em uma sessão interativa com BEGIN ou START TRANSACTION não terminou usando um comando COMMIT, ROLLBACK ou END. Isso faz com que a transação passe para o estado `idle in transaction`.

Ações

Você pode encontrar transações inativas consultando `pg_stat_activity`.

Em seu cliente SQL, execute a seguinte consulta para listar todas as conexões em estado `idle in transaction` e ordená-las por duração:

```
SELECT now() - state_change as idle_in_transaction_duration, now() - xact_start as
 xact_duration,*
FROM  pg_stat_activity
WHERE state  = 'idle in transaction'
AND   xact_start is not null
ORDER BY 1 DESC;
```

Recomendamos ações distintas dependendo dos motivos do insight.

Tópicos

- Encerrar transação
- Encerrar a conexão
- Configurar o parâmetro idle_in_transaction_session_timeout
- Verificar o status de AUTOCOMMIT
- Verificar a lógica da transação no código da aplicação

Encerrar transação

Quando você inicia uma transação em uma sessão interativa com BEGIN ou START TRANSACTION, ela passa para o estado `idle in transaction`. Ela permanecerá nesse estado até você encerrar a transação emitindo um comando COMMIT, ROLLBACK ou END, ou até desfazer completamente a conexão para reverter a transação.

Encerrar a conexão

Encerre a conexão com uma transação inativa usando a seguinte consulta:

```
SELECT pg_terminate_backend(pid);
```

pid é o ID de processo da conexão.

Configurar o parâmetro idle_in_transaction_session_timeout

Configure o parâmetro `idle_in_transaction_session_timeout` no grupo de parâmetros. A vantagem de configurar esse parâmetro é que ele não requer uma intervenção manual para encerrar transações que estão inativas há muito tempo. Para obter mais informações sobre esse parâmetro, consulte a documentação do PostgreSQL.

A mensagem a seguir será relatada no arquivo de log do PostgreSQL depois do encerramento da conexão quando uma conexão permanecer no estado idle_in_transaction por mais tempo do que o tempo especificado.

```
FATAL: terminating connection due to idle in transaction timeout
```

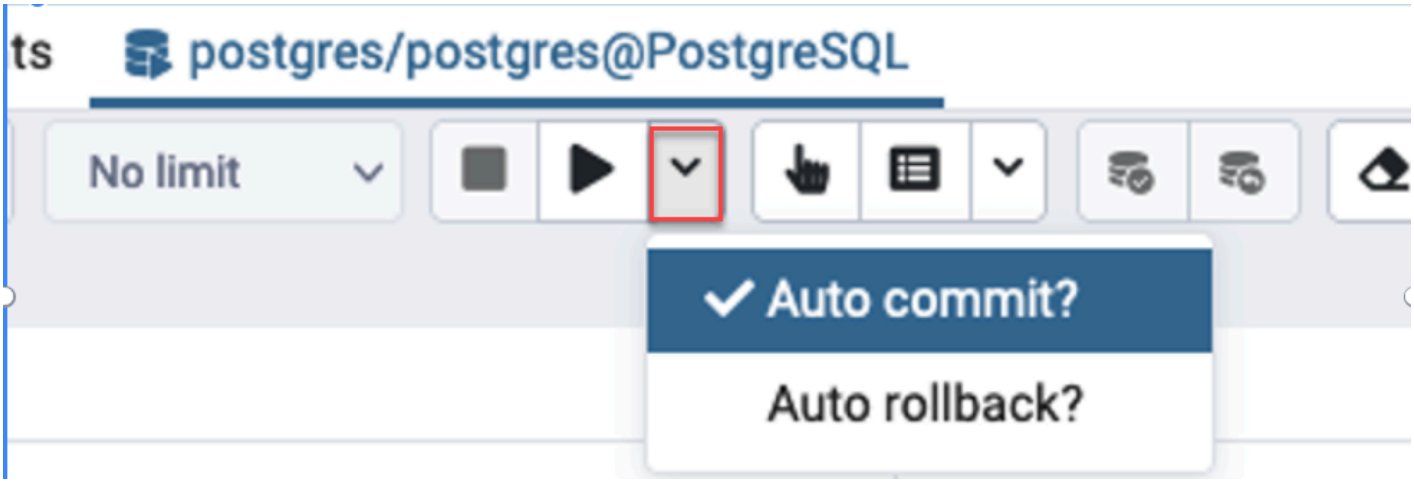
Verificar o status de AUTOCOMMIT

A opção AUTOCOMMIT está ativada por padrão. Se for acidentalmente desativada no cliente, ative-a novamente.

- No cliente psql, execute o seguinte comando:

  ```
  postgres=> \set AUTOCOMMIT on
  ```

- Em pgadmin, ative-a escolhendo a opção AUTOCOMMIT na seta para baixo.



Verificar a lógica da transação no código da aplicação

Investigue a lógica da aplicação em busca de possíveis problemas. Considere as ações a seguir:

- Verifique se a confirmação automática do JDBC está definida como true na aplicação. Além disso, considere usar comandos `COMMIT` explícitos em seu código.
- Verifique sua lógica de tratamento de erros para ver se ela fecha uma transação depois de um erro.
- Verifique se a aplicação está demorando para processar as linhas retornadas por uma consulta enquanto a transação está aberta. Nesse caso, considere programa a aplicação para fechar a transação antes de processar as linhas.
- Verifique se uma transação contém muitas operações de longa duração. Em caso afirmativo, divida uma única transação em várias transações.

Métricas relevantes

As seguintes métricas de PI estão relacionadas a esse insight:

- idle_in_transaction_count: número de sessões no estado `idle in transaction`.
- idle_in_transaction_max_time: a duração da transação que passou mais tempo no estado `idle in transaction`.

Usar extensões PostgreSQL com o Amazon RDS para PostgreSQL

É possível estender a funcionalidade do PostgreSQL instalando uma variedade de extensões e módulos. Por exemplo, para trabalhar com dados espaciais, você pode instalar e usar a extensão PostGIS. Para obter mais informações, consulte Gerenciar dados espaciais com a extensão PostGIS. Como outro exemplo, se você quiser melhorar a entrada de dados para tabelas muito grandes, considere particionar seus dados usando `pg_partman`. Para saber mais, consulte Gerenciar partições do PostgreSQL com a extensão pg_partman.

> **Note**
>
> O RDS para PostgreSQL comporta extensões de linguagem confiáveis para PostgreSQL por meio da extensão `pg_tle`, que você pode adicionar à sua instância de banco de dados. Ao usar essa extensão, os desenvolvedores podem criar suas próprias extensões do PostgreSQL em um ambiente seguro que simplifique os requisitos de instalação e configuração. Para saber mais sobre as versões do RDS para PostgreSQL que aceitam a extensão `pg_tle` e ter mais informações, consulte Trabalhar com Trusted Language Extensions para PostgreSQL.

Em alguns casos, em vez de instalar uma extensão, você pode adicionar um módulo específico à lista de `shared_preload_libraries` no grupo de parâmetros de banco de dados personalizado da instância de banco de dados do RDS para PostgreSQL. Normalmente, o grupo de parâmetros padrão do cluster de banco de dados carrega somente as `pg_stat_statements`, mas vários outros módulos estão disponíveis para serem adicionados à lista. Por exemplo, você pode incluir a capacidade de agendamento adicionando o módulo `pg_cron`, conforme detalhado em Agendar manutenção com a extensão pg_cron do PostgreSQL. Como outro exemplo, você pode registrar planos de execução de consultas carregando o módulo `auto_explain`. Para saber mais, consulte Logging execution plans of queries (Registrar em log planos de execução de consultas) no Centro de Conhecimentos da AWS.

Dependendo da versão do RDS para PostgreSQL, a instalação de uma extensão pode exigir permissões de `rds_superuser`, como a seguir:

- Para o RDS para PostgreSQL versões 12 e versões anteriores, a instalação de extensões requer privilégios de `rds_superuser`.
- Para o RDS para PostgreSQL versão 13 e versões superiores, os usuários (perfis) com permissões de criação em uma determinada instância de banco de dados podem instalar e usar

quaisquer extensões confiáveis. Para obter uma lista de extensões confiáveis, consulte Extensões confiáveis do PostgreSQL.

Também é possível especificar com precisão quais extensões podem ser instaladas na instância de banco de dados do RDS para PostgreSQL, listando-as no parâmetro `rds.allowed_extensions`. Para obter mais informações, consulte Restringir a instalação de extensões do PostgreSQL.

Para saber mais a respeito do perfil de `rds_superuser`, consulte Noções básicas de perfis e permissões do PostgreSQL.

Tópicos

- Usar funções da extensão orafce
- Usar a compatibilidade com extensões delegadas do Amazon RDS para PostgreSQL
- Gerenciar partições do PostgreSQL com a extensão pg_partman
- Usar pgAudit para registrar a atividade do banco de dados
- Agendar manutenção com a extensão pg_cron do PostgreSQL
- Usar pglogical para sincronizar dados entre instâncias
- Usar pgactive para comportar a replicação ativa-ativa
- Reduzir o inchaço em tabelas e índices com a extensão pg_repack
- Atualizar e usar a extensão PLV8
- Usar PL/Rust para escrever funções do PostgreSQL na linguagem Rust
- Gerenciar dados espaciais com a extensão PostGIS

Usar funções da extensão orafce

A extensão orafce fornece funções e operadores que emulam um subconjunto de funções e pacotes de um banco de dados Oracle. A extensão orafce permite fazer a portabilidade de uma aplicação Oracle para o PostgreSQL com mais facilidade. Essa extensão é compatível com o RDS para PostgreSQL versões 9.6.6 e posteriores. Para obter mais informações sobre a extensão, consulte o orafce no GitHub.

> **Note**
>
> O RDS para PostgreSQL não é compatível com o pacote `utl_file` que faz parte da extensão orafce. Isso ocorre porque as funções do esquema `utl_file` fornecem operações

de leitura e gravação em arquivos de texto do sistema operacional, o que exige que o superusuário acesse o host subjacente. Como um serviço gerenciado, o RDS para PostgreSQL não fornece acesso ao host.

Para usar a extensão orafce

1. Conecte-se à instância de banco de dados com o nome do usuário principal que você usou para criar a instância de banco de dados.

   Se você quiser ativar a extensão orafce para um banco de dados diferente na mesma instância de banco de dados, use o comando `/c dbname` do psql. Usando esse comando, você muda do banco de dados primário depois de iniciar a conexão.
2. Ativar a extensão orafce com a instrução `CREATE EXTENSION`.

   ```
   CREATE EXTENSION orafce;
   ```
3. Transfira propriedade do esquema oracle para a função rds_superuser com a instrução `ALTER SCHEMA`.

   ```
   ALTER SCHEMA oracle OWNER TO rds_superuser;
   ```

   Caso queira ver uma lista de proprietários do esquema oracle, use o comando de psql `\dn`.

Usar a compatibilidade com extensões delegadas do Amazon RDS para PostgreSQL

Usando a compatibilidade com extensões delegadas do PostgreSQL, é possível delegar o gerenciamento de extensões a um usuário que não precise ser um `rds_superuser`. Com esse suporte de extensão delegado, um novo perfil chamado `rds_extension` é criado, e você deve atribuí-lo a um usuário para gerenciar outras extensões. Esse perfil pode criar, atualizar e eliminar extensões.

É possível especificar quais extensões podem ser instaladas na instância de banco de dados do RDS listando-as no parâmetro `rds.allowed_extensions`. Para obter mais informações, consulte Usar extensões PostgreSQL com o Amazon RDS para PostgreSQL.

É possível restringir a lista de extensões disponíveis que podem ser gerenciadas pelo usuário com o perfil `rds_extension` usando o parâmetro `rds.allowed_delegated_extensions`.

O suporte de extensão delegado está disponível nas seguintes versões:

- Todas as versões posteriores
- 16.4 e versões 16 posteriores
- 15.8 e versões 15 posteriores
- 14.13 e versões 14 posteriores
- 13.16 e versões 13 posteriores
- 12.20 e versões 12 posteriores

Tópicos

- Ativar o suporte de extensão delegado a um usuário
- Configuração usada na compatibilidade com extensões delegadas do PostgreSQL
- Desativar o suporte para a extensão delegada
- Benefícios de usar a compatibilidade com extensões delegadas do Amazon RDS
- Limitação da compatibilidade com extensões delegadas do PostgreSQL
- Permissões necessárias para determinadas extensões
- Considerações sobre segurança
- Descartar cascata de extensão desabilitado

- Exemplos de extensões que podem ser adicionadas usando o suporte de extensão delegado

Ativar o suporte de extensão delegado a um usuário

É necessário executar o seguinte para habilitar o suporte de extensão delegado a um usuário:

1. Conceder o perfil **`rds_extension`** a um usuário: conecte-se ao banco de dados `rds_superuser` e execute o seguinte comando:

```
Postgres => grant rds_extension to user_name;
```

2. Definir a lista de extensões disponíveis para usuários delegados gerenciarem: o `rds.allowed_delegated_extensions` permite que você especifique um subconjunto das extensões disponíveis usando o parâmetro de cluster de banco de dados `rds.allowed_extensions`. É possível fazer isso em um dos seguintes níveis:

   - No cluster ou no grupo de parâmetros da instância, por meio do AWS Management Console ou da API. Para ter mais informações, consulte Grupos de parâmetros para Amazon RDS.
   - Use o seguinte comando em nível de banco de dados:

```
alter database database_name set rds.allowed_delegated_extensions =
 'extension_name_1,
                   extension_name_2,...extension_name_n';
```

   - Use o seguinte comando em nível de usuário:

```
alter user user_name set rds.allowed_delegated_extensions = 'extension_name_1,
                   extension_name_2,...extension_name_n';
```

> **Note**
>
> Não é necessário reiniciar o banco de dados depois de alterar o parâmetro dinâmico `rds.allowed_delegated_extensions`.

3. Permitir que o usuário delegado acesse os objetos criados durante o processo de criação da extensão: determinadas extensões criam objetos que exigem que permissões adicionais sejam concedidas para que o usuário com o perfil `rds_extension` possa acessá-los. O `rds_superuser` deve conceder ao usuário delegado acesso a esses objetos. Uma das opções

é usar um gatilho de eventos para conceder permissão automaticamente ao usuário delegado. Para ter mais informações, consulte o exemplo do gatilho de eventos em Desativar o suporte para a extensão delegada.

Configuração usada na compatibilidade com extensões delegadas do PostgreSQL

Nome da configuração	Descrição	Valor padrão	Observações	Quem pode modificar ou conceder permissão
`rds.allowed_delegated_extensions`	Esse parâmetro limita as extensões que um perfil rds_extension pode gerenciar em um banco de dados. Ele deve ser um subconjunto de rds.allowed_extensions.	string vazia	• Por padrão, esse parâmetro é uma string vazia, o que significa que nenhuma extensão foi delegada aos usuários com `rds_extension` . • Qualquer extensão compatível poderá ser adicionada se o usuário tiver permissões para isso. Para fazer isso, defina o parâmetro `rds.allowed_delegated_extensions` como uma string	rds_superuser

Nome da configuração	Descrição	Valor padrão	Observações	Quem pode modificar ou conceder permissão
			de nomes de extensão separados por vírgulas. Ao adicionar uma lista de extensões a esse parâmetro, você deve identificar explicitamente as extensões que o usuário com o perfil `rds_extension` pode instalar. • Quando definido como `*`, significa que todas as extensões listadas em `rds_allowed_extensions` são delegadas aos usuários com o perfil `rds_extension`. Para saber mais sobre como configurar esse	

Nome da configuração	Descrição	Valor padrão	Observações	Quem pode modificar ou conceder permissão
			parâmetro, consulte Ativar o suporte de extensão delegado a um usuário.	
`rds.allowed_extensions`	Esse parâmetro permite que o cliente limite as extensões que podem ser instaladas na instância de banco de dados do RDS. Para ter mais informações, consulte Restringir a instalação de extensões do PostgreSQL.	"*"	Por padrão, esse parâmetro é definido como “*”, o que significa que todas as extensões aceitas no RDS para PostgreSQL e no Aurora PostgreSQL podem ser criadas por usuários com os privilégios necessários. Vazio significa que nenhuma extensão pode ser instalada na instância de banco de dados do RDS.	Administrador

Nome da configuração	Descrição	Valor padrão	Observações	Quem pode modificar ou conceder permissão
`rds-delegated_extension_allow_drascac`	Esse parâmetro controla a capacidade do usuário com `rds_extension` de eliminar a extensão usando uma opção em cascata.	off	Por padrão, `rds-delegated_extension_allow_drop_cascade` é definido como `off`. Isso significa que os usuários com `rds_extension` não têm permissão para descartar uma extensão usando a opção em cascata. Para conceder essa capacidade, o parâmetro `rds.delegated_extension_allow_drop_cascade` deve ser definido como on.	rds_superuser

Desativar o suporte para a extensão delegada

Desativar parcialmente

Os usuários delegados não podem criar extensões, mas ainda podem atualizar as existentes.

- Redefina `rds.allowed_delegated_extensions` como o valor padrão no grupo de parâmetros de cluster de banco de dados.

- Use o seguinte comando em nível de banco de dados:

```
alter database database_name reset rds.allowed_delegated_extensions;
```

- Use o seguinte comando em nível de usuário:

```
alter user user_name reset rds.allowed_delegated_extensions;
```

Desativar totalmente

Revogar o perfil `rds_extension` de um usuário vai restaurar as permissões padrão do usuário. O usuário não pode mais criar, atualizar nem descartar extensões.

```
postgres => revoke rds_extension from user_name;
```

Exemplo de gatilho de eventos

Se quiser permitir que um usuário delegado com `rds_extension` use extensões que exijam permissões de configuração nos objetos criados pela criação da extensão, você poderá personalizar o exemplo abaixo de um gatilho de eventos e adicionar somente as extensões para as quais você deseja que os usuários delegados tenham acesso à funcionalidade completa. Esse gatilho de eventos pode ser criado no modelo 1 (o modelo padrão), portanto, todo banco de dados criado a partir do modelo 1 terá esse gatilho de eventos. Quando um usuário delegado instalar a extensão, esse gatilho concederá automaticamente a propriedade dos objetos criados pela extensão.

```
CREATE OR REPLACE FUNCTION create_ext()

  RETURNS event_trigger AS $$

DECLARE

  schemaname TEXT;
  databaseowner TEXT;

  r RECORD;

BEGIN

  IF tg_tag = 'CREATE EXTENSION' and current_user != 'rds_superuser' THEN
```

```
    RAISE NOTICE 'SECURITY INVOKER';
    RAISE NOTICE 'user: %', current_user;
    FOR r IN SELECT * FROM pg_event_trigger_ddl_commands()
    LOOP
        CONTINUE WHEN r.command_tag != 'CREATE EXTENSION' OR r.object_type !=
'extension';

        schemaname = (
            SELECT n.nspname
            FROM pg_catalog.pg_extension AS e
            INNER JOIN pg_catalog.pg_namespace AS n
            ON e.extnamespace = n.oid
            WHERE e.oid = r.objid
        );

        databaseowner = (
            SELECT pg_catalog.pg_get_userbyid(d.datdba)
            FROM pg_catalog.pg_database d
            WHERE d.datname = current_database()
        );
        RAISE NOTICE 'Record for event trigger %, objid: %,tag: %, current_user: %,
schema: %, database_owenr: %', r.object_identity, r.objid, tg_tag, current_user,
schemaname, databaseowner;
        IF r.object_identity = 'address_standardizer_data_us' THEN
            EXECUTE format('GRANT SELECT, UPDATE, INSERT, DELETE ON TABLE %I.us_gaz TO
%I WITH GRANT OPTION;', schemaname, databaseowner);
            EXECUTE format('GRANT SELECT, UPDATE, INSERT, DELETE ON TABLE %I.us_lex TO
%I WITH GRANT OPTION;', schemaname, databaseowner);
            EXECUTE format('GRANT SELECT, UPDATE, INSERT, DELETE ON TABLE %I.us_rules
TO %I WITH GRANT OPTION;', schemaname, databaseowner);
        ELSIF r.object_identity = 'dict_int' THEN
            EXECUTE format('ALTER TEXT SEARCH DICTIONARY %I.intdict OWNER TO %I;',
schemaname, databaseowner);
        ELSIF r.object_identity = 'pg_partman' THEN
            EXECUTE format('GRANT SELECT, UPDATE, INSERT, DELETE ON TABLE
%I.part_config TO %I WITH GRANT OPTION;', schemaname, databaseowner);
            EXECUTE format('GRANT SELECT, UPDATE, INSERT, DELETE ON TABLE
%I.part_config_sub TO %I WITH GRANT OPTION;', schemaname, databaseowner);
            EXECUTE format('GRANT SELECT, UPDATE, INSERT, DELETE ON TABLE
%I.custom_time_partitions TO %I WITH GRANT OPTION;', schemaname, databaseowner);
        ELSIF r.object_identity = 'postgis_topology' THEN
            EXECUTE format('GRANT SELECT, UPDATE, INSERT, DELETE ON ALL TABLES IN
SCHEMA topology TO %I WITH GRANT OPTION;', databaseowner);
```

```
            EXECUTE format('GRANT USAGE, SELECT ON ALL SEQUENCES IN SCHEMA topology TO
 %I WITH GRANT OPTION;', databaseowner);
            EXECUTE format('GRANT EXECUTE ON ALL FUNCTIONS IN SCHEMA topology TO %I
 WITH GRANT OPTION;', databaseowner);
            EXECUTE format('GRANT USAGE ON SCHEMA topology TO %I WITH GRANT OPTION;',
 databaseowner);
        END IF;
    END LOOP;
  END IF;
END;
$$ LANGUAGE plpgsql SECURITY DEFINER;

CREATE EVENT TRIGGER log_create_ext ON ddl_command_end EXECUTE PROCEDURE create_ext();
```

Benefícios de usar a compatibilidade com extensões delegadas do Amazon RDS

Usando a compatibilidade com extensões delegadas do PostgreSQL, é possível delegar, com segurança, o gerenciamento de extensões a usuários que não têm o perfil `rds_superuser`. Esse recurso oferece os seguintes benefícios:

- É possível delegar facilmente o gerenciamento de extensões aos usuários de sua escolha.
- Isso não exige o perfil `rds_superuser`.
- Oferece a capacidade de oferecer compatibilidade com um conjunto diferente de extensões para bancos de dados diferentes no mesmo cluster de banco de dados.

Limitação da compatibilidade com extensões delegadas do PostgreSQL

- Objetos criados durante o processo de criação da extensão podem exigir privilégios adicionais para que a extensão funcione corretamente.
- Algumas extensões não podem ser gerenciadas pelo usuário da extensão delegada por padrão, incluindo as seguintes: `log_fdw`, `pg_cron`, `pg_tle`, `pgactive`, `pglogical`, `postgis_raster`, `postgis_tiger_geocoder`, `postgis_topology`.

Permissões necessárias para determinadas extensões

Para criar, usar ou atualizar as extensões a seguir, o usuário delegado deve ter os privilégios necessários nestas funções, tabelas e esquemas.

Exter que preci de propr de ou perm s	Função	Tabelas	Schema	Dicionário de pesquisa de texto	Comentário
addre tanda er_da s	nenhuma	us_gaz, us_lex, us_lex, l.us_rules	nenhuma	nenhuma	nenhuma
amch	bt_index_check, bt_index_ parent_check	nenhuma	nenhuma	nenhuma	nenhuma
dict_i	nenhuma	nenhuma	nenhuma	intdict	nenhuma
pg_p n	nenhuma	custom_ti me_partitions, part_config, part_config_sub	nenhuma	nenhuma	nenhuma
pg_st tatem	nenhuma	nenhuma	nenhuma	nenhuma	nenhuma
Post(	st_tileenvelope	spatial_ref_sys	nenhuma	nenhuma	nenhuma
postg aster	nenhuma	nenhuma	nenhuma	nenhuma	nenhuma
postg opolo	nenhuma	topologia, camada	topologia	nenhuma	o usuário delegado deve ser o proprietá

Exter que preci de propr de ou perm s	Função	Tabelas	Schema	Dicionário de pesquisa de texto	Comentário
					rio do banco de dados
log_f	create_foreign_table_for_log_file	nenhuma	nenhuma	nenhuma	nenhuma
rds_t	role_password_encryption_type	nenhuma	nenhuma	nenhuma	nenhuma
postg iger_ oder	nenhuma	geocode_settings_default, geocode_settings	tiger	nenhuma	nenhuma
pg_fr acem	pg_freespace	nenhuma	nenhuma	nenhuma	nenhuma
pg_vi lity	pg_visibility	nenhuma	nenhuma	nenhuma	nenhuma

Considerações sobre segurança

Lembre-se de que um usuário com o perfil `rds_extension` poderá gerenciar extensões em todos os bancos de dados nos quais tiver o privilégio de conexão. Se a intenção for fazer com que um usuário delegado gerencie a extensão em um único banco de dados, uma prática recomendada é

revogar todos os privilégios públicos em cada banco de dados e, depois, conceder explicitamente o privilégio de conexão desse banco de dados específico ao usuário delegado.

Existem várias extensões que podem permitir que um usuário acesse informações de vários bancos de dados. Garanta que os usuários que receberem `rds_extension` tenham recursos de vários bancos de dados antes de adicionar essas extensões a `rds.allowed_delegated_extensions`. Por exemplo, `postgres_fdw` e `dblink` oferecem a capacidade de consultar vários bancos de dados na mesma instância ou em instâncias remotas. `log_fdw` lê os arquivos de log do mecanismo postgres, relacionados a todos os bancos de dados na instância, possivelmente contendo consultas lentas ou mensagens de erro de vários bancos de dados. `pg_cron` permite a execução de trabalhos agendados em segundo plano na instância de banco de dados e pode configurar trabalhos para execução em um banco de dados diferente.

Descartar cascata de extensão desabilitado

A capacidade de descartar a extensão com a opção em cascata por um usuário com o perfil `rds_extension` é controlada pelo parâmetro `rds.delegated_extension_allow_drop_cascade`. Por padrão, `rds-delegated_extension_allow_drop_cascade` é definido como `off`. Isso significa que os usuários com o perfil `rds_extension` não têm permissão para descartar uma extensão usando a opção em cascata, conforme mostrado na consulta a seguir.

```
DROP EXTENSION CASCADE;
```

Pois isso descartará automaticamente os objetos que dependem da extensão e, por sua vez, todos os objetos que dependem desses objetos. A tentativa de usar a opção em cascata gerará um erro.

Para conceder essa capacidade, o parâmetro `rds.delegated_extension_allow_drop_cascade` deve ser definido como on.

Alterar o parâmetro dinâmico `rds.delegated_extension_allow_drop_cascade` não requer a reinicialização do banco de dados. É possível fazer isso em um dos seguintes níveis:

- No cluster ou no grupo de parâmetros da instância, por meio do AWS Management Console ou da API.
- Usar o seguinte comando em nível de banco de dados:

```
alter database database_name set rds.delegated_extension_allow_drop_cascade = 'on';
```

- Usar o seguinte comando em nível de usuário:

```
alter role tenant_user set rds.delegated_extension_allow_drop_cascade = 'on';
```

Exemplos de extensões que podem ser adicionadas usando o suporte de extensão delegado

- `rds_tools`

```
extension_test_db=> create extension rds_tools;
CREATE EXTENSION
extension_test_db=> SELECT * from rds_tools.role_password_encryption_type() where
 rolname = 'pg_read_server_files';
ERROR: permission denied for function role_password_encryption_type
```

- `amcheck`

```
extension_test_db=> CREATE TABLE amcheck_test (id int);
CREATE TABLE
extension_test_db=> INSERT INTO amcheck_test VALUES (generate_series(1,100000));
INSERT 0 100000
extension_test_db=> CREATE INDEX amcheck_test_btree_idx ON amcheck_test USING btree
 (id);
CREATE INDEX
extension_test_db=> create extension amcheck;
CREATE EXTENSION
extension_test_db=> SELECT bt_index_check('amcheck_test_btree_idx'::regclass);
ERROR: permission denied for function bt_index_check
extension_test_db=> SELECT bt_index_parent_check('amcheck_test_btree_idx'::regclass);
ERROR: permission denied for function bt_index_parent_check
```

- `pg_freespacemap`

```
extension_test_db=> create extension pg_freespacemap;
CREATE EXTENSION
extension_test_db=> SELECT * FROM pg_freespace('pg_authid');
ERROR: permission denied for function pg_freespace
extension_test_db=> SELECT * FROM pg_freespace('pg_authid',0);
ERROR: permission denied for function pg_freespace
```

- `pg_visibility`

```
extension_test_db=> create extension pg_visibility;
CREATE EXTENSION
extension_test_db=> select * from pg_visibility('pg_database'::regclass);
ERROR: permission denied for function pg_visibility
```

- postgres_fdw

```
extension_test_db=> create extension postgres_fdw;
CREATE EXTENSION
extension_test_db=> create server myserver foreign data wrapper postgres_fdw options
 (host 'foo', dbname 'foodb', port '5432');
ERROR: permission denied for foreign-data wrapper postgres_fdw
```

Gerenciar partições do PostgreSQL com a extensão pg_partman

O particionamento de tabelas PostgreSQL fornece um framework para processamento de alta performance de entrada e relatórios de dados. Use o particionamento para bancos de dados que exigem entrada muito rápida de grandes quantidades de dados. O particionamento também fornece consultas mais rápidas de tabelas grandes. O particionamento ajuda a manter os dados sem afetar a instância do banco de dados, pois requer menos recursos de E/S.

Ao usar o particionamento, você pode dividir dados em blocos de tamanho personalizado para processamento. Por exemplo, você pode particionar dados de séries temporais para intervalos como por hora, diário, semanal, mensal, trimestral, anual, personalizado ou qualquer combinação destes. Para um exemplo de dados de séries temporais, se você particionar a tabela por hora, cada partição conterá uma hora de dados. Se você particionar a tabela de séries temporais por dia, as partições manterão dados de um dia e assim por diante. A chave de partição controla o tamanho de uma partição.

Quando você usa um comando SQL `INSERT` ou `UPDATE` em uma tabela particionada, o mecanismo de banco de dados roteia os dados para a partição apropriada. As partições de tabela PostgreSQL que armazenam os dados são tabelas filhas da tabela principal.

Durante as leituras de consulta de banco de dados, o otimizador PostgreSQL examina a cláusula `WHERE` da consulta e, se possível, direciona a verificação do banco de dados apenas para as partições relevantes.

A partir da versão 10, o PostgreSQL usa o particionamento declarativo para implementar o particionamento de tabela. Isso também é conhecido como particionamento nativo do PostgreSQL. Antes do PostgreSQL versão 10, você usou gatilhos para implementar partições.

O particionamento de tabelas PostgreSQL fornece os seguintes recursos:

- Criação de novas partições a qualquer momento.
- Intervalos variáveis de partição.
- Partições destacáveis e reanexáveis usando instruções DDL (Data Definition Language, linguagem de definição de dados).

  Por exemplo, partições destacáveis são úteis para remover dados históricos da partição principal, mas manter dados históricos para análise.
- Novas partições herdam as propriedades da tabela do banco de dados pai, incluindo:
  - Índices

  - Chaves primárias, que devem incluir a coluna de chave de partição
  - Chaves externas
  - Restrições de verificação
  - Referências
- Criação de índices para a tabela completa ou cada partição específica.

Você não pode alterar o esquema de uma partição individual. No entanto, você pode fazer uma alteração na tabela pai (adicionando uma nova coluna, por exemplo) que se propaga para as partições.

Tópicos

- Visão geral da extensão pg_partman do PostgreSQL
- Ativar a extensão pg_partman
- Configurar partições usando a função create_parent
- Configurar a manutenção da partição usando a função run_maintenance_proc

Visão geral da extensão pg_partman do PostgreSQL

Você pode usar a extensão `pg_partman` do PostgreSQL para automatizar a criação e a manutenção de partições de tabelas. Para obter mais informações gerais, consulte PG Partition Manager na documentação `pg_partman`.

> **Note**
>
> A extensão `pg_partman` é compatível com o RDS para PostgreSQL versões 12.5 e posteriores.

Em vez de ter que criar manualmente cada partição, você ajusta o `pg_partman` com as seguintes configurações:

- Tabela a ser particionada
- Tipo de partição
- Chave de partição
- Granularidade de partição

- Opções de pré-criação e gerenciamento de partições

Depois de criar uma tabela particionada do PostgreSQL, registre-a com `pg_partman` chamando a função `create_parent`. Fazer isso cria as partições necessárias com base nos parâmetros que você passa para a função.

A extensão `pg_partman` também fornece a função `run_maintenance_proc`, que você pode chamar de maneira programada para gerenciar as partições automaticamente. Para garantir que as partições adequadas sejam criadas conforme necessário, agende essa função para ser executada periodicamente (por hora, por exemplo). Você também pode garantir que as partições sejam descartadas automaticamente.

Ativar a extensão pg_partman

Se você tiver vários bancos de dados dentro da mesma instância de banco de dados PostgreSQL para a qual deseja gerenciar partições, habilite a extensão `pg_partman` separadamente para cada banco de dados. Para habilitar a extensão `pg_partman` para um banco de dados específico, crie o esquema de manutenção de partição e crie a extensão `pg_partman` da maneira a seguir.

```
CREATE SCHEMA partman;
CREATE EXTENSION pg_partman WITH SCHEMA partman;
```

> **Note**
>
> Para criar a extensão `pg_partman`, certifique-se de que você tenha privilégios `rds_superuser`.

Se você receber um erro como o seguinte, conceda os privilégios de `rds_superuser` à conta ou use sua conta de superusuário.

```
ERROR: permission denied to create extension "pg_partman"
HINT: Must be superuser to create this extension.
```

Para conceder privilégios de `rds_superuser`, conecte-se à sua conta de superusuário e execute o seguinte comando.

```
GRANT rds_superuser TO user-or-role;
```

Para os exemplos que mostram usando a extensão pg_partman, usamos a seguinte tabela de banco de dados de amostra e partição. Esse banco de dados usa uma tabela particionada com base em um carimbo de data/hora. Um esquema `data_mart` contém uma tabela chamada `events` com uma coluna chamada `created_at`. As seguintes configurações estão incluídas na tabela `events`:

- Chaves primárias `event_id` e `created_at`, que devem ter a coluna usada para orientar a partição.
- Uma restrição de verificação `ck_valid_operation` para impor valores para uma coluna `operation` da tabela.
- Duas chaves estrangeiras, onde uma (`fk_orga_membership`) aponta para a tabela externa `organization` e a outra (`fk_parent_event_id`) é uma chave estrangeira autorreferenciada.
- Dois índices, onde um (`idx_org_id`) é para a chave estrangeira e o outro (`idx_event_type`) é para o tipo de evento.

As declarações DDL a seguir criam esses objetos, que serão incluídos automaticamente em cada partição.

```
CREATE SCHEMA data_mart;
CREATE TABLE data_mart.organization ( org_id BIGSERIAL,
        org_name TEXT,
        CONSTRAINT pk_organization PRIMARY KEY (org_id)
    );

CREATE TABLE data_mart.events(
        event_id        BIGSERIAL,
        operation       CHAR(1),
        value           FLOAT(24),
        parent_event_id BIGINT,
        event_type      VARCHAR(25),
        org_id          BIGSERIAL,
        created_at      timestamp,
        CONSTRAINT pk_data_mart_event PRIMARY KEY (event_id, created_at),
        CONSTRAINT ck_valid_operation CHECK (operation = 'C' OR operation = 'D'),
        CONSTRAINT fk_orga_membership
            FOREIGN KEY(org_id)
            REFERENCES data_mart.organization (org_id),
        CONSTRAINT fk_parent_event_id
            FOREIGN KEY(parent_event_id, created_at)
            REFERENCES data_mart.events (event_id,created_at)
    ) PARTITION BY RANGE (created_at);
```

```
CREATE INDEX idx_org_id     ON  data_mart.events(org_id);
CREATE INDEX idx_event_type ON  data_mart.events(event_type);
```

Configurar partições usando a função create_parent

Depois de habilitar a extensão pg_partman, use a função `create_parent` para configurar partições dentro do esquema de manutenção de partição. Aqui é usado o exemplo de tabela `events` criado em Ativar a extensão pg_partman. Chame a função `create_parent` da seguinte forma:

```
SELECT partman.create_parent(
 p_parent_table => 'data_mart.events',
 p_control      => 'created_at',
 p_type         => 'range',
 p_interval     => '1 day',
 p_premake      => 30);
```

Os parâmetros são os seguintes:

- `p_parent_table` – a tabela particionada pai. Essa tabela já deve existir e estar totalmente qualificada, incluindo o esquema.
- `p_control` – a coluna na qual o particionamento deve ser baseado. O tipo de dados deve ser um inteiro ou baseado em tempo.
- `p_type`: o tipo é `'range'` ou `'list'`.
- `p_interval` – o intervalo ou a faixa de inteiros para cada partição. Os valores de exemplo incluem `1 day`, `1 hour` e assim por diante.
- `p_premake` – o número de partições para criar antecipadamente a fim de dar suporte a novas inserções.

Para obter uma descrição completa da função `create_parent`, consulte Funções de criação na documentação do `pg_partman`..

Configurar a manutenção da partição usando a função run_maintenance_proc

Você pode executar operações de manutenção de partição para criar automaticamente novas partições, desanexar partições ou remover partições antigas. A manutenção da partição depende da função `run_maintenance_proc` da extesão `pg_partman` e da extensão `pg_cron`, que inicia um

programador interno. O agendador pg_cron executa automaticamente instruções SQL, funções e procedimentos definidos em seus bancos de dados.

A seguir, será usado o exemplo de tabela events criado em Ativar a extensão pg_partman para definir que as operações de manutenção de partição serão executadas automaticamente. Como pré-requisito, adicione pg_cron ao parâmetro shared_preload_libraries no grupo de parâmetros da instância de banco de dados.

```
CREATE EXTENSION pg_cron;

UPDATE partman.part_config
SET infinite_time_partitions = true,
    retention = '3 months',
    retention_keep_table=true
WHERE parent_table = 'data_mart.events';
SELECT cron.schedule('@hourly', $$CALL partman.run_maintenance_proc()$$);
```

A seguir, você pode encontrar uma explicação detalhada do exemplo anterior:

1. Modifique o grupo de parâmetros associado à sua instância de banco de dados e adicione pg_cron ao valor do parâmetro shared_preload_libraries. Essa alteração exige a reinicialização da instância de banco de dados para que tenha efeito. Para obter mais informações, consulte Modificar parâmetros em um grupo de parâmetros de banco de dados no Amazon RDS.
2. Execute o comando CREATE EXTENSION pg_cron; usando uma conta que tenha as permissões de rds_superuser. Isso habilita a extensão pg_cron. Para obter mais informações, consulte Agendar manutenção com a extensão pg_cron do PostgreSQL.
3. Execute o comando UPDATE partman.part_config para ajustar as configurações pg_partman para a tabela data_mart.events.
4. Execute o comando SET . . . para configurar a tabela data_mart.events, com estas cláusulas:
   a. infinite_time_partitions = true, – configura a tabela para poder criar novas partições automaticamente, sem qualquer limite.
   b. retention = '3 months', – configura a tabela para ter uma retenção máxima de três meses.
   c. retention_keep_table=true – Configura a tabela para que, quando o período de retenção for devido, a tabela não seja excluída automaticamente. Em vez disso, as partições que são mais antigas do que o período de retenção são apenas separadas da tabela pai.

5. Execute o comando `SELECT cron.schedule . . .` para fazer uma chamada da função `pg_cron`. Esta chamada define com que frequência o programador executa o procedimento de manutenção `pg_partman`, `partman.run_maintenance_proc`. Para este exemplo, o procedimento é executado a cada hora.

Para obter uma descrição completa da função `run_maintenance_proc`, consulte Funções de manutenção na documentação do `pg_partman`.

Usar pgAudit para registrar a atividade do banco de dados

Instituições financeiras, agências governamentais e muitos setores precisam manter registros de auditoria para atender aos requisitos regulatórios. Ao usar a extensão do PostgreSQL Audit (pgAudit) com sua instância de banco de dados do RDS para PostgreSQL, você pode capturar os registros detalhados que normalmente são necessários aos auditores ou para atender aos requisitos regulatórios. Por exemplo, você pode configurar a extensão pgAudit para monitorar alterações feitas em tabelas e bancos de dados específicos, registrar o usuário que fez a alteração e muitos outros detalhes.

A extensão pgAudit se baseia na funcionalidade da infraestrutura de registro em log nativa do PostgreSQL, estendendo as mensagens de log com mais detalhes. Em outras palavras, é usada a mesma abordagem para visualizar o log de auditoria e quaisquer mensagens de log. Para obter mais informações sobre o registro em log do PostgreSQL, consulte Arquivos de log do banco de dados do RDS para PostgreSQL.

A extensão pgAudit retira dados confidenciais, como senhas de texto não criptografado, dos logs. Se sua instância de banco de dados do RDS para PostgreSQL estiver configurada para registrar declarações de linguagem de manipulação de dados (DML) conforme detalhado em Ativar o registro em log de consultas para sua instância de banco de dados do RDS para PostgreSQL, você poderá evitar o problema de senha de texto não criptografado usando a extensão do PostgreSQL Audit.

Você pode configurar a auditoria em suas instâncias de banco de dados com um alto grau de especificidade. É possível auditar todos os bancos de dados e todos os usuários. Ou você pode optar por auditar somente determinados bancos de dados, usuários e outros objetos. Também é possível excluir explicitamente da auditoria determinados usuários e bancos de dados. Para obter mais informações, consulte Excluir usuários ou bancos de dados do registro em log de auditoria.

Dada a quantidade de detalhes que podem ser capturados, recomendamos que, se você usar pgAudit, monitore seu consumo de armazenamento.

A extensão pgAudit é compatível com todas as Versões do RDS para PostgreSQL. Para obter uma lista de versões de pgAudit compatíveis com a versão do RDS para PostgreSQL, consulte Extension versions for Amazon Aurora PostgreSQL (Versões de extensão para o Amazon Aurora PostgreSQL) em Amazon RDS for PostgreSQL Release Notes (Notas de versão do Amazon RDS para PostgreSQL).

Tópicos

- Configurar a extensão pgAudit

- Auditar objetos de banco de dados
- Excluir usuários ou bancos de dados do registro em log de auditoria
- Referência para a extensão pgAudit

Configurar a extensão pgAudit

Para configurar a extensão pgAudit em sua instância de banco de dados do RDS para PostgreSQL, , primeiro adicione pgAudit às bibliotecas compartilhadas no grupo de parâmetros de banco de dados personalizado para sua instância de banco de dados do RDS para PostgreSQL. Para obter informações sobre como criar um grupo de parâmetros de banco de dados, consulte Grupos de parâmetros para Amazon RDS. Depois, instale a extensão pgAudit. Por fim, especifique os bancos de dados e os objetos que deseja auditar. Os procedimentos nesta seção mostram o procedimento. É possível usar o AWS Management Console ou a AWS CLI.

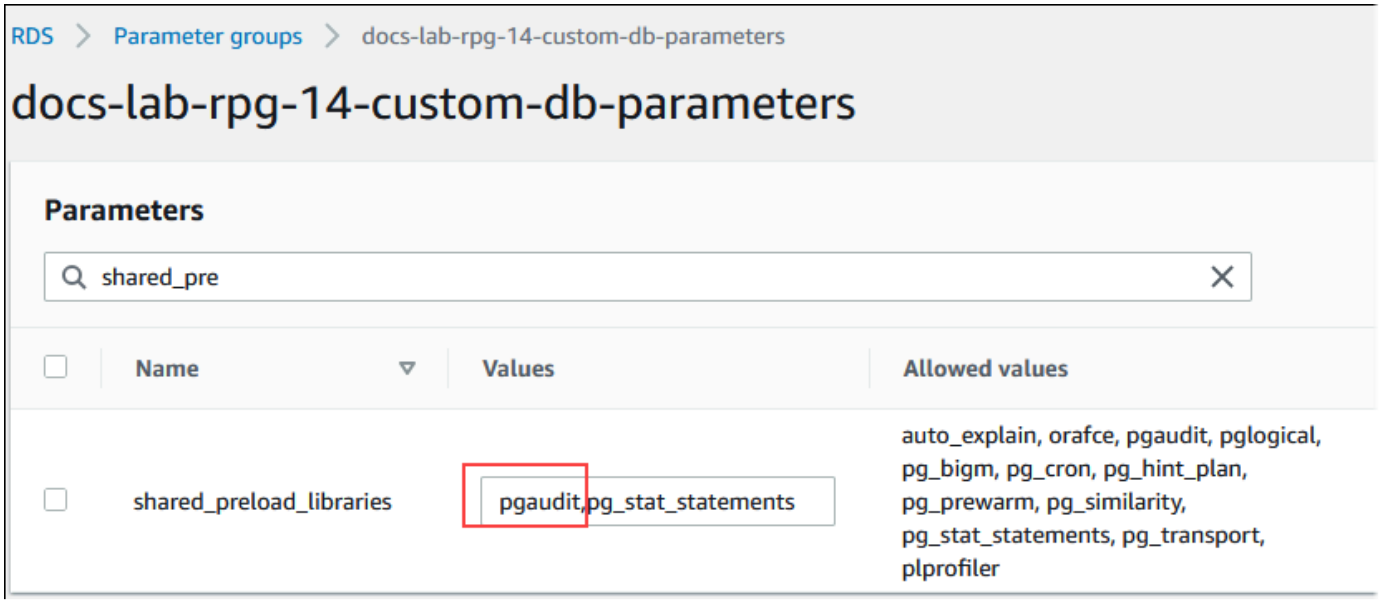
Você deve ter permissões como a função `rds_superuser` para realizar todas essas tarefas.

As etapas a seguir pressupõem que sua instância de banco de dados do RDS para PostgreSQL esteja associada a um grupo de parâmetros de banco de dados.

Console

Como configurar a extensão pgAudit

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, selecione sua instância de banco de dados do RDS para PostgreSQL.
3. Abra a guia Configuration (Configuração) para sua Instância de banco de dados do RDS para PostgreSQL. Entre os detalhes da instância, encontre o link Parameter group (Grupo de parâmetros).
4. Clique no link para abrir os parâmetros personalizados associados ao seu Instância de banco de dados do RDS para PostgreSQL.
5. No campo Parameters (Parâmetros), digite `shared_pre` para encontrar o parâmetro `shared_preload_libraries`.
6. Selecione Edit parameters (Editar parâmetros) para acessar os valores das propriedades.
7. Adicione `pgaudit` à lista no campo Values (Valores). Use uma vírgula para separar itens na lista de valores.



8. Reinicie a instância de banco de dados do RDS para PostgreSQL para que a alteração no parâmetro `shared_preload_libraries` tenha efeito.
9. Quando a instância estiver disponível, verifique se a pgAudit foi inicializada. Use `psql` para se conectar à instância de banco de dados do RDS para PostgreSQL e depois execute o comando a seguir.

```
SHOW shared_preload_libraries;
shared_preload_libraries
--------------------------
rdsutils,pgaudit
(1 row)
```

10. Com a pgAudit inicializada, agora você pode criar a extensão. Você precisa criar a extensão depois de inicializar a biblioteca porque a extensão `pgaudit` instala acionadores de eventos para auditar declarações de linguagem de definição de dados (DDL).

```
CREATE EXTENSION pgaudit;
```

11. Feche a sessão `psql`.

```
labdb=> \q
```

12. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.

13. Encontre o parâmetro `pgaudit.log` na lista e defina como o valor apropriado para o caso de uso. Por exemplo, definir o parâmetro `pgaudit.log` como `write` conforme mostrado na imagem a seguir captura inserções, atualizações, exclusões e alguns outros tipos de alterações no log.



Você também pode selecionar um dos valores a seguir para o parâmetro `pgaudit.log`.

- none: esse é o valor padrão. Nenhuma alteração no banco de dados é registrada.
- all: registra tudo (read, write, function, role, ddl, misc).
- ddl: registra todas as instruções de linguagem de definição de dados (DDL) não incluídas na classe `ROLE`.
- function: registra chamadas de função e blocos de `DO`.
- misc: registra comandos diversos, como `DISCARD`, `FETCH`, `CHECKPOINT`, `VACUUM` e `SET`.
- read: registra `SELECT` e `COPY` quando a fonte é uma relação (como uma tabela) ou uma consulta.
- role: registra declarações relacionadas a funções e privilégios, como `GRANT`, `REVOKE`, `CREATE ROLE`, `ALTER ROLE` e `DROP ROLE`.
- write: registra `INSERT`, `UPDATE`, `DELETE`, `TRUNCATE` e `COPY` quando o destino é uma relação (tabela).

14. Escolha Salvar alterações.
15. Abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
16. Selecione a instância de banco de dados do RDS para PostgreSQL na lista Bancos de dados.

AWS CLI

Como configurar a pgAudit

Para configurar a pgAudit usando a AWS CLI, chame a operação modify-db-parameter-group para modificar os parâmetros do log de auditoria em seu grupo de parâmetros personalizado, conforme mostrado no procedimento a seguir.

1. Use o comando AWS CLI a seguir para adicionar `pgaudit` ao parâmetro `shared_preload_libraries`.

```
aws rds modify-db-parameter-group \
   --db-parameter-group-name custom-param-group-name \
   --parameters
 "ParameterName=shared_preload_libraries,ParameterValue=pgaudit,ApplyMethod=pending-
reboot" \
   --region aws-region
```

2. Use o comando AWS CLI a seguir para reinicializar a instância de banco de dados do RDS para PostgreSQL para que a biblioteca da pgaudit seja inicializada.

```
aws rds reboot-db-instance \
    --db-instance-identifier your-instance \
    --region aws-region
```

3. Quando a instância estiver disponível, verifique se a `pgaudit` foi inicializada. Use `psql` para se conectar à instância de banco de dados do RDS para PostgreSQL e, depois, execute o comando a seguir.

```
SHOW shared_preload_libraries;
shared_preload_libraries
--------------------------
rdsutils,pgaudit
(1 row)
```

   Com a pgAudit inicializada, agora você pode criar a extensão.

```
CREATE EXTENSION pgaudit;
```

4. Feche a sessão `psql` para que você possa usar a AWS CLI.

```
labdb=> \q
```

5. Use o comando AWS CLI a seguir para especificar as classes de declaração que devem ser registradas pelo registro em log de auditoria da sessão. O exemplo define o parâmetro `pgaudit.log` como `write`, que captura inserções, atualizações e exclusões no log.

```
aws rds modify-db-parameter-group \
   --db-parameter-group-name custom-param-group-name \
   --parameters
 "ParameterName=pgaudit.log,ParameterValue=write,ApplyMethod=pending-reboot" \
   --region aws-region
```

Você também pode selecionar um dos valores a seguir para o parâmetro `pgaudit.log`.

- none: esse é o valor padrão. Nenhuma alteração no banco de dados é registrada.
- all: registra tudo (read, write, function, role, ddl, misc).
- ddl: registra todas as instruções de linguagem de definição de dados (DDL) não incluídas na classe `ROLE`.
- function: registra chamadas de função e blocos de `DO`.
- misc: registra comandos diversos, como `DISCARD`, `FETCH`, `CHECKPOINT`, `VACUUM` e `SET`.
- read: registra `SELECT` e `COPY` quando a fonte é uma relação (como uma tabela) ou uma consulta.
- role: registra declarações relacionadas a funções e privilégios, como `GRANT`, `REVOKE`, `CREATE ROLE`, `ALTER ROLE` e `DROP ROLE`.
- write: registra `INSERT`, `UPDATE`, `DELETE`, `TRUNCATE` e `COPY` quando o destino é uma relação (tabela).

Reinicie a instância de banco de dados do RDS para PostgreSQL usando o comando AWS CLI a seguir.

```
aws rds reboot-db-instance \
    --db-instance-identifier your-instance \
    --region aws-region
```

Auditar objetos de banco de dados

Com a pgAudit configurada em sua instância e banco de dados do RDS para PostgreSQL e configurada para seus requisitos, informações mais detalhadas são capturadas no log do PostgreSQL. Por exemplo, enquanto a configuração de registro em padrão do PostgreSQL identifica a data e a hora em que uma alteração foi feita em uma tabela do banco de dados, com a extensão pgAudit, a entrada do log pode incluir o esquema, o usuário que fez a alteração e outros detalhes, dependendo de como os parâmetros da extensão estão configurados. Você pode configurar a auditoria para monitorar as alterações das maneiras a seguir.

- Para cada sessão, por usuário. Para o nível da sessão, você pode capturar o texto do comando totalmente qualificado.
- Para cada objeto, por usuário e por banco de dados.

O recurso de auditoria de objetos é ativado quando você cria a função `rds_pgaudit` no sistema e depois a adiciona ao parâmetro `pgaudit.role` no grupo de parâmetros personalizado. Por padrão, o parâmetro `pgaudit.role` não está definido e o único valor permitido é `rds_pgaudit`. As etapas a seguir pressupõem que a `pgaudit` tenha sido inicializada e que você tenha criado a extensão `pgaudit` seguindo o procedimento em Configurar a extensão pgAudit.

```
2022-10-07 23:36:51 UTC:52.95.4.10(14410):postgres@labdb:[1374]:LOG: statement: SELECT feedback, s.sentiment,s.confidence
FROM support,aws_comprehend.detect_sentiment(feedback, 'en') s
ORDER BY s.confidence DESC;
2022-10-07 23:36:51 UTC:52.95.4.10(14410):postgres@labdb:[1374]:LOG: AUDIT: SESSION,2,1,READ,SELECT,TABLE,public.support,"SELECT
feedback, s.sentiment,s.confidence
FROM support,aws_comprehend.detect_sentiment(feedback, 'en') s
ORDER BY s.confidence DESC;",<none>
2022-10-07 23:36:51 UTC:52.95.4.10(14410):postgres@labdb:[1374]:LOG: QUERY STATISTICS
2022-10-07 23:36:51 UTC:52.95.4.10(14410):postgres@labdb:[1374]:DETAIL: ! system usage stats:
! 0.009494 s user, 0.007442 s system, 0.141985 s elapsed
! [0.022327 s user, 0.007442 s system total]
```

Conforme mostrado neste exemplo, a linha “LOG: AUDIT: SESSION” fornece informações sobre a tabela e o respectivo esquema, entre outros detalhes.

Como configurar a auditoria de objetos

1. Use `psql` para se conectar à instância de banco de dados do RDS para PostgreSQL.

   ```
   psql --host=your-instance-name.aws-region.rds.amazonaws.com --port=5432 --
   username=postgrespostgres --password --dbname=labdb
   ```

2. Crie uma função de banco de dados chamada `rds_pgaudit` usando o comando a seguir.

```
labdb=> CREATE ROLE rds_pgaudit;
CREATE ROLE
labdb=>
```

3. Feche a sessão `psql`.

```
labdb=> \q
```

Nas próximas etapas, use a AWS CLI para modificar os parâmetros de log de auditoria no grupo de parâmetros personalizado.

4. Use o comando AWS CLI a seguir para definir o parâmetro `pgaudit.role` como `rds_pgaudit`. Por padrão, esse parâmetro está vazio, e `rds_pgaudit` é o único valor permitido.

```
aws rds modify-db-parameter-group \
   --db-parameter-group-name custom-param-group-name \
   --parameters
 "ParameterName=pgaudit.role,ParameterValue=rds_pgaudit,ApplyMethod=pending-reboot"
 \
   --region aws-region
```

5. Use o comando AWS CLI a seguir para reinicializar a instância de banco de dados do RDS para PostgreSQL para que as alterações nos parâmetros tenham efeito.

```
aws rds reboot-db-instance \
    --db-instance-identifier your-instance \
    --region aws-region
```

6. Execute o comando a seguir para confirmar que `pgaudit.role` está definido como `rds_pgaudit`.

```
SHOW pgaudit.role;
pgaudit.role
------------------
rds_pgaudit
```

Para testar o registro em log da extensão pgAudit, execute vários comandos de exemplo semelhantes ao que você deseja auditar. Por exemplo, você pode executar os seguintes comandos.

```
CREATE TABLE t1 (id int);
GRANT SELECT ON t1 TO rds_pgaudit;
SELECT * FROM t1;
id
----
(0 rows)
```

Os logs do banco de dados devem conter uma entrada semelhante à seguinte.

```
...
2017-06-12 19:09:49 UTC:...:rds_test@postgres:[11701]:LOG: AUDIT:
OBJECT,1,1,READ,SELECT,TABLE,public.t1,select * from t1;
...
```

Para obter informações sobre como visualizar os logs, consulte Monitorar arquivos de log do Amazon RDS.

Para saber mais sobre a extensão pgAudit, consulte pgAudit no GitHub.

Excluir usuários ou bancos de dados do registro em log de auditoria

Conforme discutido em Arquivos de log do banco de dados do RDS para PostgreSQL, os logs do PostgreSQL consomem espaço de armazenamento. O uso da extensão pgAudit aumenta o volume de dados reunidos nos logs em vários graus, dependendo das alterações monitoradas. Talvez você não precise auditar todos os usuários nem bancos de dados no Instância de banco de dados do RDS para PostgreSQL.

Para minimizar os impactos no armazenamento e evitar a captura desnecessária de registros de auditoria, você pode excluir usuários e bancos de dados da auditoria. Você também pode alterar o registro em log em determinada sessão. Os exemplos a seguir mostram o procedimento.

> **Note**
>
> As configurações de parâmetros no nível da sessão têm precedência sobre as configurações no grupo de parâmetros de banco de dados personalizado para a instância de banco de dados do RDS para PostgreSQL. Se você não quiser que os usuários do banco de dados ignorem suas configurações de registro em log de auditoria, não se esqueça de alterar as permissões.

Suponha que sua instância de banco de dados do RDS para PostgreSQL esteja configurada) para auditar o mesmo nível de atividade para todos os usuários e bancos de dados. Depois, decida que não quer auditar o usuário `myuser`. Você pode desativar a auditoria para `myuser` com o comando SQL a seguir.

```
ALTER USER myuser SET pgaudit.log TO 'NONE';
```

Depois, você pode usar a consulta a seguir para conferir a coluna `user_specific_settings` para `pgaudit.log` a fim de confirmar se o parâmetro está definido como `NONE`.

```
SELECT
    usename AS user_name,
    useconfig AS user_specific_settings
FROM
    pg_user
WHERE
    usename = 'myuser';
```

Você deve ver a saída da forma a seguir.

```
 user_name | user_specific_settings
-----------+------------------------
 myuser    | {pgaudit.log=NONE}
(1 row)
```

Você pode desativar o registro em log de determinado usuário no meio da sessão com o banco de dados com o comando a seguir.

```
ALTER USER myuser IN DATABASE mydatabase SET pgaudit.log TO 'none';
```

Use a consulta a seguir para conferir a coluna de configurações de pgaudit.log para uma combinação específica de usuário e banco de dados.

```
SELECT
    usename AS "user_name",
    datname AS "database_name",
    pg_catalog.array_to_string(setconfig, E'\n') AS "settings"
FROM
    pg_catalog.pg_db_role_setting s
```

```
    LEFT JOIN pg_catalog.pg_database d ON d.oid = setdatabase
    LEFT JOIN pg_catalog.pg_user r ON r.usesysid = setrole
WHERE
    usename = 'myuser'
    AND datname = 'mydatabase'
ORDER BY
    1,
    2;
```

Você verá uma saída semelhante à seguinte.

```
  user_name | database_name |     settings
-----------+---------------+-----------------
 myuser    | mydatabase    | pgaudit.log=none
(1 row)
```

Depois de desativar a auditoria de `myuser`, você decide que não deseja monitorar as alterações em `mydatabase`. Você pode desativar a auditoria para esse banco de dados específico usando o comando a seguir.

```
ALTER DATABASE mydatabase SET pgaudit.log to 'NONE';
```

Depois, use a consulta a seguir para conferir a coluna database_specific_settings a fim de confirmar se pgaudit.log está definido como NONE.

```
SELECT
a.datname AS database_name,
b.setconfig AS database_specific_settings
FROM
pg_database a
FULL JOIN pg_db_role_setting b ON a.oid = b.setdatabase
WHERE
a.datname = 'mydatabase';
```

Você deve ver a saída da forma a seguir.

```
 database_name | database_specific_settings
---------------+----------------------------
 mydatabase    | {pgaudit.log=NONE}
(1 row)
```

Para restaurar as configurações padrão para myuser, use o seguinte comando:

```
ALTER USER myuser RESET pgaudit.log;
```

Para restaurar as configurações padrão para um banco de dados, use o comando a seguir.

```
ALTER DATABASE mydatabase RESET pgaudit.log;
```

Para restaurar as configurações padrão de usuário e banco de dados, use o comando a seguir.

```
ALTER USER myuser IN DATABASE mydatabase RESET pgaudit.log;
```

Você também pode capturar eventos específicos no log definindo `pgaudit.log` como um dos outros valores permitidos para o parâmetro `pgaudit.log`. Para ter mais informações, consulte Lista de configurações permitidas para o parâmetro `pgaudit.log`.

```
ALTER USER myuser SET pgaudit.log TO 'read';
ALTER DATABASE mydatabase SET pgaudit.log TO 'function';
ALTER USER myuser IN DATABASE mydatabase SET pgaudit.log TO 'read,function'
```

Referência para a extensão pgAudit

Você pode especificar o nível de detalhes que deseja para o log de auditoria alterando um ou mais dos parâmetros listados nesta seção.

Controlar o comportamento da pgAudit

Você pode controlar o registro em log de auditoria alterando um ou mais dos parâmetros listados na tabela a seguir.

Parâmetro	Descrição
`pgaudit.log`	Especifica as classes de declaração que serão registradas pelo registro em log de auditoria de sessão. Os valores permitido s incluem ddl, function, misc, read, role, write, none, all. Para obter mais informações, consulte Lista de configurações permitidas para o parâmetro `pgaudit.log` .

Parâmetro	Descrição
`pgaudit.log_catalog`	Quando ativado (definido como 1), adiciona declarações à trilha de auditoria se todas as relações em uma declaração estiverem em pg_catalog.
`pgaudit.log_level`	Especifica o nível de log que será usado para entradas de log. Valores permitidos: debug5, debug4, debug3, debug2, debug1, info, notice, warning, log
`pgaudit.log_parameter`	Quando ativado (definido como 1), os parâmetros passados com a declaração são capturados no log de auditoria.
`pgaudit.log_relation`	Quando ativado (definido como 1), o log de auditoria da sessão cria uma entrada de log separada para cada relação (TABLE, VIEW etc.) referenciada em uma declaração SELECT ou DML.
`pgaudit.log_statement_once`	Especifica se o registro incluirá o texto e os parâmetros da instrução com a primeira entrada de log para uma combinação de instrução/subinstrução ou com cada entrada.
`pgaudit.role`	Especifica a função primária a ser usada para o registro em log de auditoria de objetos. A única entrada permitida é `rds_pgaudit` .

Lista de configurações permitidas para o parâmetro **`pgaudit.log`**

Value	Descrição
nenhuma	Esse é o padrão. Nenhuma alteração no banco de dados é registrada.
tudo	Registra tudo (read, write, function, role, ddl, misc).
ddl	Registra todas as declarações de linguagem de definição de dados (DDL) não incluídas na classe `ROLE`.
função	Registra chamadas de função e blocos de `DO`.

Value	Descrição
misc	Registra comandos diversos, como `DISCARD`, `FETCH`, `CHECKPOINT` , `VACUUM` e `SET`.
leitura	Registra `SELECT` e `COPY` quando a fonte é uma relação (como uma tabela) ou uma consulta.
role (perfil)	Registra declarações relacionadas a funções e privilégios, como `GRANT`, `REVOKE`, `CREATE ROLE`, `ALTER ROLE` e `DROP ROLE`.
write	Registra `INSERT`, `UPDATE`, `DELETE`, `TRUNCATE` e `COPY` quando o destino é uma relação (tabela).

Para registrar vários tipos de eventos com auditoria de sessões, use uma lista separada por vírgulas. Para registrar todos os tipos de eventos, defina `pgaudit.log` para `ALL`. Reinicie a instância de banco de dados para aplicar as alterações.

Com a auditoria de objetos, você pode refinar o registro em log de auditoria para trabalhar com relações específicas. Por exemplo, você pode especificar que deseja o registro em log de auditoria para operações `READ` em uma ou mais tabelas.

Agendar manutenção com a extensão pg_cron do PostgreSQL

Você pode utilizar a extensão `pg_cron` do PostgreSQL para programar comandos de manutenção dentro de um banco de dados do PostgreSQL. Para obter mais informações sobre a extensão, consulte O que é pg_cron? na documentação do pg_cron.

A extensão `pg_cron` é compatível com o mecanismo do RDS para PostgreSQL versões 12.5 e posteriores.

Para saber mais sobre como usar `pg_cron`, consulte Programar trabalhos com pg_cron em bancos de dados do RDS para PostgreSQL ou compatíveis com o Aurora PostgreSQL.

Tópicos

- Configurar a extensão pg_cron
- Conceder permissões de banco de dados para usar pg_cron
- Agendar trabalhos de pg_cron
- Referência para a extensão pg_cron

Configurar a extensão pg_cron

Configure a extensão `pg_cron` da seguinte forma:

1. Modifique o grupo de parâmetros personalizado associado à sua instância de banco de dados do PostgreSQL adicionando `pg_cron` ao valor do parâmetro `shared_preload_libraries`.
   - Se a instância de banco de dados do RDS para PostgreSQL usa o parâmetro `rds.allowed_extensions` para listar explicitamente as extensões que podem ser instaladas, você precisa adicionar a extensão `pg_cron` à lista. Somente determinadas versões do RDS para PostgreSQL oferecem suporte ao parâmetro `rds.allowed_extensions`. Por padrão, todas as extensões disponíveis são permitidas. Para ter mais informações, consulte Restringir a instalação de extensões do PostgreSQL.

   Reinicie a instância de banco de dados do PostgreSQL para que as alterações no grupo de parâmetros entrem em vigor. Para saber mais sobre como trabalhar com grupos de parâmetros, consulte Modificar parâmetros em um grupo de parâmetros de banco de dados no Amazon RDS.
2. Após a reinicialização da instância de banco de dados do PostgreSQL, execute o comando a seguir usando uma conta que tenha permissões `rds_superuser`. Por exemplo, se você usou as

configurações padrão ao criar a instância de banco de dados do RDS para PostgreSQL, conecte-se como o usuário `postgres` e crie a extensão.

```
CREATE EXTENSION pg_cron;
```

O agendador do `pg_cron` é definido no banco de dados PostgreSQL padrão chamado `postgres`. Os objetos `pg_cron` são criados neste banco de dados `postgres` e todas as ações de agendamento são executadas neste banco de dados.

3. Você pode usar as configurações padrão ou programar trabalhos para serem executados em outros bancos de dados dentro de sua instância de banco de dados PostgreSQL. Para programar trabalhos a serem executados em outros bancos de dados em sua instância de banco de dados PostgreSQL, consulte o exemplo em Agendar um trabalho cron para um banco de dados diferente do banco de dados padrão .

Conceder permissões de banco de dados para usar pg_cron

A instalação da extensão `pg_cron` requer privilégios de `rds_superuser`. No entanto, as permissões para usar `pg_cron` podem ser concedidas (por um membro do grupo/perfil de `rds_superuser`) para outros usuários do banco de dados, para que eles possam programar seus próprios trabalhos. Recomendamos que você conceda permissões para o esquema `cron` somente conforme necessário se ele melhorar as operações do ambiente de produção.

Para conceder permissão a um usuário do banco de dados no esquema `cron`, execute o seguinte comando:

```
postgres=> GRANT USAGE ON SCHEMA cron TO db-user;
```

Isso concede a *`db-user`* permissão para acessar o esquema `cron` para programar trabalhos cron para os objetos que o usuário tem permissão para acessar. Se o usuário do banco de dados não tiver permissões, o trabalho falhará após a publicação da mensagem de erro no arquivo `postgresql.log`, conforme mostrado a seguir:

```
2020-12-08 16:41:00 UTC::@:[30647]:ERROR: permission denied for table table-name
2020-12-08 16:41:00 UTC::@:[27071]:LOG: background worker "pg_cron" (PID 30647) exited
 with exit code 1
```

Em outras palavras, certifique-se de que os usuários do banco de dados que tenham permissões no esquema `cron` também tenham permissões nos objetos (tabelas, esquemas e assim por diante) que planejam programar.

Os detalhes do trabalho cron e seu sucesso ou falha também são capturados na tabela `cron.job_run_details`. Para ter mais informações, consulte Tabelas para agendar trabalhos e capturar status .

Agendar trabalhos de pg_cron

As seções a seguir mostram como você pode agendar várias tarefas de gerenciamento usando trabalhos `pg_cron`.

> **Note**
>
> Ao criar trabalhos `pg_cron`, verifique se a configuração `max_worker_processes` a configuração é maior do que o número de `cron.max_running_jobs`. Um trabalho `pg_cron` falhará se ficar sem processos de operador em segundo plano. O número padrão de trabalhos `pg_cron` é `5`. Para obter mais informações, consulte Parâmetros para gerenciar a extensão pg_cron.

Tópicos

- Vacuum de tabelas
- Limpar a tabela de histórico de pg_cron
- Registrar em log erros somente no arquivo postgresql.log
- Agendar um trabalho cron para um banco de dados diferente do banco de dados padrão

Vacuum de tabelas

O autovacuum lida com manutenção de vacuum para a maioria dos casos. No entanto, você pode agendar o vacuum de uma tabela específica quando quiser.

Consulte também, Trabalhar com o autovacuum do PostgreSQL no Amazon RDS para PostgreSQL.

Veja a seguir um exemplo de uso da função `cron.schedule` para configurar um trabalho a ser usado `VACUUM FREEZE` em uma tabela específica todos os dias às 22:00 (GMT).

```
SELECT cron.schedule('manual vacuum', '0 22 * * *', 'VACUUM FREEZE pgbench_accounts');
```

```
 schedule
----------
1
(1 row)
```

Após o exemplo anterior ser executado, você pode verificar o histórico na tabela `cron.job_run_details` da seguinte forma.

```
postgres=> SELECT * FROM cron.job_run_details;
jobid  | runid | job_pid | database | username | command                        |
 status    | return_message | start_time                    | end_time
-------+-------+---------+----------+----------+--------------------------------
+-----------+----------------+--------------------------------
+-------------------------------
 1     | 1     | 3395    | postgres | adminuser| vacuum freeze pgbench_accounts
 | succeeded | VACUUM         | 2020-12-04 21:10:00.050386+00 | 2020-12-04
 21:10:00.072028+00
(1 row)
```

A seguir está uma consulta à tabela `cron.job_run_details` para ver os trabalhos que falharam.

```
postgres=> SELECT * FROM cron.job_run_details WHERE status = 'failed';
jobid | runid | job_pid | database | username | command                        | status
 | return_message                                | start_time                    |
 end_time
------+-------+---------+----------+----------+--------------------------------+--------
+-----------------------------------------------+--------------------------------
+-------------------------------
 5    | 4     | 30339   | postgres | adminuser| vacuum freeze pgbench_account | failed
 | ERROR: relation "pgbench_account" does not exist | 2020-12-04 21:48:00.015145+00 |
 2020-12-04 21:48:00.029567+00
(1 row)
```

Para ter mais informações, consulte Tabelas para agendar trabalhos e capturar status .

Limpar a tabela de histórico de pg_cron

A tabela `cron.job_run_details` contém um histórico de trabalhos cron que podem se tornar muito grandes ao longo do tempo. Recomendamos que você agende um trabalho que limpe essa tabela. Por exemplo, manter uma semana de registros pode ser suficiente para fins de solução de problemas.

O exemplo a seguir usa a função cron.schedule para agendar um trabalho que é executado todos os dias à meia-noite para limpar a tabela `cron.job_run_details`. O trabalho mantém apenas os últimos sete dias. Use sua `rds_superuser` para agendar o trabalho da seguinte forma.

```
SELECT cron.schedule('0 0 * * *', $$DELETE
    FROM cron.job_run_details
    WHERE end_time < now() - interval '7 days'$$);
```

Para obter mais informações, consulte Tabelas para agendar trabalhos e capturar status .

Registrar em log erros somente no arquivo postgresql.log

Para impedir a gravação na tabela `cron.job_run_details`, modifique o grupo de parâmetros associado à instância de banco de dados do PostgreSQL e defina o parâmetro `cron.log_run` como desativado. A extensão `pg_cron` não gravará mais na tabela e vai capturar erros somente no arquivo `postgresql.log`. Para ter mais informações, consulte Modificar parâmetros em um grupo de parâmetros de banco de dados no Amazon RDS.

Use o comando a seguir para verificar o valor do parâmetro `cron.log_run`.

```
postgres=> SHOW cron.log_run;
```

Para ter mais informações, consulte Parâmetros para gerenciar a extensão pg_cron.

Agendar um trabalho cron para um banco de dados diferente do banco de dados padrão

Os metadados para `pg_cron` são todos mantidos no banco de dados padrão PostgreSQL chamado `postgres`. Como os operadores em segundo plano são usados para executar os trabalhos cron de manutenção, você pode agendar um trabalho em qualquer um dos seus bancos de dados dentro da instância de banco de dados do PostgreSQL.

> **Note**
>
> Somente usuários com o perfil `rds_superuser` ou privilégios `rds_superuser` podem listar todos os trabalhos cron no banco de dados. Outros usuários podem visualizar somente seus próprios trabalhos na tabela `cron.job`.

1. No banco de dados cron, agende o trabalho como você normalmente faria usando a cron.schedule.

```
postgres=> SELECT cron.schedule('database1 manual vacuum', '29 03 * * *', 'vacuum
 freeze test_table');
```

2. Como um usuário com a função `rds_superuser`, atualize a coluna do banco de dados para o trabalho que você acabou de criar para que ele seja executado em outro banco de dados dentro de sua instância de banco de dados do PostgreSQL.

```
postgres=> UPDATE cron.job SET database = 'database1' WHERE jobid = 106;
```

3. Verifique consultando a tabela `cron.job`.

```
postgres=> SELECT * FROM cron.job;
jobid | schedule    | command                         | nodename  | nodeport |
 database | username  | active | jobname
------+-------------+---------------------------------+-----------+----------
+----------+----------+--------+-----------------------
106   | 29 03 * * * | vacuum freeze test_table        | localhost | 8192     |
 database1| adminuser | t      | database1 manual vacuum
  1   | 59 23 * * * | vacuum freeze pgbench_accounts | localhost | 8192     |
 postgres | adminuser | t      | manual vacuum
(2 rows)
```

> **Note**
>
> Em algumas situações, você pode adicionar um cron job que você pretende executar em um banco de dados diferente. Nesses casos, o job pode tentar executar no banco de dados padrão (`postgres`) antes de atualizar a coluna correta do banco de dados. Se o nome de usuário tiver permissões, o trabalho será executado com êxito no banco de dados padrão.

Referência para a extensão pg_cron

Você pode usar os seguintes parâmetros, funções e tabelas com a extensão `pg_cron`. Para obter mais informações, consulte O que é pg_cron? na documentação do pg_cron.

Tópicos

- Parâmetros para gerenciar a extensão pg_cron
- Referência da função: cron.schedule

- Referência da função: cron.schedule
- Tabelas para agendar trabalhos e capturar status

Parâmetros para gerenciar a extensão pg_cron

Veja a seguir uma lista de parâmetros que controlam o comportamento da extensão `pg_cron`.

Parâmetro	Descrição
cron.database_name	O banco de dados em que os metadados de `pg_cron` são mantidos.
`cron.host`	O nome do host para se conectar ao PostgresSQL. Não é possível modificar esse valor.
`cron.log_run`	Registre todos os trabalhos executados na tabela `job_run_details` . Os valores são on ou `off`. Para obter mais informações, consulte Tabelas para agendar trabalhos e capturar status .
`cron.log_statement`	Registre todas as instruções cron antes de executá-las. Os valores são on ou `off`.
`cron.max_running_jobs`	O número máximo de trabalhos que podem ser executados simultaneamente.
`cron.use_background_workers`	Use trabalhadores em segundo plano em vez de sessões de cliente. Não é possível modificar esse valor.

Use o seguinte comando SQL para exibir esses parâmetros e seus valores.

```
postgres=> SELECT name, setting, short_desc FROM pg_settings WHERE name LIKE 'cron.%'
 ORDER BY name;
```

Referência da função: cron.schedule

Essa função agenda um trabalho cron. Inicialmente, o trabalho é agendado no banco de dados `postgres` padrão. A função retorna um valor `bigint` que representa o identificador de trabalho. Para agendar trabalhos a serem executados em outros bancos de dados em sua instância de banco de dados PostgreSQL, consulte o exemplo em Agendar um trabalho cron para um banco de dados diferente do banco de dados padrão .

A função tem dois formatos de sintaxe.

Sintaxe

```
cron.schedule (job_name,
    schedule,
    command
);

cron.schedule (schedule,
    command
);
```

Parâmetros

Parâmetro	Descrição
`job_name`	O nome do trabalho cron.
`schedule`	Texto indicando a programação do trabalho cron. O formato é o formato cron padrão.
`command`	Texto do comando a ser executado.

Exemplos

```
postgres=> SELECT cron.schedule ('test','0 10 * * *', 'VACUUM pgbench_history');
 schedule
----------
      145
(1 row)
```

```
postgres=> SELECT cron.schedule ('0 15 * * *', 'VACUUM pgbench_accounts');
 schedule
----------
      146
(1 row)
```

Referência da função: cron.schedule

Esta função exclui um trabalho cron. Você pode especificar `job_name` ou `job_id`. Uma política garante que você seja o proprietário para remover a programação do trabalho. A função retorna um booleano indicando êxito ou falha.

A função tem os seguintes formatos de sintaxe.

Sintaxe

```
cron.unschedule (job_id);

cron.unschedule (job_name);
```

Parâmetros

Parâmetro	Descrição
`job_id`	Um identificador de trabalho que foi retornado da função `cron.schedule` quando o trabalho cron foi programado.
`job_name`	O nome de um trabalho cron que foi agendado com a função `cron.schedule` .

Exemplos

```
postgres=> SELECT cron.unschedule(108);
 unschedule
------------
 t
(1 row)
```

```
postgres=> SELECT cron.unschedule('test');
 unschedule
------------
 t
(1 row)
```

Tabelas para agendar trabalhos e capturar status

As tabelas a seguir são usadas para agendar os trabalhos cron e registrar como os trabalhos foram concluídos.

Tabela	Descrição
`cron.job`	Contém os metadados sobre cada trabalho agendado. A maioria das interações com esta tabela deve ser feita por meio das funções `cron.schedule` e `cron.unschedule` . **Important** Não recomendamos conceder privilégios de atualizaç ão ou inserção diretamente a essa tabela. Isso permitiria que o usuário atualizasse a coluna `username` para ser executada como `rds-superuser` .
`cron.job_run_details`	Contém informações históricas sobre trabalhos agendados passados que foram executados. Isso é útil para investigar o status, as mensagens de retorno e as horas de início e término do trabalho executado. **Note** Para evitar que esta tabela cresça indefinidamente, purgue-a regularmente. Para ver um exemplo, consulte Limpar a tabela de histórico de pg_cron.

Usar pglogical para sincronizar dados entre instâncias

Todas as versões do RDS para PostgreSQL atualmente disponíveis são compatíveis com a extensão `pglogical`. A extensão pglogical é anterior ao recurso de replicação lógica funcionalmente similar que foi introduzido pelo PostgreSQL na versão 10. Para obter mais informações, consulte Executar replicação lógica para o Amazon RDS para PostgreSQL.

A extensão `pglogical` é compatível com a replicação lógica entre dois ou mais Instâncias de banco de dados do RDS para PostgreSQL. Ela também é compatível com a replicação entre diferentes versões do PostgreSQL e entre bancos de dados executados em instâncias de banco de dados RDS para PostgreSQL e clusters de banco de dados Aurora PostgreSQL. A extensão `pglogical` usa um modelo de publicação e assinatura para replicar alterações em tabelas e outros objetos, como sequências, de um editor para um assinante. Ela depende de um slot de replicação para garantir que as alterações sejam sincronizadas de um nó do editor para um nó assinante, definido da seguinte forma.

- O nó do editor é a instância de banco de dados RDS para PostgreSQL, que é a fonte de dados a serem replicados para outros nós. O nó do editor define as tabelas a serem replicadas em um conjunto de publicações.
- O nó do assinante é a instância de banco de dados Aurora PostgreSQL que recebe atualizações WAL do editor. O assinante cria uma assinatura para se conectar ao editor e obter os dados WAL decodificados. Quando o assinante cria a assinatura, o slot de replicação é criado no nó do editor.

Depois, você pode encontrar informações sobre a configuração da extensão `pglogical`.

Tópicos

- Requisitos e limitações da extensão pglogical
- Configurar a extensão pglogical
- Configurar a replicação lógica para a instância de banco de dados RDS para PostgreSQL
- Restabelecer a replicação lógica após uma atualização principal
- Gerenciar slots de replicação lógica para RDS para PostgreSQL
- Referência de parâmetros da extensão pglogical

Requisitos e limitações da extensão pglogical

Todas as versões atualmente disponíveis do RDS para PostgreSQL são compatíveis com a extensão `pglogical`.

Tanto o nó do editor quanto o do assinante devem estar configurados para replicação lógica.

As tabelas que você deseja replicar de um publicador para um assinante devem ter os mesmos nomes e o mesmo esquema. Essas tabelas também devem conter as mesmas colunas, e as colunas devem usar os mesmos tipos de dados. As tabelas de editores e assinantes devem ter as mesmas chaves primárias. Recomendamos que você use somente a CHAVE PRIMÁRIA como restrição exclusiva.

As tabelas no nó do assinante podem ter mais restrições permissivas do que as do nó do editor para restrições CHECK e NOT NULL.

A extensão `pglogical` fornece recursos como replicação bidirecional que não são compatíveis com o recurso de replicação lógica incorporado ao PostgreSQL (versão 10 e superior). Para obter mais informações, consulte PostgreSQL bi-directional replication using pglogical (Replicação bidirecional do PostgreSQL usando pglogical).

Configurar a extensão pglogical

Para configurar a extensão `pglogical` em sua instância de banco de dados RDS para PostgreSQL , adicione `pglogical` às bibliotecas compartilhadas no grupo de parâmetros de banco de dados personalizado para sua instância de banco de dados RDS para PostgreSQL. Você também precisa definir o valor do parâmetro `rds.logical_replication` como 1, para ativar a decodificação lógica. Finalmente, você cria a extensão no banco de dados. Você pode usar o AWS Management Console ou a AWS CLI para essas tarefas.

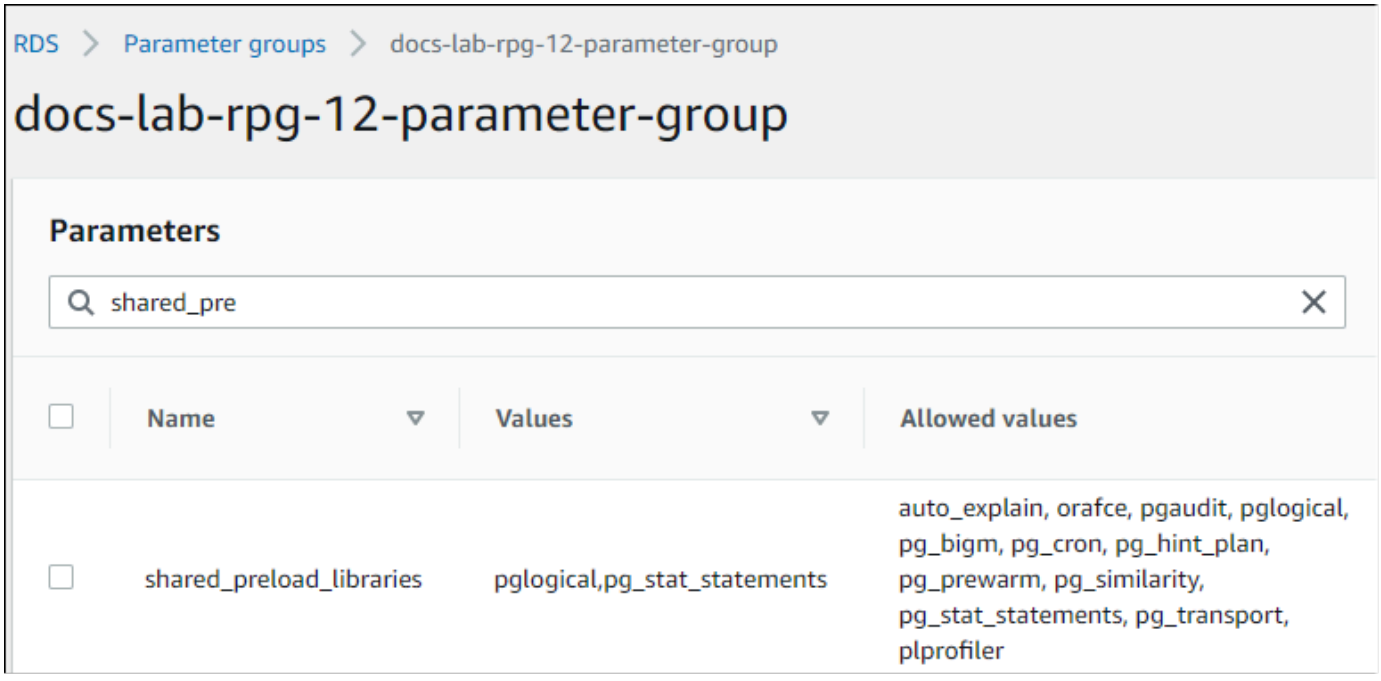
Você deve ter permissões como a função `rds_superuser` para realizar essas tarefas.

As etapas a seguir pressupõem que sua instância de banco de dados do RDS para PostgreSQL esteja associada a um grupo de parâmetros de banco de dados. Para obter informações sobre como criar um grupo de parâmetros de banco de dados, consulte Grupos de parâmetros para Amazon RDS.

Console

Como configurar a extensão pglogical

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, selecione sua instância de banco de dados do RDS para PostgreSQL.
3. Abra a guia Configuration (Configuração) para sua Instância de banco de dados do RDS para PostgreSQL. Entre os detalhes da instância, encontre o link Parameter group (Grupo de parâmetros).
4. Clique no link para abrir os parâmetros personalizados associados ao seu Instância de banco de dados do RDS para PostgreSQL.
5. No campo Parameters (Parâmetros), digite `shared_pre` para encontrar o parâmetro `shared_preload_libraries`.
6. Selecione Edit parameters (Editar parâmetros) para acessar os valores das propriedades.
7. Adicione `pglogical` à lista no campo Values (Valores). Use uma vírgula para separar itens na lista de valores.



8. Encontre o parâmetro `rds.logical_replication` e defina-o como 1 para ativar a replicação lógica.
9. Reinicialize a instância de banco de dados do RDS para PostgreSQL para que suas alterações tenham efeito.

10. Quando a instância estiver disponível, você poderá usar `psql` (ou pgAdmin) para se conectar à instância de banco de dados RDS para PostgreSQL.

```
psql --host=111122223333.aws-region.rds.amazonaws.com --port=5432 --
username=postgres --password --dbname=labdb
```

11. Para verificar se pglogical foi inicializada, execute o comando a seguir.

```
SHOW shared_preload_libraries;
shared_preload_libraries
--------------------------
rdsutils,pglogical
(1 row)
```

12. Verifique a configuração que permite a decodificação lógica da forma a seguir.

```
SHOW wal_level;
wal_level
-----------
 logical
(1 row)
```

13. Crie a extensão da forma a seguir.

```
CREATE EXTENSION pglogical;
EXTENSION CREATED
```

14. Escolha Salvar alterações.
15. Abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
16. Selecione a instância de banco de dados do RDS para PostgreSQL na lista de bancos de dados para selecioná-la e depois selecione Reboot (Reinicializar) no menu Actions (Ações).

AWS CLI

Como configurar a extensão pglogical

Para configurar a pglogical usando a AWS CLI, chame a operação modify-db-parameter-group para modificar determinados parâmetros em seu grupo de parâmetros personalizado, conforme mostrado no procedimento a seguir.

1. Use o comando AWS CLI a seguir para adicionar `pglogical` ao parâmetro `shared_preload_libraries`.

```
aws rds modify-db-parameter-group \
   --db-parameter-group-name custom-param-group-name \
   --parameters
 "ParameterName=shared_preload_libraries,ParameterValue=pglogical,ApplyMethod=pending-
reboot" \
   --region aws-region
```

2. Use o comando AWS CLI a seguir para definir `rds.logical_replication` como `1` a fim de ativar o recurso de decodificação lógica para a Instância de banco de dados do RDS para PostgreSQL.

```
aws rds modify-db-parameter-group \
   --db-parameter-group-name custom-param-group-name \
   --parameters
 "ParameterName=rds.logical_replication,ParameterValue=1,ApplyMethod=pending-
reboot" \
   --region aws-region
```

3. Use o comando AWS CLI a seguir para reinicializar a instância de banco de dados do RDS para PostgreSQL para que a biblioteca da pglogical seja inicializada.

```
aws rds reboot-db-instance \
    --db-instance-identifier your-instance \
    --region aws-region
```

4. Quando a instância estiver disponível, use `psql` para se conectar à instância de banco de dados RDS para PostgreSQL.

```
psql --host=111122223333.aws-region.rds.amazonaws.com --port=5432 --
username=postgres --password --dbname=labdb
```

5. Crie a extensão da forma a seguir.

```
CREATE EXTENSION pglogical;
EXTENSION CREATED
```

6. Reinicie a instância de banco de dados do RDS para PostgreSQL usando o comando AWS CLI a seguir.

```
aws rds reboot-db-instance \
    --db-instance-identifier your-instance \
    --region aws-region
```

Configurar a replicação lógica para a instância de banco de dados RDS para PostgreSQL

O procedimento a seguir mostra como iniciar a replicação lógica entre duas instâncias de banco de dados RDS PostgreSQL. As etapas pressupõem que tanto a fonte (editor) quanto o destino (assinante) tenham a extensão `pglogical` configurada conforme detalhado em Configurar a extensão pglogical.

> **Note**
>
> O `node_name` de um nó assinante não pode começar com `rds`.

Como criar o nó do editor e definir as tabelas a serem replicadas

Estas etapas pressupõem que a instância de banco de dados do RDS para PostgreSQL tenha um banco de dados com uma ou mais tabelas que você deseja replicar para outro nó. Você precisa recriar a estrutura da tabela do editor no assinante, então, primeiro, obtenha a estrutura da tabela, se necessário. Você pode fazer isso usando o metacomando `psql \d` *`tablename`* e criando a mesma tabela na instância do assinante. O procedimento a seguir cria uma tabela de exemplo no editor (fonte) para fins de demonstração.

1. Use `psql` para se conectar à instância que tem a tabela que você deseja usar como fonte para assinantes.

   ```
   psql --host=source-instance.aws-region.rds.amazonaws.com --port=5432 --
   username=postgres --password --dbname=labdb
   ```

   Se você não tiver uma tabela, crie uma tabela de exemplo da forma a seguir.

   a. Crie uma tabela de exemplo usando a declaração SQL a seguir.

      ```
      CREATE TABLE docs_lab_table (a int PRIMARY KEY);
      ```

b. Preencha a tabela com dados gerados usando a instrução SQL a seguir.

```
INSERT INTO docs_lab_table VALUES (generate_series(1,5000));
INSERT 0 5000
```

c. Verifique se os dados existem na tabela usando a declaração SQL a seguir.

```
SELECT count(*) FROM docs_lab_table;
```

2. Identifique essa instância de banco de dados do RDS para PostgreSQL como o nó do editor da forma a seguir.

```
SELECT pglogical.create_node(
    node_name := 'docs_lab_provider',
    dsn := 'host=source-instance.aws-region.rds.amazonaws.com port=5432
 dbname=labdb');
 create_node
-------------
   3410995529
(1 row)
```

3. Adicione a tabela que você deseja replicar ao conjunto de replicação padrão. Para obter mais informações sobre conjuntos de replicação, consulte Replication sets (Conjuntos de replicação) na documentação da pglogical.

```
SELECT pglogical.replication_set_add_table('default', 'docs_lab_table', 'true',
 NULL, NULL);
 replication_set_add_table
  --------------------------
  t
  (1 row)
```

A configuração do nó do editor está concluída. Agora você pode configurar o nó de assinante para receber as atualizações do editor.

Como configurar o nó de assinante e criar uma assinatura para receber atualizações

Estas etapas pressupõem que instância de banco de dados do RDS para PostgreSQL tenha sido configurada com a extensão `pglogical`. Para obter mais informações, consulte Configurar a extensão pglogical.

1. Use `psql` para se conectar à instância em que você deseja receber atualizações do editor.

```
psql --host=target-instance.aws-region.rds.amazonaws.com --port=5432 --
username=postgres --password --dbname=labdb
```

2. Na instância de banco de dados RDS para PostgreSQL do assinante, crie a mesma tabela que existe no editor. Neste exemplo, a tabela é `docs_lab_table`. Você pode criar a tabela da seguinte maneira.

```
CREATE TABLE docs_lab_table (a int PRIMARY KEY);
```

3. Verifique se essa tabela está vazia.

```
SELECT count(*) FROM docs_lab_table;
 count
-------
  0
(1 row)
```

4. Identifique essa instância de banco de dados RDS para PostgreSQL como o nó do assinante da forma a seguir.

```
SELECT pglogical.create_node(
    node_name := 'docs_lab_target',
    dsn := 'host=target-instance.aws-region.rds.amazonaws.com port=5432
 sslmode=require dbname=labdb user=postgres password=********');
 create_node
-------------
   2182738256
(1 row)
```

5. Crie a assinatura.

```
SELECT pglogical.create_subscription(
   subscription_name := 'docs_lab_subscription',
   provider_dsn := 'host=source-instance.aws-region.rds.amazonaws.com port=5432
 sslmode=require dbname=labdb user=postgres password=*******',
   replication_sets := ARRAY['default'],
   synchronize_data := true,
   forward_origins := '{}' );
 create_subscription
---------------------
```

```
1038357190
(1 row)
```

Ao concluir essa etapa, os dados da tabela no editor são criados na tabela no assinante. Se você quiser verificar se isso ocorreu, verifique a consulta SQL a seguir.

```
SELECT count(*) FROM docs_lab_table;
 count
-------
  5000
(1 row)
```

Desse ponto em diante, as alterações feitas na tabela do editor são replicadas na tabela do assinante.

Restabelecer a replicação lógica após uma atualização principal

Antes de realizar uma atualização de versão principal de uma instância de banco de dados do RDS para PostgreSQL que está sendo configurada como um nó de editor para replicação lógica, você deve eliminar todos os slots de replicação, mesmo aqueles que não estão ativos. Recomendamos que você desvie temporariamente as transações do banco de dados do nó do editor, elimine os slots de replicação, atualize a instância de banco de dados do RDS para PostgreSQL e, depois, restabeleça e reinicie a replicação.

Os slots de replicação são hospedados somente no nó do editor. O nó de assinante do RDS para PostgreSQL em um cenário de replicação lógica não tem slots a serem descartados, mas não pode ser atualizado para uma versão principal enquanto estiver designado como um nó de assinante com uma assinatura do editor. Antes de atualizar o nó de assinante do RDS para PostgreSQL, elimine a assinatura e o nó. Para obter mais informações, consulte Gerenciar slots de replicação lógica para RDS para PostgreSQL.

Determinar se a replicação lógica foi interrompida

Você pode determinar se o processo de replicação foi interrompido consultando o nó do editor ou o nó do assinante da forma a seguir.

Como conferir o nó do editor

- Use `psql` para se conectar ao nó do editor e, depois, consultar a função `pg_replication_slots`. Observe o valor na coluna ativa. Normalmente, isso retornará `t` (true)

mostrando que a replicação está ativa. Se a consulta retornar `f` (false), é uma indicação de que a replicação para o assinante foi interrompida.

```
SELECT slot_name,plugin,slot_type,active FROM pg_replication_slots;
                    slot_name              |      plugin      | slot_type | active
-------------------------------------------+------------------+-----------+--------
 pgl_labdb_docs_labcb4fa94_docs_lab3de412c | pglogical_output | logical   | f
(1 row)
```

Como conferir o nó do assinante

No nó do assinante, você pode conferir o status da replicação de três maneiras diferentes.

- Examine os logs do PostgreSQL no nó do assinante para encontrar mensagens de falha. O log identifica falhas com mensagens que incluem o código de saída 1, conforme mostrado a seguir.

```
2022-07-06 16:17:03 UTC::@:[7361]:LOG: background worker "pglogical apply
 16404:2880255011" (PID 14610) exited with exit code 1
2022-07-06 16:19:44 UTC::@:[7361]:LOG: background worker "pglogical apply
 16404:2880255011" (PID 21783) exited with exit code 1
```

- Consulte a função `pg_replication_origin`. Conecte-se ao banco de dados no nó do assinante usando `psql` e consulte a função `pg_replication_origin` da forma a seguir.

```
SELECT * FROM pg_replication_origin;
 roident | roname
---------+--------
(0 rows)
```

  O conjunto de resultados vazio significa que a replicação foi interrompida. Normalmente, você deve ver a saída da forma a seguir.

```
  roident |                       roname
 ---------+-----------------------------------------------------
        1 | pgl_labdb_docs_labcb4fa94_docs_lab3de412c
 (1 row)
```

- Consulte a função `pglogical.show_subscription_status` conforme exibido no exemplo a seguir.

```
SELECT subscription_name,status,slot_name FROM pglogical.show_subscription_status();
     subscription_name | status |              slot_name
-------+----------------+--------+-----------------------------------
 docs_lab_subscription | down   | pgl_labdb_docs_labcb4fa94_docs_lab3de412c
(1 row)
```

Essa saída mostra que a replicação foi interrompida. Seu status é down. Normalmente, a saída mostra o status como `replicating`.

Se seu processo de replicação lógica tiver sido interrompido, você poderá restabelecer a replicação seguindo estas etapas.

Como restabelecer a replicação lógica entre os nós do editor e do assinante

Para restabelecer a replicação, primeiro você desconecta o assinante do nó do editor e depois restabelece a assinatura, conforme descrito nestas etapas.

1. Conecte-se ao nó do assinante usando `psql` da forma a seguir.

   ```
   psql --host=222222222222.aws-region.rds.amazonaws.com --port=5432 --
   username=postgres --password --dbname=labdb
   ```

2. Desative a assinatura usando a função `pglogical.alter_subscription_disable`.

   ```
   SELECT pglogical.alter_subscription_disable('docs_lab_subscription',true);
    alter_subscription_disable
   ----------------------------
    t
   (1 row)
   ```

3. Obtenha o identificador do nó do editor consultando a `pg_replication_origin` da forma a seguir.

   ```
   SELECT * FROM pg_replication_origin;
    roident |               roname
   ---------+-----------------------------------
          1 | pgl_labdb_docs_labcb4fa94_docs_lab3de412c
   (1 row)
   ```

4. Use a resposta da etapa anterior com o comando `pg_replication_origin_create` para atribuir o identificador que pode ser usado pela assinatura quando restabelecida.

```
SELECT pg_replication_origin_create('pgl_labdb_docs_labcb4fa94_docs_lab3de412c');
  pg_replication_origin_create
------------------------------
                             1
(1 row)
```

5. Ative a assinatura passando seu nome com um status de `true`, conforme exibido no exemplo a seguir.

```
SELECT pglogical.alter_subscription_enable('docs_lab_subscription',true);
  alter_subscription_enable
---------------------------
 t
(1 row)
```

Consulte o status do nó. Seu status deve ser `replicating` conforme mostrado neste exemplo.

```
SELECT subscription_name,status,slot_name
  FROM pglogical.show_subscription_status();
            subscription_name |   status    |              slot_name
------------------------------+-------------+-------------------------------------
 docs_lab_subscription         | replicating |
 pgl_labdb_docs_lab98f517b_docs_lab3de412c
(1 row)
```

Confira o status do slot de replicação do assinante no nó do editor. A coluna `active` do slot deve retornar `t` (true), indicando que a replicação foi restabelecida.

```
SELECT slot_name,plugin,slot_type,active
  FROM pg_replication_slots;
                   slot_name              |      plugin      | slot_type | active
------------------------------------------+------------------+-----------+--------
 pgl_labdb_docs_lab98f517b_docs_lab3de412c | pglogical_output | logical   | t
(1 row)
```

Gerenciar slots de replicação lógica para RDS para PostgreSQL

Antes de realizar uma atualização de versão principal de uma uma instância de banco de dados do RDS para PostgreSQL que está atuando como um nó de editor em um cenário de replicação lógica, você deve eliminar os slots de replicação na instância. O processo de pré-conferência da atualização da versão principal notifica você de que a atualização não pode continuar até que os slots sejam eliminados.

Para eliminar slots da sua instância de banco de dados do RDS para PostgreSQL, primeiro cancele a assinatura e depois o slot.

Para identificar os slots de replicação que foram criados usando a extensão `pglogical`, faça login em cada banco de dados e obtenha o nome dos nós. Ao consultar o nó do assinante, você obtém os nós do editor e do assinante na saída, conforme mostrado neste exemplo.

```
SELECT * FROM pglogical.node;
node_id   |     node_name
------------+------------------
 2182738256 | docs_lab_target
 3410995529 | docs_lab_provider
(2 rows)
```

Você pode obter os detalhes sobre a assinatura com a consulta a seguir.

```
SELECT sub_name,sub_slot_name,sub_target
  FROM pglogical.subscription;
 sub_name |         sub_slot_name          | sub_target
----------+--------------------------------+------------
  docs_lab_subscription     | pgl_labdb_docs_labcb4fa94_docs_lab3de412c | 2182738256
(1 row)
```

Agora você pode cancelar a assinatura da forma a seguir.

```
SELECT pglogical.drop_subscription(subscription_name := 'docs_lab_subscription');
 drop_subscription
-------------------
                 1
(1 row)
```

Depois de cancelar a assinatura, você pode excluir o nó.

```
SELECT pglogical.drop_node(node_name := 'docs-lab-subscriber');
 drop_node
-----------
 t
(1 row)
```

Você pode verificar se o nó não existe mais da forma a seguir.

```
SELECT * FROM pglogical.node;
 node_id | node_name
---------+-----------
(0 rows)
```

Referência de parâmetros da extensão pglogical

Na tabela, você pode encontrar parâmetros associados à extensão `pglogical`. Parâmetros como `pglogical.conflict_log_level` e `pglogical.conflict_resolution` são usados para lidar com conflitos de atualização. Podem surgir conflitos quando alterações são feitas localmente nas mesmas tabelas que estão inscritas para receber alterações do editor. Os conflitos também podem ocorrer durante vários cenários, como replicação bidirecional ou quando vários assinantes estão se replicando do mesmo editor. Para obter mais informações, consulte PostgreSQL bi-directional replication using pglogical (Replicação bidirecional do PostgreSQL usando pglogical).

Parâmetro	Descrição
pglogical.batch_inserts	Inserções em lote, se possível. Não definido por padrão. Mude para “1” para ativar, “0” para desativar.
pglogical.conflict_log_level	Define o nível de log a ser usado para registrar em log conflitos resolvidos. Os valores de string compatíveis são debug5, debug4, debug3, debug2, debug1, info, notice, warning, error, log, fatal, panic.
pglogical.conflict_resolution	Define o método a ser usado para resolver conflitos quando eles podem ser resolvidos. Os valores de string compatíveis são error, apply_remote, keep_local, last_update_wins, first_update_wins.

Parâmetro	Descrição
pglogical.extra_connection_options	Opções de conexão para adicionar a todas as conexões de nó de pares.
pglogical.synchronous_commit	Valor de confirmação síncrona específica do pglogical
pglogical.use_spi	Use a SPI (interface de programação de servidores) em vez da API de baixo nível para aplicar alterações. Defina como “1” para ativar, “0” para desativar. Para obter mais informações sobre a SPI, consulte Server Programming Interface (Interface de programação de servidores) na documentação do PostgreSQL.

Usar pgactive para comportar a replicação ativa-ativa

A extensão `pgactive` usa replicação ativa-ativa para comportar e coordenar operações de gravação em vários bancos de dados do RDS para PostgreSQL. O Amazon RDS para PostgreSQL é compatível com a extensão `pgactive` nas seguintes versões:

- RDS para PostgreSQL 17.0 e todas as versões posteriores
- RDS para PostgreSQL 16.1 e versões 16 posteriores
- RDS para PostgreSQL 15.4-R2 e versões 15 posteriores
- RDS para PostgreSQL 14.10 e versões 14 posteriores
- RDS para PostgreSQL 13.13 e versões 13 posteriores
- RDS para PostgreSQL 12.17 e versões 12 posteriores
- RDS para PostgreSQL 11.22

> **Note**
>
> Quando há operações de gravação em mais de um banco de dados em uma configuração de replicação, existe a possibilidade de conflitos. Para obter mais informações, consulte Lidar com conflitos na replicação ativa-ativa.

Tópicos

- Limitações da extensão pgactive
- Inicializar o recurso de extensão pgactive
- Configurar a replicação ativa-ativa para as instâncias de banco de dados do RDS para PostgreSQL
- Medir o atraso de replicação entre membros pgactive
- Configurar as definições de parâmetro para a extensão pgactive
- Noções básicas sobre conflitos ativos-ativos
- Noções básicas sobre o esquema da pgactive
- Referência de funções da pgactive
- Lidar com conflitos na replicação ativa-ativa
- Lidar com sequências na replicação ativa-ativa

Limitações da extensão pgactive

- Todas as tabelas exigem uma chave primária; do contrário, atualizações e exclusões não são permitidas. Os valores na coluna Chave primária não devem ser atualizados.
- As sequências podem ter lacunas e, às vezes, podem não seguir uma ordem. As sequências não são replicadas. Para obter mais informações, consulte Lidar com sequências na replicação ativa-ativa.
- O DDL e os objetos grandes não são replicados.
- Índices secundários exclusivos podem causar divergência de dados.
- O agrupamento precisa ser idêntico em todos os nós do grupo.
- O balanceamento de carga entre os nós é um antipadrão.
- Transações grandes podem causar atraso na replicação.

Inicializar o recurso de extensão pgactive

Para inicializar o recurso de extensão `pgactive` na instância de banco de dados do RDS para PostgreSQL, defina o valor do parâmetro `rds.enable_pgactive` como `1` e, em seguida, crie a extensão no banco de dados. Isso ativa automaticamente os parâmetros `rds.logical_replication` e `track_commit_timestamp` e define o valor de `wal_level` como `logical`.

Você deve ter permissões como a função `rds_superuser` para realizar essas tarefas.

Você pode usar o AWS Management Console ou a AWS CLI para criar as instâncias necessárias do banco de dados RDS para PostgreSQL. As etapas a seguir pressupõem que a instância de banco de dados do RDS para PostgreSQL esteja associada a um grupo de parâmetros de banco de dados personalizado. Para obter mais informações sobre como criar um grupo de parâmetros de banco de dados personalizado, consulte Grupos de parâmetros para Amazon RDS.

Console

Como inicializar o recurso de extensão pgactive

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha a instância de banco de dados do RDS para PostgreSQL.

3. Abra a guia Configuração para a instância de banco de dados do RDS para PostgreSQL. Nos detalhes da instância, encontre o link do Grupo de parâmetros da instância de banco de dados.
4. Clique no link para abrir os parâmetros personalizados associados à instância de banco de dados do RDS para PostgreSQL.
5. Encontre o parâmetro `rds.enable_pgactive` e configure-o como 1 para inicializar o recurso `pgactive`.
6. Escolha Salvar alterações.
7. No painel de navegação do console do Amazon RDS, escolha Bancos de dados.
8. Selecione a instância de banco de dados do RDS para PostgreSQL e escolha Reinicializar no menu Ações.
9. Confirme a reinicialização da instância de banco de dados para que as alterações tenham efeito.
10. Quando a instância de banco de dados estiver disponível, use `psql` ou qualquer outro cliente PostgreAQL para se conectar à instância de banco de dados do RDS para PostgreSQL.

    O exemplo a seguir pressupõe que a instância de banco de dados do RDS para PostgreSQL tenha um banco de dados padrão chamado *`postgres`*.

    ```
    psql --host=mydb.111122223333.aws-region.rds.amazonaws.com --port=5432 --
    username=postgres --password=PASSWORD --dbname=postgres
    ```

11. Para verificar se pgactive foi inicializada, execute o comando a seguir.

    ```
    postgres=>SELECT setting ~ 'pgactive'
    FROM pg_catalog.pg_settings
    WHERE name = 'shared_preload_libraries';
    ```

    Se `pgactive` estiver em `shared_preload_libraries`, o comando anterior retornará o seguinte:

    ```
    ?column?
    ----------
     t
    ```

AWS CLI

Como inicializar o recurso de extensão pgactive

Para inicializar `pgactive` usando a AWS CLI, execute a operação modify-db-parameter-group para modificar determinados parâmetros no grupo parâmetros personalizado, conforme mostrado no procedimento a seguir.

1. Use o comando AWS CLI a seguir para definir `rds.enable_pgactive` como 1 para inicializar o recurso `pgactive` da instância de banco de dados do RDS para PostgreSQL.

```
postgres=>aws rds modify-db-parameter-group \
   --db-parameter-group-name custom-param-group-name \
   --parameters
 "ParameterName=rds.enable_pgactive,ParameterValue=1,ApplyMethod=pending-reboot" \
   --region aws-region
```

2. Use o comando AWS CLI a seguir para reinicializar a instância de banco de dados do RDS para PostgreSQL e inicializar a biblioteca da `pgactive`.

```
aws rds reboot-db-instance \
    --db-instance-identifier your-instance \
    --region aws-region
```

3. Quando a instância estiver disponível, use `psql` para se conectar à instância de banco de dados RDS para PostgreSQL.

```
psql --host=mydb.111122223333.aws-region.rds.amazonaws.com --port=5432 --
username=master user --password=PASSWORD --dbname=postgres
```

4. Para verificar se pgactive foi inicializada, execute o comando a seguir.

```
postgres=>SELECT setting ~ 'pgactive'
FROM pg_catalog.pg_settings
WHERE name = 'shared_preload_libraries';
```

   Se `pgactive` estiver em `shared_preload_libraries`, o comando anterior retornará o seguinte:

```
?column?
```

```
----------
 t
```

Configurar a replicação ativa-ativa para as instâncias de banco de dados do RDS para PostgreSQL

O procedimento a seguir mostra como iniciar a replicação ativa-ativa entre dois clusters de banco de dados do duas instâncias de banco de dados do RDS para PostgreSQL em que `pgactive` está disponível. Para executar o exemplo de alta disponibilidade multirregional, você precisa implantar instâncias do Amazon RDS para PostgreSQL em duas regiões diferentes e configurar o emparelhamento de VPC. Para obter mais informações, consulte Emparelhamento de VPC.

> **Note**
>
> O envio de tráfego entre várias regiões pode gerar custos adicionais.

Estas etapas pressupõem que a instância de banco de dados do RDS para PostgreSQL foi habilitada com a extensão `pgactive`. Para obter mais informações, consulte Inicializar o recurso de extensão pgactive.

Como configurar a primeira instância de banco de dados do RDS para PostgreSQL com a extensão **`pgactive`**

O exemplo a seguir ilustra como o grupo `pgactive` é criado e mostra outras etapas necessárias para criar a extensão `pgactive` na instância de banco de dados do RDS para PostgreSQL.

1. Use `psql` ou outra ferramenta cliente para se conectar à primeira instância de banco de dados do RDS para PostgreSQL.

   ```
   psql --host=firstinstance.111122223333.aws-region.rds.amazonaws.com --port=5432 --
   username=postgres --password=PASSWORD --dbname=postgres
   ```

2. Crie um banco de dados na instância do RDS para PostgreSQL usando o seguinte comando:

   ```
   postgres=> CREATE DATABASE app;
   ```

3. Alterne a conexão para o novo banco de dados usando o seguinte comando:

```
\c app
```

4. Crie e preencha uma tabela de exemplo usando a seguinte instrução SQL:

   a. Crie uma tabela de exemplo usando a declaração SQL a seguir.

   ```
   app=> CREATE SCHEMA inventory;
   CREATE TABLE inventory.products (
   id int PRIMARY KEY, product_name text NOT NULL,
   created_at timestamptz NOT NULL DEFAULT CURRENT_TIMESTAMP);
   ```

   b. Preencha a tabela com alguns dados de amostra usando a instrução SQL a seguir.

   ```
   app=> INSERT INTO inventory.products (id, product_name)
   VALUES (1, 'soap'), (2, 'shampoo'), (3, 'conditioner');
   ```

   c. Verifique se os dados existem na tabela usando a declaração SQL a seguir.

   ```
    app=>SELECT count(*) FROM inventory.products;

    count
   -------
    3
   ```

5. Crie uma extensão `pgactive` no banco de dados existente.

   ```
   app=> CREATE EXTENSION pgactive;
   ```

6. Para criar e inicializar com segurança o grupo da pgactive, use os seguintes comandos:

   ```
   app=>
   -- connection info for endpoint1
   CREATE SERVER pgactive_server_endpoint1
       FOREIGN DATA WRAPPER pgactive_fdw
       OPTIONS (host '<endpoint1>', dbname 'app');
   CREATE USER MAPPING FOR postgres
       SERVER pgactive_server_endpoint1
       OPTIONS (user 'postgres', password '<password>');
         -- connection info for endpoint2
   CREATE SERVER pgactive_server_endpoint2
       FOREIGN DATA WRAPPER pgactive_fdw
       OPTIONS (host '<endpoint2>', dbname 'app');
   ```

```
CREATE USER MAPPING FOR postgres
    SERVER pgactive_server_endpoint2
    OPTIONS (user 'postgres', password '<password>');
```

Agora você pode inicializar o grupo de replicações e adicionar esta primeira instância:

```
SELECT pgactive.pgactive_create_group(
    node_name := 'endpoint1-app',
    node_dsn := 'user_mapping=postgres
 pgactive_foreign_server=pgactive_server_endpoint1'

);
```

Use os seguintes comandos como um método alternativo, mas menos seguro, para criar e inicializar o grupo da pgactive:

```
app=> SELECT pgactive.pgactive_create_group(
    node_name := 'node1-app',
    node_dsn := 'dbname=app host=firstinstance.111122223333.aws-
region.rds.amazonaws.com user=postgres password=PASSWORD');
```

node1-app é o nome atribuído para identificar de maneira exclusiva um nó no grupo `pgactive`.

> **Note**
>
> Para realizar essa etapa com êxito em uma instância de banco de dados acessível ao público geral, você deve ativar o parâmetro `rds.custom_dns_resolution` definindo-o como 1.

7. Para verificar se a instância de banco de dados está pronta, use o seguinte comando:

```
app=> SELECT pgactive.pgactive_wait_for_node_ready();
```

Se o comando for bem-sucedido, você verá o seguinte resultado:

```
pgactive_wait_for_node_ready
```

```
-----------------------------
(1 row)
```

Como configurar a segunda instância do RDS para PostgreSQL e juntá-la ao grupo **`pgactive`**

O exemplo a seguir mostra como juntar uma instância de banco de dados do RDS para PostgreSQL ao grupo `pgactive`, bem como outras etapas necessárias para criar a extensão `pgactive` na instância de banco de dados.

Estas etapas pressupõem que as instâncias de banco de dados do RDS para PostgreSQL tenham sido configuradas com a extensão `pgactive`. Para obter mais informações, consulte Inicializar o recurso de extensão pgactive.

1. Use `psql` para se conectar à instância em que você deseja receber atualizações do editor.

```
psql --host=secondinstance.111122223333.aws-region.rds.amazonaws.com --port=5432 --
username=postgres --password=PASSWORD --dbname=postgres
```

2. Crie um banco de dados na segunda instância de banco de dados do RDS para PostgreSQL usando o seguinte comando:

```
postgres=> CREATE DATABASE app;
```

3. Alterne a conexão para o novo banco de dados usando o seguinte comando:

```
\c app
```

4. Crie a extensão `pgactive` no banco de dados existente.

```
app=> CREATE EXTENSION pgactive;
```

5. Junte a segunda instância de banco de dados do RDS para PostgreSQL ao grupo do `pgactive` de uma forma mais segura usando os seguintes comandos:

```

-- connection info for endpoint1
CREATE SERVER pgactive_server_endpoint1
    FOREIGN DATA WRAPPER pgactive_fdw
    OPTIONS (host '<endpoint1>', dbname 'app');
CREATE USER MAPPING FOR postgres
```

```
    SERVER pgactive_server_endpoint1
    OPTIONS (user 'postgres', password '<password>');

-- connection info for endpoint2
CREATE SERVER pgactive_server_endpoint2
    FOREIGN DATA WRAPPER pgactive_fdw
    OPTIONS (host '<endpoint2>', dbname 'app');
CREATE USER MAPPING FOR postgres
    SERVER pgactive_server_endpoint2
    OPTIONS (user 'postgres', password '<password>');
```

```
SELECT pgactive.pgactive_join_group(
    node_name := 'endpoint2-app',
    node_dsn := 'user_mapping=postgres
 pgactive_foreign_server=pgactive_server_endpoint2',
    join_using_dsn := 'user_mapping=postgres
 pgactive_foreign_server=pgactive_server_endpoint1'
);
```

Use os comandos a seguir como um método alternativo, mas menos seguro, para juntar a segunda instância de banco de dados do RDS for PostgreSQL ao grupo da `pgactive`:

```
app=> SELECT pgactive.pgactive_join_group(
node_name := 'node2-app',
node_dsn := 'dbname=app host=secondinstance.111122223333.aws-
region.rds.amazonaws.com user=postgres password=PASSWORD',
join_using_dsn := 'dbname=app host=firstinstance.111122223333.aws-
region.rds.amazonaws.com user=postgres password=PASSWORD');
```

node2-app é o nome atribuído para identificar de maneira exclusiva um nó no grupo `pgactive`.

6. Para verificar se a instância de banco de dados está pronta, use o seguinte comando:

```
app=> SELECT pgactive.pgactive_wait_for_node_ready();
```

Se o comando for bem-sucedido, você verá o seguinte resultado:

```
pgactive_wait_for_node_ready
-----------------------------
(1 row)
```

Se o primeiro banco de dados RDS para PostgreSQL for relativamente grande, você poderá ver `pgactive.pgactive_wait_for_node_ready()` emitindo o relatório de progresso da operação de restauração. A saída será semelhante à seguinte:

```
NOTICE:  restoring database 'app', 6% of 7483 MB complete
NOTICE:  restoring database 'app', 42% of 7483 MB complete
NOTICE:  restoring database 'app', 77% of 7483 MB complete
NOTICE:  restoring database 'app', 98% of 7483 MB complete
NOTICE:  successfully restored database 'app' from node node1-app in
 00:04:12.274956
 pgactive_wait_for_node_ready
-----------------------------
(1 row)
```

Deste ponto em diante, `pgactive` sincroniza os dados entre as duas instâncias de banco de dados.

7. Você pode usar o comando a seguir para verificar se o banco de dados da segunda instância de banco de dados tem os dados:

```
app=> SELECT count(*) FROM inventory.products;
```

Se os dados forem sincronizados com sucesso, você verá a seguinte saída:

```
 count
-------
 3
```

8. Execute o seguinte comando para inserir novos valores:

```
app=> INSERT INTO inventory.products (id, product_name) VALUES (4, 'lotion');
```

9. Conecte-se ao banco de dados da primeira instância de banco de dados e execute a seguinte consulta:

```
app=> SELECT count(*) FROM inventory.products;
```

Se a replicação ativa-ativa for inicializada, a saída será semelhante à seguinte:

```
count
-------
 4
```

Como desanexar e remover uma instância de banco de dados do grupo **`pgactive`**

É possível desanexar e remover uma instância de banco de dados do grupo `pgactive` usando estas etapas:

1. Você pode separar a segunda instância de banco de dados da primeira instância de banco de dados usando o seguinte comando:

   ```
   app=> SELECT * FROM pgactive.pgactive_detach_nodes(ARRAY['node2-app']);
   ```

2. Remova a extensão `pgactive` da segunda instância de banco de dados usando o seguinte comando:

   ```
   app=> SELECT * FROM pgactive.pgactive_remove();
   ```

   Para remover a extensão forçosamente:

   ```
   app=> SELECT * FROM pgactive.pgactive_remove(true);
   ```

3. Descarte a extensão usando o seguinte comando:

   ```
   app=> DROP EXTENSION pgactive;
   ```

Medir o atraso de replicação entre membros pgactive

Você pode usar a consulta a seguir para visualizar o atraso de replicação entre os membros de `pgactive`. Execute essa consulta em cada nó de `pgactive` para ter a imagem completa.

```
app=> SELECT * FROM pgactive.pgactive_get_replication_lag_info();
#-[ RECORD 1 ]--------+-------------------------------------------
#node_name            | node2-app
#node_sysid           | 7481018224801653637
#application_name     | pgactive:7481018224801653637:send
#slot_name            | pgactive_16385_7481018224801653637_0_16385__
#active               | t
#active_pid           | 783486
#pending_wal_decoding | 0
#pending_wal_to_apply | 0
#restart_lsn          | 0/2108150
#confirmed_flush_lsn  | 0/2154690
#sent_lsn             | 0/2154690
#write_lsn            | 0/2154690
#flush_lsn            | 0/2154690
#replay_lsn           | 0/2154690
#-[ RECORD 2 ]--------+-------------------------------------------
#node_name            | node1-app
#node_sysid           | 7481018033434600853
#application_name     | pgactive:7481018033434600853:send
#slot_name            | pgactive_16385_7481018033434600853_0_16385__
#active               | t
#active_pid           | 783488
#pending_wal_decoding | 0
#pending_wal_to_apply | 0
#restart_lsn          | 0/20F5AD0
#confirmed_flush_lsn  | 0/214EF68
#sent_lsn             | 0/214EF68
#write_lsn            | 0/214EF68
#flush_lsn            | 0/214EF68
#replay_lsn           | 0/214EF68
```

Monitore, no mínimo, os seguintes diagnósticos:

active

Configure alertas quando active for false, o que indica que o slot não está em uso no momento (a instância do assinante foi desconectada do publicador).

pending_wal_decoding

Na replicação lógica do PostgreSQL, os arquivos WAL são armazenados em formato binário. O publicador deve decodificar essas alterações do WAL e convertê-las em alterações lógicas (como operações de inserção, atualização ou exclusão).

A métrica pending_wal_decoding mostra o número de arquivos WAL aguardando decodificação do lado do publicador.

Esse número pode aumentar devido a estes fatores:

- Quando o assinante não estiver conectado, o status active será false e pending_wal_decoding aumentará.
- O slot está ativo, mas o publicador não consegue acompanhar o volume de alterações do WAL.

pending_wal_to_apply

A métrica pending_wal_apply indica o número de arquivos WAL que ainda precisam ser aplicados do lado do assinante.

Vários fatores podem impedir que o assinante aplique alterações e possivelmente provocar um cenário de disco cheio:

- Diferenças de esquema: por exemplo, quando você tem alterações no fluxo do WAL para uma tabela chamada sample e essa tabela não existe do lado do assinante.
- Valores nas colunas da chave primária foram atualizados.
- Índices secundários exclusivos podem causar divergência de dados.

Configurar as definições de parâmetro para a extensão pgactive

É possível usar a consulta a seguir para visualizar todos os parâmetros associados à extensão `pgactive`.

```
app=> SELECT * FROM pg_settings WHERE name LIKE 'pgactive.%';
```

Você pode configurar a extensão `pgactive` usando vários parâmetros. Esses parâmetros podem ser definidos por meio do AWS Management Console ou da interface da CLI AWS.

Principais parâmetros da extensão pgactive

A seguinte tabela apresenta uma referência para os principais parâmetros da extensão `pgactive`:

Parâmetro	Unidade	Padrão	Descrição
`pgactive.conflict_`	`boolean`	–	Registra em log informaçõ

Parâmetro	Unidade	Padrão	Descrição
`logging_include_tuples`			es completas da tupla para a extensão `pgactive`. **Note** É necessário reinicializar o servidor para que as alterações entrem em vigor.

Parâmetro	Unidade	Padrão	Descrição
`pgactive.log_conflicts_to_table`	`boolean`	–	Determina se a extensão `pgactive` registra em log os conflitos detectados na tabela `pgactive.pgactive_conflict_history`. Para ter mais informações e detalhes, consulte “Registro em log de conflitos”. **Note** É necessário reinicializar o servidor para que as alterações entrem em vigor.

Parâmetro	Unidade	Padrão	Descrição
`pgactive.log_conflicts_to_logfile`	`boolean`	–	Determina se a extensão `pgactive` registra em log os conflitos detectados no arquivo de log do PostgreSQL. Para ter mais informações e detalhes, consulte “Registro em log de conflitos”. **Note** É necessário reinicializar o servidor para que as alterações entrem em vigor.

Parâmetro	Unidade	Padrão	Descrição
`pgactive.synchronous_commit`	`boolean`	off	Determina o comportamento de confirmação dos operadores apply da pgactive. Quando desabilitados (off), os operadores apply executam confirmações assíncronas, o que melhora o throughput do PostgreSQL durante as operações apply, mas atrasa as confirmações de reprodução para o precedente. Configurá-los como `off` é sempre seguro e não causará perda ou omissão de transações. Essa configuração afeta apenas o tempo de liberação do disco no nó subsequente

Parâmetro	Unidade	Padrão	Descrição
			e quando as confirmações são enviadas ao precedent e. O sistema atrasa o envio das confirmaç ões de liberação de reproduçã o até que as confirmações sejam liberadas para o disco por meio de operações não relacionadas, como pontos de verificaç ão ou trabalho periódico. No entanto, se o precedente tiver o subsequen te listado em `synchrono us_standb y_names` , configurá-lo como `off` fará com que as confirmações síncronas no precedente demorem mais para relatar êxito

Parâmetro	Unidade	Padrão	Descrição
			ao cliente. Nesse caso, configure o parâmetro como on. **Note** Mesmo quando esse parâmetro é definido como on com os nós listados em `synchronous_standby_names`, conflitos de replicação ainda podem ocorrer em configurações ativas-ativas. Isso ocorre porque o sistema

Parâmetro	Unidade	Padrão	Descrição
			não tem bloqueio entre nós e gerenciamento global de snapshots, permitindo que transações simultâneas em nós diferentes modifiquem a mesma tupla. Além disso, as transações só iniciam a replicação depois de serem confirmadas no nó precedente. Habilitar

Parâmetro	Unidade	Padrão	Descrição
			a confirmação síncrona não transforma a extensão pgactive em um sistema sempre consistente.

Parâmetro	Unidade	Padrão	Descrição
pgactive.temp_dump_directory	string	–	Define o caminho de armazenamento temporário necessário para as operações de clonagem do banco de dados durante a configuração inicial. Esse diretório deve ser gravável pelo usuário do Postgres e ter espaço de armazenamento suficiente para conter um despejo completo do banco de dados. O sistema usa esse local somente durante a configuração inicial do banco de dados com operações de cópia lógica. Esse parâmetro não é usado pelo pgactive_

Parâmetro	Unidade	Padrão	Descrição
			`init_copy` `command`.

Parâmetro	Unidade	Padrão	Descrição
`pgactive.max_ddl_lock_delay`	`milliseconds`	`-1`	Especifica o tempo máximo de espera para o bloqueio de DDL antes de interromper à força as transações de gravação simultâneas. O valor padrão é `-1`, que adota o valor definido em `max_standby_streaming_delay` . Esse parâmetro aceita unidades de tempo. Por exemplo, você pode configurá-lo como 10s (10 segundos). Durante esse período de espera, o sistema tenta adquirir bloqueios de DDL enquanto aguarda a confirmação ou reversão das transações de

Parâmetro	Unidade	Padrão	Descrição
			gravação em andamento. Para ter mais informações, consulte “DDL Locking”.

Parâmetro	Unidade	Padrão	Descrição
pgactive.ddl_lock_timeout	milliseconds	-1	Especifica por quanto tempo uma tentativa de bloqueio de DDL espera para obter o bloqueio. O valor padrão é -1, que usa o valor especificado em lock_timeout. Você pode definir esse parâmetro usando unidades de tempo, como 10s (10 segundos). Esse temporizador controla apenas o período de espera para obter um bloqueio de DDL. Quando o sistema obtém o bloqueio e inicia a operação de DDL, o temporizador para. Esse parâmetro não limita a duração

Parâmetro	Unidade	Padrão	Descrição
			total em que um bloqueio de DDL pode ser mantido ou o tempo geral de operação de DDL. Para controlar a duração total da operação, use `statement_timeout` em vez disso. Para ter mais informações, consulte “DDL Locking”.

Parâmetro	Unidade	Padrão	Descrição
`pgactive.debug_trace_ddl_locks_level`	`boolean`	–	Substitui o nível de log de depuração padrão para operações de bloqueio de DDL na extensão `pgactive`. Quando configurada, essa definição faz com que mensagens relacionadas ao bloqueio de DDL sejam emitidas no nível de depuração de LOG em vez de no nível padrão. Use esse parâmetro para monitorar a atividade de bloqueio de DDL sem habilitar os níveis detalhados de log `DEBUG1` ou `DEBUG2` em todo o servidor. Níveis de log, em ordem crescente de detalhamento:

Parâmetro	Unidade	Padrão	Descrição
			• none: as mensagens de bloqueio de DDL aparecem somente nos níveis de log do servidor DEBUG1 e inferiores. • statement: adiciona saída de LOG para tentativas de aquisição de bloqueio de DDL. • acquire_release: registra eventos de aquisição, liberação e recusa de bloqueios, bem como aplicação de bloqueios de DDL remotos em nós pares. • peers: oferece detalhes adicionais sobre

Parâmetro	Unidade	Padrão	Descrição
			negociações de bloqueio de DDL entre nós pares. • debug; registra em log todas as atividades relacionadas a bloqueios de DDL no nível de LOG. Para ter mais informações sobre as opções de monitoram ento, consulte “Monitoring global DDL locks”. Note As alteraçõe s nessa configura ção entram em vigor quando você a recarrega . Não é necessári

Parâmetro	Unidade	Padrão	Descrição
			o reiniciar o servidor.

Parâmetros adicionais da extensão pgactive

A tabela a seguir apresenta as opções de configuração internas e menos frequentemente usadas disponíveis para a extensão `pgactive`.

Parâmetro	Unidade	Padrão	Descrição
`pgactive.debug_apply_delay`	`integer`	–	Define um atraso de apply (em milissegundos) para conexões configuradas que não têm um atraso de apply explícito na respectiva entrada `pgactive.pgactive_connections` . Esse atraso é definido durante a criação do nó ou o tempo de junção, e a pgactive não reproduzirá uma transação

Parâmetro	Unidade	Padrão	Descrição
			em nós pares enquanto pelo menos o número especificado de milissegu ndos não tiver decorrido desde que ela foi confirmada. Usado principal mente para simular redes de alta latência em ambientes de teste para facilitar a criação de conflitos. Por exemplo, com um atraso de 500 ms nos nós A e B, você terá pelo menos 500 ms para realizar uma inserção conflitante no nó B depois de inserir um valor no nó A. Note Requer o recarrega

Parâmetro	Unidade	Padrão	Descrição
			mento do servidor ou a reinicial ização dos operadore s apply para entrar em vigor.

Parâmetro	Unidade	Padrão	Descrição
pgactive.connectability_check_duration	integer	–	Especifica a duração (em segundos) em que um operador de banco de dados tenta estabelecer conexões durante tentativas malsucedidas. O operador faz 1 tentativa de conexão por segundo até ter êxito ou atingir esse valor de tempo limite. Essa configuração é útil quando o mecanismo de banco de dados é iniciado antes de o operador estar pronto para estabelecer conexões.

Parâmetro	Unidade	Padrão	Descrição
`pgactive.skip_ddl_replication`	`boolean`	on	Controla como as alterações de DDL são replicadas ou tratadas no Amazon RDS com `pgactive` habilitada. Quando definido como on, o nó processa as alterações de DDL como um nó não pgactive. Os seguintes requisitos se aplicam ao trabalhar com esse parâmetro: • Novos nós não podem se juntar a um grupo da pgactive se o respectivo valor de `skip_ddl_replication` for diferente do nó precedente. • Os nós existentes não podem iniciar

Parâmetro	Unidade	Padrão	Descrição
			operadores da pgactive se o valor do parâmetro não correspon der ao nó precedente. • Todos os membros da pgactive devem usar o mesmo valor de parâmetro. Você pode modificar esse parâmetro de duas maneiras com privilégios de superusuário: globalmente e localmente (nível de sessão). **Note** A alteração incorreta desse parâmetro pode desfazer suas configura

Parâmetro	Unidade	Padrão	Descrição
			ções de replicaçã o.
pgactive. do_not_re plicate	boolean	–	Esse parâmetro é somente para uso interno. Ao definir esse parâmetro em uma transação , as alteraçõe s não são replicadas para outros nós no cluster de banco de dados. **Note** A alteração incorreta desse parâmetro pode desfazer suas configura ções de replicaçã o.

Parâmetro	Unidade	Padrão	Descrição
pgactive.discard_mismatched_row_attributes	boolean	–	Esse parâmetro destina-se somente a especialistas. Recomendamos usá-lo somente ao solucionar problemas específicos de replicação. Use esse parâmetro quando: • O fluxo de replicação de entrada contiver linhas com mais colunas do que a tabela local. • Essas linhas remotas contêm valores não nulos. Essa configuração substitui a seguinte mensagem de erro e permite que surjam divergências

Parâmetro	Unidade	Padrão	Descrição
			de dados para que a replicação continue: `cannot right-pad mismatched attributes; attno %u is missing in local table and remote row has non-null, non-dropped value for this attribute .` **Note** A alteração incorreta desse parâmetro pode desfazer suas configurações de replicação.

Parâmetro	Unidade	Padrão	Descrição
pgactive.debug_trace_replay	boolean	–	Quando definido como on, ele emite uma mensagem de log para cada ação remota que os operadores apply subsequentes processam. Os logs incluem: • Alterar tipo • Nome da tabela afetada • Número de alterações desde o início da transação • LSN de confirmação da transação • Carimbo de data/hora da confirmação • Identificador do nó precedente • Identificador do nó de encaminhamento (se aplicável)

Parâmetro	Unidade	Padrão	Descrição
			Os logs também capturam comandos de DDL enfileirados e exclusões de tabela. para> Por padrão, os logs não incluem o conteúdo de campos de linha. Para incluir valores de linha nos logs, você deve recompilar com os seguintes sinalizadores habilitados: • VERBOSE_INSERT • VERBOSE_UPDATE • VERBOSE_DELETE Note A habilitação dessa configuração de

Parâmetro	Unidade	Padrão	Descrição
			registro em log pode afetar o desempenho. Recomendamos habilitá-la somente quando necessário para solucionar problemas. As alterações nessa configuração entram em vigor quando você a recarrega. Não é necessário reiniciar o servidor.

Parâmetro	Unidade	Padrão	Descrição
`pgactive.extra_apply_connection_options`		–	Você pode configurar os parâmetros de conexão para todas as conexões de nós pares com nós da pgactive. Esses parâmetros controlam configurações, como keepalives e modos SSL. Por padrão, a pgactive usa os seguintes parâmetros de conexão: • connect_timeout=30 • keepalives=1 • keepalives_idle=20 • keepalives_interval=20 • keepalives_count=5 Para substituir os parâmetros padrão, use o comando

Parâmetro	Unidade	Padrão	Descrição
			semelhante abaixo: `pgactive.extra_apply_connection_options = 'keepalives=0'` As strings de conexão de nós individuais têm precedência sobre essas configurações e sobre as opções de conexão integradas da pgactive. Para ter mais informações sobre formatos de string de conexão, consulte as strings de conexão libpq. Recomendamos manter as configurações padrão de keepalive habilitadas.

Parâmetro	Unidade	Padrão	Descrição
			Só desabilite os keepalives se você tiver problemas para concluir grandes transações em redes não confiáveis. Note Recomendamos manter as configurações padrão de keepalive habilitadas. Só desabilite os keepalives se você tiver problemas para concluir grandes transações em redes

Parâmetro	Unidade	Padrão	Descrição
			não confiáveis. As alterações nessa configuração entram em vigor quando você a recarrega. Não é necessário reiniciar o servidor.

Parâmetro	Unidade	Padrão	Descrição
`pgactive.init_node_parallel_jobs (int)`		–	Especifica o número de trabalhos paralelos que `pg_dump` e `pg_restore` podem usar durante junções de nós lógicos com a função `pgactive.pgactive_join_group` . As alterações nessa configuração entram em vigor quando você a recarrega. Não é necessário reiniciar o servidor.

Parâmetro	Unidade	Padrão	Descrição
`pgactive.max_nodes`	`int`	4	Especifica o número máximo de nós permitidos em um grupo da extensão pgactive. O valor padrão é quatro nós. Você deve considerar o seguinte ao definir o valor desse parâmetro: • Todos os nós em um grupo da extensão pgactive devem usar o mesmo valor de parâmetro. • Um novo nó não poderá se juntar se o valor do respectivo parâmetro for diferente do nó precedente. • Os nós existentes não poderão iniciar operadores da extensão

Parâmetro	Unidade	Padrão	Descrição
			pgactive se o valor do respectivo parâmetro for diferente do nó precedente. • Grupos maiores exigem esforços adicionais de monitoramento e manutenção; portanto, defina o valor desse parâmetro de maneira sensata. Você pode definir esse parâmetro de duas maneiras: no arquivo de configuração e usando o comando `ALTER SYSTEM SET`. O valor padrão para esse parâmetro é 4; ou seja, pode haver no

Parâmetro	Unidade	Padrão	Descrição
			máximo quatro nós permitidos no grupo da extensão `pgactive` em qualquer momento específico. **Note** A alteração entra em vigor após a reinicialização do servidor.
`pgactive.permit_node_identifier_getter_function_creation`	`boolean`	–	Esse parâmetro destina-se somente para uso interno. Quando habilitada, a extensão `pgactive` permite a criação da função getter do identificador de nó da pgactive.

Noções básicas sobre conflitos ativos-ativos

Quando você usa pgactive no modo ativo-ativo, gravar nas mesmas tabelas por meio de vários nós pode criar conflitos de dados. Embora alguns sistemas de clustering usem bloqueios distribuídos para impedir o acesso simultâneo, a pgactive adota uma abordagem otimista que é mais adequada para aplicações distribuídas geograficamente.

Alguns sistemas de clustering de banco de dados impedem o acesso simultâneo aos dados por meio de bloqueios distribuídos. Embora essa abordagem funcione quando os servidores estão próximos, não é possível usá-la para aplicações distribuídas geograficamente porque, para oferecer um bom desempenho, é necessária uma latência extremamente baixa. Em vez de usar bloqueios distribuídos (uma abordagem pessimista), a extensão pgactive usa uma abordagem otimista. Isso significa que ela:

- Ajuda a evitar conflitos quando possível.
- Possibilita que determinados tipos de conflito ocorram.
- Oferece resolução quando ocorrem conflitos.

Essa abordagem oferece maior flexibilidade ao criar aplicações distribuídas.

Como os conflitos acontecem

Os conflitos entre nós surgem de sequências de eventos que não poderiam acontecer se todas as transações envolvidas ocorressem simultaneamente no mesmo nó. Como os nós só trocam alterações após a confirmação das transações, cada transação é válida individualmente no nó em que foi confirmada, mas não seria válida se executada em outro nó que tivesse feito outro trabalho nesse meio tempo. Como o apply da pgactive basicamente reproduz a transação nos outros nós, a operação de repetição pode falhar se houver um conflito entre uma transação que está sendo aplicada e uma transação que foi confirmada no nó receptor.

O motivo pelo qual a maioria dos conflitos não pode acontecer quando todas as transações são executadas em um único nó é que o PostgreSQL tem mecanismos de comunicação entre transações para evitá-los, como:

- Índices UNIQUE
- SEQUENCEs
- Bloqueio de linha e relação
- Rastreamento de dependências SERIALIZÁVEL

Todos esses mecanismos são formas de comunicação entre transações para evitar problemas indesejáveis de simultaneidade.

A pgactive alcança baixa latência e lida bem com partições de rede porque não usa um gerenciador de transações distribuídas ou gerenciador de bloqueios. Mas isso significa que as transações em nós diferentes são executadas de maneira totalmente isolada umas das outras. Embora o isolamento normalmente melhore a consistência do banco de dados, nesse caso, você precisa reduzir o isolamento para evitar conflitos.

Tipos de conflito

Os conflitos que podem ocorrer incluem:

Tópicos

- Conflitos de CHAVE PRIMÁRIA ou UNIQUE
- Conflitos INSERT/INSERT
- INSERTs que violam várias restrições UNIQUE
- Conflitos UPDATE/UPDATE
- Conflitos UPDATE na CHAVE PRIMÁRIA
- UPDATEs que violam várias restrições UNIQUE
- Conflitos UPDATE/DELETE
- Conflitos INSERT/UPDATE
- Conflitos DELETE/DELETE
- Conflitos de restrição de chave estrangeira
- Conflitos de restrição de exclusão
- Conflitos de dados globais
- Conflitos de bloqueio e interrupções decorrentes de deadlock
- Conflitos divergentes

Conflitos de CHAVE PRIMÁRIA ou UNIQUE

Os conflitos de linha ocorrem quando várias operações tentam modificar a mesma chave de linha de uma maneira não possíveis em um único nó. Esses conflitos representam o tipo mais comum de conflito de dados.

A pgactive resolve conflitos detectados por meio da abordagem de última atualização prevalece ou de seu manipulador de conflitos personalizado.

Os conflitos de linha incluem:

- INSERT versus INSERT
- INSERT versus UPDATE
- UPDATE versus DELETE
- INSERT versus DELETE
- DELETE versus DELETE
- INSERT versus DELETE

Conflitos INSERT/INSERT

Esse conflito mais comum ocorre quando INSERTs em dois nós diferentes criam uma tupla com os mesmos valores de CHAVE PRIMÁRIA (ou valores de restrição UNIQUE idênticos quando não existe CHAVE PRIMÁRIA).

O pgactivelink resolve conflitos de INSERT usando o carimbo de data/hora do host de origem para manter a tupla mais recente. Você pode substituir esse comportamento padrão com seu manipulador de conflitos personalizado. Embora esse processo não exija nenhuma ação especial do administrador, esteja ciente de que o pgactivelink descarta uma das operações INSERT em todos os nós. Nenhuma mesclagem automática de dados ocorre, a menos que seu manipulador personalizado a implemente.

O pgactivelink só pode resolver conflitos envolvendo uma única violação de restrição. Se um INSERT violar várias restrições UNIQUE, você deverá implementar estratégias adicionais de resolução de conflitos.

INSERTs que violam várias restrições UNIQUE

Um conflito INSERT/INSERT pode violar várias restrições UNIQUE, incluindo a CHAVE PRIMÁRIA. O pgactivelink só pode lidar com conflitos que envolvam uma única restrição UNIQUE. Quando os conflitos violarem várias restrições UNIQUE, o operador apply falhará e retornará o seguinte erro:

```
multiple unique constraints violated by remotely INSERTed tuple.
```

Em versões mais antigas, essa situação gerava um erro de “conflito de exclusividade divergente”.

Para resolver esses conflitos, você deve executar uma ação manual. Use DELETE para excluir as tuplas locais conflitantes ou UPDATE para atualizá-las e remover os conflitos com a nova tupla remota. Esteja ciente de que talvez você precise resolver várias tuplas conflitantes. No momento, o pgactivelink não oferece nenhuma funcionalidade integrada para ignorar, descartar ou mesclar tuplas que violem várias restrições únicas.

> **Note**
>
> Para ter mais informações, consulte UPDATEs que violam várias restrições UNIQUE.

Conflitos UPDATE/UPDATE

Esse conflito ocorre quando dois nós modificam simultaneamente a mesma tupla sem alterar a CHAVE PRIMÁRIA. O pgactivelink resolve esses conflitos usando a lógica de última atualização prevalece ou seu manipulador de conflitos personalizado, se definido. Uma CHAVE PRIMÁRIA é essencial para correspondência de tuplas e resolução de conflitos. Para tabelas sem CHAVE PRIMÁRIA, o pgactivelink rejeita as operações UPDATE com o seguinte erro:

```
Cannot run UPDATE or DELETE on table (tablename) because it does not have a primary key.
```

Conflitos UPDATE na CHAVE PRIMÁRIA

A pgactive tem limitações para lidar com atualizações de CHAVE PRIMÁRIA. Embora você possa realizar a operação UPDATE em uma CHAVE PRIMÁRIA, a pgactive não consegue resolver conflitos automaticamente usando a lógica de última atualização prevalece para essas operações. Você deve garantir que as atualizações da CHAVE PRIMÁRIA não entrem em conflito com os valores existentes. Se ocorrerem conflitos durante as atualizações da CHAVE PRIMÁRIA, eles se tornarão conflitos divergentes e exigirão intervenção manual. Para ter mais informações sobre como lidar com essas situações, consulte Conflitos divergentes.

UPDATEs que violam várias restrições UNIQUE

O pgactivelink não consegue aplicar a resolução de conflitos de última atualização prevalece quando um UPDATE recebido viola várias restrições UNIQUE ou valores de CHAVE PRIMÁRIA. Esse comportamento é semelhante às operações INSERT com várias violações de restrição. Essas situações criam conflitos divergentes que exigem intervenção manual. Para obter mais informações, consulte Conflitos divergentes.

Conflitos UPDATE/DELETE

Esses conflitos ocorrem quando um nó executa UPDATE em uma linha e outro nó executa simultaneamente DELETE nessa linha. Nesse caso, ocorre um conflito UPDATE/DELETE na reprodução. A solução é descartar qualquer UPDATE que chegue após um DELETE, a menos que seu manipulador de conflitos personalizado especifique o contrário.

O pgactivelink requer uma CHAVE PRIMÁRIA para combinar tuplas e resolver conflitos. Para tabelas sem CHAVE PRIMÁRIA, ele rejeita operações DELETE com o seguinte erro:

```
Cannot run UPDATE or DELETE on table (tablename) because it does not have a primary key.
```

> **Note**
>
> O pgactivelink não consegue distinguir entre os conflitos UPDATE/DELETE e INSERT/UPDATE. Em ambos os casos, um UPDATE afeta uma linha inexistente. Devido à replicação assíncrona e à falta de ordem de reprodução entre os nós, o pgactivelink não consegue determinar se o UPDATE é para uma nova linha (INSERT ainda não recebido) ou uma linha excluída. Em ambos os cenários, o pgactivelink descarta o UPDATE.

Conflitos INSERT/UPDATE

Esse conflito pode ocorrer em ambientes com vários nós. Isso acontece quando um nó insere uma linha, um segundo nó a atualiza e um terceiro nó recebe o UPDATE antes do INSERT original. Por padrão, o pgactivelink resolve esses conflitos descartando o UPDATE, a menos que seu acionador de conflito personalizado especifique o contrário. Esteja ciente de que esse método de resolução pode provocar inconsistências de dados entre os nós. Para ter mais informações sobre cenários semelhantes e o respectivo tratamento, consulte Conflitos UPDATE/DELETE.

Conflitos DELETE/DELETE

Esse conflito ocorre quando dois nós diferentes excluem simultaneamente a mesma tupla. O pgactivelink considera esses conflitos não prejudiciais porque as duas operações DELETE têm o mesmo resultado final. Nesse cenário, o pgactivelink ignora com segurança uma das operações DELETE sem afetar a consistência de dados.

Conflitos de restrição de chave estrangeira

As restrições de CHAVE ESTRANGEIRA podem causar conflitos ao aplicar transações remotas aos dados locais existentes. Esses conflitos geralmente ocorrem quando as transações são aplicadas em uma sequência diferente da ordem lógica nos nós de origem.

Por padrão, a pgactive aplica alterações com session_replication_role como `replica`, o que ignora as verificações de chave estrangeira durante a replicação. Em configurações ativas-ativas, isso pode provocar violações de chave estrangeira. A maioria das violações é temporária e resolvida quando a replicação é concluída. Mas pode haver chaves estrangeiras pendentes porque a pgactive não permite o bloqueio de linhas entre nós.

Esse comportamento é inerente aos sistemas ativos-ativos assíncronos com tolerância à partição. Por exemplo, o nó A pode inserir uma nova linha secundária enquanto o nó B exclui simultaneamente a linha principal. O sistema não consegue impedir esse tipo de modificação simultânea entre os nós.

Para minimizar conflitos de chave estrangeira, recomendamos o seguinte:

- Limite as relações de chave estrangeira a entidades estreitamente relacionadas.
- Modifique as entidades relacionadas por meio de um único nó quando possível.
- Escolha entidades que raramente exijam modificação.
- Implemente o controle de simultaneidade em nível de aplicação para modificações.

Conflitos de restrição de exclusão

O pgactivelink não permite restrições de exclusão e limita a criação de restrições.

> **Note**
>
> Se você converter um banco de dados independente existente em um banco de dados pgactivelink, remova manualmente todas as restrições de exclusão.

Em um sistema assíncrono distribuído, não é possível garantir que nenhum conjunto de linhas viole a restrição. Isso ocorre porque todas as transações em nós diferentes são totalmente isoladas. As restrições de exclusão podem gerar deadlocks de reprodução, impedindo que a reprodução progrida de um nó para outro devido a violações da restrição de exclusão.

Se você forçar o pgactivelink a criar uma restrição de exclusão ou se não remover as existentes ao converter um banco de dados independente em pgactivelink, é provável que a replicação seja interrompida. Para restaurar o progresso da replicação, remova ou altere as tuplas locais que estão em conflito com uma tupla remota de entrada para que a transação remota possa ser aplicada.

Conflitos de dados globais

Ao usar o pgactivelink, conflitos podem ocorrer quando os nós têm diferentes dados globais do sistema geral do PostgreSQL, como funções. Esses conflitos podem fazer com que as operações, principalmente de DDL, sejam bem-sucedidas e confirmadas em um nó, mas falhem ao serem aplicadas a outros nós.

Se um usuário existir em um nó e não em outro, podem ocorrer problemas de replicação:

- O Node1 tem um usuário chamado `fred`, mas esse usuário não existe no Node2.
- Quando `fred` cria uma tabela no Node1, a tabela é replicada com `fred` como proprietário.
- Quando esse comando de DDL é aplicado ao Node2, ele falha porque o usuário `fred` não existe.
- Essa falha gera um ERRO nos logs do PostgreSQL no Node2 e incrementa o contador `pgactive.pgactive_stats.nr_rollbacks`.

Resolução: crie o usuário `fred` no Node2. O usuário não precisa de permissões idênticas, mas deve existir em ambos os nós.

Se uma tabela existir em um nó e não em outro, as operações de modificação de dados vão falhar:

- O Node1 tem uma tabela chamada `foo` que não existe no Node2.
- Qualquer operação de DML na tabela `foo` no Node1 falhará quando replicada para o Node2.

Resolução: crie a tabela `foo` no Node2 com a mesma estrutura.

> **Note**
>
> No momento, o pgactivelink não replica comandos CREATE USER ou operações de DDL. A replicação de DDL está programada para uma versão futura.

Conflitos de bloqueio e interrupções decorrentes de deadlock

Como os processos apply da pgactive operam como sessões normais de usuário, eles seguem as regras padrão de bloqueio de linhas e tabelas. Isso pode fazer com que os processos apply do pgactivelink aguardem bloqueios mantidos por transações de usuário ou por outros processos apply.

Os seguintes tipos de bloqueio podem afetar os processos apply:

- Bloqueio explícito em nível de tabela (LOCK TABLE…) por sessão de usuário
- Bloqueio explícito em nível de linha (SELECT… FOR UPDATE/FOR SHARE) por sessão de usuário
- Bloqueio de chaves estrangeiras
- Bloqueio implícito devido operações de UPDATE, INSERT ou DELETE de linhas, seja da atividade local ou do operador apply de outros servidores

Deadlocks podem ocorrer entre:

- Um processo apply do pgactivelink e uma transação do usuário
- Dois processos apply

Quando ocorrem deadlocks, o detector de deadlock do PostgreSQL encerra uma das transações problemáticas. Se o processo do operador apply do pgactivelink for encerrado, ele fará uma nova tentativa automaticamente e em geral terá êxito.

> **Note**
>
> - Esses problemas são temporários e normalmente não exigem intervenção do administrador. Se um processo apply for bloqueado por um longo período devido ao bloqueio de uma sessão de usuário ociosa, você poderá encerrar a sessão do usuário para retomar a replicação. Essa situação é semelhante a quando um usuário mantém um bloqueio longo que afeta a sessão de outro usuário.
> - Para identificar atrasos na reprodução relacionados ao bloqueio, habilite o recurso `log_lock_waits` no PostgreSQL.

Conflitos divergentes

Conflitos divergentes ocorrem quando dados que deveriam ser idênticos entre os nós diferem inesperadamente. Embora esses conflitos não devam acontecer, nem todos podem ser evitados de forma confiável na implementação atual.

> **Note**
>
> A modificação da CHAVE PRIMÁRIA de uma linha poderá causar conflitos divergentes se outro nó alterar a chave da mesma linha antes de todos os nós processarem a alteração. Evite alterar as chaves primárias ou restrinja as alterações a um nó designado. Para obter mais informações, consulte the section called “Conflitos UPDATE na CHAVE PRIMÁRIA ”.

Conflitos divergentes envolvendo dados de linha geralmente exigem intervenção do administrador. Para resolver esses conflitos, você deve ajustar manualmente os dados em um nó para corresponder a outro e, ao mesmo tempo, desabilitar temporariamente a replicação usando `pgactive.pgactive_do_not_replicate`. Esses conflitos não devem ocorrer quando você usa a pgactive conforme documentado e evita configurações ou funções marcadas como inseguras.

Como administrador, você deve resolver esses conflitos manualmente.
Dependendo do tipo de conflito, você precisará usar opções avançadas, como `pgactive.pgactive_do_not_replicate`. Use essas opções com cuidado, pois o uso indevido pode piorar a situação. Devido à variedade de possíveis conflitos, não podemos fornecer instruções gerais de resolução.

Conflitos divergentes ocorrem quando dados que deveriam ser idênticos entre diferentes nós diferem inesperadamente. Embora esses conflitos não devam acontecer, nem todos podem ser evitados de forma confiável na implementação atual.

Evitar ou permitir conflitos

Na maioria dos casos, você pode usar o design de aplicação adequado para evitar conflitos ou tornar a aplicação tolerante a conflitos.

Os conflitos só ocorrem quando acontecem operações simultâneas em vários nós. Como evitar conflitos:

- Grave em apenas um nó.

- Grave em subconjuntos de banco de dados independentes em cada nó (por exemplo, atribua um esquema separado a cada nó).

Quanto a conflitos INSERT versus INSERT, use sequências globais para evitar conflitos por completo.

Se os conflitos não forem aceitáveis para seu caso de uso, considere implementar o bloqueio distribuído no nível da aplicação. Muitas vezes, a melhor abordagem é projetar a aplicação para funcionar com os mecanismos de resolução de conflitos da pgactive, em vez de tentar evitar todos os conflitos. Para obter mais informações, consulte the section called “Tipos de conflito”.

Registro em log de conflitos

O pgactivelink registra incidentes de conflito na tabela `pgactive.pgactive_conflict_history` para ajudar você a diagnosticar e lidar com conflitos ativos-ativos. O registro em log de conflitos nessa tabela ocorre somente quando você define `pgactive.log_conflicts_to_table` como verdadeiro. A extensão pgactive também registra em log conflitos no arquivo de log do PostgreSQL quando log_min_messages é definido como `LOG` ou `lower`, independentemente da configuração `pgactive.log_conflicts_to_table`.

Use a tabela do histórico de conflitos para:

- Medir a frequência com que a aplicação cria conflitos.
- Identificar onde os conflitos ocorrem.
- Melhorar a aplicação para reduzir as taxas de conflito.
- Detectar casos em que as resoluções de conflito não produzem os resultados desejados.
- Determinar onde você precisa de acionadores de conflito definidos pelo usuário ou alterações no design da aplicação.

Quanto a conflitos de linha, você pode, opcionalmente, registrar em log valores de linha. Isso é controlado pela configuração `pgactive.log_conflicts_to_table`. Observe que:

- Essa opção é global para todo o banco de dados.
- Não há controle por tabela sobre o registro em log de valores de linha.
- Nenhum limite é aplicado a números de campo, elementos de matriz ou comprimentos de campo.
- A habilitação desse recurso pode não ser aconselhável se você trabalha com linhas de vários megabytes que podem desencadear conflitos.

Como a tabela do histórico de conflitos contém dados de todas as tabelas do banco de dados (cada uma com esquemas possivelmente diferentes), os valores das linhas registradas em log são armazenados como campos JSON. O JSON é criado usando `row_to_json`, de forma semelhante a chamá-lo diretamente do SQL. O PostgreSQL não oferece uma função `json_to_row`, então você precisará de um código específico de tabela (em PL/pgSQL, PL/Python, PL/Perl etc.) para reconstruir uma tupla de tipo composto por meio do JSON registrado em log.

> **Note**
>
> O suporte a conflitos definidos pelo usuário está programado como um futuro recurso de extensão.

Noções básicas sobre o esquema da pgactive

O esquema da pgactive gerencia a replicação ativa-ativa no RDS para PostgreSQL. Esse esquema contém tabelas que armazenam informações de configuração e status da replicação.

> **Note**
>
> O esquema da pgactive está evoluindo e está sujeito a alterações. Não modifique os dados nessas tabelas diretamente.

As tabelas principais no esquema da pgactive incluem:

- `pgactive_nodes`: armazena informações sobre os nós no grupo de replicação ativa-ativa.
- `pgactive_connections`: armazena detalhes da conexão para cada nó.

pgactive_nodes

A pgactive_nodes armazena informações sobre os nós que participam do grupo de replicação ativa-ativa.

Coluna	Tipo	Collation (Agrupamento)	Anulável	Padrão
node_sysid	text	–	não nulo	–
node_timeline	oid	–	não nulo	–
node_dboid	oid	–	não nulo	–
node_status	char	–	não nulo	–
node_name	text	–	não nulo	–
node_dsn	text	–	não nulo	–
node_init_from_dsn	text	–	não nulo	–
node_read_only	boolean	–	–	false
node_seq_id	smallint	–	não nulo	–

node_sysid

ID exclusivo para um nó, gerado durante `pgactive_create_group` ou `pgactive_join_group`.

node_status

Prontidão do nó:

- b: iniciando a configuração
- i: inicializando
- c: recuperando
- o: criando slots de saída
- r: pronto
- k: encerrado

> **Note**
>
> Essa coluna não indica se um nó está conectado ou desconectado.

node_name

Nome de nó exclusivo fornecido pelo usuário.

node_dsn

String de conexão ou nome de mapeamento do usuário

node_init_from_dsn

DSN com base no qual esse nó foi criado.

pgactive_connection

A pgactive_connections armazena detalhes da conexão para cada nó.

Coluna	Tipo	Collation (Agrupamento)	Anulável	Padrão	
conn_sysid	text	nenhuma	não nulo	nenhuma	
conn_timeline	oid	nenhuma	não nulo	nenhuma	
conn_dboid	oid	nenhuma	não nulo	nenhuma	
conn_dsn	text	nenhuma	não nulo	nenhuma	
conn_apply_delay	integer	nenhuma	nenhuma	nenhuma	
conn_replication_sets	text	nenhuma	nenhuma	nenhuma	

conn_sysid

Identificador do nó ao qual essa entrada se refere.

conn_dsn

O mesmo que `node_dsn` de pgactive.pgactive_nodes.

conn_apply_delay

Se definido, milissegundos de espera antes de aplicar cada transação por meio do nó remoto. Principalmente para depuração. Se nulo, o padrão global se aplica.

Trabalhar com conjuntos de replicações

Os conjuntos de replicações determinam quais tabelas incluir ou excluir das operações de replicação. Por padrão, todas as tabelas são replicadas, a menos que você especifique o contrário usando as seguintes funções:

- `pgactive_exclude_table_replication_set()`: exclui tabelas especificadas da replicação.
- `pgactive_include_table_replication_set()`: inclui tabelas especificadas na replicação.

> **Note**
>
> Antes de configurar os conjuntos de replicações, considere o seguinte:
>
> - Você pode configurar a inclusão ou exclusão da tabela somente depois de executar `pgactive_create_group()`, mas antes de `pgactive_join_group()`.
> - Depois de usar `pgactive_exclude_table_replication_set()`, você não pode usar `pgactive_include_table_replication_set()`.
> - Depois de usar `pgactive_include_table_replication_set()`, você não pode usar `pgactive_exclude_table_replication_set()`.

O sistema trata as tabelas recém-criadas de forma diferente com base na sua configuração inicial:

- Se você excluiu tabelas: todas as outras tabelas criadas depois de `pgactive_join_group()` serão automaticamente incluídas na replicação.
- Se você incluiu tabelas: todas as outras tabelas criadas depois de `pgactive_join_group()` serão automaticamente excluídas da replicação.

Para visualizar a configuração do conjunto de replicações de uma tabela específica, use a função `pgactive.pgactive_get_table_replication_sets()`.

Referência de funções da pgactive

Abaixo, você pode encontrar uma lista de funções da pgactive com os respectivos parâmetros, valores de retorno e observações práticas de uso para ajudar você a usá-las de forma eficaz:

get_last_applied_xact_info

Recupera as informações da última transação aplicada para um nó especificado.

Arguments (Argumentos)

- sysid (text): OID de carimbo de data/hora
- dboid (OID)

Tipo de retorno

Ela registra o seguinte:

- last_applied_xact_id (OID)
- last_applied_xact_committs (carimbo de data/hora com fuso horário)
- last_applied_xact_at (carimbo de data/hora com fuso horário)

Observações de uso

Use esta função para recuperar as informações da última transação aplicada para um nó especificado.

pgactive_apply_pause

Pausa o processo apply da replicação.

Arguments (Argumentos)

Nenhum

Tipo de retorno

boolean

Observações de uso

Chame esta função para pausar o processo apply da replicação.

pgactive_apply_resume

Retoma o processo apply da replicação.

Arguments (Argumentos)

Nenhum

Tipo de retorno

nulo

Observações de uso

Chame essa função para retomar o processo apply da replicação.

pgactive_is_apply_paused

Verifica se o processo apply da replicação está em pausa no momento.

Arguments (Argumentos)

Nenhum

Tipo de retorno

boolean

Observações de uso

Use essa função para verificar se o processo apply da replicação está em pausa no momento.

pgactive_create_group

Cria um grupo da pgactive convertendo um banco de dados independente no nó inicial.

Arguments (Argumentos)

- node_name (text)
- node_dsn (text)
- apply_delay integer DEFAULT NULL::integer - replication_sets text[] DEFAULT ARRAY[‘default’::text]

Tipo de retorno

nulo

Observações de uso

Cria um grupo da pgactive convertendo um banco de dados independente no nó inicial. A função realiza verificações de integridade antes de transformar o nó em um nó pgactive. Antes de usar essa função, o cluster do PostgreSQL deve ter `max_worker_processes` suficientes para comportar operadores em segundo plano da pgactive.

pgactive_detach_nodes

Remove os nós especificados do grupo da pgactive.

Arguments (Argumentos)

- p_nodes (text[])

Tipo de retorno

nulo

Observações de uso

Use essa função para remover os nós especificados do grupo da pgactive.

pgactive_exclude_table_replication_set

Exclui uma tabela específica da replicação.

Arguments (Argumentos)

- p_relation (regclass)

Tipo de retorno

nulo

Observações de uso

Use essa função para excluir uma tabela específica da replicação.

pgactive_get_replication_lag_info

Recupera informações detalhadas do atraso da replicação, incluindo detalhes do nó, status do WAL e valores de LSN.

Arguments (Argumentos)

Nenhum

Tipo de retorno

Registro SETOF: node_name text, node_sysid text, application_name text, slot_name text, active boolean, active_pid integer, pending_wal_decoding bigint (tamanho aproximado do WAL em bytes a ser decodificado no nó emissor), pending_wal_to_apply bigint (tamanho aproximado do WAL em bytes a ser decodificado no nó receptor), restart_lsn pg_lsn, confirmed_flush_lsn pg_lsn, sent_lsn pg_lsn, write_lsn pg_lsn, flush_lsn pg_lsn, replay_lsn pg_lsn

Observações de uso

Chame esta função para recuperar informações sobre o atraso da replicação, incluindo detalhes do nó, status do WAL e valores de LSN.

pgactive_get_stats

Recupera estatísticas de replicação da pgactive.

Arguments (Argumentos)

Nenhum

Tipo de retorno

Registro SETOF: rep_node_id oid, rilocalid oid, riremoteid text, nr_commit bigint, nr_rollback bigint, nr_insert bigint, nr_insert_conflict bigint, nr_update bigint, nr_update_conflict bigint, nr_delete bigint, nr_delete_conflict bigint, nr_disconnect bigint

Observações de uso

Use esta função para recuperar estatísticas de replicação da pgactive.

pgactive_get_table_replication_sets

Obtém a configuração do conjunto de replicações para uma relação específica.

Arguments (Argumentos)

- relation (regclass)

Tipo de retorno

Registro SETOF

Observações de uso

Chame esta função para obter a configuração do conjunto de replicações para uma relação específica.

pgactive_include_table_replication_set

Inclui uma tabela específica na replicação.

Arguments (Argumentos)

- p_relation (regclass)

Tipo de retorno

nulo

Observações de uso

Use esta função para incluir uma tabela específica na replicação.

pgactive_join_group

Adiciona um nó a um grupo existente da pgactive.

Arguments (Argumentos)

- node_name (text)
- node_dsn (text)
- join_using_dsn (text)
- apply_delay (integer, optional)
- replication_sets (text[], default: ['default'])
- bypass_collation_check (boolean, default: false)
- bypass_node_identifier_creation (boolean, default: false)
- bypass_user_tables_check (boolean, default: false)

Tipo de retorno

nulo

Observações de uso

Chame essa função para adicionar um nó a um grupo existente da pgactive. O cluster do PostgreSQL deve ter max_worker_processes suficientes para os operadores em segundo plano da pgactive.

pgactive_remove

Remove todos os componentes da pgactive do nó local.

Arguments (Argumentos)

- force (boolean, default: false)

Tipo de retorno

nulo

Observações de uso

Chame esta função para remover todos os componentes da pgactive do nó local.

pgactive_snowflake_id_nextval

Gera valores de sequência exclusivos específicos do nó.

Arguments (Argumentos)

- regclass

Tipo de retorno

bigint

Observações de uso

Use esta função para gerar valores de sequência exclusivos específicos do nó.

pgactive_update_node_conninfo

Atualiza as informações de conexão de um nó da pgactive.

Arguments (Argumentos)

- node_name_to_update (text)
- node_dsn_to_update (text)

Tipo de retorno

nulo

Observações de uso

Usa esta função para atualizar as informações de conexão de um nó da pgactive.

pgactive_wait_for_node_ready

Monitora o progresso das operações de criação de grupo ou junção.

Arguments (Argumentos)

- timeout (integer, default: 0)
- progress_interval (integer, default: 60)

Tipo de retorno

nulo

Observações de uso

Chame esta função para monitorar o progresso das operações de criação de grupo ou junção.

Lidar com conflitos na replicação ativa-ativa

A extensão `pgactive` funciona por banco de dados e não por cluster. Cada instância de banco de dados que usa `pgactive` é uma instância independente e pode aceitar alterações de dados de qualquer fonte. Quando uma alteração é enviada a uma instância de banco de dados, o PostgreSQL a confirma localmente e depois usa `pgactive` para replicar a alteração de forma assíncrona para outras instâncias de banco de dados. Quando duas instâncias de banco de dados do PostgreSQL atualizam o mesmo registro quase ao mesmo tempo, pode ocorrer um conflito.

A extensão `pgactive` fornece mecanismos para detecção de conflitos e resolução automática. Ela rastreia o carimbo de data/hora em que a transação foi confirmada em ambas as instâncias de banco de dados e aplica automaticamente a alteração com o carimbo de data/hora mais

recente. A extensão `pgactive` também registra em log quando ocorre um conflito na tabela `pgactive.pgactive_conflict_history`.

O `pgactive.pgactive_conflict_history` continuará crescendo. Talvez você queira definir uma política de limpeza. Isso pode ser feito excluindo alguns registros regularmente ou definindo um esquema de particionamento para essa relação (e depois separando, descartando e truncando as partições de interesse). Para implementar a política de limpeza regularmente, uma opção é usar a extensão `pg_cron`. Veja as informações a seguir de um exemplo para a tabela de histórico `pg_cron`, Agendar manutenção com a extensão pg_cron do PostgreSQL.

Lidar com sequências na replicação ativa-ativa

Uma instância de banco de dados do RDS para PostgreSQL com a extensão `pgactive` usa dois mecanismos de sequência diferentes para gerar valores exclusivos.

Sequências globais

Para usar uma sequência global, crie uma sequência local com a instrução `CREATE SEQUENCE`. Use `pgactive.pgactive_snowflake_id_nextval(seqname)` em vez de `usingnextval(seqname)` para obter o próximo valor exclusivo da sequência.

O exemplo a seguir cria uma rede global.

```
app=> CREATE TABLE gstest (
      id bigint primary key,
      parrot text
    );
```

```
app=>CREATE SEQUENCE gstest_id_seq OWNED BY gstest.id;
```

```
app=> ALTER TABLE gstest \
      ALTER COLUMN id SET DEFAULT \
      pgactive.pgactive_snowflake_id_nextval('gstest_id_seq');
```

Sequências particionadas

Em sequências divididas ou particionadas, uma sequência normal do PostgreSQL é usada em cada nó. Cada sequência é incrementada na mesma quantidade e começa com diferentes deslocamentos. Por exemplo, com a etapa 100, o nó 1 gera a sequência como 101, 201, 301 e assim por diante, e o

nó 2 gera a sequência como 102, 202, 302 e assim por diante. Esse esquema funciona bem mesmo que os nós não possam se comunicar por longos períodos, mas exige que o designer especifique um número máximo de nós ao estabelecer o esquema e requer configuração por nó. Erros podem facilmente levar à sobreposição de sequências.

É relativamente simples configurar essa abordagem com `pgactive` criando a sequência desejada em um nó da seguinte maneira:

```
CREATE TABLE some_table (generated_value bigint primary key);
```

```
app=> CREATE SEQUENCE some_seq INCREMENT 100 OWNED BY some_table.generated_value;
```

```
app=> ALTER TABLE some_table ALTER COLUMN generated_value SET DEFAULT
 nextval('some_seq');
```

Em seguida, chame `setval` em cada nó para fornecer um valor inicial de deslocamento diferente da forma a seguir.

```
app=>
-- On node 1
SELECT setval('some_seq', 1);

-- On node 2
SELECT setval('some_seq', 2);
```

Reduzir o inchaço em tabelas e índices com a extensão pg_repack

É possível usar a extensão `pg_repack` para remover a sobrecarga de tabelas e índices como uma alternativa a `VACUUM FULL`. Esta extensão é compatível com o RDS para PostgreSQL versões 9.6.3 e posteriores. Para ter mais informações sobre a extensão `pg_repack` e a recriação da tabela inteira, consulte a documentação de projetos do GitHub.

Ao contrário de `VACUUM FULL`, a extensão `pg_repack` requer um bloqueio exclusivo (AccessExclusiveLock) somente por um curto período durante a operação de recriação da tabela nos seguintes casos:

- Criação inicial da tabela de logs: uma tabela de logs é criada para registrar as alterações que ocorrem durante a cópia inicial dos dados, conforme mostrado no seguinte exemplo:

```
postgres=>\dt+ repack.log_*
List of relations
-[ RECORD 1 ]-+----------
Schema        | repack
Name          | log_16490
Type          | table
Owner         | postgres
Persistence   | permanent
Access method | heap
Size          | 65 MB
Description   |
```

- Fase final de troca e descarte.

Para o restante da operação de recriação, ela só precisa de um bloqueio `ACCESS SHARE` na tabela original para copiar as linhas dela para a nova tabela. Isso ajuda as operações INSERT, UPDATE e DELETE a prosseguir normalmente.

Recomendações

As recomendações a seguir se aplicam ao remover a sobrecarga das tabelas e dos índices usando a extensão `pg_repack`:

- Realize a recriação fora do horário comercial ou durante uma janela de manutenção para minimizar o impacto na performance de outras atividades do banco de dados.

- Monitore atentamente as sessões de bloqueio durante a atividade de recriação e garanta que não haja nenhuma atividade na tabela original que possa bloquear `pg_repack`, especificamente durante a fase final de troca e descarte, quando ela precisa de um bloqueio exclusivo na tabela original. Para ter mais informações, consulte Como identificar o que está bloqueando uma consulta.

  Ao ver uma sessão de bloqueio, é possível encerrá-la usando o comando a seguir após uma análise cuidadosa. Isso ajuda na continuação da `pg_repack` para concluir a recriação:

  ```
  SELECT pg_terminate_backend(pid);
  ```

- Ao aplicar as alterações acumuladas da tabela de logs de `pg_repack's` em sistemas com uma taxa de transação muito alta, o processo de aplicação pode não ser capaz de acompanhar a taxa de alterações. Nesses casos, `pg_repack` não conseguiria concluir o processo de aplicação. Para ter mais informações, consulte Monitorar a nova tabela durante a recriação. Se os índices estiverem muito sobrecarregados, uma solução alternativa será realizar uma recriação somente de índices. Isso também ajuda os ciclos de limpeza do índice do VACUUM a terminar mais rapidamente.

  É possível ignorar a fase de limpeza do índice usando o VACUUM manual do PostgreSQL versão 12, e ela é ignorada automaticamente durante o autovacuum de emergência do PostgreSQL versão 14. Isso ajuda o VACUUM a ser concluído mais rapidamente sem remover a sobrecarga do índice e serve apenas para situações de emergência, como evitar o VACUUM de encapsulamento. Para ter mais informações, consulte Evitar a sobrecarga nos índices no Guia do usuário do Amazon Aurora.

Pré-requisitos

- A tabela deve ter PRIMARY KEY ou a restrição UNIQUE não nula.
- A versão da extensão deve ser a mesma para o cliente e para o servidor.
- Garanta que a instância do RDS tenha mais `FreeStorageSpace` do que o tamanho total da tabela sem a sobrecarga. Por exemplo, pense no tamanho total da tabela, incluindo TOAST e índices, como 2 TB, e a sobrecarga total na tabela, como 1 TB. O `FreeStorageSpace` necessário deve ser maior do que o valor exibido pelo seguinte cálculo:

  `2TB (Table size) - 1TB (Table bloat) = 1TB`

É possível usar a consulta a seguir para conferir o tamanho total da tabela e usar `pgstattuple` para gerar a sobrecarga. Para ter mais informações, consulte Diagnosticar a sobrecarga na tabela e no índice no Guia do usuário do Amazon Aurora.

```
SELECT pg_size_pretty(pg_total_relation_size('table_name')) AS total_table_size;
```

Esse espaço é recuperado após a conclusão da atividade.

- Garanta que a instância do RDS tenha capacidade computacional e de E/S suficientes para lidar com a operação de recriação. Pense na possibilidade de aumentar a escala da classe de instância verticalmente para conseguir o equilíbrio ideal de performance.

Como usar a extensão **`pg_repack`**

1. Instale a extensão `pg_repack` na instância de banco de dados RDS for PostgreSQL executando o comando a seguir.

   ```
   CREATE EXTENSION pg_repack;
   ```

2. Execute os comandos a seguir para conceder acesso de gravação para recriar as tabelas de logs temporárias criadas pelo `pg_repack`.

   ```
   ALTER DEFAULT PRIVILEGES IN SCHEMA repack GRANT INSERT ON TABLES TO PUBLIC;
   ALTER DEFAULT PRIVILEGES IN SCHEMA repack GRANT USAGE, SELECT ON SEQUENCES TO
    PUBLIC;
   ```

3. Conecte-se ao banco de dados usando o utilitário de cliente `pg_repack`. Use uma conta que tenha privilégios `rds_superuser`. Como exemplo, suponha que a função `rds_test` tenha os privilégios `rds_superuser`. A sintaxe a seguir realiza a `pg_repack` para tabelas completas, incluindo todos os índices da tabela no banco de dados `postgres`.

   ```
   pg_repack -h db-instance-name.111122223333.aws-region.rds.amazonaws.com -U rds_test
    -k postgres
   ```

   > **Note**
   >
   > É necessário se conectar usando a opção -k. Não há suporte para a opção -a.

A resposta do cliente `pg_repack` fornece informações sobre as tabelas recriadas na instância de banco de dados.

```
INFO: repacking table "pgbench_tellers"
INFO: repacking table "pgbench_accounts"
INFO: repacking table "pgbench_branches"
```

4. A sintaxe a seguir recria uma única tabela `orders` incluindo índices no banco de dados `postgres`.

```
pg_repack -h db-instance-name.111122223333.aws-region.rds.amazonaws.com -U rds_test
 --table orders -k postgres
```

A sintaxe a seguir recria somente os índices da tabela `orders` no banco de dados `postgres`.

```
pg_repack -h db-instance-name.111122223333.aws-region.rds.amazonaws.com -U rds_test
 --table orders --only-indexes -k postgres
```

Monitorar a nova tabela durante a recriação

- O tamanho do banco de dados é aumentado pelo tamanho total da tabela menos a sobrecarga, até a fase de troca e descarte da recriação. É possível monitorar a taxa de crescimento do tamanho do banco de dados, calcular a velocidade da recriação e estimar aproximadamente o tempo necessário para concluir a transferência inicial dos dados.

  Por exemplo, pense no tamanho total da tabela como 2 TB, o tamanho do banco de dados como 4 TB e a sobrecarga total na tabela como 1 TB. O valor do tamanho total do banco de dados exibido pelo cálculo no final da operação de recriação é o seguinte:

  `2TB (Table size) + 4 TB (Database size) - 1TB (Table bloat) = 5TB`

  É possível calcular aproximadamente a velocidade da operação de recriação criando uma amostra da taxa de crescimento em bytes entre dois pontos no tempo. Se a taxa de crescimento for de 1 GB por minuto, poderá levar mil minutos ou 16,6 horas aproximadamente para concluir a operação inicial de criação da tabela. Além da criação inicial da tabela, a `pg_repack` também precisa aplicar as alterações acumuladas. O tempo necessário depende da taxa de aplicação das alterações em andamento, além das alterações acumuladas.

> **Note**
>
> É possível usar a extensão `pgstattuple` para calcular a sobrecarga na tabela. Para ter mais informações, consulte pgstattuple.

- O número de linhas na tabela de logs `pg_repack's`, no esquema de recriação, representa o volume de alterações pendentes para serem aplicadas à nova tabela após o carregamento inicial.

  É possível conferir a tabela de logs `pg_repack's` em `pg_stat_all_tables` para monitorar as alterações aplicadas à nova tabela. `pg_stat_all_tables.n_live_tup` indica o número de registros pendentes a serem aplicados à nova tabela. Para ter mais informações, consulte pg_stat_all_tables.

```
postgres=>SELECT relname,n_live_tup FROM pg_stat_all_tables WHERE schemaname =
 'repack' AND relname ILIKE '%log%';

-[ RECORD 1 ]---------
relname    | log_16490
n_live_tup | 2000000
```

- É possível usar a extensão `pg_stat_statements` para descobrir o tempo gasto em cada etapa da operação de recriação. Isso é útil na preparação para aplicar a mesma operação de recriação em um ambiente de produção. É possível ajustar a cláusula `LIMIT` para estender ainda mais a saída.

```
postgres=>SELECT
     SUBSTR(query, 1, 100) query,
     round((round(total_exec_time::numeric, 6) / 1000 / 60),4)
 total_exec_time_in_minutes
 FROM
     pg_stat_statements
 WHERE
     query ILIKE '%repack%'
 ORDER BY
     total_exec_time DESC LIMIT 5;

 query                                                                |
 total_exec_time_in_minutes
```

```
------------------------------------------------------------------------
+---------------------------
 CREATE UNIQUE INDEX index_16493 ON repack.table_16490 USING btree (a) |
        6.8627
 INSERT INTO repack.table_16490 SELECT a FROM ONLY public.t1           |
        6.4150
 SELECT repack.repack_apply($1, $2, $3, $4, $5, $6)                    |
        0.5395
 SELECT repack.repack_drop($1, $2)                                     |
        0.0004
 SELECT repack.repack_swap($1)                                         |
        0.0004
(5 rows)
```

A recriação é uma operação extraordinária, portanto, a tabela original não é afetada e não prevemos nenhum desafio inesperado que exija a recuperação da tabela original. Se a recriação falhar inesperadamente, você deverá inspecionar a causa do erro e resolvê-lo.

Depois que o problema for resolvido, descarte e recrie a extensão `pg_repack` no banco de dados em que a tabela existe e repita a etapa `pg_repack`. Além disso, a disponibilidade de recursos computacionais e a acessibilidade simultânea da tabela desempenham um papel crucial na conclusão oportuna da operação de recriação.

Atualizar e usar a extensão PLV8

O PLV8 é uma extensão de linguagem Javascript confiável para o PostgreSQL. Você pode usá-lo para procedimentos armazenados, gatilhos e outros códigos processuais que podem ser chamados do SQL. Essa extensão de idioma é compatível com todas as versões atuais do PostgreSQL.

Se você usar o PLV8 e atualizar o PostgreSQL para uma nova versão do PLV8, você poderá aproveitar imediatamente a nova extensão. Tome as medidas a seguir para sincronizar os metadados do catálogo com a nova versão do PLV8. Estas etapas são opcionais, mas recomendamos fortemente que você as conclua para evitar avisos de incompatibilidade dos metadados.

O processo de atualização elimina todas as suas funções PLV8 existentes. Portanto, recomendamos que você crie um snapshot da sua instância de banco de dados do RDS para PostgreSQL antes de atualizar. Para ter mais informações, consulte Criar um snapshot de banco de dados para uma instância de banco de dados single-AZ para o Amazon RDS.

> **Important**
>
> A partir da versão 18 do PostgreSQL, o Amazon RDS para PostgreSQL descontinuará as extensões `plcoffee` e `plls` do PostgreSQL. Recomendamos que você pare de usar o CoffeeScript e o LiveScript em suas aplicações para garantir um caminho de atualização para futuras atualizações de versão do mecanismo.

Para sincronizar os metadados do catálogo com uma nova versão do PLV8

1. Verifique se você precisa atualizar. Para fazer isso, execute o comando a seguir enquanto conectado à instância.

   ```
   SELECT * FROM pg_available_extensions WHERE name IN ('plv8','plls','plcoffee');
   ```

   Se os resultados contiverem valores para uma versão instalada que seja um número menor que a versão padrão, continue com este procedimento para atualizar suas extensões. Por exemplo, o seguinte conjunto de resultados indica que você deve atualizar.

   ```
   name    | default_version | installed_version |                    comment
   --------+-----------------+-------------------
   +-------------------------------------------------
   ```

```
plls    | 2.1.0           | 1.5.3             | PL/LiveScript (v8) trusted
 procedural language
plcoffee| 2.1.0           | 1.5.3             | PL/CoffeeScript (v8) trusted
 procedural language
plv8    | 2.1.0           | 1.5.3             | PL/JavaScript (v8) trusted
 procedural language
(3 rows)
```

2. Crie um snapshot da sua instância de banco de dados do RDS para PostgreSQL se você ainda não tiver feito isso. Você pode continuar com as seguintes etapas enquanto o snapshot estiver sendo criado.
3. Obtenha uma contagem do número de funções do PLV8 em sua instância de banco de dados para que você possa confirmar se todas estão implementadas depois da atualização. Por exemplo, a consulta SQL a seguir retorna o número de funções gravadas em plv8, plcoffee e plls.

```
SELECT proname, nspname, lanname
FROM pg_proc p, pg_language l, pg_namespace n
WHERE p.prolang = l.oid
AND n.oid = p.pronamespace
AND lanname IN ('plv8','plcoffee','plls');
```

4. Use o pg_dump para criar um arquivo de despejo de somente esquema. Por exemplo, crie um arquivo na máquina do cliente no diretório `/tmp`.

```
./pg_dump -Fc --schema-only -U master postgres >/tmp/test.dmp
```

Esse exemplo usa as seguintes opções:

- `-Fc`: formato personalizado
- --schema-only: fará o despejo apenas dos comandos necessários para criar o esquema (funções em nosso caso)
- `-U`: o nome do usuário mestre do RDS
- `database`: o nome do banco de dados de nossa instância de banco de dados

Para obter mais informações sobre pg_dump, consulte pg_dump na documentação do PostgreSQL.

5. Extraia a declaração DDL "CREATE FUNCTION" que está presente no arquivo de despejo. O exemplo a seguir usa o comando `grep` para extrair a instrução DDL que cria as funções e as salva em um arquivo. Use isso em etapas subsequentes para recriar as funções.

```
./pg_restore -l /tmp/test.dmp | grep FUNCTION > /tmp/function_list
```

   Para obter mais informações sobre pg_restore, consulte pg_restore na documentação do PostgreSQL.

6. Descarte as funções e as extensões. O exemplo a seguir descarta todos os objetos baseados em PLV8. A opção de cascata garante que qualquer dependente seja descartado.

```
DROP EXTENSION plv8 CASCADE;
```

   Se sua instância do PostgreSQL contiver objetos baseados em plcoffee ou plls, repita essa etapa para essas extensões.

7. Crie as extensões. O exemplo a seguir cria as extensões plv8, plcoffee e plls.

```
CREATE EXTENSION plv8;
CREATE EXTENSION plcoffee;
CREATE EXTENSION plls;
```

8. Crie as funções usando o arquivo de despejo e o arquivo do “driver”.

   O exemplo a seguir recria as funções que você extraiu anteriormente.

```
./pg_restore -U master -d postgres -Fc -L /tmp/function_list /tmp/test.dmp
```

9. Verifique se todas as suas funções foram recriadas usando a seguinte consulta.

```
SELECT * FROM pg_available_extensions WHERE name IN ('plv8','plls','plcoffee');
```

   A versão 2 do PLV8 adiciona a seguinte linha extra ao conjunto de resultados:

```
    proname    |  nspname   | lanname
---------------+------------+----------
 plv8_version  | pg_catalog | plv8
```

Usar PL/Rust para escrever funções do PostgreSQL na linguagem Rust

O PL/Rust é uma extensão de linguagem Rust confiável para o PostgreSQL. Você pode usá-lo para procedimentos armazenados, funções e outros códigos processuais que podem ser chamados do SQL. A extensão de linguagem PL/Rust está disponível nas seguintes versões:

- RDS para PostgreSQL 17.1 e versões 17 posteriores
- RDS para PostgreSQL 16.1 e versões 16 posteriores
- RDS para PostgreSQL 15.2-R2 e versões 15 posteriores
- RDS para PostgreSQL 14.9 e versões 14 posteriores
- RDS para PostgreSQL 13.12 e versões 13 posteriores

Para obter mais informações, consulte PL/Rust no GitHub.

Tópicos

- Configurar o PL/Rust
- Criar funções com o PL/Rust
- Usar caixas com PL/Rust
- Limitações do PL/Rust

Configurar o PL/Rust

Para instalar a extensão plrust da instância de banco de dados, adicione o plrust ao parâmetro `shared_preload_libraries` no grupo de parâmetros de banco de dados associado à instância de banco de dados. Com a extensão plrust instalada, é possível criar funções.

Para modificar o parâmetro `shared_preload_libraries`, a instância de banco de dados deve estar associada a um grupo de parâmetros personalizado. Para obter mais informações sobre como criar um grupo de parâmetros de banco de dados personalizado, consulte Grupos de parâmetros para Amazon RDS.

É possível instalar a extensão plrust usando o AWS Management Console ou a AWS CLI.

As etapas a seguir pressupõem que a instância de banco de dados esteja associada a um grupo de parâmetros de banco de dados personalizado.

Console

Instalar a extensão plrust no parâmetro **`shared_preload_libraries`**

Conclua as etapas a seguir usando uma conta que seja membro do grupo (perfil) `rds_superuser`.

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Bancos de dados.
3. Escolha o nome da instância de banco de dados para mostrar os detalhes.
4. Abra a guia Configuração da instância de banco de dados e encontre o link do grupo de parâmetros da instância de banco de dados.
5. Clique no link para abrir os parâmetros personalizados associados à instância de banco.
6. No campo Parameters (Parâmetros), digite `shared_pre` para encontrar o parâmetro **`shared_preload_libraries`**.
7. Selecione Edit parameters (Editar parâmetros) para acessar os valores das propriedades.
8. Adicione plrust à lista no campo Valores. Use uma vírgula para separar itens na lista de valores.
9. Reinicialize a instância de banco de dados para que a alteração no parâmetro `shared_preload_libraries` tenha efeito. A reinicialização inicial pode exigir mais tempo para ser concluída.
10. Quando a instância estiver disponível, verifique se plrust foi inicializado. Use `psql` para se conectar à instância de banco de dados e execute o comando a seguir.

    ```
    SHOW shared_preload_libraries;
    ```

    O resultado deve ser semelhante ao seguinte:

    ```
    shared_preload_libraries
    --------------------------
    rdsutils,plrust
    (1 row)
    ```

AWS CLI

Instalar a extensão plrust no parâmetro shared_preload_libraries

Conclua as etapas a seguir usando uma conta que seja membro do grupo (perfil) `rds_superuser`.

1. Utilize o comando modify-db-parameter-group da AWS CLI para adicionar plrust ao parâmetro `shared_preload_libraries`.

```
aws rds modify-db-parameter-group \
   --db-parameter-group-name custom-param-group-name \
   --parameters
 "ParameterName=shared_preload_libraries,ParameterValue=plrust,ApplyMethod=pending-
reboot" \
   --region aws-region
```

2. Use o comando reboot-db-instance da AWS CLI para reinicializar a instância de banco de dados e inicializar a biblioteca plrust. A reinicialização inicial pode exigir mais tempo para ser concluída.

```
aws rds reboot-db-instance \
    --db-instance-identifier your-instance \
    --region aws-region
```

3. Quando a instância estiver disponível, é possível verificar se plrust foi inicializado.
   Use `psql` para se conectar à instância de banco de dados e execute o comando a seguir.

```
SHOW shared_preload_libraries;
```

   O resultado deve ser semelhante ao seguinte:

```
shared_preload_libraries
--------------------------
rdsutils,plrust
(1 row)
```

Criar funções com o PL/Rust

O PL/Rust compilará a função como uma biblioteca dinâmica, a carregará e a executará.

A função do Rust a seguir filtra múltiplos de uma matriz.

```
postgres=> CREATE LANGUAGE plrust;
CREATE EXTENSION
```

```
CREATE OR REPLACE FUNCTION filter_multiples(a BIGINT[], multiple BIGINT) RETURNS
 BIGINT[]
    IMMUTABLE STRICT
    LANGUAGE PLRUST AS
$$
    Ok(Some(a.into_iter().filter(|x| x.unwrap() % multiple != 0).collect()))
$$;

WITH gen_values AS (
SELECT ARRAY(SELECT * FROM generate_series(1,100)) as arr)
SELECT filter_multiples(arr, 3)
from gen_values;
```

Usar caixas com PL/Rust

No RDS para PostgreSQL versões 16.3-R2 e posterior, 15.7-R2 e versões 15 posteriores, 14.12-R2 e versões 14 posteriores e 13.15-R2 e versões 13 posteriores, o PL/Rust é compatível com caixas adicionais:

- `url`
- `regex`
- `serde`
- `serde_json`

No RDS para PostgreSQL versões 15.5-R2 e posterior, 14.10-R2 e versões 14 posteriores e 13.13-R2 e versões 13 posteriores, o PL/Rust é compatível com duas caixas adicionais:

- `croaring-rs`
- `num-bigint`

A partir das versões 15.4, 14.9 e 13.12 do Amazon RDS para PostgreSQL, o PL/Rust comporta as seguintes caixas:

- `aes`
- `ctr`
- `rand`

Somente os recursos padrão são comportados para essas caixas. As novas versões do RDS para PostgreSQL poderão conter versões de caixa atualizadas, ao passo que as mais antigas podem deixar de ter suporte.

Siga as práticas recomendadas para realizar uma atualização de versão principal e testar se suas funções do PL/Rust são compatíveis com a nova versão principal. Para obter mais informações, consulte a publicação de blog Práticas recomendadas para atualizar o Amazon RDS para versões principais e secundárias do PostgreSQL e Atualizar o mecanismo de banco de dados PostgreSQL para Amazon RDS no “Guia do usuário do Amazon RDS”.

Exemplos de uso de dependências ao criar uma função PL/Rust estão disponíveis em Usar dependências.

Limitações do PL/Rust

Por padrão, os usuários do banco de dados não podem usar PL/Rust. Para fornecer acesso ao PL/Rust, conecte-se como usuário com o privilégio rds_superuser e execute o seguinte comando:

```
postgres=> GRANT USAGE ON LANGUAGE PLRUST TO user;
```

Gerenciar dados espaciais com a extensão PostGIS

PostGIS é uma extensão do PostgreSQL para armazenar e gerenciar informações espaciais. Para saber mais sobre a extensão PostGIS, consulte Postgis.net.

Desde a versão 10.5, o PostgreSQL é compatível com a biblioteca libprotobuf 1.3.0 usada pelo PostGIS para trabalhar com dados de blocos vetoriais do Mapbox.

A configuração da extensão PostGIS exige privilégios `rds_superuser`. Recomendamos criar um usuário (perfil) para gerenciar a extensão PostGIS e os dados espaciais. A extensão PostGIS e seus componentes relacionados adicionam milhares de funções ao PostgreSQL. Considere criar a extensão PostGIS em seu próprio esquema se isso fizer sentido para o seu caso de uso. O exemplo a seguir mostra como instalar a extensão em seu próprio banco de dados, mas isso não é necessário.

Tópicos

- Etapa 1: Criar um usuário (função) para gerenciar a extensão PostGIS
- Etapa 2: Carregar as extensões PostGIS
- Etapa 3: Transferir a propriedade dos esquemas de extensão
- Etapa 4: Transferir a propriedade dos objetos PostGIS
- Etapa 5: Testar as extensões
- Etapa 6: Atualize a extensão PostGIS
- Versões de extensão PostGIS
- Upgrade do PostGIS 2 para o PostGIS 3

Etapa 1: Criar um usuário (função) para gerenciar a extensão PostGIS

Primeiro, conecte-se a uma instância de banco de dados do RDS para PostgreSQL como um usuário que tem privilégios `rds_superuser`. Se você manteve o nome padrão ao configurar a instância, se conectará como `postgres`.

```
psql --host=111122223333.aws-region.rds.amazonaws.com --port=5432 --username=postgres
 --password
```

Crie um perfil separado (usuário) para administrar a extensão PostGIS.

```
postgres=>  CREATE ROLE gis_admin LOGIN PASSWORD 'change_me';
```

```
CREATE ROLE
```

Conceda a esse perfil privilégios `rds_superuser` para permitir que ele instale a extensão.

```
postgres=> GRANT rds_superuser TO gis_admin;
GRANT
```

Crie um banco de dados a ser usado para seus artefatos PostGIS. Esta etapa é opcional. Como alternativa, você pode criar um esquema em seu banco de dados de usuário para as extensões PostGIS, mas isso também não é necessário.

```
postgres=> CREATE DATABASE lab_gis;
CREATE DATABASE
```

Conceda a `gis_admin` todos os privilégios no banco de dados `lab_gis`.

```
postgres=> GRANT ALL PRIVILEGES ON DATABASE lab_gis TO gis_admin;
GRANT
```

Saia da sessão e reconecte-se a uma instância de banco de dados do RDS para PostgreSQL como `gis_admin`.

```
postgres=> psql --host=111122223333.aws-region.rds.amazonaws.com --port=5432 --
username=gis_admin --password --dbname=lab_gis
Password for user gis_admin:...
lab_gis=>
```

Continue a configurar a extensão conforme detalhado nas próximas etapas.

Etapa 2: Carregar as extensões PostGIS

A extensão PostGIS inclui várias extensões relacionadas que funcionam juntas para fornecer funcionalidade geoespacial. Dependendo do seu caso de uso, talvez você não precise de todas as extensões criadas nesta etapa.

Use instruções `CREATE EXTENSION` para carregar as extensões PostGIS.

```
CREATE EXTENSION postgis;
CREATE EXTENSION
CREATE EXTENSION postgis_raster;
```

```
CREATE EXTENSION
CREATE EXTENSION fuzzystrmatch;
CREATE EXTENSION
CREATE EXTENSION postgis_tiger_geocoder;
CREATE EXTENSION
CREATE EXTENSION postgis_topology;
CREATE EXTENSION
CREATE EXTENSION address_standardizer_data_us;
CREATE EXTENSION
```

É possível verificar os resultados executando a consulta SQL mostrada no exemplo a seguir, que lista as extensões e seus proprietários.

```
SELECT n.nspname AS "Name",
  pg_catalog.pg_get_userbyid(n.nspowner) AS "Owner"
  FROM pg_catalog.pg_namespace n
  WHERE n.nspname !~ '^pg_' AND n.nspname <> 'information_schema'
  ORDER BY 1;
List of schemas
     Name     |   Owner
--------------+-----------
 public       | postgres
 tiger        | rdsadmin
 tiger_data   | rdsadmin
 topology     | rdsadmin
(4 rows)
```

Etapa 3: Transferir a propriedade dos esquemas de extensão

Use as instruções ALTER SCHEMA para transferir a propriedade dos esquemas à função `gis_admin`.

```
ALTER SCHEMA tiger OWNER TO gis_admin;
ALTER SCHEMA
ALTER SCHEMA tiger_data OWNER TO gis_admin;
ALTER SCHEMA
ALTER SCHEMA topology OWNER TO gis_admin;
ALTER SCHEMA
```

Se você quiser confirmar a alteração de propriedade, realize a consulta SQL a seguir. Ou é possível usar o metacomando `\dn` na linha de comando do psql.

```
SELECT n.nspname AS "Name",
  pg_catalog.pg_get_userbyid(n.nspowner) AS "Owner"
  FROM pg_catalog.pg_namespace n
  WHERE n.nspname !~ '^pg_' AND n.nspname <> 'information_schema'
  ORDER BY 1;

       List of schemas
     Name     |     Owner
--------------+---------------
 public       | postgres
 tiger        | gis_admin
 tiger_data   | gis_admin
 topology     | gis_admin
(4 rows)
```

Etapa 4: Transferir a propriedade dos objetos PostGIS

> **Note**
>
> Não altere a propriedade das funções do PostGIS. A operação adequada e as futuras atualizações do PostGIS exigem que essas funções retenham a propriedade original. Para ter mais informações sobre permissões do PostGIS, consulte Segurança do PostgreSQL.

Use a seguinte função para transferir a propriedade das tabelas do PostGIS ao perfil `gis_admin`. Execute a seguinte instrução no prompt psql para criar a função.

```
CREATE FUNCTION exec(text) returns text language plpgsql volatile AS $f$ BEGIN EXECUTE
 $1; RETURN $1; END; $f$;
CREATE FUNCTION
```

Depois, execute a consulta a seguir para executar a função `exec` que, por sua vez, executa as instruções e altera as permissões.

```
SELECT exec('ALTER TABLE ' || quote_ident(s.nspname) || '.' || quote_ident(s.relname)
 || ' OWNER TO gis_admin;')
  FROM (
    SELECT nspname, relname
    FROM pg_class c JOIN pg_namespace n ON (c.relnamespace = n.oid)
    WHERE nspname in ('tiger','topology') AND
```

```
    relkind IN ('r','S','v') ORDER BY relkind = 'S')
s;
```

Etapa 5: Testar as extensões

Para evitar a necessidade de especificar o nome do esquema, adicione o esquema `tiger` ao seu caminho de pesquisa usando o seguinte comando.

```
SET search_path=public,tiger;
SET
```

Teste o esquema `tiger` usando a seguinte instrução SELECT.

```
SELECT address, streetname, streettypeabbrev, zip
 FROM normalize_address('1 Devonshire Place, Boston, MA 02109') AS na;
address | streetname | streettypeabbrev |  zip
---------+------------+------------------+-------
       1 | Devonshire | Pl               | 02109
(1 row)
```

Para saber mais sobre essa extensão, consulte Geocodificador Tiger na documentação do PostGIS.

Teste o acesso ao esquema `topology` usando a seguinte instrução `SELECT`. Isso chama a função `createtopology` para registrar um novo objeto de topologia (my_new_topo) com o identificador de referência espacial especificado (26986) e a tolerância padrão (0,5). Para saber mais, consulte CreateTopology na documentação do PostGIS.

```
SELECT topology.createtopology('my_new_topo',26986,0.5);
 createtopology
----------------
              1
(1 row)
```

Etapa 6: Atualize a extensão PostGIS

Cada nova versão do PostgreSQL oferece suporte a uma ou mais versões da extensão PostGIS compatíveis com essa versão. A atualização do mecanismo PostgreSQL para uma nova versão não atualiza automaticamente a extensão PostGIS. Antes de atualizar o mecanismo PostgreSQL, faça upgrade do PostGIS para a versão mais recente disponível para a versão atual do PostgreSQL. Para obter detalhes, consulte Versões de extensão PostGIS.

Após a atualização do mecanismo PostgreSQL, faça upgrade da extensão PostGIS novamente, desta vez para a versão compatível com a versão recém-atualizada do mecanismo PostgreSQL. Para obter mais informações sobre como fazer upgrade do mecanismo PostgreSQL, consulte Como atualizar a versão principal do RDS para PostgreSQL.

Você pode verificar as atualizações de versão da extensão PostGIS disponíveis na sua instância de banco de dados do RDS para PostgreSQL a qualquer momento. Para fazer isso, execute o comando a seguir. Esta função está disponível com PostGIS 2.5.0 e versões posteriores.

```
SELECT postGIS_extensions_upgrade();
```

Se a sua aplicação não oferecer suporte à versão mais recente do PostGIS, você poderá instalar uma versão mais antiga do PostGIS que esteja disponível na sua versão principal conforme o exposto a seguir.

```
CREATE EXTENSION postgis VERSION "2.5.5";
```

Se quiser fazer upgrade para uma versão específica do PostGIS usando uma versão mais antiga, também poderá usar o comando a seguir.

```
ALTER EXTENSION postgis UPDATE TO "2.5.5";
```

Dependendo de sua versão atual antes do upgrade, talvez você precise usar essa função novamente. O resultado da primeira execução da função determina a necessidade de uma função de atualização adicional. Por exemplo, isso acontece em caso de upgrade do PostGIS 2 para o PostGIS 3. Para ter mais informações, consulte Upgrade do PostGIS 2 para o PostGIS 3.

Se você atualizou essa extensão para se preparar para uma atualização da versão principal do mecanismo PostgreSQL, poderá continuar com outras tarefas preliminares. Para obter mais informações, consulte Como atualizar a versão principal do RDS para PostgreSQL.

Versões de extensão PostGIS

Recomendamos que você instale as versões de todas as extensões, como PostGIS, conforme listado em "Extension versions for Amazon RDS para PostgreSQL" (Versões de extensões para o Amazon RDS para PostgreSQL) nas Notas de lançamento do Amazon RDS para PostgreSQL. Você pode conferir quais versões estão disponíveis na sua versão usando o comando a seguir.

```
SELECT * FROM pg_available_extension_versions WHERE name='postgis';
```

Informações sobre versões estão disponíveis nas seções a seguir das Notas de lançamento do Amazon RDS para PostgreSQL:

- Extensões do PostgreSQL versão 16 compatíveis com o Amazon RDS
- Extensões do PostgreSQL versão 15 compatíveis com o Amazon RDS
- Extensões do PostgreSQL versão 14 compatíveis com o Amazon RDS
- Extensões do PostgreSQL versão 13 compatíveis com o Amazon RDS
- Extensões do PostgreSQL versão 12 compatíveis com o Amazon RDS
- Extensões do PostgreSQL versão 11 compatíveis com o Amazon RDS
- Extensões do PostgreSQL versão 10 compatíveis com o Amazon RDS
- Extensões do PostgreSQL versão 9.6.x compatíveis com o Amazon RDS

Upgrade do PostGIS 2 para o PostGIS 3

A partir da versão 3.0, a funcionalidade de rasterização do PostGIS é uma extensão separada, `postgis_raster`. Essa extensão tem seu próprio caminho de instalação e upgrade. Isso remove dezenas de funções, tipos de dados e outros artefatos necessários para o processamento de imagens rasterizadas da extensão `postgis` principal. Isso significa que, se o seu caso de uso não exigir processamento de rasterização, você não precisará instalar a extensão `postgis_raster`.

No exemplo de upgrade a seguir, o primeiro comando de upgrade extrai a funcionalidade de rasterização na extensão `postgis_raster`. Um segundo comando de upgrade é necessário para atualizar `postgis_raster` para a nova versão.

Como fazer upgrade do PostGIS 2 para o PostGIS 3

1. Identifique a versão padrão do PostGIS que está disponível para a versão do PostgreSQL em seu Instância de banco de dados do RDS para PostgreSQL. Para fazer isso, execute a consulta a seguir.

```
SELECT * FROM pg_available_extensions
    WHERE default_version > installed_version;
  name   | default_version | installed_version |                          comment
---------+-----------------+-------------------
+-------------------------------------------------------------
 postgis | 3.1.4           | 2.3.7             | PostGIS geometry and geography
 spatial types and functions
```

```
(1 row)
```

2. Identifique as versões do PostGIS instaladas em cada banco de dados na sua instância de banco de dados do RDS para PostgreSQL. Em outras palavras, consulte cada banco de dados do usuário da seguinte forma.

```
SELECT
    e.extname AS "Name",
    e.extversion AS "Version",
    n.nspname AS "Schema",
    c.description AS "Description"
FROM
    pg_catalog.pg_extension e
    LEFT JOIN pg_catalog.pg_namespace n ON n.oid = e.extnamespace
    LEFT JOIN pg_catalog.pg_description c ON c.objoid = e.oid
    AND c.classoid = 'pg_catalog.pg_extension'::pg_catalog.regclass
WHERE
    e.extname LIKE '%postgis%'
ORDER BY
    1;
  Name   | Version | Schema |                            Description
---------+---------+--------
+------------------------------------------------------------------------
 postgis | 2.3.7   | public | PostGIS geometry, geography, and raster spatial types
 and functions
(1 row)
```

Essa divergência entre a versão padrão (PostGIS 3.1.4) e a versão instalada (PostGIS 2.3.7) significa que você precisa atualizar a extensão PostGIS.

```
ALTER EXTENSION postgis UPDATE;
ALTER EXTENSION
WARNING: unpackaging raster
WARNING: PostGIS Raster functionality has been unpackaged
```

3. Execute a consulta a seguir para verificar se a funcionalidade de rasterização já está em seu próprio pacote.

```
SELECT
    probin,
    count(*)
FROM
```

```
    pg_proc
WHERE
    probin LIKE '%postgis%'
GROUP BY
    probin;
          probin          | count
--------------------------+-------
 $libdir/rtpostgis-2.3    | 107
 $libdir/postgis-3        | 487
(2 rows)
```

O resultado mostra que ainda há uma diferença entre as versões. As funções do PostGIS são da versão 3 (postgis-3), enquanto as funções de rasterização (rtpostgis) são da versão 2 (rtpostgis-2.3). Para concluir a atualização, execute o comando de upgrade novamente, da seguinte forma.

```
postgres=> SELECT postgis_extensions_upgrade();
```

Você pode ignorar as mensagens de aviso. Execute a consulta a seguir novamente para verificar se a atualização foi concluída. A atualização é concluída quando o PostGIS e todas as extensões relacionadas deixam de estar sinalizadas como necessitando de atualização.

```
SELECT postgis_full_version();
```

4. Use a consulta a seguir para ver o processo de atualização concluído e as extensões empacotadas separadamente, e verifique se as versões correspondem.

```
SELECT
    e.extname AS "Name",
    e.extversion AS "Version",
    n.nspname AS "Schema",
    c.description AS "Description"
FROM
    pg_catalog.pg_extension e
    LEFT JOIN pg_catalog.pg_namespace n ON n.oid = e.extnamespace
    LEFT JOIN pg_catalog.pg_description c ON c.objoid = e.oid
        AND c.classoid = 'pg_catalog.pg_extension'::pg_catalog.regclass
WHERE
    e.extname LIKE '%postgis%'
ORDER BY
    1;
```

```
      Name      | Version | Schema |                                Description
----------------+---------+--------
+-------------------------------------------------------------------------
 postgis        | 3.1.5   | public | PostGIS geometry, geography, and raster
 spatial types and functions
 postgis_raster | 3.1.5   | public | PostGIS raster types and functions
(2 rows)
```

A saída mostra que a extensão PostGIS 2 foi atualizada para PostGIS 3, e tanto `postgis` quanto a extensão `postgis_raster` agora separada estão na versão 3.1.5.

Depois que essa atualização for concluída, se você não planejar usar a funcionalidade de rasterização, poderá descartar a extensão da seguinte forma.

```
DROP EXTENSION postgis_raster;
```

Trabalhar com os invólucros de dados externos compatíveis do Amazon RDS for PostgreSQL

Um Foreign Data Wraper (FDW – Empacotador de dados externos) é um tipo específico de extensão que fornece acesso a dados externos. Por exemplo, a extensão `oracle_fdw` permite que o cluster de banco de dados do RDS for PostgreSQL funcione com bancos de dados Oracle. Citando outro exemplo, se você usar a extensão `postgres_fdw` nativa do PostgreSQL, poderá acessar dados armazenados em instâncias de banco de dados do PostgreSQL externas à instância de banco de dados do RDS for PostgreSQL.

A seguir, você encontrará informações sobre vários invólucros de dados externos do PostgreSQL compatíveis.

Tópicos

- Usar a extensão log_fdw para acessar o log de banco de dados usando SQL
- Usar a extensão postgres_fdw para acessar dados externos
- Trabalhar com bancos de dados MySQL usando a extensão mysql_fdw
- Trabalhar com um banco de dados Oracle usando a extensão oracle_fdw
- Trabalhar com bancos de dados do SQL Server usando a extensão tds_fdw

Usar a extensão log_fdw para acessar o log de banco de dados usando SQL

O RDS for PostgreSQL é compatível com a extensão , que permite acessar o log do mecanismo de banco de dados usando uma interface SQL. A extensão `log_fdw` apresenta duas novas funções que facilitam a criação de tabelas externas para logs de banco de dados:

- `list_postgres_log_files` – lista os arquivos no diretório do log do banco de dados e o tamanho do arquivo em bytes.
- `create_foreign_table_for_log_file(table_name text, server_name text, log_file_name text)` – cria uma tabela externa para o arquivo especificado no banco de dados atual.

Todas as funções criadas por `log_fdw` são de propriedade do `rds_superuser`. Os membros da função `rds_superuser` podem conceder acesso a essas funções para outros usuários do banco de dados.

Por padrão, os arquivos de log são gerados pelo Amazon RDS no formato (erro padrão) `stderr`, conforme especificado no parâmetro `log_destination`. Existem apenas duas opções para esse parâmetro, `stderr` e `csvlog` (valores separados por vírgula, CSV). Se você adicionar a opção `csvlog` ao parâmetro, o Amazon RDS gerará os dois logs, `stderr` e `csvlog`. Isso pode afetar a capacidade de armazenamento em seu cluster de banco de dados. Portanto, você precisa estar ciente dos outros parâmetros que afetam o processamento de logs. Para ter mais informações, consulte Definir o destino dos logs (`stderr`, `csvlog`).

Um benefício de gerar logs do `csvlog` é que a extensão `log_fdw` permite criar tabelas externas com dados divididos ordenadamente em várias colunas. Para fazer isso, sua instância precisa estar associada a um grupo de parâmetros de banco de dados personalizado para que você possa alterar a configuração para `log_destination`. Para obter mais informações sobre como fazer isso, consulte Trabalhar com parâmetros na instância de banco de dados do RDS para PostgreSQL.

O exemplo a seguir pressupõe que o parâmetro `log_destination` inclua `cvslog`.

Para usar a extensão log_fdw

1. Instale a extensão `log_fdw`.

```
postgres=> CREATE EXTENSION log_fdw;
CREATE EXTENSION
```

2. Crie o servidor de log como um wrapper externo de dados.

```
postgres=> CREATE SERVER log_server FOREIGN DATA WRAPPER log_fdw;
CREATE SERVER
```

3. Selecione todos os arquivos de log na lista.

```
postgres=> SELECT * FROM list_postgres_log_files() ORDER BY 1;
```

   A seguir você encontra um exemplo de resposta.

```
          file_name           | file_size_bytes
------------------------------+-----------------
```

```
 postgresql.log.2023-08-09-22.csv |           1111
 postgresql.log.2023-08-09-23.csv |           1172
 postgresql.log.2023-08-10-00.csv |           1744
 postgresql.log.2023-08-10-01.csv |           1102
(4 rows)
```

4. Crie uma tabela com uma única coluna "log_entry" para o arquivo selecionado.

```
postgres=> SELECT create_foreign_table_for_log_file('my_postgres_error_log',
     'log_server', 'postgresql.log.2023-08-09-22.csv');
```

A resposta não fornece nenhum detalhe além de que a tabela agora existe.

```
---------------------------------
(1 row)
```

5. Selecione um exemplo de arquivo de log. O código a seguir recupera o horário do log e a descrição da mensagem de erro.

```
postgres=> SELECT log_time, message FROM my_postgres_error_log ORDER BY 1;
```

A seguir você encontra um exemplo de resposta.

```
             log_time             |                                 message
----------------------------------
+-------------------------------------------------------------------------------
Tue Aug 09 15:45:18.172 2023 PDT | ending log output to stderr
Tue Aug 09 15:45:18.175 2023 PDT | database system was interrupted; last known up
 at 2023-08-09 22:43:34 UTC
Tue Aug 09 15:45:18.223 2023 PDT | checkpoint record is at 0/90002E0
Tue Aug 09 15:45:18.223 2023 PDT | redo record is at 0/90002A8; shutdown FALSE
Tue Aug 09 15:45:18.223 2023 PDT | next transaction ID: 0/1879; next OID: 24578
Tue Aug 09 15:45:18.223 2023 PDT | next MultiXactId: 1; next MultiXactOffset: 0
Tue Aug 09 15:45:18.223 2023 PDT | oldest unfrozen transaction ID: 1822, in
 database 1
(7 rows)
```

Usar a extensão postgres_fdw para acessar dados externos

Você pode acessar dados em uma tabela em um servidor de banco de dados remoto com a extensão postgres_fdw. Se você configurar uma conexão remota usando a instância de banco de dados do PostgreSQL, o acesso também estará disponível para a réplica de leitura.

Para usar postgres_fdw para acessar um servidor de banco de dados remoto

1. Instale a extensão postgres_fdw.

```
CREATE EXTENSION postgres_fdw;
```

2. Crie um servidor de dados externo usando CREATE SERVER.

```
CREATE SERVER foreign_server
FOREIGN DATA WRAPPER postgres_fdw
OPTIONS (host 'xxx.xx.xxx.xx', port '5432', dbname 'foreign_db');
```

3. Crie um mapeamento de usuário para identificar a função a ser usada no servidor remoto.

> **Important**
>
> Para ocultar a senha a fim de que ela não apareça nos logs, defina `log_statement=none` no nível da sessão. A configuração no nível do parâmetro não oculta a senha.

```
CREATE USER MAPPING FOR local_user
SERVER foreign_server
OPTIONS (user 'foreign_user', password 'password');
```

4. Crie uma tabela que mapeia para a tabela no servidor remoto.

```
CREATE FOREIGN TABLE foreign_table (
        id integer NOT NULL,
        data text)
SERVER foreign_server
OPTIONS (schema_name 'some_schema', table_name 'some_table');
```

Trabalhar com bancos de dados MySQL usando a extensão mysql_fdw

Para acessar um banco de dados compatível com MySQL pela instância de banco de dados RDS para PostgreSQL, você pode instalar e usar a extensão `mysql_fdw`. Esse invólucro de dados externos permite que você trabalhe com o RDS for MySQL, o Aurora MySQL, o MariaDB e outros bancos de dados compatíveis com MySQL. A conexão do RDS for PostgreSQL com o banco de dados MySQL é criptografada com base no melhor esforço, dependendo das configurações do cliente e do servidor. No entanto, você pode impor a criptografia, se quiser. Para ter mais informações, consulte Usar criptografia em trânsito com a extensão.

A extensão `mysql_fdw` é compatível com o Amazon RDS para PostgreSQL versão 14.2, 13.6 e posteriores. Ela é compatível com seleções, inserções, atualizações e exclusões de um banco de dados do RDS for PostgreSQL para tabelas em uma instância de banco de dados compatível com MySQL.

Tópicos

- Configurar o banco de dados do RDS for PostgreSQL para usar a extensão mysql_fdw
- Exemplo: trabalhar com um banco de dados do RDS para MySQL pelo RDS para PostgreSQL
- Usar criptografia em trânsito com a extensão

Configurar o banco de dados do RDS for PostgreSQL para usar a extensão mysql_fdw

Para configurar a extensão `mysql_fdw` na instância de banco de dados do RDS para PostgreSQL, é necessário carregar a extensão na instância de banco de dados e, em seguida, criar o ponto de conexão com a instância de banco de dados do MySQL. Para essa tarefa, você precisa ter os seguintes detalhes sobre a instância de banco de dados do MySQL:

- Nome de host ou endpoint. Para uma instância de banco de dados do RDS for MySQL, você pode encontrar o endpoint usando o console. Escolha a guia “Connectivity & security” (Conectividade e segurança) e procure na seção “Endpoint and port” (Endpoint e porta).
- Número da porta. O número da porta padrão do MySQL é 3306.
- O nome do banco de dados. O identificador do banco de dados.

Você também precisa fornecer acesso no grupo de segurança ou na lista de controle de acesso (ACL) para a porta 3306 do MySQL. Tanto o cluster de MySQL precisam acessar a porta 3306. Se

o acesso não estiver configurado corretamente, ao tentar se conectar à tabela compatível com o MySQL, será exibida uma mensagem de erro semelhante à seguinte:

```
ERROR: failed to connect to MySQL: Can't connect to MySQL server on 'hostname.aws-
region.rds.amazonaws.com:3306' (110)
```

No procedimento a seguir, você (como a conta `rds_superuser`) cria o servidor externo. Depois, você concede acesso ao servidor externo a usuários específicos. Em seguida, esses usuários criam seus próprios mapeamentos para as contas de usuário apropriadas do MySQL para trabalhar com a instância de banco de dados do MySQL.

Para usar mysql_fdw a fim de acessar um servidor de banco de dados MySQL

1. Conecte-se à instância de banco de dados do PostgreSQL usando uma conta que tenha a função `rds_superuser`. Se tiver aceitado os padrões ao criar a instância de banco de dados do RDS para PostgreSQL, o nome de usuário será `postgres` e você poderá se conectar usando a ferramenta da linha de comando `psql` da seguinte forma:

   ```
   psql --host=your-DB-instance.aws-region.rds.amazonaws.com --port=5432 --
   username=postgres --password
   ```

2. Instale a extensão `mysql_fdw` da seguinte forma:

   ```
   postgres=> CREATE EXTENSION mysql_fdw;
   CREATE EXTENSION
   ```

Depois que a extensão é instalada na instância de banco de dados do RDS for PostgreSQL, você configura o servidor externo que fornece a conexão com um banco de dados do MySQL.

Para criar o servidor externo

Execute essas tarefas na instância de banco de dados do RDS para PostgreSQL. As etapas presumem que você esteja conectado como usuário com privilégios `rds_superuser`, como `postgres`.

1. Crie um servidor externo na instância de banco de dados do RDS para PostgreSQL:

   ```
   postgres=> CREATE SERVER mysql-db FOREIGN DATA WRAPPER mysql_fdw OPTIONS (host 'db-
   name.111122223333.aws-region.rds.amazonaws.com', port '3306');
   ```

```
CREATE SERVER
```

2. Conceda aos usuários apropriados acesso ao servidor externo. Eles devem ser usuários não administradores, ou seja, usuários sem a função `rds_superuser`.

```
postgres=> GRANT USAGE ON FOREIGN SERVER mysql-db to user1;
GRANT
```

Os usuários do PostgreSQL criam e gerenciam suas próprias conexões com o banco de dados do MySQL por meio do servidor externo.

Exemplo: trabalhar com um banco de dados do RDS para MySQL pelo RDS para PostgreSQL

Suponha que você tenha uma tabela simples em uma instância de banco de dados do RDS para PostgreSQL. Os usuários do RDS para PostgreSQL querem consultar os itens (`SELECT`), `INSERT`, `UPDATE` e `DELETE` nessa tabela. Suponha que a extensão `mysql_fdw` foi criada na instância de banco de dados do RDS for PostgreSQL, conforme detalhado no procedimento anterior. Depois de se conectar à instância de banco de dados do RDS for PostgreSQL como um usuário com privilégios `rds_superuser`, é possível prosseguir com as etapas abaixo.

1. Crie um servidor externo na instância de banco de dados do RDS para PostgreSQL:

```
test=> CREATE SERVER mysqldb FOREIGN DATA WRAPPER mysql_fdw OPTIONS (host 'your-
DB.aws-region.rds.amazonaws.com', port '3306');
CREATE SERVER
```

2. Conceda uso a um usuário que não tenha permissões `rds_superuser`; por exemplo, `user1`:

```
test=> GRANT USAGE ON FOREIGN SERVER mysqldb TO user1;
GRANT
```

3. Conecte-se como *`user1`* e, em seguida, crie um mapeamento para o usuário do MySQL:

```
test=> CREATE USER MAPPING FOR user1 SERVER mysqldb OPTIONS (username 'myuser',
 password 'mypassword');
CREATE USER MAPPING
```

4. Crie uma tabela externa vinculada a uma tabela do MySQL:

```
test=> CREATE FOREIGN TABLE mytab (a int, b text) SERVER mysqldb OPTIONS (dbname
 'test', table_name '');
CREATE FOREIGN TABLE
```

5. Execute uma consulta simples na tabela externa:

```
test=> SELECT * FROM mytab;
a |   b
---+-------
1 | apple
(1 row)
```

6. Você pode adicionar, alterar e remover dados da tabela do MySQL. Por exemplo:

```
test=> INSERT INTO mytab values (2, 'mango');
INSERT 0 1
```

   Execute a consulta SELECT novamente para ver os resultados:

```
test=> SELECT * FROM mytab ORDER BY 1;
 a |   b
---+-------
1 | apple
2 | mango
(2 rows)
```

Usar criptografia em trânsito com a extensão

A conexão com o MySQL pelo RDS for PostgreSQL usa criptografia em trânsito (TLS/SSL) por padrão. No entanto, a conexão volta para não criptografada quando a configuração do cliente e do servidor é diferente. Você pode impor a criptografia para todas as conexões de saída especificando a opção REQUIRE SSL nas contas de usuário do RDS for MySQL. Essa mesma abordagem também funciona para contas de usuário do MariaDB e do Aurora MySQL.

Para contas de usuário do MySQL configuradas como REQUIRE SSL, a tentativa de conexão falhará se não for possível estabelecer uma conexão segura.

Para aplicar criptografia a contas de usuário do banco de dados do MySQL existentes, você pode usar o comando ALTER USER. A sintaxe varia, dependendo da versão do MySQL, conforme

mostrado na tabela a seguir. Para obter mais informações, consulte ALTER USER no Manual de referência do MySQL.

MySQL 5.7, MySQL 8.0	MySQL 5.6
`ALTER USER 'user'@'%' REQUIRE SSL;`	`GRANT USAGE ON *.* to 'user'@'%' REQUIRE SSL;`

Para obter mais informações sobre a extensão `mysql_fdw`, consulte a documentação do mysql_fdw.

Trabalhar com um banco de dados Oracle usando a extensão oracle_fdw

Para acessar um banco de dados Oracle pela instância de banco de dados do RDS for PostgreSQL, você pode instalar e usar a extensão `oracle_fdw`. Essa extensão é um invólucro de dados externos para bancos de dados Oracle. Para saber mais sobre a extensão, consulte a documentação do oracle_fdw.

A extensão `oracle_fdw` é compatível com o RDS for PostgreSQL 12.7, 13.3 e versões posteriores.

Tópicos

- Ativação da extensão oracle_fdw
- Exemplo: usar um servidor externo vinculado a um banco de dados Amazon RDS for Oracle
- Como trabalhar com criptografia em trânsito
- Noções básicas da visualização e das permissões de pg_user_mappings

Ativação da extensão oracle_fdw

Para usar a extensão oracle_fdw, realize o procedimento a seguir.

Para ativar a extensão oracle_fdw

- Execute o comando a seguir usando uma conta que tenha as permissões de `rds_superuser`.

```
CREATE EXTENSION oracle_fdw;
```

Exemplo: usar um servidor externo vinculado a um banco de dados Amazon RDS for Oracle

O exemplo a seguir mostra o uso de um servidor externo vinculado a um banco de dados Amazon RDS for Oracle.

Para criar um servidor externo vinculado a um banco de dados do RDS for Oracle

1. Na instância de banco de dados do RDS for Oracle, observe:
   - Endpoint
   - Port
   - Database name
2. Crie um servidor externo.

```
test=> CREATE SERVER oradb FOREIGN DATA WRAPPER oracle_fdw OPTIONS (dbserver
 '//endpoint:port/DB_name');
CREATE SERVER
```

3. Conceda uso a um usuário que não tenha privilégios `rds_superuser`, por exemplo `user1`.

```
test=> GRANT USAGE ON FOREIGN SERVER oradb TO user1;
GRANT
```

4. Conecte-se como `user1` e crie um mapeamento para um usuário Oracle.

```
test=> CREATE USER MAPPING FOR user1 SERVER oradb OPTIONS (user 'oracleuser',
 password 'mypassword');
CREATE USER MAPPING
```

5. Crie uma tabela estrangeira vinculada a uma tabela Oracle.

```
test=> CREATE FOREIGN TABLE mytab (a int) SERVER oradb OPTIONS (table 'MYTABLE');
CREATE FOREIGN TABLE
```

6. Consulte a tabela externa.

```
test=>  SELECT * FROM mytab;
a
---
1
```

```
(1 row)
```

Se a consulta relatar o seguinte erro, verifique seu grupo de segurança e a lista de controle de acesso (ACL) para verificar se as duas instâncias podem se comunicar.

```
ERROR: connection for foreign table "mytab" cannot be established
DETAIL: ORA-12170: TNS:Connect timeout occurred
```

Como trabalhar com criptografia em trânsito

A criptografia PostgreSQL-to-Oracle em trânsito é baseada em uma combinação de parâmetros de configuração de cliente e servidor. Para obter um exemplo usando o Oracle 21c, consulte About the Values for Negotiating Encryption and Integrity (Sobre os valores para negociar criptografia e integridade) na documentação do Oracle. Como o cliente usado para oracle_fdw no Amazon RDS está configurado com `ACCEPTED`, a criptografia depende da configuração do servidor de banco de dados do Oracle e utiliza a Oracle Security Library (libnnz) para isso.

Se o banco de dados estiver no RDS for Oracle, consulte Criptografia de rede nativa do Oracle para configurar a criptografia.

Noções básicas da visualização e das permissões de pg_user_mappings

O catálogo PostgreSQL `pg_user_mapping` armazena o mapeamento de um usuário do RDS for PostgreSQL para o usuário em um servidor de dados externo (remoto). O acesso ao catálogo é restrito, mas você usa a visualização `pg_user_mappings` para ver os mapeamentos. Veja a seguir um exemplo que mostra como as permissões se aplicam a um banco de dados Oracle de exemplo, mas essas informações se aplicam de forma mais geral a qualquer wrapper de dados externo.

Na saída a seguir, você pode encontrar funções e permissões mapeadas para três usuários de exemplo diferentes. Usuários eo `rdssu1` e `rdssu2` são membros da função `rds_superuser`, e `user1` não é. O exemplo usa o metacomando `\du` do `psql` para listar as funções existentes.

```
test=>  \du
                                                    List of roles
    Role name    |                         Attributes                         |
                 Member of
-----------------+------------------------------------------------------------
+-----------------------------------------------------------
```

```
 rdssu1          |                                                          |
 {rds_superuser}
 rdssu2          |                                                          |
 {rds_superuser}
 user1           |                                                          | {}
```

Todos os usuários, incluindo aqueles com privilégios `rds_superuser`, têm permissão para visualizar seus próprios mapeamentos de usuário (`umoptions`) na tabela `pg_user_mappings`. Como mostrado no exemplo a seguir, quando `rdssu1` tenta obter todos os mapeamentos do usuário, é gerado um erro, mesmo com privilégios `rdssu1rds_superuser`:

```
test=> SELECT * FROM pg_user_mapping;
ERROR: permission denied for table pg_user_mapping
```

Veja a seguir alguns exemplos:

```
test=> SET SESSION AUTHORIZATION rdssu1;
SET
test=> SELECT * FROM pg_user_mappings;
 umid  | srvid | srvname | umuser | usename    |            umoptions
-------+-------+---------+--------+------------+---------------------------------
 16414 | 16411 | oradb   |  16412 | user1      |
 16423 | 16411 | oradb   |  16421 | rdssu1     | {user=oracleuser,password=mypwd}
 16424 | 16411 | oradb   |  16422 | rdssu2     |
 (3 rows)

test=> SET SESSION AUTHORIZATION rdssu2;
SET
test=> SELECT * FROM pg_user_mappings;
 umid  | srvid | srvname | umuser | usename    |            umoptions
-------+-------+---------+--------+------------+---------------------------------
 16414 | 16411 | oradb   |  16412 | user1      |
 16423 | 16411 | oradb   |  16421 | rdssu1     |
 16424 | 16411 | oradb   |  16422 | rdssu2     | {user=oracleuser,password=mypwd}
 (3 rows)

test=> SET SESSION AUTHORIZATION user1;
SET
test=> SELECT * FROM pg_user_mappings;
 umid  | srvid | srvname | umuser | usename    |           umoptions
-------+-------+---------+--------+------------+-------------------------------
 16414 | 16411 | oradb   |  16412 | user1      | {user=oracleuser,password=mypwd}
```

```
 16423 | 16411 | oradb   |  16421 | rdssu1     |
 16424 | 16411 | oradb   |  16422 | rdssu2     |
 (3 rows)
```

Devido às diferenças na implementação de `information_schema._pg_user_mappings` e `pg_catalog.pg_user_mappings`, um `rds_superuser` criado manualmente requer outras permissões para visualizar senhas no `pg_catalog.pg_user_mappings`.

Nenhuma permissão adicional é necessária para um `rds_superuser` visualizar senhas no `information_schema._pg_user_mappings`.

Usuários que não tenham a função `rds_superuser` poderão visualizar senhas em `pg_user_mappings` somente nas seguintes condições:

- O usuário atual é o usuário que está sendo mapeado e é proprietário do servidor ou mantém o privilégio `USAGE` sobre ele.
- O usuário atual é o proprietário do servidor, e o mapeamento é para `PUBLIC`.

Trabalhar com bancos de dados do SQL Server usando a extensão tds_fdw

Você pode usar a extensão `tds_fdw` do PostgreSQL para acessar bancos de dados compatíveis com o protocolo de fluxo de dados tabular (TDS), como bancos de dados do Sybase e do Microsoft SQL Server. Esse invólucro de dados externos permite que você se conecte pela instância de banco de dados do RDS for PostgreSQL a bancos de dados que usam o protocolo TDS, incluindo o Amazon RDS for Microsoft SQL Server. Para obter mais informações, consulte a documentação do tds-fdw/tds_fdw no GitHub.

A extensão `tds_fdw` é compatível com o Amazon RDS for PostgreSQL versões 14.2, 13.6 e posteriores.

Configurar o banco de dados do RDS for PostgreSQL para usar a extensão tds_fdw

Nos procedimentos a seguir, você pode encontrar um exemplo de configuração e uso de `tds_fdw` com uma instância de banco de dados do RDS for PostgreSQL. Antes de se conectar a um banco de dados do SQL Server usando `tds_fdw`, é necessário obter os seguintes detalhes da instância:

- Nome de host ou endpoint. Para uma instância de banco de dados do RDS for SQL Server, você pode encontrar o endpoint usando o console. Escolha a guia “Connectivity & security” (Conectividade e segurança) e procure na seção “Endpoint and port” (Endpoint e porta).

- Número da porta. O número da porta padrão para o Microsoft SQL Server é 1433.
- O nome do banco de dados. O identificador do banco de dados.

Você também precisa fornecer acesso no grupo de segurança ou na lista de controle de acesso (ACL) para a porta 1433 do SQL Server. Tanto a instância de banco de dados do RDS for PostgreSQL como a instância de banco de dados do RDS for SQL Server precisam de acesso à porta 1433. Se o acesso não estiver configurado corretamente, ao tentar consultar o Microsoft SQL Server, será exibida a seguinte mensagem de erro:

```
ERROR: DB-Library error: DB #: 20009, DB Msg: Unable to connect:
Adaptive Server is unavailable or does not exist (mssql2019.aws-
region.rds.amazonaws.com), OS #: 0, OS Msg: Success, Level: 9
```

Para usar tds_fdw a fim de se conectar a um banco de dados do SQL Server

1. Conecte-se à instância de banco de dados do PostgreSQL usando uma conta que tenha a função `rds_superuser`:

   ```
   psql --host=your-DB-instance.aws-region.rds.amazonaws.com --port=5432 --
   username=test --password
   ```

2. Instale a extensão `tds_fdw`:

   ```
   test=> CREATE EXTENSION tds_fdw;
   CREATE EXTENSION
   ```

Depois que a extensão for instalada na instância de banco de dados do RDS for PostgreSQL, configure o servidor externo.

Para criar o servidor externo

Execute essas tarefas na instância de banco de dados do RDS for PostgreSQL usando uma conta com privilégios `rds_superuser`.

1. Crie um servidor externo na instância de banco de dados do RDS para PostgreSQL:

   ```
   test=> CREATE SERVER sqlserverdb FOREIGN DATA WRAPPER tds_fdw OPTIONS
    (servername 'mssql2019.aws-region.rds.amazonaws.com', port '1433', database
    'tds_fdw_testing');
   ```

```
CREATE SERVER
```

Para acessar dados não ASCII do lado do SQLServer, crie um link de servidor com a opção character_set na instância de banco de dados do RDS para PostgreSQL:

```
test=> CREATE SERVER sqlserverdb FOREIGN DATA WRAPPER tds_fdw OPTIONS (servername
 'mssql2019.aws-region.rds.amazonaws.com', port '1433', database 'tds_fdw_testing',
 character_set 'UTF-8');
CREATE SERVER
```

2. Conceda uso a um usuário que não tenha permissões da função `rds_superuser`; por exemplo, `user1`:

```
test=> GRANT USAGE ON FOREIGN SERVER sqlserverdb TO user1;
```

3. Conecte-se como user1 e, em seguida, crie um mapeamento para um usuário do SQL Server:

```
test=> CREATE USER MAPPING FOR user1 SERVER sqlserverdb OPTIONS (username
 'sqlserveruser', password 'password');
CREATE USER MAPPING
```

4. Crie uma tabela externa vinculada a uma tabela do SQL Server:

```
test=> CREATE FOREIGN TABLE mytab (a int) SERVER sqlserverdb OPTIONS (table
 'MYTABLE');
CREATE FOREIGN TABLE
```

5. Consulte a tabela externa:

```
test=> SELECT * FROM mytab;
 a
---
 1
(1 row)
```

Usar criptografia em trânsito para a conexão

A conexão do RDS for PostgreSQL com o SQL Server usa criptografia em trânsito (TLS/SSL), dependendo da configuração do banco de dados do SQL Server. Se o SQL Server não estiver

configurado para criptografia, o cliente do RDS for PostgreSQL que faz a solicitação ao banco de dados do SQL Server retornará ao estado de não criptografado.

Você pode aplicar a criptografia para a conexão com instâncias de banco de dados do RDS for SQL Server definindo o parâmetro `rds.force_ssl`. Para saber como, consulte Forçar conexões com a instância de banco de dados para usar SSL. Para obter mais informações sobre a configuração de SSL/TLS para o RDS for SQL Server, consulte Usar SSL com uma instância de banco de dados do Microsoft SQL Server.

Trabalhar com Trusted Language Extensions para PostgreSQL

O Trusted Language Extensions para PostgreSQL é um kit de desenvolvimento de código aberto para criar extensões do PostgreSQL. Ele permite que você crie extensões do PostgreSQL de alta performance e as execute com segurança em sua instância de banco de dados do RDS para PostgreSQL. Ao usar o Trusted Language Extensions (TLE) para PostgreSQL, você pode criar extensões do PostgreSQL que sigam a abordagem documentada para estender a funcionalidade do PostgreSQL. Para obter mais informações, consulte Packaging Related Objects into an Extension (Compactar objetos relacionados em uma extensão) na documentação do PostgreSQL.

Um dos principais benefícios do TLE é que você pode usá-lo em ambientes que não fornecem acesso ao sistema de arquivos subjacente à instância do PostgreSQL. Anteriormente, a instalação de uma nova extensão exigia acesso ao sistema de arquivos. O TLE remove essa restrição. Ele fornece um ambiente de desenvolvimento para criar extensões para qualquer banco de dados do PostgreSQL, como as executadas em suas instâncias de banco de dados do RDS para PostgreSQL.

O TLE foi projetado para impedir o acesso a recursos inseguros para as extensões criadas com o uso do TLE. Seu ambiente de execução limita o impacto de qualquer defeito de extensão em uma única conexão de banco de dados. O TLE também oferece aos administradores de banco de dados um controle refinado sobre quem pode instalar extensões e fornece um modelo de permissões para executá-las.

O TLE é compatível com as seguintes versões do RDS para PostgreSQL:

- Versão 17.1 e versões 17 posteriores
- Versão 16.1 e versões 16 posteriores
- Versão 15.2 e versões 15 posteriores
- Versão 14.5 e versões 14 posteriores
- Versão 13.12 e versões 13 posteriores

O ambiente de desenvolvimento e de execução do Trusted Language Extensions são empacotados como a extensão `pg_tle` do PostgreSQL, versão 1.0.1. É compatível com a criação de extensões em JavaScript, Perl, Tcl, PL/pgSQL e SQL. Você instala a extensão `pg_tle` em sua instância de banco de dados do RDS para PostgreSQL da mesma forma que instala outras extensões do PostgreSQL. Depois de configurar `pg_tle`, os desenvolvedores podem usá-lo para criar extensões do PostgreSQL, conhecidas como extensões TLE.

Nos tópicos a seguir, você encontrará informações sobre como configurar Trusted Language Extensions e como começar a criar suas próprias extensões TLE.

Tópicos

- Terminologia
- Requisitos para usar Trusted Language Extensions para PostgreSQL
- Configurar o Trusted Language Extensions em sua instância de banco de dados do RDS para PostgreSQL
- Visão geral do Trusted Language Extensions para PostgreSQL
- Criar extensões TLE para RDS para PostgreSQL
- Descartar suas extensões TLE de um banco de dados
- Desinstalar o Trusted Language Extensions para PostgreSQL
- Usar ganchos do PostgreSQL com suas extensões TLE
- Usar tipos de dados personalizados no TLE
- Referência de funções para Trusted Language Extensions para PostgreSQL
- Referência de ganchos para Trusted Language Extensions para PostgreSQL

Terminologia

Para ajudar você a entender melhor o Trusted Language Extensions, consulte o glossário a seguir para conhecer os termos usados neste tópico.

Trusted Language Extensions para PostgreSQL

: Trusted Language Extensions para PostgreSQL é o nome oficial do kit de desenvolvimento de código aberto que vem com a extensão `pg_tle`. Ele está disponível para uso em qualquer sistema PostgreSQL. Para obter mais informações, consulte aws/pg_tle no GitHub.

Trusted Language Extensions

: Trusted Language Extensions é o nome abreviado de Trusted Language Extensions para PostgreSQL. Esse nome curto e sua abreviatura (TLE) também são utilizados nesta documentação.

linguagem confiável

: Uma linguagem confiável é uma linguagem de programação ou script que tem atributos de segurança específicos. Por exemplo, as linguagens confiáveis geralmente restringem o acesso

ao sistema de arquivos e limitam o uso de propriedades de rede especificadas. O kit de desenvolvimento TLE foi projetado para oferecer suporte a linguagens confiáveis. O PostgreSQL é compatível com várias linguagens diferentes que são usadas para criar extensões confiáveis ou não confiáveis. Por exemplo, consulte Trusted and Untrusted PL/Perl (PL/Perl confiável e não confiável) na documentação do PostgreSQL. Quando você cria uma extensão com o uso de Trusted Language Extensions, a extensão usa inerentemente mecanismos de linguagem confiáveis.

Extensão TLE

Uma extensão TLE é uma extensão do PostgreSQL criada com o uso do kit de desenvolvimento Trusted Language Extensions (TLE).

Requisitos para usar Trusted Language Extensions para PostgreSQL

A seguir são apresentados os requisitos para configurar e usar o kit de desenvolvimento TLE.

- Versões do RDS para PostgreSQL: o Trusted Language Extensions só é compatível com o RDS para PostgreSQL versões 13.12 e versões 13 posteriores, 14.5 e versões 14 posteriores e 15.2 e versões posteriores.
  - Se você precisar atualizar sua instância do RDS para PostgreSQL, consulte Atualizações do mecanismo de banco de dados do RDS para PostgreSQL.
  - Se você ainda não tem uma instância de banco de dados do Amazon RDS que execute o PostgreSQL, você pode criar uma. Para ter mais informações, consulte Instância de banco de dados do RDS para PostgreSQL, consulte Criar uma instância de banco de dados PostgreSQL e conectar-se a ela.
- Requer privilégios **`rds_superuser`**: para instalar e configurar a extensão `pg_tle`, sua função de usuário do banco de dados deve ter as permissões da função `rds_superuser`. Por padrão, essa função é concedida ao usuário `postgres` que cria o Instância de banco de dados do RDS para PostgreSQL.
- Requer um grupo de parâmetros de banco de dados personalizado: sua instância de banco de dados do RDS para PostgreSQL deve ser configurada com um grupo de parâmetros de banco de dados personalizado.
  - Se sua instância do RDS para PostgreSQL não estiver configurada com um grupo de parâmetros de banco de dados personalizado, você deverá criar um e associá-lo à sua instância de banco de dados do RDS para PostgreSQL. Para obter um breve resumo das etapas, consulte Criar e aplicar um grupo de parâmetros de banco de dados personalizado.

- Se sua instância de banco de dados do RDS para PostgreSQL já estiver configurada usando um grupo de parâmetros de banco de dados, você poderá configurar o Trusted Language Extensions. Para obter detalhes, consulte Configurar o Trusted Language Extensions em sua instância de banco de dados do RDS para PostgreSQL.

Criar e aplicar um grupo de parâmetros de banco de dados personalizado

Use as etapas a seguir para criar um grupo de parâmetros de banco de dados personalizado e configurar sua instância de banco de dados do RDS para PostgreSQL para usá-lo.

Console

Como criar um grupo de parâmetros de banco de dados personalizado e usá-lo com sua instância de banco de dados do RDS para PostgreSQL

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. Selecione Parameter groups (Grupos de parâmetros) no menu do Amazon RDS
3. Escolha Create parameter group (Criar grupo de parâmetros).
4. Na página Parameter group details (Detalhes do grupo de parâmetros), insira as informações a seguir.
   - Em Parameter group family (Família de grupos de parâmetros), selecione postgres14.
   - Em Type (Tipo), selecione DB Parameter Group (Grupo de parâmetros de banco de dados).
   - Em Group name (Nome do grupo), forneça ao seu grupo de parâmetros um nome significativo no contexto de suas operações.
   - Em Description (Descrição), insira uma descrição útil para que outras pessoas de sua equipe possam encontrá-lo facilmente.
5. Escolha Criar. Seu grupo de parâmetros de banco de dados personalizado é criado em sua Região da AWS. Agora você pode modificar sua instância do RDS para PostgreSQL para usá-la seguindo as próximas etapas.
6. Selecione Databases (Bancos de dados) no menu do Amazon RDS.
7. Selecione a instância de banco de dados do RDS para PostgreSQL que você deseja usar com o TLE dentre as opções listadas e, depois, selecione Modify (Modificar).
8. Na Na página Modify DB instance settings (Modificar configurações da instância de banco de dados), encontre Database options (Opções de banco de dados) na seção Additional

configuration (Configuração adicional) e selecione seu grupo de parâmetros de banco de dados personalizado no seletor.

9. Selecione Continue (Continuar) para salvar a alteração.
10. Selecione Apply immediately (Aplicar imediatamente) para que você possa continuar configurando da instância de banco de dados do RDS para PostgreSQL para usar o TLE.

Para continuar configurando seu sistema para Trusted Language Extensions, consulte Configurar o Trusted Language Extensions em sua instância de banco de dados do RDS para PostgreSQL.

Para obter mais informações sobre como trabalhar com Grupos de parâmetros do banco de dados, consulte Grupos de parâmetros de banco de dados para instâncias de banco de dados do Amazon RDS.

AWS CLI

Você pode evitar especificar o argumento `--region` ao usar comandos da CLI configurando sua AWS CLI com sua Região da AWS padrão. Para obter mais informações, consulte Conceitos básicos de configuração da no Guia do usuário do AWS Command Line Interface.

Como criar um grupo de parâmetros de banco de dados personalizado e usá-lo com sua instância de banco de dados do RDS para PostgreSQL

1. Use o comando create-db-parameter-group da AWS CLI para criar um grupo de parâmetros de banco de dados personalizado com base em postgres14 para sua Região da AWS.

   Para Linux, macOS ou Unix:

   ```
   aws rds create-db-parameter-group \
     --region aws-region \
     --db-parameter-group-name custom-params-for-pg-tle \
     --db-parameter-group-family postgres14 \
     --description "My custom DB parameter group for Trusted Language Extensions"
   ```

   Para Windows:

   ```
   aws rds create-db-parameter-group ^
     --region aws-region ^
     --db-parameter-group-name custom-params-for-pg-tle ^
     --db-parameter-group-family postgres14 ^
   ```

```
  --description "My custom DB parameter group for Trusted Language Extensions"
```

Seu grupo de parâmetros de banco de dados personalizado está disponível em sua Região da AWS, para que você possa modificar a instância de banco de dados do RDS para PostgreSQL para usá-lo.

2. Use o comando modify-db-instance da AWS CLI para aplicar seu grupo de parâmetros de banco de dados personalizado à instância de banco de dados do RDS para PostgreSQL. Esse comando reinicia imediatamente a instância ativa.

   Para Linux, macOS ou Unix:

```
aws rds modify-db-instance \
  --region aws-region \
  --db-instance-identifier your-instance-name \
  --db-parameter-group-name custom-params-for-pg-tle \
  --apply-immediately
```

   Para Windows:

```
aws rds modify-db-instance ^
  --region aws-region ^
  --db-instance-identifier your-instance-name ^
  --db-parameter-group-name custom-params-for-pg-tle ^
  --apply-immediately
```

Para continuar configurando seu sistema para Trusted Language Extensions, consulte Configurar o Trusted Language Extensions em sua instância de banco de dados do RDS para PostgreSQL.

Para obter mais informações, consulte Grupos de parâmetros para Amazon RDS.

Configurar o Trusted Language Extensions em sua instância de banco de dados do RDS para PostgreSQL

As etapas a seguir pressupõem que sua instância de banco de dados do RDS para PostgreSQL esteja associada a um grupo de parâmetros de banco de dados. Você pode usar o AWS Management Console ou a AWS CLI para essas etapas.

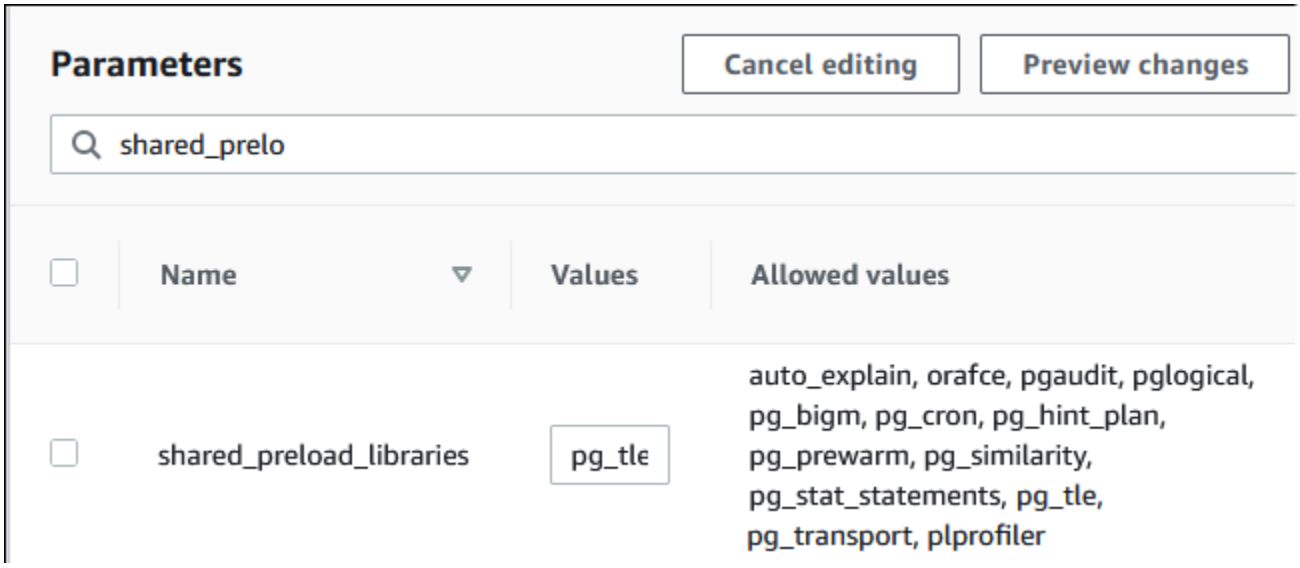
Ao configurar o Trusted Language Extensions em sua instância de banco de dados do RDS para PostgreSQL, você o instala em um banco de dados específico para uso pelos usuários do banco de dados que têm permissões nesse banco de dados.

Console

Como configurar o Trusted Language Extensions

Execute as etapas a seguir usando uma conta que seja membro do grupo `rds_superuser` (função).

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, selecione sua instância de banco de dados do RDS para PostgreSQL.
3. Abra a guia Configuration (Configuração) para sua Instância de banco de dados do RDS para PostgreSQL. Entre os detalhes da instância, encontre o link Parameter group (Grupo de parâmetros).
4. Clique no link para abrir os parâmetros personalizados associados ao seu Instância de banco de dados do RDS para PostgreSQL.
5. No campo Parameters (Parâmetros), digite `shared_pre` para encontrar o parâmetro `shared_preload_libraries`.
6. Selecione Edit parameters (Editar parâmetros) para acessar os valores das propriedades.
7. Adicione `pg_tle` à lista no campo Values (Valores). Use uma vírgula para separar itens na lista de valores.



8. Reinicie a instância de banco de dados do RDS para PostgreSQL para que a alteração no parâmetro `shared_preload_libraries` tenha efeito.
9. Quando a instância estiver disponível, verifique se `pg_tle` foi inicializado. Use `psql` para se conectar à instância de banco de dados do RDS para PostgreSQL e depois execute o comando a seguir.

```
SHOW shared_preload_libraries;
shared_preload_libraries
--------------------------
rdsutils,pg_tle
(1 row)
```

10. Com a extensão `pg_tle` inicializada, agora você pode criar a extensão.

```
CREATE EXTENSION pg_tle;
```

Para verificar se a extensão está instalada, você pode usar o metacomando `psql` a seguir.

```
labdb=> \dx
                          List of installed extensions
  Name   | Version |   Schema   |                Description
---------+---------+------------+-------------------------------------------
 pg_tle  | 1.0.1   | pgtle      | Trusted-Language Extensions for PostgreSQL
 plpgsql | 1.0     | pg_catalog | PL/pgSQL procedural language
```

11. Conceda a função `pgtle_admin` ao nome de usuário principal que você criou para sua instância de banco de dados do RDS para PostgreSQL ao configurá-la. Se você aceitou o padrão, é `postgres`.

```
labdb=> GRANT pgtle_admin TO postgres;
GRANT ROLE
```

É possível verificar se a concessão ocorreu usando o metacomando `psql` conforme mostrado no exemplo a seguir. Somente as funções `pgtle_admin` e `postgres` são mostradas na saída. Para ter mais informações, consulte Noções básicas sobre o perfil rds_superuser.

```
labdb=> \du
                           List of roles
    Role name    |           Attributes            |              Member of
```

```
----------------+----------------------------------
+----------------------------------
pgtle_admin     | Cannot login                      | {}
postgres        | Create role, Create DB          +| {rds_superuser,pgtle_admin}
                | Password valid until infinity    |...
```

12. Feche a sessão `psql` usando o metacomando `\q`.

```
\q
```

Para começar a criar extensões TLE, consulte Exemplo: Criar uma extensão de linguagem confiável usando SQL.

AWS CLI

Você pode evitar especificar o argumento `--region` ao usar comandos da CLI configurando sua AWS CLI com sua Região da AWS padrão. Para obter mais informações, consulte Conceitos básicos de configuração da no Guia do usuário do AWS Command Line Interface.

Como configurar o Trusted Language Extensions

1. Utilize o comando modify-db-parameter-group AWS CLI para adicionar `pg_tle` ao parâmetro `shared_preload_libraries`.

```
aws rds modify-db-parameter-group \
   --db-parameter-group-name custom-param-group-name \
   --parameters
 "ParameterName=shared_preload_libraries,ParameterValue=pg_tle,ApplyMethod=pending-
reboot" \
   --region aws-region
```

2. Utilize o comando reboot-db-instance da AWS CLI para reinicializar a instância de banco de dados do RDS para PostgreSQL e inicialize a biblioteca `pg_tle`.

```
aws rds reboot-db-instance \
    --db-instance-identifier your-instance \
    --region aws-region
```

3. Quando a instância estiver disponível, verifique se a `pg_tle` foi inicializada. Use `psql` para se conectar à instância de banco de dados do RDS para PostgreSQL e, depois, execute o comando a seguir.

```
SHOW shared_preload_libraries;
shared_preload_libraries
--------------------------
rdsutils,pg_tle
(1 row)
```

Com `pg_tle` inicializado, agora você pode criar a extensão.

```
CREATE EXTENSION pg_tle;
```

4. Conceda a função `pgtle_admin` ao nome de usuário principal que você criou para sua instância de banco de dados do RDS para PostgreSQL ao configurá-la. Se você aceitou o padrão, é `postgres`.

```
GRANT pgtle_admin TO postgres;
GRANT ROLE
```

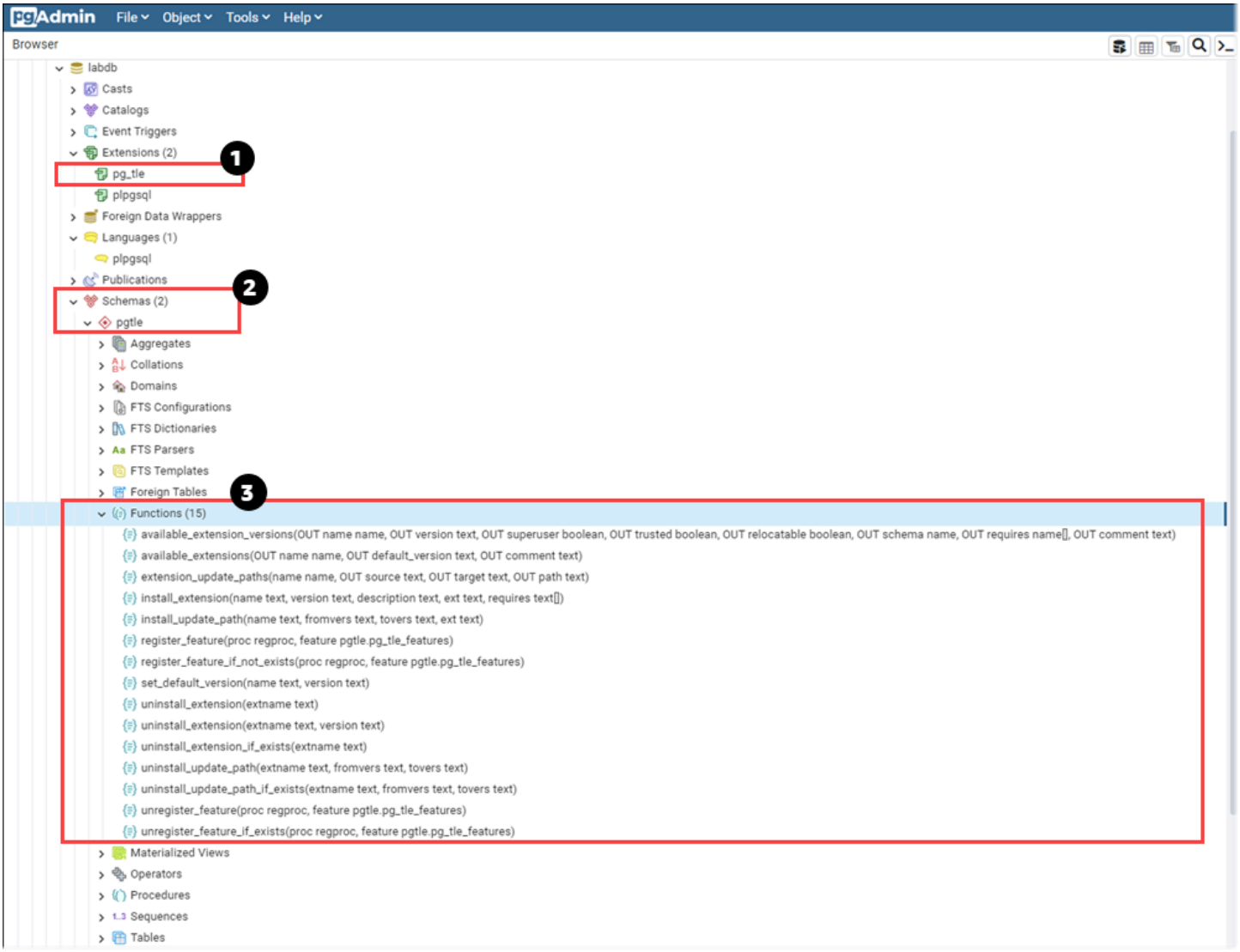
5. Feche a sessão `psql` da forma a seguir.

```
labdb=> \q
```

Para começar a criar extensões TLE, consulte Exemplo: Criar uma extensão de linguagem confiável usando SQL.

Visão geral do Trusted Language Extensions para PostgreSQL

O Trusted Language Extensions para PostgreSQL é uma extensão do PostgreSQL que você instala em sua instância de banco de dados do RDS para PostgreSQL da mesma forma que configura outras extensões do PostgreSQL. Na imagem a seguir de um exemplo de banco de dados na ferramenta cliente pgAdmin, você pode ver alguns dos componentes que compõem a extensão `pg_tle`.



É possível ver os detalhes a seguir.

1. O kit de desenvolvimento Trusted Language Extensions para PostgreSQL é embalado como a extensão `pg_tle`. Assim, o `pg_tle` é adicionado às extensões disponíveis para o banco de dados no qual está instalado.
2. O TLE tem seu próprio esquema, `pgtle`. Esse esquema contém funções auxiliares (3) para instalar e gerenciar as extensões criadas.
3. O TLE fornece mais de uma dúzia de funções auxiliares para instalar, registrar e gerenciar suas extensões. Para saber mais sobre essas funções, consulte Referência de funções para Trusted Language Extensions para PostgreSQL.

São outros componentes da extensão `pg_tle`:

- A função **`pgtle_admin`**: a função `pgtle_admin` é criada quando a extensão `pg_tle` é instalada. Essa função é privilegiada e deve ser tratada como tal. É altamente recomendável seguir o princípio de privilégio mínimo ao conceder a função `pgtle_admin` a usuários de banco de dados. Em outras palavras, conceda a função `pgtle_admin` somente aos usuários do banco de dados que tenham permissão para criar, instalar e gerenciar novas extensões TLE, como `postgres`.
- A tabela **`pgtle.feature_info`**: `pgtle.feature_info` é uma tabela protegida que contém informações sobre seus TLEs, ganchos e as funções e os procedimentos armazenados personalizados utilizados. Se você tiver privilégios `pgtle_admin`, use as funções de Trusted Language Extensions a seguir para adicionar e atualizar essas informações na tabela.
  - pgtle.register_feature
  - pgtle.register_feature_if_not_exists
  - pgtle.unregister_feature
  - pgtle.unregister_feature_if_exists

Criar extensões TLE para RDS para PostgreSQL

Você pode instalar qualquer extensão criada com o TLE em qualquer instância de banco de dados do RDS para PostgreSQL que tenha a extensão `pg_tle` instalada. A extensão `pg_tle` tem como escopo o banco de dados PostgreSQL no qual ela está instalada. As extensões que você cria usando o TLE têm como escopo o mesmo banco de dados.

Use as várias funções `pgtle` para instalar o código que compõe sua extensão TLE. As funções do Trusted Language Extensions a seguir exigem a função `pgtle_admin`.

- pgtle.install_extension
- pgtle.install_update_path
- pgtle.register_feature
- pgtle.register_feature_if_not_exists
- pgtle.set_default_version
- pgtle.uninstall_extension (nome)
- pgtle.uninstall_extension (nome, versão)
- pgtle.uninstall_extension_if_exists
- pgtle.uninstall_update_path
- pgtle.uninstall_update_path_if_exists

- pgtle.unregister_feature
- pgtle.unregister_feature_if_exists

Exemplo: Criar uma extensão de linguagem confiável usando SQL

O exemplo a seguir mostra como criar uma extensão TLE chamada `pg_distance` que contém algumas funções SQL para calcular distâncias usando fórmulas diferentes. Na lista, você pode encontrar a função para calcular a distância de Manhattan e a função para calcular a distância euclidiana. Para obter mais informações sobre a diferença entre essas fórmulas, consulte Geometria taxicab Geometria euclidiana na Wikipedia.

Você poderá usar esse exemplo em sua própria instância de banco de dados do RDS para PostgreSQL se tiver a extensão `pg_tle` configurada conforme detalhado em Configurar o Trusted Language Extensions em sua instância de banco de dados do RDS para PostgreSQL.

> **Note**
>
> Você precisa ter os privilégios da função `pgtle_admin` para seguir esse procedimento.

Como criar o exemplo de extensão TLE

As etapas a seguir usam um exemplo de banco de dados chamado `labdb`. Esse banco de dados é de propriedade do usuário primário `postgres`. A função `postgres` também tem as permissões da função `pgtle_admin`.

1. Use o `psql` para se conectar à Instância de banco de dados do RDS para PostgreSQL.

   ```
   psql --host=db-instance-123456789012.aws-region.rds.amazonaws.com
   --port=5432 --username=postgres --password --dbname=labdb
   ```

2. Crie uma extensão TLE denominada `pg_distance` copiando o código a seguir e colando-o no console da sessão `psql`.

   ```
   SELECT pgtle.install_extension
   (
    'pg_distance',
    '0.1',
     'Distance functions for two points',
   $_pg_tle_$
   ```

```
    CREATE FUNCTION dist(x1 float8, y1 float8, x2 float8, y2 float8, norm int)
    RETURNS float8
    AS $$
      SELECT (abs(x2 - x1) ^ norm + abs(y2 - y1) ^ norm) ^ (1::float8 / norm);
    $$ LANGUAGE SQL;

    CREATE FUNCTION manhattan_dist(x1 float8, y1 float8, x2 float8, y2 float8)
    RETURNS float8
    AS $$
      SELECT dist(x1, y1, x2, y2, 1);
    $$ LANGUAGE SQL;

    CREATE FUNCTION euclidean_dist(x1 float8, y1 float8, x2 float8, y2 float8)
    RETURNS float8
    AS $$
      SELECT dist(x1, y1, x2, y2, 2);
    $$ LANGUAGE SQL;
$_pg_tle_$
);
```

Você verá a saída da forma a seguir.

```
install_extension
---------------
 t
(1 row)
```

Os artefatos que compõem a extensão `pg_distance` agora estão instalados em seu banco de dados. Esses artefatos incluem o arquivo de controle e o código da extensão, que são itens que precisam estar presentes para que a extensão possa ser criada usando o comando `CREATE EXTENSION`. Em outras palavras, você ainda precisa criar a extensão para disponibilizar suas funções aos usuários do banco de dados.

3. Para criar a extensão, use o comando `CREATE EXTENSION` como você faz com qualquer outra extensão. Assim como acontece com outras extensões, o usuário do banco de dados precisa ter as permissões `CREATE` no banco de dados.

```
CREATE EXTENSION pg_distance;
```

4. Para testar a extensão TLE `pg_distance`, você pode usá-la para calcular a distância de Manhattan entre quatro pontos.

```
labdb=> SELECT manhattan_dist(1, 1, 5, 5);
8
```

Para calcular a distância euclidiana entre o mesmo conjunto de pontos, você pode usar o seguinte.

```
labdb=> SELECT euclidean_dist(1, 1, 5, 5);
5.656854249492381
```

A extensão `pg_distance` carrega as funções no banco de dados e as disponibiliza para qualquer usuário com permissões no banco de dados.

Modificar a extensão TLE

Para melhorar a performance da consulta para as funções empacotadas nessa extensão TLE, adicione os dois atributos do PostgreSQL a seguir às suas especificações.

- `IMMUTABLE`: o atributo `IMMUTABLE` garante que o otimizador de consultas possa usar otimizações para melhorar os tempos de resposta da consulta. Para obter mais informações, consulte Function Volatility Categories (Categorias de volatilidade de funções) na documentação do PostgreSQL.
- `PARALLEL SAFE`: o atributo `PARALLEL SAFE` é outro atributo que permite que o PostgreSQL execute a função no modo paralelo. Para obter mais informações, consulte CREATE FUNCTION na documentação do PostgreSQL.

No exemplo a seguir, você pode ver como a função `pgtle.install_update_path` é usada para adicionar esses atributos a cada função para criar uma versão `0.2` da extensão TLE `pg_distance`. Para ter mais informações sobre essa função, consulte pgtle.install_update_path. Você precisa ter a função `pgtle_admin` para realizar essa tarefa.

Como atualizar uma extensão TLE existente e especificar a versão padrão

1. Conecte-se à instância de banco de dados do RDS para PostgreSQL usando `psql` ou outra ferramenta de cliente, como o pgAdmin

```
psql --host=db-instance-123456789012.aws-region.rds.amazonaws.com
--port=5432 --username=postgres --password --dbname=labdb
```

2. Modifique a extensão TLE existente copiando o código a seguir e colando-o no console da sessão `psql`.

```
SELECT pgtle.install_update_path
(
 'pg_distance',
 '0.1',
 '0.2',
$_pg_tle_$
    CREATE OR REPLACE FUNCTION dist(x1 float8, y1 float8, x2 float8, y2 float8,
 norm int)
    RETURNS float8
    AS $$
      SELECT (abs(x2 - x1) ^ norm + abs(y2 - y1) ^ norm) ^ (1::float8 / norm);
    $$ LANGUAGE SQL IMMUTABLE PARALLEL SAFE;

    CREATE OR REPLACE FUNCTION manhattan_dist(x1 float8, y1 float8, x2 float8, y2
 float8)
    RETURNS float8
    AS $$
      SELECT dist(x1, y1, x2, y2, 1);
    $$ LANGUAGE SQL IMMUTABLE PARALLEL SAFE;

    CREATE OR REPLACE FUNCTION euclidean_dist(x1 float8, y1 float8, x2 float8, y2
 float8)
    RETURNS float8
    AS $$
      SELECT dist(x1, y1, x2, y2, 2);
    $$ LANGUAGE SQL IMMUTABLE PARALLEL SAFE;
$_pg_tle_$
);
```

Você verá uma resposta semelhante ao seguinte.

```
install_update_path
---------------------
 t
(1 row)
```

Você pode tornar essa versão da extensão a versão padrão, para que os usuários do banco de dados não precisem especificar uma versão ao criar ou atualizar a extensão em seu banco de dados.

3. Para especificar que a versão modificada (versão 0.2) de sua extensão TLE é a versão padrão, use a função `pgtle.set_default_version` conforme mostrado no exemplo a seguir.

```
SELECT pgtle.set_default_version('pg_distance', '0.2');
```

   Para ter mais informações sobre essa função, consulte pgtle.set_default_version.

4. Com o código implementado, você pode atualizar a extensão TLE instalada da maneira usual, utilizando o comando `ALTER EXTENSION ... UPDATE`, conforme mostrado aqui:

```
ALTER EXTENSION pg_distance UPDATE;
```

Descartar suas extensões TLE de um banco de dados

Você pode descartar suas extensões TLE usando o comando `DROP EXTENSION` da mesma forma que faz com outras extensões do PostgreSQL. Descartar a extensão não remove os arquivos de instalação que a compõem, o que permite aos usuários recriar a extensão. Para remover a extensão e seus arquivos de instalação, execute o processo de duas etapas a seguir.

Como descartar a extensão TLE e remover seus arquivos de instalação

1. Use `psql` ou outra ferramenta de cliente para se conectar à instância de banco de dados do RDS para PostgreSQL.

```
psql --host=.111122223333.aws-region.rds.amazonaws.com --port=5432 --
username=postgres --password --dbname=dbname
```

2. Descarte a extensão como você faria com qualquer extensão do PostgreSQL.

```
DROP EXTENSION your-TLE-extension
```

   Por exemplo, se você criar a extensão `pg_distance` conforme detalhado em Exemplo: Criar uma extensão de linguagem confiável usando SQL, poderá descartar a extensão da forma a seguir.

```
DROP EXTENSION pg_distance;
```

   Você vê uma saída confirmando que a extensão foi descartada da forma a seguir.

```
DROP EXTENSION
```

Neste ponto, a extensão não estará mais ativa no banco de dados. No entanto, seus arquivos de instalação e arquivo de controle ainda estão disponíveis no banco de dados, portanto, os usuários do banco de dados poderão criar a extensão novamente, se desejarem.

- Se quiser deixar os arquivos de extensão intactos para que os usuários do banco de dados possam criar sua extensão TLE, você pode parar neste ponto.
- Se quiser remover todos os arquivos que compõem a extensão, siga para a próxima etapa.

3. Para remover todos os arquivos de instalação da sua extensão, use a função `pgtle.uninstall_extension`. Essa função remove todos os arquivos de código e controle de sua extensão.

```
SELECT pgtle.uninstall_extension('your-tle-extension-name');
```

Por exemplo, para remover todos os arquivos de instalação `pg_distance`, use o comando a seguir.

```
SELECT pgtle.uninstall_extension('pg_distance');
 uninstall_extension
---------------------
 t
(1 row)
```

Desinstalar o Trusted Language Extensions para PostgreSQL

Se você não quiser mais criar suas próprias extensões TLE usando o TLE, poderá descartar a extensão `pg_tle` e remover todos os artefatos. Essa ação inclui descartar todas as extensões TLE no banco de dados e o esquema `pgtle`.

Como descartar a extensão **`pg_tle`** e seu esquema de um banco de dados

1. Use `psql` ou outra ferramenta de cliente para se conectar à instância de banco de dados do RDS para PostgreSQL.

```
psql --host=.111122223333.aws-region.rds.amazonaws.com --port=5432 --
username=postgres --password --dbname=dbname
```

2. Descarte a extensão `pg_tle` do banco de dados. Se o banco de dados tiver suas próprias extensões TLE ainda em execução no banco de dados, você também precisará descartar essas extensões. Para isso, você pode usar a palavra-chave `CASCADE`, conforme mostrado a seguir.

   ```
   DROP EXTENSION pg_tle CASCADE;
   ```

   Se a extensão `pg_tle` ainda não estiver ativa no banco de dados, você não precisará usar a palavra-chave `CASCADE`.

3. Descarte o esquema `pgtle`. Essa ação remove todas as funções de gerenciamento do banco de dados.

   ```
   DROP SCHEMA pgtle CASCADE;
   ```

   O comando retornará o seguinte quando o processo for concluído.

   ```
   DROP SCHEMA
   ```

   A extensão `pg_tle`, seu esquema e funções e todos os artefatos são removidos. Para criar extensões usando o TLE, execute o processo de configuração novamente. Para ter mais informações, consulte Configurar o Trusted Language Extensions em sua instância de banco de dados do RDS para PostgreSQL.

Usar ganchos do PostgreSQL com suas extensões TLE

Um gancho é um mecanismo de retorno de chamada disponível no PostgreSQL que possibilita aos desenvolvedores chamar funções personalizadas ou outras rotinas durante operações regulares de banco de dados. O kit de desenvolvimento TLE é compatível com ganchos do PostgreSQL para que você possa integrar funções personalizadas com o comportamento do PostgreSQL no ambiente de execução. Por exemplo, você pode usar um gancho para associar o processo de autenticação ao seu próprio código personalizado ou modificar o processo de planejamento e execução de consultas de acordo com suas necessidades específicas.

Suas extensões TLE podem usar ganchos. Se um gancho tiver escopo global, ele se aplicará a todos os bancos de dados. Portanto, se sua extensão TLE usar um gancho global, você precisará criar sua extensão TLE em todos os bancos de dados que seus usuários puderem acessar.

Ao usar a extensão `pg_tle` para criar seu próprio Trusted Language Extensions, você pode usar os ganchos disponíveis de uma API SQL para criar as funções de sua extensão. Você deve registrar todos os ganchos com `pg_tle`. Para alguns ganchos, talvez você também precise definir vários parâmetros de configuração. Por exemplo, o gancho de verificação `passcode` pode ser configurado como ativado, desativado ou obrigatório. Para obter mais informações sobre os requisitos específicos dos ganchos `pg_tle` disponíveis, consulte Referência de ganchos para Trusted Language Extensions para PostgreSQL.

Exemplo: Criar uma extensão que use um gancho do PostgreSQL

O exemplo abordado nesta seção usa um gancho do PostgreSQL para conferir a senha fornecida durante operações específicas de SQL e impede que os usuários do banco de dados definam suas senhas como qualquer uma das contidas na tabela `password_check.bad_passwords`. A tabela contém as dez opções de senhas mais usadas, mas facilmente identificáveis.

Para configurar esse exemplo em sua instância de banco de dados do RDS para PostgreSQL, você já deve ter instalado o Trusted Language Extensions. Para obter detalhes, consulte Configurar o Trusted Language Extensions em sua instância de banco de dados do RDS para PostgreSQL.

Como configurar o exemplo de gancho de verificação de senha

1. Use o `psql` para se conectar à Instância de banco de dados do RDS para PostgreSQL.

```
psql --host=db-instance-123456789012.aws-region.rds.amazonaws.com
--port=5432 --username=postgres --password --dbname=labdb
```

2. Copie o código do Lista de códigos do gancho de verificação de senha e cole-o no banco de dados.

```
SELECT pgtle.install_extension (
  'my_password_check_rules',
  '1.0',
  'Do not let users use the 10 most commonly used passwords',
$_pgtle_$
  CREATE SCHEMA password_check;
  REVOKE ALL ON SCHEMA password_check FROM PUBLIC;
  GRANT USAGE ON SCHEMA password_check TO PUBLIC;
```

```
 CREATE TABLE password_check.bad_passwords (plaintext) AS
 VALUES
   ('123456'),
   ('password'),
   ('12345678'),
   ('qwerty'),
   ('123456789'),
   ('12345'),
   ('1234'),
   ('111111'),
   ('1234567'),
   ('dragon');
 CREATE UNIQUE INDEX ON password_check.bad_passwords (plaintext);

 CREATE FUNCTION password_check.passcheck_hook(username text, password text,
password_type pgtle.password_types, valid_until timestamptz, valid_null boolean)
 RETURNS void AS $$
   DECLARE
     invalid bool := false;
   BEGIN
     IF password_type = 'PASSWORD_TYPE_MD5' THEN
       SELECT EXISTS(
         SELECT 1
         FROM password_check.bad_passwords bp
         WHERE ('md5' || md5(bp.plaintext || username)) = password
       ) INTO invalid;
       IF invalid THEN
         RAISE EXCEPTION 'Cannot use passwords from the common password
dictionary';
       END IF;
     ELSIF password_type = 'PASSWORD_TYPE_PLAINTEXT' THEN
       SELECT EXISTS(
         SELECT 1
         FROM password_check.bad_passwords bp
         WHERE bp.plaintext = password
       ) INTO invalid;
       IF invalid THEN
         RAISE EXCEPTION 'Cannot use passwords from the common password
dictionary';
       END IF;
     END IF;
   END
 $$ LANGUAGE plpgsql SECURITY DEFINER;
```

```
  GRANT EXECUTE ON FUNCTION password_check.passcheck_hook TO PUBLIC;

  SELECT pgtle.register_feature('password_check.passcheck_hook', 'passcheck');
$_pgtle_$
);
```

Quando a extensão for carregada em seu banco de dados, você verá uma saída como a seguinte.

```
 install_extension
-------------------
 t
(1 row)
```

3. Enquanto ainda estiver conectado ao banco de dados, agora você poderá criar a extensão.

```
CREATE EXTENSION my_password_check_rules;
```

4. Você pode confirmar que a extensão foi criada no banco de dados usando o metacomando `psql` a seguir.

```
\dx
                        List of installed extensions
          Name           | Version |   Schema   |
 Description
-------------------------+---------+------------
+-------------------------------------------------------------
 my_password_check_rules | 1.0     | public     | Prevent use of any of the top-ten
 most common bad passwords
 pg_tle                  | 1.0.1   | pgtle      | Trusted-Language Extensions for
 PostgreSQL
 plpgsql                 | 1.0     | pg_catalog | PL/pgSQL procedural language
(3 rows)
```

5. Abra outra sessão do terminal para trabalhar com o AWS CLI. Você precisa modificar seu grupo de parâmetros de banco de dados personalizado para ativar o gancho de verificação de senha. Para isso, use o comando modify-db-parameter-group da CLI, conforme mostrado no exemplo a seguir.

```
aws rds modify-db-parameter-group \
```

```
    --region aws-region \
    --db-parameter-group-name your-custom-parameter-group \
    --parameters
 "ParameterName=pgtle.enable_password_check,ParameterValue=on,ApplyMethod=immediate"
```

Quando o parâmetro é ativado com êxito, você vê uma saída como a seguinte.

```
(
    "DBParameterGroupName": "docs-lab-parameters-for-tle"
}
```

Poderá levar alguns minutos para que a alteração na configuração do grupo de parâmetros tenha efeito. No entanto, esse parâmetro é dinâmico, portanto, você não precisa reiniciar a instância de banco de dados do RDS para PostgreSQL para que a configuração tenha efeito.

6. Abra a sessão `psql` e consulte o banco de dados para verificar se o gancho password_check foi ativado.

```
labdb=> SHOW pgtle.enable_password_check;
pgtle.enable_password_check
----------------------------
on
(1 row)
```

O gancho de verificação de senha agora está ativo. É possível testá-lo criando uma função e usando uma das senhas incorretas, conforme mostrado no exemplo a seguir.

```
CREATE ROLE test_role PASSWORD 'password';
ERROR:  Cannot use passwords from the common password dictionary
CONTEXT:  PL/pgSQL function
 password_check.passcheck_hook(text,text,pgtle.password_types,timestamp with time
 zone,boolean) line 21 at RAISE
SQL statement "SELECT password_check.passcheck_hook(
    $1::pg_catalog.text,
    $2::pg_catalog.text,
    $3::pgtle.password_types,
    $4::pg_catalog.timestamptz,
    $5::pg_catalog.bool)"
```

A saída foi formatada para facilitar a leitura.

O exemplo a seguir mostra que o comportamento do metacomando interativo `pgsql` `\password` também é afetado pelo gancho password_check.

```
postgres=> SET password_encryption TO 'md5';
SET
postgres=> \password
Enter new password for user "postgres":*****
Enter it again:*****
ERROR:  Cannot use passwords from the common password dictionary
CONTEXT:  PL/pgSQL function
 password_check.passcheck_hook(text,text,pgtle.password_types,timestamp with time
 zone,boolean) line 12 at RAISE
SQL statement "SELECT password_check.passcheck_hook($1::pg_catalog.text,
 $2::pg_catalog.text, $3::pgtle.password_types, $4::pg_catalog.timestamptz,
 $5::pg_catalog.bool)"
```

Você poderá descartar essa extensão TLE e desinstalar seus arquivos de origem, se desejar. Para obter mais informações, consulte Descartar suas extensões TLE de um banco de dados.

Lista de códigos do gancho de verificação de senha

O código de exemplo mostrado aqui define a especificação da extensão TLE `my_password_check_rules`. Quando você copia esse código e o cola em seu banco de dados, o código da extensão `my_password_check_rules` é carregado no banco de dados e o gancho `password_check` é registrado para uso pela extensão.

```
SELECT pgtle.install_extension (
  'my_password_check_rules',
  '1.0',
  'Do not let users use the 10 most commonly used passwords',
$_pgtle_$
  CREATE SCHEMA password_check;
  REVOKE ALL ON SCHEMA password_check FROM PUBLIC;
  GRANT USAGE ON SCHEMA password_check TO PUBLIC;

  CREATE TABLE password_check.bad_passwords (plaintext) AS
  VALUES
    ('123456'),
    ('password'),
    ('12345678'),
    ('qwerty'),
    ('123456789'),
```

```
    ('12345'),
    ('1234'),
    ('111111'),
    ('1234567'),
    ('dragon');
  CREATE UNIQUE INDEX ON password_check.bad_passwords (plaintext);

  CREATE FUNCTION password_check.passcheck_hook(username text, password text,
 password_type pgtle.password_types, valid_until timestamptz, valid_null boolean)
  RETURNS void AS $$
    DECLARE
      invalid bool := false;
    BEGIN
      IF password_type = 'PASSWORD_TYPE_MD5' THEN
        SELECT EXISTS(
          SELECT 1
          FROM password_check.bad_passwords bp
          WHERE ('md5' || md5(bp.plaintext || username)) = password
        ) INTO invalid;
        IF invalid THEN
          RAISE EXCEPTION 'Cannot use passwords from the common password dictionary';
        END IF;
      ELSIF password_type = 'PASSWORD_TYPE_PLAINTEXT' THEN
        SELECT EXISTS(
          SELECT 1
          FROM password_check.bad_passwords bp
          WHERE bp.plaintext = password
        ) INTO invalid;
        IF invalid THEN
          RAISE EXCEPTION 'Cannot use passwords from the common password dictionary';
        END IF;
      END IF;
    END
  $$ LANGUAGE plpgsql SECURITY DEFINER;

  GRANT EXECUTE ON FUNCTION password_check.passcheck_hook TO PUBLIC;

  SELECT pgtle.register_feature('password_check.passcheck_hook', 'passcheck');
$_pgtle_$
);
```

Usar tipos de dados personalizados no TLE

O PostgreSQL comporta comandos para registrar novos tipos básicos (também conhecidos como tipos escalares) para lidar de forma eficiente com estruturas de dados complexas no banco de dados. Um tipo básico permite que você personalize como os dados são armazenados internamente e como convertê-los de e em uma representação textual externa. Esses tipos de dados personalizados são úteis ao estender o PostgreSQL para comportar domínios funcionais em que um tipo integrado, como número ou texto, não pode fornecer semântica de pesquisa suficiente.

O RDS para PostgreSQL permite que você crie tipos de dados personalizados em sua extensão de linguagem confiável e defina funções que comportem operações de SQL e indexação para esses novos tipos de dados. Os tipos de dados estão disponíveis para as seguintes versões:

- RDS para PostgreSQL 15.7 e as versões 15 posteriores
- RDS para PostgreSQL 14.9 e versões 14 posteriores
- RDS para PostgreSQL 13.2 e versões 13 posteriores

Para obter mais informações, consulte Tipos de Trusted Language Base.

Referência de funções para Trusted Language Extensions para PostgreSQL

Veja a documentação de referência a seguir sobre as funções disponíveis em Trusted Language Extensions para PostgreSQL. Use essas funções para instalar, registrar, atualizar e gerenciar suas extensões TLE, ou seja, as extensões do PostgreSQL que você desenvolve usando o kit de desenvolvimento Trusted Language Extensions.

Funções

- pgtle.available_extensions
- pgtle.available_extension_versions
- pgtle.extension_update_paths
- pgtle.install_extension
- pgtle.install_update_path
- pgtle.register_feature
- pgtle.register_feature_if_not_exists

- pgtle.set_default_version
- pgtle.uninstall_extension (nome)
- pgtle.uninstall_extension (nome, versão)
- pgtle.uninstall_extension_if_exists
- pgtle.uninstall_update_path
- pgtle.uninstall_update_path_if_exists
- pgtle.unregister_feature
- pgtle.unregister_feature_if_exists

pgtle.available_extensions

A função `pgtle.available_extensions` é uma função de retorno de conjunto. Ela retorna todas as extensões TLE disponíveis no banco de dados. Cada linha retornada contém informações sobre uma única extensão TLE.

Protótipo de função

```
pgtle.available_extensions()
```

Função

Nenhum.

Argumentos

Nenhum.

Saída

- `name`: o nome da extensão TLE.
- `default_version`: a versão da extensão TLE a ser usada quando `CREATE EXTENSION` é chamada sem uma versão especificada.
- `description`: uma descrição mais detalhada sobre a extensão TLE.

Exemplo de uso

```
SELECT * FROM pgtle.available_extensions();
```

pgtle.available_extension_versions

A função `available_extension_versions` é de retorno de conjunto. Ela retorna uma lista de todas as extensões TLE disponíveis e suas versões. Cada linha contém informações sobre uma versão específica de determinada extensão TLE, incluindo se ela requer uma função específica.

Protótipo de função

```
pgtle.available_extension_versions()
```

Função

Nenhum.

Argumentos

Nenhum.

Saída

- `name`: o nome da extensão TLE.
- `version`: a versão da extensão TLE.
- `superuser`: esse valor é sempre `false` para suas extensões TLE. As permissões necessárias para criar a extensão TLE ou atualizá-la são as mesmas para criar outros objetos em determinado banco de dados.
- `trusted`: esse valor é sempre `false` para uma extensão TLE.
- `relocatable`: esse valor é sempre `false` para uma extensão TLE.
- `schema`: especifica o nome do esquema no qual a extensão TLE está instalada.
- `requires`: uma matriz contendo os nomes de outras extensões necessárias para essa extensão TLE.
- `description`: uma descrição detalhada da extensão TLE.

Para obter mais informações sobre valores de saída, consulte Packaging Related Objects into an Extension > Extension Files (Compactação de objetos relacionados em uma extensão > arquivos de extensão) na documentação do PostgreSQL.

Exemplo de uso

```
SELECT * FROM pgtle.available_extension_versions();
```

pgtle.extension_update_paths

A função `extension_update_paths` é de retorno de conjunto. Ela retorna uma lista de todos os caminhos de atualização possíveis para uma extensão TLE. Cada linha inclui as atualizações ou downgrades disponíveis para essa extensão TLE.

Protótipo de função

```
pgtle.extension_update_paths(name)
```

Função

Nenhum.

Argumentos

`name`: o nome da extensão TLE da qual obter caminhos de atualização.

Saída

- `source`: a versão de origem de uma atualização.
- `target`: a versão de destino de uma atualização.
- `path`: o caminho de atualização usado para atualizar uma extensão TLE da versão `source` para a `target`, por exemplo, `0.1--0.2`.

Exemplo de uso

```
SELECT * FROM pgtle.extension_update_paths('your-TLE');
```

pgtle.install_extension

A função `install_extension` permite que você instale os artefatos que compõem sua extensão TLE no banco de dados, após o qual ela pode ser criada usando o comando `CREATE EXTENSION`.

Protótipo de função

```
pgtle.install_extension(name text, version text, description text, ext text, requires
 text[] DEFAULT NULL::text[])
```

Função

Nenhum.

Argumentos

- `name`: o nome da extensão TLE. Esse valor é usado ao chamar`CREATE EXTENSION`.
- `version`: a versão da extensão TLE.
- `description`: uma descrição detalhada da extensão TLE. Essa descrição é exibida no campo `comment` em `pgtle.available_extensions()`.
- `ext`: o conteúdo da extensão TLE. Esse valor contém objetos, como funções.
- `requires`: um parâmetro opcional que especifica dependências para essa extensão TLE. A extensão `pg_tle` é adicionada automaticamente como uma dependência.

Muitos desses argumentos são iguais aos incluídos em um arquivo de controle de extensão para instalar uma extensão do PostgreSQL no sistema de arquivos de uma instância do PostgreSQL. Para obter mais informações, consulte Extension Files (Arquivos de extensão) em Packaging Related Objects into an Extension (Compactação de objetos relacionados em uma extensão) na documentação do PostgreSQL.

Saída

Essa função retorna `OK` em caso de sucesso e `NULL` em caso de erro.

- `OK`: a extensão TLE foi instalada com êxito no banco de dados.
- `NULL`: a extensão TLE não foi instalada com êxito no banco de dados.

Exemplo de uso

```
SELECT pgtle.install_extension(
 'pg_tle_test',
 '0.1',
 'My first pg_tle extension',
```

```
$_pgtle_$
  CREATE FUNCTION my_test()
  RETURNS INT
  AS $$
    SELECT 42;
  $$ LANGUAGE SQL IMMUTABLE;
$_pgtle_$
);
```

pgtle.install_update_path

A função `install_update_path` fornece um caminho de atualização entre duas versões diferentes de uma extensão TLE. Essa função permite que os usuários de sua extensão TLE atualizem sua versão usando a sintaxe `ALTER EXTENSION ... UPDATE`.

Protótipo de função

```
pgtle.install_update_path(name text, fromvers text, tovers text, ext text)
```

Função

`pgtle_admin`

Argumentos

- `name`: o nome da extensão TLE. Esse valor é usado ao chamar`CREATE EXTENSION`.
- `fromvers`: a versão de origem da extensão TLE para a atualização.
- `tovers`: a versão de destino da extensão TLE para a atualização.
- `ext`: o conteúdo da atualização. Esse valor contém objetos, como funções.

Saída

Nenhum.

Exemplo de uso

```
SELECT pgtle.install_update_path('pg_tle_test', '0.1', '0.2',
  $_pgtle_$
    CREATE OR REPLACE FUNCTION my_test()
    RETURNS INT
    AS $$
```

```
      SELECT 21;
    $$ LANGUAGE SQL IMMUTABLE;
  $_pgtle_$
);
```

pgtle.register_feature

A função `register_feature` adiciona o recurso interno especificado do PostgreSQL à tabela `pgtle.feature_info`. Os ganchos do PostgreSQL são um exemplo de um recurso interno do PostgreSQL. O kit de desenvolvimento Trusted Language Extensions é compatível com o uso de ganchos do PostgreSQL. Atualmente, essa função é compatível com o recurso a seguir.

- `passcheck`: registra o gancho de verificação de senha com seu procedimento ou função que personaliza o comportamento de verificação de senha do PostgreSQL.

Protótipo de função

```
pgtle.register_feature(proc regproc, feature pg_tle_feature)
```

Função

`pgtle_admin`

Argumentos

- `proc`: o nome de uma função ou um procedimento armazenado a ser usado para o recurso.
- `feature`: o nome do recurso `pg_tle` (como`passcheck`) a ser registrado na função.

Saída

Nenhum.

Exemplo de uso

```
SELECT pgtle.register_feature('pw_hook', 'passcheck');
```

pgtle.register_feature_if_not_exists

A função `pgtle.register_feature_if_not_exists` adiciona o recurso do PostgreSQL especificado à tabela `pgtle.feature_info` e identifica a extensão TLE ou outro procedimento

ou função que usa o recurso. Para obter mais informações sobre ganchos e Trusted Language Extensions, consulteUsar ganchos do PostgreSQL com suas extensões TLE.

Protótipo de função

```
pgtle.register_feature_if_not_exists(proc regproc, feature pg_tle_feature)
```

Função

`pgtle_admin`

Argumentos

- `proc`: o nome de uma função ou um procedimento armazenado que contém a lógica (código) a ser usada como um recurso para sua extensão TLE. Por exemplo, o código `pw_hook`.
- `feature`: o nome do recurso do PostgreSQL a ser registrado na função TLE. Atualmente, o único recurso disponível é o gancho `passcheck`. Para ter mais informações, consulte Gancho de verificação de senha (passcheck).

Saída

Retorna `true` após registrar o recurso para a extensão especificada. Retorna `false` se o recurso já estiver registrado.

Exemplo de uso

```
SELECT pgtle.register_feature_if_not_exists('pw_hook', 'passcheck');
```

pgtle.set_default_version

A função `set_default_version` permite que você especifique uma `default_version` para sua extensão TLE. Você pode usar essa função para definir um caminho de atualização e designar a versão como padrão para sua extensão TLE. Quando os usuários do banco de dados especificam sua extensão TLE nos comandos `CREATE EXTENSION` e `ALTER EXTENSION ... UPDATE`, essa versão da extensão TLE é criada no banco de dados para esse usuário.

Essa função retorna `true` em caso de êxito. Se a extensão TLE especificada no argumento `name` não existir, a função retornará um erro. Da mesma forma, se a `version` da extensão TLE não existir, ela retornará um erro.

Protótipo de função

```
pgtle.set_default_version(name text, version text)
```

Função

`pgtle_admin`

Argumentos

- `name`: o nome da extensão TLE. Esse valor é usado ao chamar`CREATE EXTENSION`.
- `version`: a versão da extensão TLE para definir o padrão.

Saída

- `true`: quando a configuração da versão padrão é bem-sucedida, a função retorna `true`.
- `ERROR`: retornará uma mensagem de erro se uma extensão TLE com o nome ou versão especificados não existir.

Exemplo de uso

```
SELECT * FROM pgtle.set_default_version('my-extension', '1.1');
```

pgtle.uninstall_extension (nome)

A função `uninstall_extension` remove todas as versões de uma extensão TLE de um banco de dados. Essa função impede que futuras chamadas de `CREATE EXTENSION` instalem a extensão TLE. Se a extensão TLE não existir no banco de dados, um erro será gerado.

A função `uninstall_extension` não descartará uma extensão TLE se ela estiver atualmente ativa no banco de dados. Para remover uma extensão TLE que está ativa no momento, você precisa chamar explicitamente `DROP EXTENSION` para removê-la.

Protótipo de função

```
pgtle.uninstall_extension(extname text)
```

Função

`pgtle_admin`

Argumentos

- `extname`: o nome da extensão TLE a ser desinstalada. Esse nome é o mesmo usado com `CREATE EXTENSION` para carregar a extensão TLE para uso em determinado banco de dados.

Saída

Nenhum.

Exemplo de uso

```
SELECT * FROM pgtle.uninstall_extension('pg_tle_test');
```

pgtle.uninstall_extension (nome, versão)

A função `uninstall_extension(name, version)` remove a versão especificada da extensão TLE do banco de dados. Essa função impede que `CREATE EXTENSION` e `ALTER EXTENSION` instalem ou atualizem uma extensão TLE para a versão especificada. Essa função também remove todos os caminhos de atualização para a versão especificada da extensão TLE. Essa função não desinstalará a extensão TLE se ela estiver atualmente ativa no banco de dados. Você deve chamar explicitamente `DROP EXTENSION` para remover a extensão TLE. Para desinstalar todas as versões de uma extensão TLE, consultepgtle.uninstall_extension (nome).

Protótipo de função

```
pgtle.uninstall_extension(extname text, version text)
```

Função

`pgtle_admin`

Argumentos

- `extname`: o nome da extensão TLE. Esse valor é usado ao chamar`CREATE EXTENSION`.
- `version`: a versão da extensão TLE a ser desinstalada do banco de dados.

Saída

Nenhum.

Exemplo de uso

```
SELECT * FROM pgtle.uninstall_extension('pg_tle_test', '0.2');
```

pgtle.uninstall_extension_if_exists

A função `uninstall_extension_if_exists` remove todas as versões de uma extensão TLE de determinado banco de dados. Se a extensão TLE não existir, a função retornará silenciosamente (nenhuma mensagem de erro será gerada). Se a extensão especificada estiver atualmente ativa em um banco de dados, essa função não a descartará. Você deve chamar explicitamente `DROP EXTENSION` para remover a extensão TLE antes de usar essa função para desinstalar seus artefatos.

Protótipo de função

```
pgtle.uninstall_extension_if_exists(extname text)
```

Função

`pgtle_admin`

Argumentos

- `extname`: o nome da extensão TLE. Esse valor é usado ao chamar `CREATE EXTENSION`.

Saída

A função `uninstall_extension_if_exists` retorna `true` após a desinstalação da extensão especificada. Se a extensão especificada não existir, a função retornará `false`.

- `true`: retorna `true` após a desinstalação da extensão TLE.
- `false`: retorna `false` quando a extensão TLE não existe no banco de dados.

Exemplo de uso

```
SELECT * FROM pgtle.uninstall_extension_if_exists('pg_tle_test');
```

pgtle.uninstall_update_path

A função `uninstall_update_path` remove o caminho de atualização especificado de uma extensão TLE. Isso impede `ALTER EXTENSION ... UPDATE TO` de usar isso como um caminho de atualização.

Se a extensão TLE estiver sendo usada atualmente por uma das versões desse caminho de atualização, ela permanecerá no banco de dados.

Se o caminho de atualização especificado não existir, essa função gerará um erro.

Protótipo de função

```
pgtle.uninstall_update_path(extname text, fromvers text, tovers text)
```

Função

`pgtle_admin`

Argumentos

- `extname`: o nome da extensão TLE. Esse valor é usado ao chamar`CREATE EXTENSION`.
- `fromvers`: a versão de origem da extensão TLE usada no caminho de atualização.
- `tovers`: a versão de destino da extensão TLE usada no caminho de atualização.

Saída

Nenhum.

Exemplo de uso

```
SELECT * FROM pgtle.uninstall_update_path('pg_tle_test', '0.1', '0.2');
```

pgtle.uninstall_update_path_if_exists

A função `uninstall_update_path_if_exists` é semelhante a `uninstall_update_path` no sentido de remover o caminho de atualização especificado de uma extensão TLE. No entanto, se o caminho de atualização não existir, essa função não gerará uma mensagem de erro. Em vez disso, a função retornará `false`.

Protótipo de função

```
pgtle.uninstall_update_path_if_exists(extname text, fromvers text, tovers text)
```

Função

`pgtle_admin`

Argumentos

- `extname`: o nome da extensão TLE. Esse valor é usado ao chamar`CREATE EXTENSION`.
- `fromvers`: a versão de origem da extensão TLE usada no caminho de atualização.
- `tovers`: a versão de destino da extensão TLE usada no caminho de atualização.

Saída

- `true`: a função atualizou com êxito o caminho da extensão TLE.
- `false`: a função não conseguiu atualizar o caminho da extensão TLE.

Exemplo de uso

```
SELECT * FROM pgtle.uninstall_update_path_if_exists('pg_tle_test', '0.1', '0.2');
```

pgtle.unregister_feature

A função `unregister_feature` fornece uma maneira de remover funções que foram registradas para usar recursos `pg_tle`, como ganchos. Para obter informações sobre como registrar uma recurso, consulte pgtle.register_feature.

Protótipo de função

```
pgtle.unregister_feature(proc regproc, feature pg_tle_features)
```

Função

`pgtle_admin`

Argumentos

- `proc`: o nome de uma função armazenada a ser registrada em um recurso `pg_tle`.

- `feature`: o nome do recurso `pg_tle` a ser registrado na função. Por exemplo, `passcheck` é um recurso que pode ser registrado para uso pelas extensões de linguagem confiáveis desenvolvidas por você. Para ter mais informações, consulte Gancho de verificação de senha (passcheck).

Saída

Nenhum.

Exemplo de uso

```
SELECT * FROM pgtle.unregister_feature('pw_hook', 'passcheck');
```

pgtle.unregister_feature_if_exists

A função `unregister_feature` fornece uma maneira de remover funções que foram registradas para usar recursos `pg_tle`, como ganchos. Para ter mais informações, consulte Usar ganchos do PostgreSQL com suas extensões TLE. Retorna `true` após cancelar o registro do recurso com êxito. Retorna `false` se o recurso não foi registrado.

Para obter informações sobre como registrar recursos `pg_tle` para suas extensões TLE, consulte pgtle.register_feature.

Protótipo de função

```
pgtle.unregister_feature_if_exists('proc regproc', 'feature pg_tle_features')
```

Função

`pgtle_admin`

Argumentos

- `proc`: o nome da função armazenada que foi registrada para incluir um recurso `pg_tle`.
- `feature`: o nome do recurso `pg_tle` que foi registrado com a extensão de linguagem confiável.

Saída

Retorna `true` ou `false` da maneira a seguir.

- `true`: a função cancelou com êxito o registro do recurso da extensão.

- `false`: a função não conseguiu cancelar o registro do recurso da extensão TLE.

Exemplo de uso

```
SELECT * FROM pgtle.unregister_feature_if_exists('pw_hook', 'passcheck');
```

Referência de ganchos para Trusted Language Extensions para PostgreSQL

O Trusted Language Extensions para PostgreSQL é compatível com ganchos do PostgreSQL. Um gancho é um mecanismo interno de retorno de chamada disponível para que os desenvolvedores estendam a funcionalidade principal do PostgreSQL. Usando ganchos, os desenvolvedores podem implementar suas próprias funções ou procedimentos para uso durante várias operações de banco de dados, modificando assim o comportamento do PostgreSQL de alguma forma. Por exemplo, você pode usar um gancho `passcheck` para personalizar a forma como o PostgreSQL manipula as senhas fornecidas ao criar ou alterar senhas para usuários (funções).

Veja a documentação a seguir para saber mais sobre o hook passcheck disponível para as extensões TLE. Para saber mais sobre os hooks disponíveis, incluindo o hook de autenticação do cliente, consulte Trusted Language Extensions hooks.

Gancho de verificação de senha (passcheck)

O gancho `passcheck` é usado para personalizar o comportamento do PostgreSQL durante o processo de verificação de senha para os comandos SQL e o metacomando `psql` a seguir.

- `CREATE ROLE` *`username`* `...PASSWORD`: para obter mais informações, consulte CREATE ROLE na documentação do PostgreSQL.
- `ALTER ROLE` *`username`*`...PASSWORD`: para obter mais informações, consulte ALTER ROLE na documentação do PostgreSQL.
- `\password` *`username`*: esse metacomando interativo `psql` altera com segurança a senha do usuário especificado usando o hash da senha antes de usar a sintaxe `ALTER ROLE ... PASSWORD` de forma transparente. O metacomando é um invólucro seguro para o comando `ALTER ROLE ... PASSWORD`, portanto, o gancho se aplica ao comportamento do metacomando `psql`.

Para obter um exemplo, consulte Lista de códigos do gancho de verificação de senha.

Sumário

- Protótipo de função
- Argumentos
- Configuração
- Observações de uso

Protótipo de função

```
passcheck_hook(username text, password text, password_type pgtle.password_types,
 valid_until timestamptz, valid_null boolean)
```

Argumentos

A função de gancho `passcheck` usa os seguintes argumentos:

- `username`: o nome (como texto) da função (nome de usuário) que está definindo uma senha.
- `password`: o texto simples ou a senha com hash. A senha digitada deve corresponder ao tipo especificado em `password_type`.
- `password_type`: especifique o formato `pgtle.password_type` da senha. Esse formato pode ser uma das opções a seguir.
  - `PASSWORD_TYPE_PLAINTEXT`: uma senha de texto simples.
  - `PASSWORD_TYPE_MD5`: uma senha que foi criptografada usando o algoritmo MD5 (resumo de mensagens 5).
  - `PASSWORD_TYPE_SCRAM_SHA_256`: uma senha que foi criptografada usando o algoritmo SCRAM-SHA-256.
- `valid_until`: especifique a hora em que a senha se torna inválida. Esse argumento é opcional. Se você usar esse argumento, especifique a hora como um valor `timestamptz`.
- `valid_null`: se esse valor booliano estiver definido como `true`, a opção `valid_until` será definida como `NULL`.

Configuração

A função `pgtle.enable_password_check` controla se o gancho passcheck está ativo. O gancho passcheck tem três configurações possíveis.

- `off`: desativa o gancho de verificação de senha `passcheck`. Este é o valor padrão.
- on: ativa o gancho de verificação de senha `passcode` para que as senhas sejam conferidas na tabela.
- `require`: requer que um gancho de verificação de senha seja definido.

Observações de uso

Para ativar ou desativar o gancho `passcheck`, você precisa modificar o grupo de parâmetros de banco de dados personalizado para a instância de banco de dados do RDS para PostgreSQL.

Para Linux, macOS ou Unix:

```
aws rds modify-db-parameter-group \
    --region aws-region \
    --db-parameter-group-name your-custom-parameter-group \
    --parameters
 "ParameterName=pgtle.enable_password_check,ParameterValue=on,ApplyMethod=immediate"
```

Para Windows:

```
aws rds modify-db-parameter-group ^
    --region aws-region ^
    --db-parameter-group-name your-custom-parameter-group ^
    --parameters
 "ParameterName=pgtle.enable_password_check,ParameterValue=on,ApplyMethod=immediate"
```

Exemplos de código para o Amazon RDS usando AWS SDKs

Os exemplos de código a seguir mostram como usar o Amazon RDS com um Kit de desenvolvimento de software (SDK) da AWS.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço específicas, é possível ver as ações contextualizadas em seus devidos cenários e exemplos entre serviços.

Cenários são exemplos de código que mostram como realizar uma tarefa específica chamando várias funções dentro do mesmo serviço.

Exemplos entre serviços são amostras de aplicações que funcionam em vários Serviços da AWS.

Para obter uma lista completa dos Guias do desenvolvedor do SDK da AWS e exemplos de código, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Conceitos básicos

Olá, Amazon RDS

O exemplo de código a seguir mostra como começar a usar o Amazon RDS.

.NET

SDK para .NET

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
using System;
using System.Threading.Tasks;
using Amazon.RDS;
```

```
using Amazon.RDS.Model;

namespace RDSActions;

public static class HelloRds
{
    static async Task Main(string[] args)
    {
        var rdsClient = new AmazonRDSClient();

        Console.WriteLine($"Hello Amazon RDS! Following are some of your DB
 instances:");
        Console.WriteLine();

        // You can use await and any of the async methods to get a response.
        // Let's get the first twenty DB instances.
        var response = await rdsClient.DescribeDBInstancesAsync(
            new DescribeDBInstancesRequest()
            {
                MaxRecords = 20 // Must be between 20 and 100.
            });

        foreach (var instance in response.DBInstances)
        {
            Console.WriteLine($"\tDB name: {instance.DBName}");
            Console.WriteLine($"\tArn: {instance.DBInstanceArn}");
            Console.WriteLine($"\tIdentifier: {instance.DBInstanceIdentifier}");
            Console.WriteLine();
        }
    }
}
```

- Para obter detalhes da API, consulte DescribeDBInstances na Referência da API AWS SDK para .NET.

C++

SDK para C++

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

Código para o arquivo CMakeLists.txt do CMake.

```
# Set the minimum required version of CMake for this project.
cmake_minimum_required(VERSION 3.13)

# Set the AWS service components used by this project.
set(SERVICE_COMPONENTS rds)

# Set this project's name.
project("hello_rds")

# Set the C++ standard to use to build this target.
# At least C++ 11 is required for the AWS SDK for C++.
set(CMAKE_CXX_STANDARD 11)

# Use the MSVC variable to determine if this is a Windows build.
set(WINDOWS_BUILD ${MSVC})

if (WINDOWS_BUILD) # Set the location where CMake can find the installed
 libraries for the AWS SDK.
    string(REPLACE ";" "/aws-cpp-sdk-all;" SYSTEM_MODULE_PATH
 "${CMAKE_SYSTEM_PREFIX_PATH}/aws-cpp-sdk-all")
    list(APPEND CMAKE_PREFIX_PATH ${SYSTEM_MODULE_PATH})
endif ()

# Find the AWS SDK for C++ package.
find_package(AWSSDK REQUIRED COMPONENTS ${SERVICE_COMPONENTS})

if (WINDOWS_BUILD AND AWSSDK_INSTALL_AS_SHARED_LIBS)
     # Copy relevant AWS SDK for C++ libraries into the current binary directory
 for running and debugging.
```

```
    # set(BIN_SUB_DIR "/Debug") # If you are building from the command line, you
 may need to uncomment this
                                    # and set the proper subdirectory to the
 executables' location.

    AWSSDK_CPY_DYN_LIBS(SERVICE_COMPONENTS ""
 ${CMAKE_CURRENT_BINARY_DIR}${BIN_SUB_DIR})
endif ()

add_executable(${PROJECT_NAME}
        hello_rds.cpp)

target_link_libraries(${PROJECT_NAME}
        ${AWSSDK_LINK_LIBRARIES})
```

Código para o arquivo de origem hello_rds.cpp.

```
#include <aws/core/Aws.h>
#include <aws/rds/RDSClient.h>
#include <aws/rds/model/DescribeDBInstancesRequest.h>
#include <iostream>

/*
 *  A "Hello Rds" starter application which initializes an Amazon Relational
 Database Service (Amazon RDS) client and
 *  describes the Amazon RDS instances.
 *
 *  main function
 *
 *  Usage: 'hello_rds'
 *
 */

int main(int argc, char **argv) {
    Aws::SDKOptions options;
    // Optionally change the log level for debugging.
//   options.loggingOptions.logLevel = Utils::Logging::LogLevel::Debug;
    Aws::InitAPI(options); // Should only be called once.
    int result = 0;
    {
        Aws::Client::ClientConfiguration clientConfig;
        // Optional: Set to the AWS Region (overrides config file).
```

```
        // clientConfig.region = "us-east-1";

        Aws::RDS::RDSClient rdsClient(clientConfig);
        Aws::String marker;
        std::vector<Aws::String> instanceDBIDs;

        do {
            Aws::RDS::Model::DescribeDBInstancesRequest request;

            if (!marker.empty()) {
                request.SetMarker(marker);
            }

            Aws::RDS::Model::DescribeDBInstancesOutcome outcome =
                    rdsClient.DescribeDBInstances(request);

            if (outcome.IsSuccess()) {
                for (auto &instance: outcome.GetResult().GetDBInstances()) {
                    instanceDBIDs.push_back(instance.GetDBInstanceIdentifier());
                }
                marker = outcome.GetResult().GetMarker();
            } else {
                result = 1;
                std::cerr << "Error with RDS::DescribeDBInstances. "
                          << outcome.GetError().GetMessage()
                          << std::endl;
                break;
            }
        } while (!marker.empty());

        std::cout << instanceDBIDs.size() << " RDS instances found." <<
 std::endl;
        for (auto &instanceDBID: instanceDBIDs) {
            std::cout << "   Instance: " << instanceDBID << std::endl;
        }
    }

    Aws::ShutdownAPI(options); // Should only be called once.
    return result;
}
```

- Para obter detalhes da API, consulte DescribeDBInstances na Referência da API AWS SDK para C++.

Go

SDK para Go V2

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
package main

import (
 "context"
 "fmt"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/config"
 "github.com/aws/aws-sdk-go-v2/service/rds"
)

// main uses the AWS SDK for Go V2 to create an Amazon Relational Database
 Service (Amazon RDS)
// client and list up to 20 DB instances in your account.
// This example uses the default settings specified in your shared credentials
// and config files.
func main() {
 sdkConfig, err := config.LoadDefaultConfig(context.TODO())
 if err != nil {
  fmt.Println("Couldn't load default configuration. Have you set up your AWS
 account?")
  fmt.Println(err)
  return
 }
 rdsClient := rds.NewFromConfig(sdkConfig)
 const maxInstances = 20
 fmt.Printf("Let's list up to %v DB instances.\n", maxInstances)
```

```
 output, err := rdsClient.DescribeDBInstances(context.TODO(),
  &rds.DescribeDBInstancesInput{MaxRecords: aws.Int32(maxInstances)})
 if err != nil {
  fmt.Printf("Couldn't list DB instances: %v\n", err)
  return
 }
 if len(output.DBInstances) == 0 {
  fmt.Println("No DB instances found.")
 } else {
  for _, instance := range output.DBInstances {
   fmt.Printf("DB instance %v has database %v.\n",
 *instance.DBInstanceIdentifier,
    *instance.DBName)
  }
 }
}
```

- Para obter detalhes da API, consulte DescribeDBInstances na Referência da API AWS SDK para Go.

Java

SDK para Java 2.x

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.rds.RdsClient;
import software.amazon.awssdk.services.rds.model.DescribeDbInstancesResponse;
import software.amazon.awssdk.services.rds.model.DBInstance;
import software.amazon.awssdk.services.rds.model.RdsException;
import java.util.List;

/**
 * Before running this Java V2 code example, set up your development
```

```
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class DescribeDBInstances {

    public static void main(String[] args) {
        Region region = Region.US_EAST_1;
        RdsClient rdsClient = RdsClient.builder()
                .region(region)
                .build();

        describeInstances(rdsClient);
        rdsClient.close();
    }

    public static void describeInstances(RdsClient rdsClient) {
        try {
            DescribeDbInstancesResponse response =
 rdsClient.describeDBInstances();
            List<DBInstance> instanceList = response.dbInstances();
            for (DBInstance instance : instanceList) {
                System.out.println("Instance ARN is: " +
 instance.dbInstanceArn());
                System.out.println("The Engine is " + instance.engine());
                System.out.println("Connection endpoint is" +
 instance.endpoint().address());
            }

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
    }
}
```

- Para obter detalhes da API, consulte DescribeDBInstances na Referência da API AWS SDK for Java 2.x.

Python

SDK para Python (Boto3).

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
"""
Purpose

Shows how to use the AWS SDK for Python (Boto3) with the Amazon Relational
 Database Service
(Amazon RDS) to list the databases in your account.
"""

import boto3
from botocore.exceptions import ClientError

# Create an RDS client
rds_client = boto3.client("rds")

# Create a paginator for the describe_db_instances operation
paginator = rds_client.get_paginator("describe_db_instances")

try:
    # Use the paginator to get a list of DB instances
    response_iterator = paginator.paginate(
        PaginationConfig={
            "MaxItems": 123,
            "PageSize": 50,  # Adjust PageSize as needed
            "StartingToken": None,
        }
    )

    # Iterate through the pages of the response
    instances_found = False
    for page in response_iterator:
        if "DBInstances" in page and page["DBInstances"]:
            instances_found = True
```

```
            print("Your RDS instances are:")
            for db in page["DBInstances"]:
                print(db["DBInstanceIdentifier"])

    if not instances_found:
        print("No RDS instances found!")

except ClientError as e:
    print(f"Couldn't list RDS instances. Here's why: {e.response['Error']
['Message']}")
```

- Para obter detalhes da API, consulte DescribeDBInstances na Referência da API AWS SDK para Python (Boto3).

Ruby

SDK para Ruby

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
require 'aws-sdk-rds'
require 'logger'

# RDSManager is a class responsible for managing RDS operations
# such as listing all RDS DB instances in the current AWS account.
class RDSManager
  def initialize(client)
    @client = client
    @logger = Logger.new($stdout)
  end

  # Lists and prints all RDS DB instances in the current AWS account.
  def list_db_instances
    @logger.info('Listing RDS DB instances')
```

```
    paginator = @client.describe_db_instances
    instances = []

    paginator.each_page do |page|
      instances.concat(page.db_instances)
    end

    if instances.empty?
      @logger.info('No instances found.')
    else
      @logger.info("Found #{instances.count} instance(s):")
      instances.each do |instance|
        @logger.info(" * #{instance.db_instance_identifier}
 (#{instance.db_instance_status})")
      end
    end
  end
end

if $PROGRAM_NAME == __FILE__
  rds_client = Aws::RDS::Client.new(region: 'us-west-2')
  manager = RDSManager.new(rds_client)
  manager.list_db_instances
end
```

- Para obter detalhes da API, consulte DescribeDBInstances na Referência da API AWS SDK para Ruby.

Exemplos de código

- Ações para o Amazon SNS usando AWS SDKs
  - Usar CreateDBInstance com o AWS SDK ou a CLI
  - Usar CreateDBParameterGroup com o AWS SDK ou a CLI
  - Usar CreateDBSnapshot com o AWS SDK ou a CLI
  - Usar DeleteDBInstance com o AWS SDK ou a CLI
  - Usar DeleteDBParameterGroup com o AWS SDK ou a CLI
  - Usar DescribeAccountAttributes com o AWS SDK ou a CLI

- Usar DescribeDBEngineVersions com o AWS SDK ou a CLI
- Usar DescribeDBInstances com o AWS SDK ou a CLI
- Usar DescribeDBParameterGroups com o AWS SDK ou a CLI
- Usar DescribeDBParameters com o AWS SDK ou a CLI
- Usar DescribeDBSnapshots com o AWS SDK ou a CLI
- Usar DescribeOrderableDBInstanceOptions com o AWS SDK ou a CLI
- Usar GenerateRDSAuthToken com o AWS SDK ou a CLI
- Usar ModifyDBInstance com o AWS SDK ou a CLI
- Usar ModifyDBParameterGroup com o AWS SDK ou a CLI
- Usar RebootDBInstance com o AWS SDK ou a CLI
- Cenários do Amazon RDS usando AWS SDKs
  - Começar a usar instâncias de banco de dados do Amazon RDS usando um AWS SDK
- Exemplos de tecnologia servidor para o Amazon RDS usando SDKs da AWS
  - Como se conectar a um banco de dados do Amazon RDS em uma função do Lambda
- Exemplos do Amazon RDS entre serviços usando AWS SDKs
  - Crie um rastreador de itens de trabalho do Aurora Sem Servidor

Ações para o Amazon SNS usando AWS SDKs

Os exemplos de código a seguir demonstram como realizar ações específicas do Amazon RDS com AWS SDKs. Esses trechos chamam a API do Amazon RDS e são trechos de código de programas maiores que devem ser executados no contexto. Cada exemplo inclui um link para o GitHub, em que é possível encontrar instruções para configurar e executar o código.

Os exemplos a seguir incluem apenas as ações mais utilizadas. Para obter uma lista completa, consulte a Referência de API do Amazon Relational Database Service (Amazon RDS).

Exemplos

- Usar CreateDBInstance com o AWS SDK ou a CLI
- Usar CreateDBParameterGroup com o AWS SDK ou a CLI
- Usar CreateDBSnapshot com o AWS SDK ou a CLI
- Usar DeleteDBInstance com o AWS SDK ou a CLI

- Usar DeleteDBParameterGroup com o AWS SDK ou a CLI
- Usar DescribeAccountAttributes com o AWS SDK ou a CLI
- Usar DescribeDBEngineVersions com o AWS SDK ou a CLI
- Usar DescribeDBInstances com o AWS SDK ou a CLI
- Usar DescribeDBParameterGroups com o AWS SDK ou a CLI
- Usar DescribeDBParameters com o AWS SDK ou a CLI
- Usar DescribeDBSnapshots com o AWS SDK ou a CLI
- Usar DescribeOrderableDBInstanceOptions com o AWS SDK ou a CLI
- Usar GenerateRDSAuthToken com o AWS SDK ou a CLI
- Usar ModifyDBInstance com o AWS SDK ou a CLI
- Usar ModifyDBParameterGroup com o AWS SDK ou a CLI
- Usar RebootDBInstance com o AWS SDK ou a CLI

Usar **`CreateDBInstance`** com o AWS SDK ou a CLI

Os exemplos de código a seguir mostram como usar o `CreateDBInstance`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação no contexto no seguinte exemplo de código:

- Começar a usar instâncias de banco de dados

.NET

SDK para .NET

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
/// <summary>
```

```
   /// Create an RDS DB instance with a particular set of properties. Use the
action DescribeDBInstancesAsync
   /// to determine when the DB instance is ready to use.
   /// </summary>
   /// <param name="dbName">Name for the DB instance.</param>
   /// <param name="dbInstanceIdentifier">DB instance identifier.</param>
   /// <param name="parameterGroupName">DB parameter group to associate with the
instance.</param>
   /// <param name="dbEngine">The engine for the DB instance.</param>
   /// <param name="dbEngineVersion">Version for the DB instance.</param>
   /// <param name="instanceClass">Class for the DB instance.</param>
   /// <param name="allocatedStorage">The amount of storage in gibibytes (GiB)
to allocate to the DB instance.</param>
   /// <param name="adminName">Admin user name.</param>
   /// <param name="adminPassword">Admin user password.</param>
   /// <returns>DB instance object.</returns>
   public async Task<DBInstance> CreateDBInstance(string dbName, string
dbInstanceIdentifier,
       string parameterGroupName, string dbEngine, string dbEngineVersion,
       string instanceClass, int allocatedStorage, string adminName, string
adminPassword)
   {
       var response = await _amazonRDS.CreateDBInstanceAsync(
           new CreateDBInstanceRequest()
           {
               DBName = dbName,
               DBInstanceIdentifier = dbInstanceIdentifier,
               DBParameterGroupName = parameterGroupName,
               Engine = dbEngine,
               EngineVersion = dbEngineVersion,
               DBInstanceClass = instanceClass,
               AllocatedStorage = allocatedStorage,
               MasterUsername = adminName,
               MasterUserPassword = adminPassword
           });

       return response.DBInstance;
   }
```

- Para obter detalhes da API, consulte CreateDBInstance na Referência da API AWS SDK para .NET.

C++

SDK para C++

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
        Aws::Client::ClientConfiguration clientConfig;
        // Optional: Set to the AWS Region (overrides config file).
        // clientConfig.region = "us-east-1";

    Aws::RDS::RDSClient client(clientConfig);

        Aws::RDS::Model::CreateDBInstanceRequest request;
        request.SetDBName(DB_NAME);
        request.SetDBInstanceIdentifier(DB_INSTANCE_IDENTIFIER);
        request.SetDBParameterGroupName(PARAMETER_GROUP_NAME);
        request.SetEngine(engineVersion.GetEngine());
        request.SetEngineVersion(engineVersion.GetEngineVersion());
        request.SetDBInstanceClass(dbInstanceClass);
        request.SetStorageType(DB_STORAGE_TYPE);
        request.SetAllocatedStorage(DB_ALLOCATED_STORAGE);
        request.SetMasterUsername(administratorName);
        request.SetMasterUserPassword(administratorPassword);

        Aws::RDS::Model::CreateDBInstanceOutcome outcome =
                client.CreateDBInstance(request);

        if (outcome.IsSuccess()) {
            std::cout << "The DB instance creation has started."
                      << std::endl;
        }
        else {
            std::cerr << "Error with RDS::CreateDBInstance. "
                      << outcome.GetError().GetMessage()
                      << std::endl;
            cleanUpResources(PARAMETER_GROUP_NAME, "", client);
            return false;
        }
```

- Para obter detalhes da API, consulte CreateDBInstance na Referência da API AWS SDK para C++.

CLI

AWS CLI

Como criar uma instância de banco de dados

O exemplo de `create-db-instance` a seguir usa as opções necessárias para iniciar uma nova instância de banco de dados.

```
aws rds create-db-instance \
    --db-instance-identifier test-mysql-instance \
    --db-instance-class db.t3.micro \
    --engine mysql \
    --master-username admin \
    --master-user-password secret99 \
    --allocated-storage 20
```

Saída:

```
{
    "DBInstance": {
        "DBInstanceIdentifier": "test-mysql-instance",
        "DBInstanceClass": "db.t3.micro",
        "Engine": "mysql",
        "DBInstanceStatus": "creating",
        "MasterUsername": "admin",
        "AllocatedStorage": 20,
        "PreferredBackupWindow": "12:55-13:25",
        "BackupRetentionPeriod": 1,
        "DBSecurityGroups": [],
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sg-12345abc",
                "Status": "active"
            }
        ],
        "DBParameterGroups": [
```

```
            {
                "DBParameterGroupName": "default.mysql5.7",
                "ParameterApplyStatus": "in-sync"
            }
        ],
        "DBSubnetGroup": {
            "DBSubnetGroupName": "default",
            "DBSubnetGroupDescription": "default",
            "VpcId": "vpc-2ff2ff2f",
            "SubnetGroupStatus": "Complete",
            "Subnets": [
                {
                    "SubnetIdentifier": "subnet-########",
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2c"
                    },
                    "SubnetStatus": "Active"
                },
                {
                    "SubnetIdentifier": "subnet-########",
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2d"
                    },
                    "SubnetStatus": "Active"
                },
                {
                    "SubnetIdentifier": "subnet-########",
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2a"
                    },
                    "SubnetStatus": "Active"
                },
                {
                    "SubnetIdentifier": "subnet-########",
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2b"
                    },
                    "SubnetStatus": "Active"
                }
            ]
        },
        "PreferredMaintenanceWindow": "sun:08:07-sun:08:37",
        "PendingModifiedValues": {
            "MasterUserPassword": "****"
```

```
        },
        "MultiAZ": false,
        "EngineVersion": "5.7.22",
        "AutoMinorVersionUpgrade": true,
        "ReadReplicaDBInstanceIdentifiers": [],
        "LicenseModel": "general-public-license",
        "OptionGroupMemberships": [
            {
                "OptionGroupName": "default:mysql-5-7",
                "Status": "in-sync"
            }
        ],
        "PubliclyAccessible": true,
        "StorageType": "gp2",
        "DbInstancePort": 0,
        "StorageEncrypted": false,
        "DbiResourceId": "db-5555EXAMPLE44444444EXAMPLE",
        "CACertificateIdentifier": "rds-ca-2019",
        "DomainMemberships": [],
        "CopyTagsToSnapshot": false,
        "MonitoringInterval": 0,
        "DBInstanceArn": "arn:aws:rds:us-west-2:123456789012:db:test-mysql-
instance",
        "IAMDatabaseAuthenticationEnabled": false,
        "PerformanceInsightsEnabled": false,
        "DeletionProtection": false,
        "AssociatedRoles": []
    }
}
```

Para obter mais informações, consulte Criar uma instância de banco de dados do Amazon RDS no Guia do usuário do Amazon RDS.

- Para obter detalhes da API, consulte CreateDBInstance na Referência de comandos da AWS CLI.

Go

SDK para Go V2

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
type DbInstances struct {
 RdsClient *rds.Client
}



// CreateInstance creates a DB instance.
func (instances *DbInstances) CreateInstance(instanceName string, dbName string,
 dbEngine string, dbEngineVersion string, parameterGroupName string,
 dbInstanceClass string,
 storageType string, allocatedStorage int32, adminName string, adminPassword
 string) (
 *types.DBInstance, error) {
 output, err := instances.RdsClient.CreateDBInstance(context.TODO(),
 &rds.CreateDBInstanceInput{
  DBInstanceIdentifier: aws.String(instanceName),
  DBName:               aws.String(dbName),
  DBParameterGroupName: aws.String(parameterGroupName),
  Engine:               aws.String(dbEngine),
  EngineVersion:        aws.String(dbEngineVersion),
  DBInstanceClass:      aws.String(dbInstanceClass),
  StorageType:          aws.String(storageType),
  AllocatedStorage:     aws.Int32(allocatedStorage),
  MasterUsername:       aws.String(adminName),
  MasterUserPassword:   aws.String(adminPassword),
 })
 if err != nil {
  log.Printf("Couldn't create instance %v: %v\n", instanceName, err)
  return nil, err
 } else {
  return output.DBInstance, nil
```

```
 }
}
```

- Para obter detalhes da API, consulte CreateDBInstance na Referência da API AWS SDK para Go.

Java

SDK para Java 2.x

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
import com.google.gson.Gson;
import
 software.amazon.awssdk.auth.credentials.EnvironmentVariableCredentialsProvider;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.rds.RdsClient;
import software.amazon.awssdk.services.rds.model.DescribeDbInstancesRequest;
import software.amazon.awssdk.services.rds.model.CreateDbInstanceRequest;
import software.amazon.awssdk.services.rds.model.CreateDbInstanceResponse;
import software.amazon.awssdk.services.rds.model.RdsException;
import software.amazon.awssdk.services.rds.model.DescribeDbInstancesResponse;
import software.amazon.awssdk.services.rds.model.DBInstance;
import software.amazon.awssdk.services.secretsmanager.SecretsManagerClient;
import
 software.amazon.awssdk.services.secretsmanager.model.GetSecretValueRequest;
import
 software.amazon.awssdk.services.secretsmanager.model.GetSecretValueResponse;

import java.util.List;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
```

```
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 *
 * This example requires an AWS Secrets Manager secret that contains the
 * database credentials. If you do not create a
 * secret, this example will not work. For more details, see:
 *
 * https://docs.aws.amazon.com/secretsmanager/latest/userguide/integrating_how-
services-use-secrets_RS.html
 *
 *
 */

public class CreateDBInstance {
    public static long sleepTime = 20;

    public static void main(String[] args) {
        final String usage = """

                Usage:
                    <dbInstanceIdentifier> <dbName> <secretName>

                Where:
                    dbInstanceIdentifier - The database instance identifier.\s
                    dbName - The database name.\s
                    secretName - The name of the AWS Secrets Manager secret that
 contains the database credentials."
                """;

        if (args.length != 3) {
            System.out.println(usage);
            System.exit(1);
        }

        String dbInstanceIdentifier = args[0];
        String dbName = args[1];
        String secretName = args[2];
        Gson gson = new Gson();
        User user = gson.fromJson(String.valueOf(getSecretValues(secretName)),
 User.class);
        Region region = Region.US_WEST_2;
        RdsClient rdsClient = RdsClient.builder()
```

```
                .region(region)
                .build();

        createDatabaseInstance(rdsClient, dbInstanceIdentifier, dbName,
user.getUsername(), user.getPassword());
        waitForInstanceReady(rdsClient, dbInstanceIdentifier);
        rdsClient.close();
    }

    private static SecretsManagerClient getSecretClient() {
        Region region = Region.US_WEST_2;
        return SecretsManagerClient.builder()
                .region(region)

.credentialsProvider(EnvironmentVariableCredentialsProvider.create())
                .build();
    }

    private static String getSecretValues(String secretName) {
        SecretsManagerClient secretClient = getSecretClient();
        GetSecretValueRequest valueRequest = GetSecretValueRequest.builder()
                .secretId(secretName)
                .build();

        GetSecretValueResponse valueResponse =
secretClient.getSecretValue(valueRequest);
        return valueResponse.secretString();
    }

    public static void createDatabaseInstance(RdsClient rdsClient,
            String dbInstanceIdentifier,
            String dbName,
            String userName,
            String userPassword) {

        try {
            CreateDbInstanceRequest instanceRequest =
CreateDbInstanceRequest.builder()
                    .dbInstanceIdentifier(dbInstanceIdentifier)
                    .allocatedStorage(100)
                    .dbName(dbName)
                    .engine("mysql")
                    .dbInstanceClass("db.m4.large")
                    .engineVersion("8.0")
```

```
                    .storageType("standard")
                    .masterUsername(userName)
                    .masterUserPassword(userPassword)
                    .build();

            CreateDbInstanceResponse response =
rdsClient.createDBInstance(instanceRequest);
            System.out.print("The status is " +
response.dbInstance().dbInstanceStatus());

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
    }

    // Waits until the database instance is available.
    public static void waitForInstanceReady(RdsClient rdsClient, String
dbInstanceIdentifier) {
        boolean instanceReady = false;
        String instanceReadyStr;
        System.out.println("Waiting for instance to become available.");
        try {
            DescribeDbInstancesRequest instanceRequest =
DescribeDbInstancesRequest.builder()
                    .dbInstanceIdentifier(dbInstanceIdentifier)
                    .build();

            // Loop until the cluster is ready.
            while (!instanceReady) {
                DescribeDbInstancesResponse response =
rdsClient.describeDBInstances(instanceRequest);
                List<DBInstance> instanceList = response.dbInstances();
                for (DBInstance instance : instanceList) {
                    instanceReadyStr = instance.dbInstanceStatus();
                    if (instanceReadyStr.contains("available"))
                        instanceReady = true;
                    else {
                        System.out.print(".");
                        Thread.sleep(sleepTime * 1000);
                    }
                }
            }
            System.out.println("Database instance is available!");
```

```
        } catch (RdsException | InterruptedException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }
```

- Para obter detalhes da API, consulte CreateDBInstance na Referência da API AWS SDK for Java 2.x.

Kotlin

SDK para Kotlin

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
suspend fun createDatabaseInstance(
    dbInstanceIdentifierVal: String?,
    dbNamedbVal: String?,
    masterUsernameVal: String?,
    masterUserPasswordVal: String?,
) {
    val instanceRequest =
        CreateDbInstanceRequest {
            dbInstanceIdentifier = dbInstanceIdentifierVal
            allocatedStorage = 100
            dbName = dbNamedbVal
            engine = "mysql"
            dbInstanceClass = "db.m4.large"
            engineVersion = "8.0"
            storageType = "standard"
            masterUsername = masterUsernameVal
            masterUserPassword = masterUserPasswordVal
        }

    RdsClient { region = "us-west-2" }.use { rdsClient ->
```

```
        val response = rdsClient.createDbInstance(instanceRequest)
        print("The status is ${response.dbInstance?.dbInstanceStatus}")
    }
}

// Waits until the database instance is available.
suspend fun waitForInstanceReady(dbInstanceIdentifierVal: String?) {
    val sleepTime: Long = 20
    var instanceReady = false
    var instanceReadyStr: String
    println("Waiting for instance to become available.")

    val instanceRequest =
        DescribeDbInstancesRequest {
            dbInstanceIdentifier = dbInstanceIdentifierVal
        }

    RdsClient { region = "us-west-2" }.use { rdsClient ->
        while (!instanceReady) {
            val response = rdsClient.describeDbInstances(instanceRequest)
            val instanceList = response.dbInstances
            if (instanceList != null) {
                for (instance in instanceList) {
                    instanceReadyStr = instance.dbInstanceStatus.toString()
                    if (instanceReadyStr.contains("available")) {
                        instanceReady = true
                    } else {
                        println("...$instanceReadyStr")
                        delay(sleepTime * 1000)
                    }
                }
            }
        }
        println("Database instance is available!")
    }
}
```

- Para obter detalhes da API, consulte CreateDBInstance na Referência da API AWS SDK para Kotlin.

PHP

SDK para PHP

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
require __DIR__ . '/vendor/autoload.php';

use Aws\Exception\AwsException;



$rdsClient = new Aws\Rds\RdsClient([
    'region' => 'us-east-2'
]);

$dbIdentifier = '<<{{db-identifier}}>>';
$dbClass = 'db.t2.micro';
$storage = 5;
$engine = 'MySQL';
$username = 'MyUser';
$password = 'MyPassword';

try {
    $result = $rdsClient->createDBInstance([
        'DBInstanceIdentifier' => $dbIdentifier,
        'DBInstanceClass' => $dbClass,
        'AllocatedStorage' => $storage,
        'Engine' => $engine,
        'MasterUsername' => $username,
        'MasterUserPassword' => $password,
    ]);
    var_dump($result);
} catch (AwsException $e) {
    echo $e->getMessage();
    echo "\n";
}
```

- Para obter detalhes da API, consulte CreateDBInstance na Referência da API AWS SDK para PHP.

Python

SDK para Python (Boto3).

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
class InstanceWrapper:
    """Encapsulates Amazon RDS DB instance actions."""

    def __init__(self, rds_client):
        """
        :param rds_client: A Boto3 Amazon RDS client.
        """
        self.rds_client = rds_client

    @classmethod
    def from_client(cls):
        """
        Instantiates this class from a Boto3 client.
        """
        rds_client = boto3.client("rds")
        return cls(rds_client)


    def create_db_instance(
        self,
        db_name,
        instance_id,
        parameter_group_name,
        db_engine,
        db_engine_version,
```

```
        instance_class,
        storage_type,
        allocated_storage,
        admin_name,
        admin_password,
    ):
        """
        Creates a DB instance.

        :param db_name: The name of the database that is created in the DB
instance.
        :param instance_id: The ID to give the newly created DB instance.
        :param parameter_group_name: A parameter group to associate with the DB
instance.
        :param db_engine: The database engine of a database to create in the DB
instance.
        :param db_engine_version: The engine version for the created database.
        :param instance_class: The DB instance class for the newly created DB
instance.
        :param storage_type: The storage type of the DB instance.
        :param allocated_storage: The amount of storage allocated on the DB
instance, in GiBs.
        :param admin_name: The name of the admin user for the created database.
        :param admin_password: The admin password for the created database.
        :return: Data about the newly created DB instance.
        """
        try:
            response = self.rds_client.create_db_instance(
                DBName=db_name,
                DBInstanceIdentifier=instance_id,
                DBParameterGroupName=parameter_group_name,
                Engine=db_engine,
                EngineVersion=db_engine_version,
                DBInstanceClass=instance_class,
                StorageType=storage_type,
                AllocatedStorage=allocated_storage,
                MasterUsername=admin_name,
                MasterUserPassword=admin_password,
            )
            db_inst = response["DBInstance"]
        except ClientError as err:
            logger.error(
                "Couldn't create DB instance %s. Here's why: %s: %s",
                instance_id,
```

```
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return db_inst
```

- Para ter detalhes da API, consulte CreateDBInstance na Referência de API do AWS SDK para Python (Boto3).

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`CreateDBParameterGroup`** com o AWS SDK ou a CLI

Os exemplos de código a seguir mostram como usar o `CreateDBParameterGroup`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação no contexto no seguinte exemplo de código:

- Começar a usar instâncias de banco de dados

.NET

SDK para .NET

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
   /// <summary>
   /// Create a new DB parameter group. Use the action
DescribeDBParameterGroupsAsync
   /// to determine when the DB parameter group is ready to use.
```

```
    /// </summary>
    /// <param name="name">Name of the DB parameter group.</param>
    /// <param name="family">Family of the DB parameter group.</param>
    /// <param name="description">Description of the DB parameter group.</param>
    /// <returns>The new DB parameter group.</returns>
    public async Task<DBParameterGroup> CreateDBParameterGroup(
        string name, string family, string description)
    {
        var response = await _amazonRDS.CreateDBParameterGroupAsync(
            new CreateDBParameterGroupRequest()
            {
                DBParameterGroupName = name,
                DBParameterGroupFamily = family,
                Description = description
            });
        return response.DBParameterGroup;
    }
```

- Para obter detalhes da API, consulte CreateDBParameterGroup na Referência da API AWS SDK para .NET.

C++

SDK para C++

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
    Aws::Client::ClientConfiguration clientConfig;
    // Optional: Set to the AWS Region (overrides config file).
    // clientConfig.region = "us-east-1";

Aws::RDS::RDSClient client(clientConfig);

    Aws::RDS::Model::CreateDBParameterGroupRequest request;
    request.SetDBParameterGroupName(PARAMETER_GROUP_NAME);
```

```
        request.SetDBParameterGroupFamily(dbParameterGroupFamily);
        request.SetDescription("Example parameter group.");

        Aws::RDS::Model::CreateDBParameterGroupOutcome outcome =
                client.CreateDBParameterGroup(request);

        if (outcome.IsSuccess()) {
            std::cout << "The DB parameter group was successfully created."
                      << std::endl;
        }
        else {
            std::cerr << "Error with RDS::CreateDBParameterGroup. "
                      << outcome.GetError().GetMessage()
                      << std::endl;
            return false;
        }
```

- Para obter detalhes da API, consulte CreateDBParameterGroup na Referência da API AWS SDK para C++.

CLI

AWS CLI

Como criar um grupo de parâmetros de banco de dados

O exemplo de `create-db-parameter-group` a seguir cria um grupo de parâmetros de banco de dados.

```
aws rds create-db-parameter-group \
    --db-parameter-group-name mydbparametergroup \
    --db-parameter-group-family MySQL5.6 \
    --description "My new parameter group"
```

Saída:

```
{
    "DBParameterGroup": {
        "DBParameterGroupName": "mydbparametergroup",
        "DBParameterGroupFamily": "mysql5.6",
        "Description": "My new parameter group",
```

```
        "DBParameterGroupArn": "arn:aws:rds:us-
east-1:123456789012:pg:mydbparametergroup"
    }
}
```

Para obter mais informações, consulte Criar um grupo de parâmetros de banco de dados no Guia do usuário do Amazon RDS.

- Para obter detalhes da API, consulte CreateDBParameterGroup na Referência de comandos da AWS CLI.

Go

SDK para Go V2

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
type DbInstances struct {
 RdsClient *rds.Client
}



// CreateParameterGroup creates a DB parameter group that is based on the
 specified
// parameter group family.
func (instances *DbInstances) CreateParameterGroup(
 parameterGroupName string, parameterGroupFamily string, description string) (
 *types.DBParameterGroup, error) {

 output, err := instances.RdsClient.CreateDBParameterGroup(context.TODO(),
  &rds.CreateDBParameterGroupInput{
   DBParameterGroupName:   aws.String(parameterGroupName),
   DBParameterGroupFamily: aws.String(parameterGroupFamily),
   Description:            aws.String(description),
  })
 if err != nil {
```

```
  log.Printf("Couldn't create parameter group %v: %v\n", parameterGroupName, err)
  return nil, err
 } else {
  return output.DBParameterGroup, err
 }
}
```

- Para obter detalhes da API, consulte CreateDBParameterGroup na Referência da API AWS SDK para Go.

Java

SDK para Java 2.x

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
   public static void createDBParameterGroup(RdsClient rdsClient, String
dbGroupName, String dbParameterGroupFamily) {
       try {
           CreateDbParameterGroupRequest groupRequest =
CreateDbParameterGroupRequest.builder()
                   .dbParameterGroupName(dbGroupName)
                   .dbParameterGroupFamily(dbParameterGroupFamily)
                   .description("Created by using the AWS SDK for Java")
                   .build();

           CreateDbParameterGroupResponse response =
rdsClient.createDBParameterGroup(groupRequest);
           System.out.println("The group name is " +
response.dbParameterGroup().dbParameterGroupName());

       } catch (RdsException e) {
           System.out.println(e.getLocalizedMessage());
           System.exit(1);
       }
```

```
    }
```

- Para obter detalhes da API, consulte CreateDBParameterGroup na Referência da API AWS SDK for Java 2.x.

Python

SDK para Python (Boto3).

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
class InstanceWrapper:
    """Encapsulates Amazon RDS DB instance actions."""

    def __init__(self, rds_client):
        """
        :param rds_client: A Boto3 Amazon RDS client.
        """
        self.rds_client = rds_client

    @classmethod
    def from_client(cls):
        """
        Instantiates this class from a Boto3 client.
        """
        rds_client = boto3.client("rds")
        return cls(rds_client)


    def create_parameter_group(
        self, parameter_group_name, parameter_group_family, description
    ):
        """
        Creates a DB parameter group that is based on the specified parameter
 group
        family.
```

```
        :param parameter_group_name: The name of the newly created parameter
group.
        :param parameter_group_family: The family that is used as the basis of
the new
                                       parameter group.
        :param description: A description given to the parameter group.
        :return: Data about the newly created parameter group.
        """
        try:
            response = self.rds_client.create_db_parameter_group(
                DBParameterGroupName=parameter_group_name,
                DBParameterGroupFamily=parameter_group_family,
                Description=description,
            )
        except ClientError as err:
            logger.error(
                "Couldn't create parameter group %s. Here's why: %s: %s",
                parameter_group_name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return response
```

- Para ter detalhes da API, consulte CreateDBParameterGroup na Referência de API do AWS SDK para Python (Boto3).

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`CreateDBSnapshot`** com o AWS SDK ou a CLI

Os exemplos de código a seguir mostram como usar o `CreateDBSnapshot`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação no contexto no seguinte exemplo de código:

- Começar a usar instâncias de banco de dados

.NET

SDK para .NET

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
    /// <summary>
    /// Create a snapshot of a DB instance.
    /// </summary>
    /// <param name="dbInstanceIdentifier">DB instance identifier.</param>
    /// <param name="snapshotIdentifier">Identifier for the snapshot.</param>
    /// <returns>DB snapshot object.</returns>
    public async Task<DBSnapshot> CreateDBSnapshot(string dbInstanceIdentifier,
string snapshotIdentifier)
    {
        var response = await _amazonRDS.CreateDBSnapshotAsync(
            new CreateDBSnapshotRequest()
            {
                DBSnapshotIdentifier = snapshotIdentifier,
                DBInstanceIdentifier = dbInstanceIdentifier
            });

        return response.DBSnapshot;
    }
```

- Para obter detalhes da API, consulte CreateDBSnapshot na Referência da API AWS SDK para .NET.

C++

SDK para C++

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
        Aws::Client::ClientConfiguration clientConfig;
        // Optional: Set to the AWS Region (overrides config file).
        // clientConfig.region = "us-east-1";

    Aws::RDS::RDSClient client(clientConfig);

            Aws::RDS::Model::CreateDBSnapshotRequest request;
            request.SetDBInstanceIdentifier(DB_INSTANCE_IDENTIFIER);
            request.SetDBSnapshotIdentifier(snapshotID);

            Aws::RDS::Model::CreateDBSnapshotOutcome outcome =
                    client.CreateDBSnapshot(request);

            if (outcome.IsSuccess()) {
                std::cout << "Snapshot creation has started."
                          << std::endl;
            }
            else {
                std::cerr << "Error with RDS::CreateDBSnapshot. "
                          << outcome.GetError().GetMessage()
                          << std::endl;
                cleanUpResources(PARAMETER_GROUP_NAME, DB_INSTANCE_IDENTIFIER,
client);
                return false;
            }
```

- Para obter detalhes da API, consulte CreateDBSnapshot na Referência da API AWS SDK para C++.

CLI

AWS CLI

Como criar um snapshot de banco de dados

O exemplo de `create-db-snapshot` a seguir cria um snapshot de banco de dados.

```
aws rds create-db-snapshot \
    --db-instance-identifier database-mysql \
    --db-snapshot-identifier mydbsnapshot
```

Saída:

```
{
    "DBSnapshot": {
        "DBSnapshotIdentifier": "mydbsnapshot",
        "DBInstanceIdentifier": "database-mysql",
        "Engine": "mysql",
        "AllocatedStorage": 100,
        "Status": "creating",
        "Port": 3306,
        "AvailabilityZone": "us-east-1b",
        "VpcId": "vpc-6594f31c",
        "InstanceCreateTime": "2019-04-30T15:45:53.663Z",
        "MasterUsername": "admin",
        "EngineVersion": "5.6.40",
        "LicenseModel": "general-public-license",
        "SnapshotType": "manual",
        "Iops": 1000,
        "OptionGroupName": "default:mysql-5-6",
        "PercentProgress": 0,
        "StorageType": "io1",
        "Encrypted": true,
        "KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/
AKIAIOSFODNN7EXAMPLE",
        "DBSnapshotArn": "arn:aws:rds:us-
east-1:123456789012:snapshot:mydbsnapshot",
        "IAMDatabaseAuthenticationEnabled": false,
        "ProcessorFeatures": [],
        "DbiResourceId": "db-AKIAIOSFODNN7EXAMPLE"
    }
}
```

Para obter mais informações, consulte Criar um snapshot de banco de dados no Guia do usuário do Amazon RDS.

- Para obter detalhes da API, consulte CreateDBSnapshot na Referência de comandos da AWS CLI.

Go

SDK para Go V2

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
type DbInstances struct {
 RdsClient *rds.Client
}



// CreateSnapshot creates a snapshot of a DB instance.
func (instances *DbInstances) CreateSnapshot(instanceName string, snapshotName
 string) (
 *types.DBSnapshot, error) {
 output, err := instances.RdsClient.CreateDBSnapshot(context.TODO(),
 &rds.CreateDBSnapshotInput{
  DBInstanceIdentifier: aws.String(instanceName),
  DBSnapshotIdentifier: aws.String(snapshotName),
 })
 if err != nil {
  log.Printf("Couldn't create snapshot %v: %v\n", snapshotName, err)
  return nil, err
 } else {
  return output.DBSnapshot, nil
 }
}
```

- Para obter detalhes da API, consulte CreateDBSnapshot na Referência da API AWS SDK para Go.

Java

SDK para Java 2.x

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
   // Create an Amazon RDS snapshot.
   public static void createSnapshot(RdsClient rdsClient, String
dbInstanceIdentifier, String dbSnapshotIdentifier) {
       try {
           CreateDbSnapshotRequest snapshotRequest =
CreateDbSnapshotRequest.builder()
                   .dbInstanceIdentifier(dbInstanceIdentifier)
                   .dbSnapshotIdentifier(dbSnapshotIdentifier)
                   .build();

           CreateDbSnapshotResponse response =
rdsClient.createDBSnapshot(snapshotRequest);
           System.out.println("The Snapshot id is " +
response.dbSnapshot().dbiResourceId());

       } catch (RdsException e) {
           System.out.println(e.getLocalizedMessage());
           System.exit(1);
       }
   }
```

- Para obter detalhes da API, consulte CreateDBSnapshot na Referência da API AWS SDK for Java 2.x.

PHP

SDK para PHP

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
require __DIR__ . '/vendor/autoload.php';

use Aws\Exception\AwsException;



$rdsClient = new Aws\Rds\RdsClient([
    'region' => 'us-east-2'
]);

$dbIdentifier = '<<{{db-identifier}}>>';
$snapshotName = '<<{{backup_2018_12_25}}>>';

try {
    $result = $rdsClient->createDBSnapshot([
        'DBInstanceIdentifier' => $dbIdentifier,
        'DBSnapshotIdentifier' => $snapshotName,
    ]);
    var_dump($result);
} catch (AwsException $e) {
    echo $e->getMessage();
    echo "\n";
}
```

- Para obter detalhes da API, consulte CreateDBSnapshot na Referência da API AWS SDK para PHP.

Python

SDK para Python (Boto3).

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
class InstanceWrapper:
    """Encapsulates Amazon RDS DB instance actions."""

    def __init__(self, rds_client):
        """
        :param rds_client: A Boto3 Amazon RDS client.
        """
        self.rds_client = rds_client

    @classmethod
    def from_client(cls):
        """
        Instantiates this class from a Boto3 client.
        """
        rds_client = boto3.client("rds")
        return cls(rds_client)


    def create_snapshot(self, snapshot_id, instance_id):
        """
        Creates a snapshot of a DB instance.

        :param snapshot_id: The ID to give the created snapshot.
        :param instance_id: The ID of the DB instance to snapshot.
        :return: Data about the newly created snapshot.
        """
        try:
            response = self.rds_client.create_db_snapshot(
                DBSnapshotIdentifier=snapshot_id,
 DBInstanceIdentifier=instance_id
            )
            snapshot = response["DBSnapshot"]
```

```
        except ClientError as err:
            logger.error(
                "Couldn't create snapshot of %s. Here's why: %s: %s",
                instance_id,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return snapshot
```

- Para obter detalhes da API, consulte CreateDBSnapshot na Referência da API AWS SDK para Python (Boto3).

Ruby

SDK para Ruby

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
require "aws-sdk-rds"  # v2: require 'aws-sdk'

# Create a snapshot for an Amazon Relational Database Service (Amazon RDS)
# DB instance.
#
# @param rds_resource [Aws::RDS::Resource] The resource containing SDK logic.
# @param db_instance_name [String] The name of the Amazon RDS DB instance.
# @return [Aws::RDS::DBSnapshot, nil] The snapshot created, or nil if error.
def create_snapshot(rds_resource, db_instance_name)
  id = "snapshot-#{rand(10**6)}"
  db_instance = rds_resource.db_instance(db_instance_name)
  db_instance.create_snapshot({
                                db_snapshot_identifier: id
                              })
rescue Aws::Errors::ServiceError => e
```

```
  puts "Couldn't create DB instance snapshot #{id}:\n #{e.message}"
end
```

- Para ter detalhes da API, consulte CreateDBSnapshot na Referência de API do AWS SDK para Ruby.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`DeleteDBInstance`** com o AWS SDK ou a CLI

Os exemplos de código a seguir mostram como usar o `DeleteDBInstance`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação no contexto no seguinte exemplo de código:

- Começar a usar instâncias de banco de dados

.NET

SDK para .NET

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
/// <summary>
/// Delete a particular DB instance.
/// </summary>
/// <param name="dbInstanceIdentifier">DB instance identifier.</param>
/// <returns>DB instance object.</returns>
public async Task<DBInstance> DeleteDBInstance(string dbInstanceIdentifier)
{
    var response = await _amazonRDS.DeleteDBInstanceAsync(
        new DeleteDBInstanceRequest()
```

```
            {
                DBInstanceIdentifier = dbInstanceIdentifier,
                SkipFinalSnapshot = true,
                DeleteAutomatedBackups = true
            });

        return response.DBInstance;
    }
```

- Para obter detalhes da API, consulte DeleteDBInstance na Referência da API AWS SDK para .NET.

C++

SDK para C++

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
        Aws::Client::ClientConfiguration clientConfig;
        // Optional: Set to the AWS Region (overrides config file).
        // clientConfig.region = "us-east-1";

    Aws::RDS::RDSClient client(clientConfig);

            Aws::RDS::Model::DeleteDBInstanceRequest request;
            request.SetDBInstanceIdentifier(dbInstanceIdentifier);
            request.SetSkipFinalSnapshot(true);
            request.SetDeleteAutomatedBackups(true);

            Aws::RDS::Model::DeleteDBInstanceOutcome outcome =
                    client.DeleteDBInstance(request);

            if (outcome.IsSuccess()) {
                std::cout << "DB instance deletion has started."
                          << std::endl;
```

```
            }
            else {
                std::cerr << "Error with RDS::DeleteDBInstance. "
                          << outcome.GetError().GetMessage()
                          << std::endl;
                result = false;
            }
```

- Para obter detalhes da API, consulte DeleteDBInstance na Referência da API AWS SDK para C++.

CLI

AWS CLI

Como excluir uma instância de banco de dados

O exemplo de `delete-db-instance` a seguir exclui a instância de banco de dados especificada depois de criar um snapshot de banco de dados final chamado `test-instance-final-snap`.

```
aws rds delete-db-instance \
    --db-instance-identifier test-instance \
    --final-db-snapshot-identifier test-instance-final-snap
```

Saída:

```
{
    "DBInstance": {
        "DBInstanceIdentifier": "test-instance",
        "DBInstanceStatus": "deleting",
        ...some output truncated...
    }
}
```

- Para obter detalhes da API, consulte DeleteDBInstance na Referência de comandos da AWS CLI.

Go

SDK para Go V2

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
type DbInstances struct {
 RdsClient *rds.Client
}



// DeleteInstance deletes a DB instance.
func (instances *DbInstances) DeleteInstance(instanceName string) error {
 _, err := instances.RdsClient.DeleteDBInstance(context.TODO(),
 &rds.DeleteDBInstanceInput{
  DBInstanceIdentifier:   aws.String(instanceName),
  SkipFinalSnapshot:      true,
  DeleteAutomatedBackups: aws.Bool(true),
 })
 if err != nil {
  log.Printf("Couldn't delete instance %v: %v\n", instanceName, err)
  return err
 } else {
  return nil
 }
}
```

- Para obter detalhes da API, consulte DeleteDBInstance na Referência da API AWS SDK para Go.

Java

SDK para Java 2.x

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.rds.RdsClient;
import software.amazon.awssdk.services.rds.model.DeleteDbInstanceRequest;
import software.amazon.awssdk.services.rds.model.DeleteDbInstanceResponse;
import software.amazon.awssdk.services.rds.model.RdsException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class DeleteDBInstance {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                    <dbInstanceIdentifier>\s

                Where:
                    dbInstanceIdentifier - The database instance identifier\s
                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String dbInstanceIdentifier = args[0];
```

```
        Region region = Region.US_WEST_2;
        RdsClient rdsClient = RdsClient.builder()
                .region(region)
                .build();

        deleteDatabaseInstance(rdsClient, dbInstanceIdentifier);
        rdsClient.close();
    }

    public static void deleteDatabaseInstance(RdsClient rdsClient, String
 dbInstanceIdentifier) {
        try {
            DeleteDbInstanceRequest deleteDbInstanceRequest =
 DeleteDbInstanceRequest.builder()
                    .dbInstanceIdentifier(dbInstanceIdentifier)
                    .deleteAutomatedBackups(true)
                    .skipFinalSnapshot(true)
                    .build();

            DeleteDbInstanceResponse response =
 rdsClient.deleteDBInstance(deleteDbInstanceRequest);
            System.out.print("The status of the database is " +
 response.dbInstance().dbInstanceStatus());

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
    }
}
```

- Para obter detalhes da API, consulte DeleteDBInstance na Referência da API AWS SDK for Java 2.x.

Kotlin

SDK para Kotlin

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
suspend fun deleteDatabaseInstance(dbInstanceIdentifierVal: String?) {
    val deleteDbInstanceRequest =
        DeleteDbInstanceRequest {
            dbInstanceIdentifier = dbInstanceIdentifierVal
            deleteAutomatedBackups = true
            skipFinalSnapshot = true
        }

    RdsClient { region = "us-west-2" }.use { rdsClient ->
        val response = rdsClient.deleteDbInstance(deleteDbInstanceRequest)
        print("The status of the database is
 ${response.dbInstance?.dbInstanceStatus}")
    }
}
```

- Para obter detalhes da API, consulte DeleteDBInstance na Referência da API AWS SDK para Kotlin.

PHP

SDK para PHP

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
require __DIR__ . '/vendor/autoload.php';

use Aws\Exception\AwsException;


//Create an RDSClient
$rdsClient = new Aws\Rds\RdsClient([
    'region' => 'us-east-1'
]);

$dbIdentifier = '<<{{db-identifier}}>>';

try {
    $result = $rdsClient->deleteDBInstance([
        'DBInstanceIdentifier' => $dbIdentifier,
    ]);
    var_dump($result);
} catch (AwsException $e) {
    echo $e->getMessage();
    echo "\n";
}
```

- Para obter detalhes da API, consulte DeleteDBInstance na Referência da API AWS SDK para PHP.

Python

SDK para Python (Boto3).

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
class InstanceWrapper:
    """Encapsulates Amazon RDS DB instance actions."""

    def __init__(self, rds_client):
```

```
        """
        :param rds_client: A Boto3 Amazon RDS client.
        """
        self.rds_client = rds_client

    @classmethod
    def from_client(cls):
        """
        Instantiates this class from a Boto3 client.
        """
        rds_client = boto3.client("rds")
        return cls(rds_client)


    def delete_db_instance(self, instance_id):
        """
        Deletes a DB instance.

        :param instance_id: The ID of the DB instance to delete.
        :return: Data about the deleted DB instance.
        """
        try:
            response = self.rds_client.delete_db_instance(
                DBInstanceIdentifier=instance_id,
                SkipFinalSnapshot=True,
                DeleteAutomatedBackups=True,
            )
            db_inst = response["DBInstance"]
        except ClientError as err:
            logger.error(
                "Couldn't delete DB instance %s. Here's why: %s: %s",
                instance_id,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return db_inst
```

- Para ter detalhes da API, consulte DeleteDBInstance na Referência de API do AWS SDK para Python (Boto3).

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`DeleteDBParameterGroup`** com o AWS SDK ou a CLI

Os exemplos de código a seguir mostram como usar o `DeleteDBParameterGroup`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação no contexto no seguinte exemplo de código:

- Começar a usar instâncias de banco de dados

.NET

SDK para .NET

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
   /// <summary>
   /// Delete a DB parameter group. The group cannot be a default DB parameter
group
   /// or be associated with any DB instances.
   /// </summary>
   /// <param name="name">Name of the DB parameter group.</param>
   /// <returns>True if successful.</returns>
   public async Task<bool> DeleteDBParameterGroup(string name)
   {
       var response = await _amazonRDS.DeleteDBParameterGroupAsync(
           new DeleteDBParameterGroupRequest()
           {
               DBParameterGroupName = name,
           });
       return response.HttpStatusCode == HttpStatusCode.OK;
   }
```

- Para obter detalhes da API, consulte DeleteDBParameterGroup na Referência da API AWS SDK para .NET.

C++

SDK para C++

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
    Aws::Client::ClientConfiguration clientConfig;
    // Optional: Set to the AWS Region (overrides config file).
    // clientConfig.region = "us-east-1";

Aws::RDS::RDSClient client(clientConfig);

    Aws::RDS::Model::DeleteDBParameterGroupRequest request;
    request.SetDBParameterGroupName(parameterGroupName);

    Aws::RDS::Model::DeleteDBParameterGroupOutcome outcome =
            client.DeleteDBParameterGroup(request);

    if (outcome.IsSuccess()) {
        std::cout << "The DB parameter group was successfully deleted."
                  << std::endl;
    }
    else {
        std::cerr << "Error with RDS::DeleteDBParameterGroup. "
                  << outcome.GetError().GetMessage()
                  << std::endl;
        result = false;
    }
```

- Para obter detalhes da API, consulte DeleteDBParameterGroup na Referência da API AWS SDK para C++.

CLI

AWS CLI

Como excluir um grupo de parâmetros de banco de dados

O exemplo de command a seguir exclui um grupo de parâmetros de banco de dados.

```
aws rds delete-db-parameter-group \
    --db-parameter-group-name mydbparametergroup
```

Este comando não produz saída.

Para obter mais informações, consulte Como trabalhar com grupos de parâmetros de banco de dados no Guia do usuário do Amazon RDS.

- Para obter detalhes da API, consulte DeleteDBParameterGroup na Referência de comandos da AWS CLI.

Go

SDK para Go V2

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
type DbInstances struct {
 RdsClient *rds.Client
}



// DeleteParameterGroup deletes the named DB parameter group.
func (instances *DbInstances) DeleteParameterGroup(parameterGroupName string)
 error {
 _, err := instances.RdsClient.DeleteDBParameterGroup(context.TODO(),
  &rds.DeleteDBParameterGroupInput{
   DBParameterGroupName: aws.String(parameterGroupName),
```

```
  })
 if err != nil {
  log.Printf("Couldn't delete parameter group %v: %v\n", parameterGroupName, err)
  return err
 } else {
  return nil
 }
}
```

- Para obter detalhes da API, consulte DeleteDBParameterGroup na Referência da API AWS SDK para Go.

Java

SDK para Java 2.x

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
   // Delete the parameter group after database has been deleted.
   // An exception is thrown if you attempt to delete the para group while
database
   // exists.
   public static void deleteParaGroup(RdsClient rdsClient, String dbGroupName,
String dbARN)
           throws InterruptedException {
       try {
           boolean isDataDel = false;
           boolean didFind;
           String instanceARN;

           // Make sure that the database has been deleted.
           while (!isDataDel) {
               DescribeDbInstancesResponse response =
rdsClient.describeDBInstances();
               List<DBInstance> instanceList = response.dbInstances();
```

```
                int listSize = instanceList.size();
                didFind = false;
                int index = 1;
                for (DBInstance instance : instanceList) {
                    instanceARN = instance.dbInstanceArn();
                    if (instanceARN.compareTo(dbARN) == 0) {
                        System.out.println(dbARN + " still exists");
                        didFind = true;
                    }
                    if ((index == listSize) && (!didFind)) {
                        // Went through the entire list and did not find the
database ARN.
                        isDataDel = true;
                    }
                    Thread.sleep(sleepTime * 1000);
                    index++;
                }
            }

            // Delete the para group.
            DeleteDbParameterGroupRequest parameterGroupRequest =
DeleteDbParameterGroupRequest.builder()
                    .dbParameterGroupName(dbGroupName)
                    .build();

            rdsClient.deleteDBParameterGroup(parameterGroupRequest);
            System.out.println(dbGroupName + " was deleted.");

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
    }
```

- Para obter detalhes da API, consulte DeleteDBParameterGroup na Referência da API AWS SDK for Java 2.x.

Python

SDK para Python (Boto3).

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
class InstanceWrapper:
    """Encapsulates Amazon RDS DB instance actions."""

    def __init__(self, rds_client):
        """
        :param rds_client: A Boto3 Amazon RDS client.
        """
        self.rds_client = rds_client

    @classmethod
    def from_client(cls):
        """
        Instantiates this class from a Boto3 client.
        """
        rds_client = boto3.client("rds")
        return cls(rds_client)


    def delete_parameter_group(self, parameter_group_name):
        """
        Deletes a DB parameter group.

        :param parameter_group_name: The name of the parameter group to delete.
        :return: Data about the parameter group.
        """
        try:
            self.rds_client.delete_db_parameter_group(
                DBParameterGroupName=parameter_group_name
            )
        except ClientError as err:
            logger.error(
                "Couldn't delete parameter group %s. Here's why: %s: %s",
```

```
                parameter_group_name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
```

- Para ter detalhes da API, consulte DeleteDBParameterGroup na Referência de API do AWS SDK para Python (Boto3).

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`DescribeAccountAttributes`** com o AWS SDK ou a CLI

Os exemplos de código a seguir mostram como usar o `DescribeAccountAttributes`.

CLI

AWS CLI

Como descrever os atributos da conta

O exemplo de `describe-account-attributes` a seguir recupera os atributos da conta da AWS atual.

```
aws rds describe-account-attributes
```

Saída:

```
{
    "AccountQuotas": [
        {
            "Max": 40,
            "Used": 4,
            "AccountQuotaName": "DBInstances"
        },
        {
            "Max": 40,
            "Used": 0,
```

```
            "AccountQuotaName": "ReservedDBInstances"
        },
        {
            "Max": 100000,
            "Used": 40,
            "AccountQuotaName": "AllocatedStorage"
        },
        {
            "Max": 25,
            "Used": 0,
            "AccountQuotaName": "DBSecurityGroups"
        },
        {
            "Max": 20,
            "Used": 0,
            "AccountQuotaName": "AuthorizationsPerDBSecurityGroup"
        },
        {
            "Max": 50,
            "Used": 1,
            "AccountQuotaName": "DBParameterGroups"
        },
        {
            "Max": 100,
            "Used": 3,
            "AccountQuotaName": "ManualSnapshots"
        },
        {
            "Max": 20,
            "Used": 0,
            "AccountQuotaName": "EventSubscriptions"
        },
        {
            "Max": 50,
            "Used": 1,
            "AccountQuotaName": "DBSubnetGroups"
        },
        {
            "Max": 20,
            "Used": 1,
            "AccountQuotaName": "OptionGroups"
        },
        {
            "Max": 20,
```

```
            "Used": 6,
            "AccountQuotaName": "SubnetsPerDBSubnetGroup"
        },
        {
            "Max": 5,
            "Used": 0,
            "AccountQuotaName": "ReadReplicasPerMaster"
        },
        {
            "Max": 40,
            "Used": 1,
            "AccountQuotaName": "DBClusters"
        },
        {
            "Max": 50,
            "Used": 0,
            "AccountQuotaName": "DBClusterParameterGroups"
        },
        {
            "Max": 5,
            "Used": 0,
            "AccountQuotaName": "DBClusterRoles"
        }
    ]
}
```

- Para obter detalhes da API, consulte DescribeAccountAttributes na Referência de comandos da AWS CLI.

Java

SDK para Java 2.x

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.rds.RdsClient;
```

```
import software.amazon.awssdk.services.rds.model.AccountQuota;
import software.amazon.awssdk.services.rds.model.RdsException;
import
 software.amazon.awssdk.services.rds.model.DescribeAccountAttributesResponse;
import java.util.List;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class DescribeAccountAttributes {
    public static void main(String[] args) {
        Region region = Region.US_WEST_2;
        RdsClient rdsClient = RdsClient.builder()
                .region(region)
                .build();

        getAccountAttributes(rdsClient);
        rdsClient.close();
    }

    public static void getAccountAttributes(RdsClient rdsClient) {
        try {
            DescribeAccountAttributesResponse response =
 rdsClient.describeAccountAttributes();
            List<AccountQuota> quotasList = response.accountQuotas();
            for (AccountQuota quotas : quotasList) {
                System.out.println("Name is: " + quotas.accountQuotaName());
                System.out.println("Max value is " + quotas.max());
            }

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
    }
}
```

- Para obter detalhes da API, consulte DescribeAccountAttributes, na Referência da API AWS SDK for Java 2.x.

Kotlin

SDK para Kotlin

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
suspend fun getAccountAttributes() {
    RdsClient { region = "us-west-2" }.use { rdsClient ->
        val response =
 rdsClient.describeAccountAttributes(DescribeAccountAttributesRequest {})
        response.accountQuotas?.forEach { quotas ->
            val response = response.accountQuotas
            println("Name is: ${quotas.accountQuotaName}")
            println("Max value is ${quotas.max}")
        }
    }
}
```

- Para ter detalhes da API, consulte DescribeAccountAttributes na Referência de API do AWS SDK for Kotlin.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`DescribeDBEngineVersions`** com o AWS SDK ou a CLI

Os exemplos de código a seguir mostram como usar o `DescribeDBEngineVersions`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação no contexto no seguinte exemplo de código:

- Começar a usar instâncias de banco de dados

.NET

SDK para .NET

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
    /// <summary>
    /// Get a list of DB engine versions for a particular DB engine.
    /// </summary>
    /// <param name="engine">Name of the engine.</param>
    /// <param name="dbParameterGroupFamily">Optional parameter group family
name.</param>
    /// <returns>List of DBEngineVersions.</returns>
    public async Task<List<DBEngineVersion>> DescribeDBEngineVersions(string
engine,
        string dbParameterGroupFamily = null)
    {
        var response = await _amazonRDS.DescribeDBEngineVersionsAsync(
            new DescribeDBEngineVersionsRequest()
            {
                Engine = engine,
                DBParameterGroupFamily = dbParameterGroupFamily
            });
        return response.DBEngineVersions;
    }
```

- Para obter detalhes da API, consulte DescribeDBEngineVersions na Referência da API AWS SDK para .NET.

C++

SDK para C++

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
        Aws::Client::ClientConfiguration clientConfig;
        // Optional: Set to the AWS Region (overrides config file).
        // clientConfig.region = "us-east-1";

    Aws::RDS::RDSClient client(clientConfig);


//! Routine which gets available DB engine versions for an engine name and
//! an optional parameter group family.
/*!
 \sa getDBEngineVersions()
 \param engineName: A DB engine name.
 \param parameterGroupFamily: A parameter group family name, ignored if empty.
 \param engineVersionsResult: Vector of 'DBEngineVersion' objects returned by the
 routine.
 \param client: 'RDSClient' instance.
 \return bool: Successful completion.
 */
bool AwsDoc::RDS::getDBEngineVersions(const Aws::String &engineName,
                                      const Aws::String &parameterGroupFamily,

 Aws::Vector<Aws::RDS::Model::DBEngineVersion> &engineVersionsResult,
                                      const Aws::RDS::RDSClient &client) {
    Aws::RDS::Model::DescribeDBEngineVersionsRequest request;
    request.SetEngine(engineName);
    if (!parameterGroupFamily.empty()) {
        request.SetDBParameterGroupFamily(parameterGroupFamily);
    }

    engineVersionsResult.clear();
    Aws::String marker; // Used for pagination.
```

```
    do {
        if (!marker.empty()) {
            request.SetMarker(marker);
        }


        Aws::RDS::Model::DescribeDBEngineVersionsOutcome outcome =
                client.DescribeDBEngineVersions(request);

        if (outcome.IsSuccess()) {
            auto &engineVersions = outcome.GetResult().GetDBEngineVersions();
            engineVersionsResult.insert(engineVersionsResult.end(),
 engineVersions.begin(),
                                        engineVersions.end());
            marker = outcome.GetResult().GetMarker();
        }
        else {
            std::cerr << "Error with RDS::DescribeDBEngineVersionsRequest. "
                      << outcome.GetError().GetMessage()
                      << std::endl;
            return false;
        }

    } while (!marker.empty());


    return true;
}
```

- Para obter detalhes da API, consulte DescribeDBEngineVersions na Referência da API AWS SDK para C++.

CLI

AWS CLI

Como descrever as versões do mecanismo de banco de dados do mecanismo de banco de dados do MySQL

O exemplo de `describe-db-engine-versions` a seguir exibe os detalhes de cada uma das versões do mecanismo de banco de dados para o mecanismo de banco de dados especificado.

```
aws rds describe-db-engine-versions \
    --engine mysql
```

Saída:

```
{
    "DBEngineVersions": [
        {
            "Engine": "mysql",
            "EngineVersion": "5.5.46",
            "DBParameterGroupFamily": "mysql5.5",
            "DBEngineDescription": "MySQL Community Edition",
            "DBEngineVersionDescription": "MySQL 5.5.46",
            "ValidUpgradeTarget": [
                {
                    "Engine": "mysql",
                    "EngineVersion": "5.5.53",
                    "Description": "MySQL 5.5.53",
                    "AutoUpgrade": false,
                    "IsMajorVersionUpgrade": false
                },
                {
                    "Engine": "mysql",
                    "EngineVersion": "5.5.54",
                    "Description": "MySQL 5.5.54",
                    "AutoUpgrade": false,
                    "IsMajorVersionUpgrade": false
                },
                {
                    "Engine": "mysql",
                    "EngineVersion": "5.5.57",
                    "Description": "MySQL 5.5.57",
                    "AutoUpgrade": false,
                    "IsMajorVersionUpgrade": false
                },
                ...some output truncated...
            ]
        }
```

Para obter mais informações sobre o Amazon RDS, consulte O que é o Amazon Relational Database Service (Amazon RDS)? no Guia do usuário do Amazon RDS.

- Para obter detalhes da API, consulte DescribeDBEngineVersions na Referência de comandos da AWS CLI.

Go

SDK para Go V2

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
type DbInstances struct {
 RdsClient *rds.Client
}



// GetEngineVersions gets database engine versions that are available for the
 specified engine
// and parameter group family.
func (instances *DbInstances) GetEngineVersions(engine string,
 parameterGroupFamily string) (
 []types.DBEngineVersion, error) {
 output, err := instances.RdsClient.DescribeDBEngineVersions(context.TODO(),
  &rds.DescribeDBEngineVersionsInput{
   Engine:                 aws.String(engine),
   DBParameterGroupFamily: aws.String(parameterGroupFamily),
  })
 if err != nil {
  log.Printf("Couldn't get engine versions for %v: %v\n", engine, err)
  return nil, err
 } else {
  return output.DBEngineVersions, nil
 }
}
```

- Para obter detalhes da API, consulte DescribeDBEngineVersions na Referência da API AWS SDK para Go.

Java

SDK para Java 2.x

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
   public static void describeDBEngines(RdsClient rdsClient) {
       try {
           DescribeDbEngineVersionsRequest engineVersionsRequest =
DescribeDbEngineVersionsRequest.builder()
                   .defaultOnly(true)
                   .engine("mysql")
                   .maxRecords(20)
                   .build();

           DescribeDbEngineVersionsResponse response =
rdsClient.describeDBEngineVersions(engineVersionsRequest);
           List<DBEngineVersion> engines = response.dbEngineVersions();

           // Get all DBEngineVersion objects.
           for (DBEngineVersion engineOb : engines) {
               System.out.println("The name of the DB parameter group family for
the database engine is "
                       + engineOb.dbParameterGroupFamily());
               System.out.println("The name of the database engine " +
engineOb.engine());
               System.out.println("The version number of the database engine " +
engineOb.engineVersion());
           }

       } catch (RdsException e) {
           System.out.println(e.getLocalizedMessage());
```

```
            System.exit(1);
        }
    }
```

- Para obter detalhes da API, consulte DescribeDBEngineVersions na Referência da API AWS SDK for Java 2.x.

Python

SDK para Python (Boto3).

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
class InstanceWrapper:
    """Encapsulates Amazon RDS DB instance actions."""

    def __init__(self, rds_client):
        """
        :param rds_client: A Boto3 Amazon RDS client.
        """
        self.rds_client = rds_client

    @classmethod
    def from_client(cls):
        """
        Instantiates this class from a Boto3 client.
        """
        rds_client = boto3.client("rds")
        return cls(rds_client)


    def get_engine_versions(self, engine, parameter_group_family=None):
        """
        Gets database engine versions that are available for the specified engine
        and parameter group family.
```

```
        :param engine: The database engine to look up.
        :param parameter_group_family: When specified, restricts the returned
list of
                                       engine versions to those that are
compatible with
                                       this parameter group family.
        :return: The list of database engine versions.
        """
        try:
            kwargs = {"Engine": engine}
            if parameter_group_family is not None:
                kwargs["DBParameterGroupFamily"] = parameter_group_family
            response = self.rds_client.describe_db_engine_versions(**kwargs)
            versions = response["DBEngineVersions"]
        except ClientError as err:
            logger.error(
                "Couldn't get engine versions for %s. Here's why: %s: %s",
                engine,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return versions
```

- Para ter detalhes da API, consulte DescribeDBEngineVersions na Referência de API do AWS SDK para Python (Boto3).

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`DescribeDBInstances`** com o AWS SDK ou a CLI

Os exemplos de código a seguir mostram como usar o `DescribeDBInstances`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação no contexto no seguinte exemplo de código:

- Começar a usar instâncias de banco de dados

.NET

SDK para .NET

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
   /// <summary>
   /// Returns a list of DB instances.
   /// </summary>
   /// <param name="dbInstanceIdentifier">Optional name of a specific DB
instance.</param>
   /// <returns>List of DB instances.</returns>
   public async Task<List<DBInstance>> DescribeDBInstances(string
dbInstanceIdentifier = null)
   {
       var results = new List<DBInstance>();
       var instancesPaginator = _amazonRDS.Paginators.DescribeDBInstances(
           new DescribeDBInstancesRequest
           {
               DBInstanceIdentifier = dbInstanceIdentifier
           });
       // Get the entire list using the paginator.
       await foreach (var instances in instancesPaginator.DBInstances)
       {
           results.Add(instances);
       }
       return results;
   }
```

- Para obter detalhes da API, consulte DescribeDBInstances na Referência da API AWS SDK para .NET.

C++

SDK para C++

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
        Aws::Client::ClientConfiguration clientConfig;
        // Optional: Set to the AWS Region (overrides config file).
        // clientConfig.region = "us-east-1";

    Aws::RDS::RDSClient client(clientConfig);


//! Routine which gets a DB instance description.
/*!
 \sa describeDBInstance()
 \param dbInstanceIdentifier: A DB instance identifier.
 \param instanceResult: The 'DBInstance' object containing the description.
 \param client: 'RDSClient' instance.
 \return bool: Successful completion.
 */
bool AwsDoc::RDS::describeDBInstance(const Aws::String &dbInstanceIdentifier,
                                     Aws::RDS::Model::DBInstance &instanceResult,
                                     const Aws::RDS::RDSClient &client) {
    Aws::RDS::Model::DescribeDBInstancesRequest request;
    request.SetDBInstanceIdentifier(dbInstanceIdentifier);

    Aws::RDS::Model::DescribeDBInstancesOutcome outcome =
            client.DescribeDBInstances(request);

    bool result = true;
    if (outcome.IsSuccess()) {
        instanceResult = outcome.GetResult().GetDBInstances()[0];
    }
    else if (outcome.GetError().GetErrorType() !=
             Aws::RDS::RDSErrors::D_B_INSTANCE_NOT_FOUND_FAULT) {
        result = false;
        std::cerr << "Error with RDS::DescribeDBInstances. "
```

```
                  << outcome.GetError().GetMessage()
                  << std::endl;
    }
        // This example does not log an error if the DB instance does not exist.
        // Instead, instanceResult is set to empty.
    else {
        instanceResult = Aws::RDS::Model::DBInstance();
    }

    return result;
}
```

- Para obter detalhes da API, consulte DescribeDBInstances na Referência da API AWS SDK para C++.

CLI

AWS CLI

Como descrever uma instância de banco de dados

O exemplo de `describe-db-instances` a seguir recupera detalhes sobre a instância de banco de dados especificada.

```
aws rds describe-db-instances \
    --db-instance-identifier mydbinstancecf
```

Saída:

```
{
    "DBInstances": [
        {
            "DBInstanceIdentifier": "mydbinstancecf",
            "DBInstanceClass": "db.t3.small",
            "Engine": "mysql",
            "DBInstanceStatus": "available",
            "MasterUsername": "masterawsuser",
            "Endpoint": {
                "Address": "mydbinstancecf.abcexample.us-
east-1.rds.amazonaws.com",
                "Port": 3306,
```

```
                "HostedZoneId": "Z2R2ITUGPM61AM"
            },
            ...some output truncated...
        }
    ]
}
```

- Para obter detalhes da API, consulte DescribeDBInstances na Referência de comandos da AWS CLI.

Go

SDK para Go V2

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
type DbInstances struct {
 RdsClient *rds.Client
}



// GetInstance gets data about a DB instance.
func (instances *DbInstances) GetInstance(instanceName string) (
 *types.DBInstance, error) {
 output, err := instances.RdsClient.DescribeDBInstances(context.TODO(),
  &rds.DescribeDBInstancesInput{
   DBInstanceIdentifier: aws.String(instanceName),
  })
 if err != nil {
  var notFoundError *types.DBInstanceNotFoundFault
  if errors.As(err, &notFoundError) {
   log.Printf("DB instance %v does not exist.\n", instanceName)
   err = nil
  } else {
   log.Printf("Couldn't get instance %v: %v\n", instanceName, err)
```

```
  }
  return nil, err
 } else {
  return &output.DBInstances[0], nil
 }
}
```

- Para obter detalhes da API, consulte DescribeDBInstances na Referência da API AWS SDK para Go.

Java

SDK para Java 2.x

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.rds.RdsClient;
import software.amazon.awssdk.services.rds.model.DescribeDbInstancesResponse;
import software.amazon.awssdk.services.rds.model.DBInstance;
import software.amazon.awssdk.services.rds.model.RdsException;
import java.util.List;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class DescribeDBInstances {

    public static void main(String[] args) {
```

```
        Region region = Region.US_EAST_1;
        RdsClient rdsClient = RdsClient.builder()
                .region(region)
                .build();

        describeInstances(rdsClient);
        rdsClient.close();
    }

    public static void describeInstances(RdsClient rdsClient) {
        try {
            DescribeDbInstancesResponse response =
 rdsClient.describeDBInstances();
            List<DBInstance> instanceList = response.dbInstances();
            for (DBInstance instance : instanceList) {
                System.out.println("Instance ARN is: " +
 instance.dbInstanceArn());
                System.out.println("The Engine is " + instance.engine());
                System.out.println("Connection endpoint is" +
 instance.endpoint().address());
            }

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
    }
}
```

- Para obter detalhes da API, consulte DescribeDBInstances na Referência da API AWS SDK for Java 2.x.

Kotlin

SDK para Kotlin

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
suspend fun describeInstances() {
    RdsClient { region = "us-west-2" }.use { rdsClient ->
        val response = rdsClient.describeDbInstances(DescribeDbInstancesRequest
 {})
        response.dbInstances?.forEach { instance ->
            println("Instance Identifier is ${instance.dbInstanceIdentifier}")
            println("The Engine is ${instance.engine}")
            println("Connection endpoint is ${instance.endpoint?.address}")
        }
    }
}
```

- Para obter detalhes da API, consulte DescribeDBInstances na Referência da API AWS SDK para Kotlin.

PHP

SDK para PHP

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
require __DIR__ . '/vendor/autoload.php';

use Aws\Exception\AwsException;


//Create an RDSClient
$rdsClient = new Aws\Rds\RdsClient([
    'region' => 'us-east-2'
]);

try {
    $result = $rdsClient->describeDBInstances();
    foreach ($result['DBInstances'] as $instance) {
        print('<p>DB Identifier: ' . $instance['DBInstanceIdentifier']);
```

```
        print('<br />Endpoint: ' . $instance['Endpoint']["Address"]
            . ':' . $instance['Endpoint']["Port"]);
        print('<br />Current Status: ' . $instance["DBInstanceStatus"]);
        print('</p>');
    }
    print(" Raw Result ");
    var_dump($result);
} catch (AwsException $e) {
    echo $e->getMessage();
    echo "\n";
}
```

- Para obter detalhes da API, consulte DescribeDBInstances na Referência da API AWS SDK para PHP.

Python

SDK para Python (Boto3).

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
class InstanceWrapper:
    """Encapsulates Amazon RDS DB instance actions."""

    def __init__(self, rds_client):
        """
        :param rds_client: A Boto3 Amazon RDS client.
        """
        self.rds_client = rds_client

    @classmethod
    def from_client(cls):
        """
        Instantiates this class from a Boto3 client.
        """
```

```
        rds_client = boto3.client("rds")
        return cls(rds_client)


    def get_db_instance(self, instance_id):
        """
        Gets data about a DB instance.

        :param instance_id: The ID of the DB instance to retrieve.
        :return: The retrieved DB instance.
        """
        try:
            response = self.rds_client.describe_db_instances(
                DBInstanceIdentifier=instance_id
            )
            db_inst = response["DBInstances"][0]
        except ClientError as err:
            if err.response["Error"]["Code"] == "DBInstanceNotFound":
                logger.info("Instance %s does not exist.", instance_id)
            else:
                logger.error(
                    "Couldn't get DB instance %s. Here's why: %s: %s",
                    instance_id,
                    err.response["Error"]["Code"],
                    err.response["Error"]["Message"],
                )
                raise
        else:
            return db_inst
```

- Para obter detalhes da API, consulte DescribeDBInstances na Referência da API AWS SDK para Python (Boto3).

Ruby

SDK para Ruby

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
require "aws-sdk-rds"  # v2: require 'aws-sdk'

# List all Amazon Relational Database Service (Amazon RDS) DB instances.
#
# @param rds_resource [Aws::RDS::Resource] An SDK for Ruby Amazon RDS resource.
# @return [Array, nil] List of all DB instances, or nil if error.
def list_instances(rds_resource)
  db_instances = []
  rds_resource.db_instances.each do |i|
    db_instances.append({
                          "name": i.id,
                          "status": i.db_instance_status
                        })
  end
  db_instances
rescue Aws::Errors::ServiceError => e
  puts "Couldn't list instances:\n#{e.message}"
end
```

- Para ter detalhes da API, consulte DescribeDBInstances na Referência de API do AWS SDK para Ruby.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`DescribeDBParameterGroups`** com o AWS SDK ou a CLI

Os exemplos de código a seguir mostram como usar o `DescribeDBParameterGroups`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação no contexto no seguinte exemplo de código:

- Começar a usar instâncias de banco de dados

.NET

SDK para .NET

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
    /// <summary>
    /// Get descriptions of DB parameter groups.
    /// </summary>
    /// <param name="name">Optional name of the DB parameter group to describe.</
param>
    /// <returns>The list of DB parameter group descriptions.</returns>
    public async Task<List<DBParameterGroup>> DescribeDBParameterGroups(string
 name = null)
    {
        var response = await _amazonRDS.DescribeDBParameterGroupsAsync(
            new DescribeDBParameterGroupsRequest()
            {
                DBParameterGroupName = name
            });
        return response.DBParameterGroups;
    }
```

- Para obter detalhes da API, consulte DescribeDBParameterGroups na Referência da API AWS SDK para .NET.

C++

SDK para C++

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
        Aws::Client::ClientConfiguration clientConfig;
        // Optional: Set to the AWS Region (overrides config file).
        // clientConfig.region = "us-east-1";

    Aws::RDS::RDSClient client(clientConfig);

        Aws::RDS::Model::DescribeDBParameterGroupsRequest request;
        request.SetDBParameterGroupName(PARAMETER_GROUP_NAME);

        Aws::RDS::Model::DescribeDBParameterGroupsOutcome outcome =
                client.DescribeDBParameterGroups(request);

        if (outcome.IsSuccess()) {
            std::cout << "DB parameter group named '" <<
                      PARAMETER_GROUP_NAME << "' already exists." << std::endl;
            dbParameterGroupFamily = outcome.GetResult().GetDBParameterGroups()
[0].GetDBParameterGroupFamily();
        }

        else {
            std::cerr << "Error with RDS::DescribeDBParameterGroups. "
                      << outcome.GetError().GetMessage()
                      << std::endl;
            return false;
        }
```

- Para obter detalhes da API, consulte DescribeDBParameterGroups na Referência da API AWS SDK para C++.

CLI

AWS CLI

Como descrever um grupo de parâmetros de banco de dados

O exemplo de `describe-db-parameter-groups` a seguir recupera detalhes sobre os grupos de parâmetros de banco de dados.

```
aws rds describe-db-parameter-groups
```

Saída:

```
{
    "DBParameterGroups": [
        {
            "DBParameterGroupName": "default.aurora-mysql5.7",
            "DBParameterGroupFamily": "aurora-mysql5.7",
            "Description": "Default parameter group for aurora-mysql5.7",
            "DBParameterGroupArn": "arn:aws:rds:us-
east-1:123456789012:pg:default.aurora-mysql5.7"
        },
        {
            "DBParameterGroupName": "default.aurora-postgresql9.6",
            "DBParameterGroupFamily": "aurora-postgresql9.6",
            "Description": "Default parameter group for aurora-postgresql9.6",
            "DBParameterGroupArn": "arn:aws:rds:us-
east-1:123456789012:pg:default.aurora-postgresql9.6"
        },
        {
            "DBParameterGroupName": "default.aurora5.6",
            "DBParameterGroupFamily": "aurora5.6",
            "Description": "Default parameter group for aurora5.6",
            "DBParameterGroupArn": "arn:aws:rds:us-
east-1:123456789012:pg:default.aurora5.6"
        },
        {
            "DBParameterGroupName": "default.mariadb10.1",
            "DBParameterGroupFamily": "mariadb10.1",
            "Description": "Default parameter group for mariadb10.1",
            "DBParameterGroupArn": "arn:aws:rds:us-
east-1:123456789012:pg:default.mariadb10.1"
        },
```

```
        ...some output truncated...
    ]
}
```

Para obter mais informações, consulte Como trabalhar com grupos de parâmetros de banco de dados no Guia do usuário do Amazon RDS.

- Para obter detalhes da API, consulte DescribeDBParameterGroups na Referência de comandos da AWS CLI.

Go

SDK para Go V2

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
type DbInstances struct {
 RdsClient *rds.Client
}



// GetParameterGroup gets a DB parameter group by name.
func (instances *DbInstances) GetParameterGroup(parameterGroupName string) (
 *types.DBParameterGroup, error) {
 output, err := instances.RdsClient.DescribeDBParameterGroups(
  context.TODO(), &rds.DescribeDBParameterGroupsInput{
   DBParameterGroupName: aws.String(parameterGroupName),
  })
 if err != nil {
  var notFoundError *types.DBParameterGroupNotFoundFault
  if errors.As(err, &notFoundError) {
   log.Printf("Parameter group %v does not exist.\n", parameterGroupName)
   err = nil
  } else {
   log.Printf("Error getting parameter group %v: %v\n", parameterGroupName, err)
  }
```

```
  return nil, err
 } else {
  return &output.DBParameterGroups[0], err
 }
}
```

- Para obter detalhes da API, consulte DescribeDBParameterGroups na Referência da API AWS SDK para Go.

Java

SDK para Java 2.x

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
   public static void describeDbParameterGroups(RdsClient rdsClient, String
dbGroupName) {
       try {
           DescribeDbParameterGroupsRequest groupsRequest =
DescribeDbParameterGroupsRequest.builder()
                   .dbParameterGroupName(dbGroupName)
                   .maxRecords(20)
                   .build();

           DescribeDbParameterGroupsResponse response =
rdsClient.describeDBParameterGroups(groupsRequest);
           List<DBParameterGroup> groups = response.dbParameterGroups();
           for (DBParameterGroup group : groups) {
               System.out.println("The group name is " +
group.dbParameterGroupName());
               System.out.println("The group description is " +
group.description());
           }

       } catch (RdsException e) {
```

```
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
    }
```

- Para obter detalhes da API, consulte DescribeDBParameterGroups na Referência da API AWS SDK for Java 2.x.

Python

SDK para Python (Boto3).

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
class InstanceWrapper:
    """Encapsulates Amazon RDS DB instance actions."""

    def __init__(self, rds_client):
        """
        :param rds_client: A Boto3 Amazon RDS client.
        """
        self.rds_client = rds_client

    @classmethod
    def from_client(cls):
        """
        Instantiates this class from a Boto3 client.
        """
        rds_client = boto3.client("rds")
        return cls(rds_client)


    def get_parameter_group(self, parameter_group_name):
        """
        Gets a DB parameter group.
```

```
        :param parameter_group_name: The name of the parameter group to retrieve.
        :return: The parameter group.
        """
        try:
            response = self.rds_client.describe_db_parameter_groups(
                DBParameterGroupName=parameter_group_name
            )
            parameter_group = response["DBParameterGroups"][0]
        except ClientError as err:
            if err.response["Error"]["Code"] == "DBParameterGroupNotFound":
                logger.info("Parameter group %s does not exist.",
parameter_group_name)
            else:
                logger.error(
                    "Couldn't get parameter group %s. Here's why: %s: %s",
                    parameter_group_name,
                    err.response["Error"]["Code"],
                    err.response["Error"]["Message"],
                )
                raise
        else:
            return parameter_group
```

- Para obter detalhes da API, consulte DescribeDBParameterGroups na referência da API do AWS SDK para Python (Boto3).

Ruby

SDK para Ruby

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
require "aws-sdk-rds"  # v2: require 'aws-sdk'

# List all Amazon Relational Database Service (Amazon RDS) parameter groups.
```

```
#
# @param rds_resource [Aws::RDS::Resource] An SDK for Ruby Amazon RDS resource.
# @return [Array, nil] List of all parameter groups, or nil if error.
def list_parameter_groups(rds_resource)
  parameter_groups = []
  rds_resource.db_parameter_groups.each do |p|
    parameter_groups.append({
                              "name": p.db_parameter_group_name,
                              "description": p.description
                            })
  end
  parameter_groups
rescue Aws::Errors::ServiceError => e
  puts "Couldn't list parameter groups:\n #{e.message}"
end
```

- Para ter detalhes da API, consulte DescribeDBParameterGroups na Referência de API do AWS SDK para Ruby.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`DescribeDBParameters`** com o AWS SDK ou a CLI

Os exemplos de código a seguir mostram como usar o `DescribeDBParameters`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação no contexto no seguinte exemplo de código:

- Começar a usar instâncias de banco de dados

.NET

SDK para .NET

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
   /// <summary>
   /// Get a list of DB parameters from a specific parameter group.
   /// </summary>
   /// <param name="dbParameterGroupName">Name of a specific DB parameter
group.</param>
   /// <param name="source">Optional source for selecting parameters.</param>
   /// <returns>List of parameter values.</returns>
   public async Task<List<Parameter>> DescribeDBParameters(string
dbParameterGroupName, string source = null)
   {
       var results = new List<Parameter>();
       var paginateParameters = _amazonRDS.Paginators.DescribeDBParameters(
           new DescribeDBParametersRequest()
           {
               DBParameterGroupName = dbParameterGroupName,
               Source = source
           });
       // Get the entire list using the paginator.
       await foreach (var parameters in paginateParameters.Parameters)
       {
           results.Add(parameters);
       }
       return results;
   }
```

- Para obter detalhes da API, consulte DescribeDBParameters na Referência da API AWS SDK para .NET.

C++

SDK para C++

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
        Aws::Client::ClientConfiguration clientConfig;
        // Optional: Set to the AWS Region (overrides config file).
        // clientConfig.region = "us-east-1";

    Aws::RDS::RDSClient client(clientConfig);


//! Routine which gets DB parameters using the 'DescribeDBParameters' api.
/*!
 \sa getDBParameters()
 \param parameterGroupName: The name of the parameter group.
 \param namePrefix: Prefix string to filter results by parameter name.
 \param source: A source such as 'user', ignored if empty.
 \param parametersResult: Vector of 'Parameter' objects returned by the routine.
 \param client: 'RDSClient' instance.
 \return bool: Successful completion.
 */
bool AwsDoc::RDS::getDBParameters(const Aws::String &parameterGroupName,
                                  const Aws::String &namePrefix,
                                  const Aws::String &source,
                                  Aws::Vector<Aws::RDS::Model::Parameter>
 &parametersResult,
                                  const Aws::RDS::RDSClient &client) {
    Aws::String marker;
    do {
        Aws::RDS::Model::DescribeDBParametersRequest request;
        request.SetDBParameterGroupName(PARAMETER_GROUP_NAME);
        if (!marker.empty()) {
            request.SetMarker(marker);
        }
        if (!source.empty()) {
            request.SetSource(source);
```

```
        }

        Aws::RDS::Model::DescribeDBParametersOutcome outcome =
                client.DescribeDBParameters(request);

        if (outcome.IsSuccess()) {
            const Aws::Vector<Aws::RDS::Model::Parameter> &parameters =
                    outcome.GetResult().GetParameters();
            for (const Aws::RDS::Model::Parameter &parameter: parameters) {
                if (!namePrefix.empty()) {
                    if (parameter.GetParameterName().find(namePrefix) == 0) {
                        parametersResult.push_back(parameter);
                    }
                }
                else {
                    parametersResult.push_back(parameter);
                }
            }

            marker = outcome.GetResult().GetMarker();
        }
        else {
            std::cerr << "Error with RDS::DescribeDBParameters. "
                      << outcome.GetError().GetMessage()
                      << std::endl;
            return false;
        }
    } while (!marker.empty());

    return true;
}
```

- Para obter detalhes da API, consulte DescribeDBParameters na Referência da API AWS SDK para C++.

CLI

AWS CLI

Como descrever os parâmetros de um grupo de parâmetros de banco de dados

O exemplo de `describe-db-parameters` a seguir recupera detalhes sobre o grupo de parâmetros de banco de dados especificado.

```
aws rds describe-db-parameters \
    --db-parameter-group-name mydbpg
```

Saída:

```
{
    "Parameters": [
        {
            "ParameterName": "allow-suspicious-udfs",
            "Description": "Controls whether user-defined functions that have
 only an xxx symbol for the main function can be loaded",
            "Source": "engine-default",
            "ApplyType": "static",
            "DataType": "boolean",
            "AllowedValues": "0,1",
            "IsModifiable": false,
            "ApplyMethod": "pending-reboot"
        },
        {
            "ParameterName": "auto_generate_certs",
            "Description": "Controls whether the server autogenerates SSL key and
 certificate files in the data directory, if they do not already exist.",
            "Source": "engine-default",
            "ApplyType": "static",
            "DataType": "boolean",
            "AllowedValues": "0,1",
            "IsModifiable": false,
            "ApplyMethod": "pending-reboot"
        },
        ...some output truncated...
    ]
}
```

Para obter mais informações, consulte Como trabalhar com grupos de parâmetros de banco de dados no Guia do usuário do Amazon RDS.

- Para obter detalhes da API, consulte DescribeDBParameters na Referência de comandos da AWS CLI.

Go

SDK para Go V2

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
type DbInstances struct {
 RdsClient *rds.Client
}



// GetParameters gets the parameters that are contained in a DB parameter group.
func (instances *DbInstances) GetParameters(parameterGroupName string, source
 string) (
 []types.Parameter, error) {

 var output *rds.DescribeDBParametersOutput
 var params []types.Parameter
 var err error
 parameterPaginator := rds.NewDescribeDBParametersPaginator(instances.RdsClient,
  &rds.DescribeDBParametersInput{
   DBParameterGroupName: aws.String(parameterGroupName),
   Source:               aws.String(source),
  })
 for parameterPaginator.HasMorePages() {
  output, err = parameterPaginator.NextPage(context.TODO())
  if err != nil {
   log.Printf("Couldn't get parameters for %v: %v\n", parameterGroupName, err)
   break
  } else {
   params = append(params, output.Parameters...)
  }
 }
 return params, err
}
```

- Para obter detalhes da API, consulte DescribeDBParameters na Referência da API AWS SDK para Go.

Java

SDK para Java 2.x

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
   // Retrieve parameters in the group.
   public static void describeDbParameters(RdsClient rdsClient, String
dbGroupName, int flag) {
       try {
           DescribeDbParametersRequest dbParameterGroupsRequest;
           if (flag == 0) {
               dbParameterGroupsRequest = DescribeDbParametersRequest.builder()
                       .dbParameterGroupName(dbGroupName)
                       .build();
           } else {
               dbParameterGroupsRequest = DescribeDbParametersRequest.builder()
                       .dbParameterGroupName(dbGroupName)
                       .source("user")
                       .build();
           }

           DescribeDbParametersResponse response =
rdsClient.describeDBParameters(dbParameterGroupsRequest);
           List<Parameter> dbParameters = response.parameters();
           String paraName;
           for (Parameter para : dbParameters) {
               // Only print out information about either auto_increment_offset
or
               // auto_increment_increment.
               paraName = para.parameterName();
               if ((paraName.compareTo("auto_increment_offset") == 0)
```

```
                        || (paraName.compareTo("auto_increment_increment ") ==
0)) {
                    System.out.println("*** The parameter name is  " + paraName);
                    System.out.println("*** The parameter value is  " +
para.parameterValue());
                    System.out.println("*** The parameter data type is " +
para.dataType());
                    System.out.println("*** The parameter description is " +
para.description());
                    System.out.println("*** The parameter allowed values  is " +
para.allowedValues());
                }
            }

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
    }
```

- Para obter detalhes da API, consulte DescribeDBParameters na Referência da API AWS SDK for Java 2.x.

Python

SDK para Python (Boto3).

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
class InstanceWrapper:
    """Encapsulates Amazon RDS DB instance actions."""

    def __init__(self, rds_client):
        """
        :param rds_client: A Boto3 Amazon RDS client.
        """
```

```
        self.rds_client = rds_client

    @classmethod
    def from_client(cls):
        """
        Instantiates this class from a Boto3 client.
        """
        rds_client = boto3.client("rds")
        return cls(rds_client)


    def get_parameters(self, parameter_group_name, name_prefix="", source=None):
        """
        Gets the parameters that are contained in a DB parameter group.

        :param parameter_group_name: The name of the parameter group to query.
        :param name_prefix: When specified, the retrieved list of parameters is
filtered
                            to contain only parameters that start with this
prefix.
        :param source: When specified, only parameters from this source are
retrieved.
                       For example, a source of 'user' retrieves only parameters
that
                       were set by a user.
        :return: The list of requested parameters.
        """
        try:
            kwargs = {"DBParameterGroupName": parameter_group_name}
            if source is not None:
                kwargs["Source"] = source
            parameters = []
            paginator = self.rds_client.get_paginator("describe_db_parameters")
            for page in paginator.paginate(**kwargs):
                parameters += [
                    p
                    for p in page["Parameters"]
                    if p["ParameterName"].startswith(name_prefix)
                ]
        except ClientError as err:
            logger.error(
                "Couldn't get parameters for %s. Here's why: %s: %s",
                parameter_group_name,
                err.response["Error"]["Code"],
```

```
                err.response["Error"]["Message"],
            )
            raise
        else:
            return parameters
```

- Para obter detalhes da API, consulte DescribeDBParameters na Referência da API AWS SDK para Python (Boto3).

Ruby

SDK para Ruby

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
require "aws-sdk-rds"  # v2: require 'aws-sdk'

# List all Amazon Relational Database Service (Amazon RDS) parameter groups.
#
# @param rds_resource [Aws::RDS::Resource] An SDK for Ruby Amazon RDS resource.
# @return [Array, nil] List of all parameter groups, or nil if error.
def list_parameter_groups(rds_resource)
  parameter_groups = []
  rds_resource.db_parameter_groups.each do |p|
    parameter_groups.append({
                              "name": p.db_parameter_group_name,
                              "description": p.description
                            })
  end
  parameter_groups
rescue Aws::Errors::ServiceError => e
  puts "Couldn't list parameter groups:\n #{e.message}"
end
```

- Para ter detalhes da API, consulte DescribeDBParameters na Referência de API do AWS SDK para Ruby.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`DescribeDBSnapshots`** com o AWS SDK ou a CLI

Os exemplos de código a seguir mostram como usar o `DescribeDBSnapshots`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação no contexto no seguinte exemplo de código:

- Começar a usar instâncias de banco de dados

.NET

SDK para .NET

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
   /// <summary>
   /// Return a list of DB snapshots for a particular DB instance.
   /// </summary>
   /// <param name="dbInstanceIdentifier">DB instance identifier.</param>
   /// <returns>List of DB snapshots.</returns>
   public async Task<List<DBSnapshot>> DescribeDBSnapshots(string
dbInstanceIdentifier)
   {
       var results = new List<DBSnapshot>();
       var snapshotsPaginator = _amazonRDS.Paginators.DescribeDBSnapshots(
           new DescribeDBSnapshotsRequest()
           {
```

```
                DBInstanceIdentifier = dbInstanceIdentifier
            });

        // Get the entire list using the paginator.
        await foreach (var snapshots in snapshotsPaginator.DBSnapshots)
        {
            results.Add(snapshots);
        }
        return results;
    }
```

- Para obter detalhes da API, consulte DescribeDBSnapshots na Referência da API AWS SDK para .NET.

C++

SDK para C++

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
    Aws::Client::ClientConfiguration clientConfig;
    // Optional: Set to the AWS Region (overrides config file).
    // clientConfig.region = "us-east-1";

Aws::RDS::RDSClient client(clientConfig);

        Aws::RDS::Model::DescribeDBSnapshotsRequest request;
        request.SetDBSnapshotIdentifier(snapshotID);

        Aws::RDS::Model::DescribeDBSnapshotsOutcome outcome =
                client.DescribeDBSnapshots(request);

        if (outcome.IsSuccess()) {
            snapshot = outcome.GetResult().GetDBSnapshots()[0];
        }
```

```
            else {
                std::cerr << "Error with RDS::DescribeDBSnapshots. "
                          << outcome.GetError().GetMessage()
                          << std::endl;
                cleanUpResources(PARAMETER_GROUP_NAME, DB_INSTANCE_IDENTIFIER,
 client);
                return false;
            }
```

- Para obter detalhes da API, consulte DescribeDBSnapshots na Referência da API AWS SDK para C++.

CLI

AWS CLI

Exemplo 1: como descrever um snapshot de banco de dados de uma instância de banco de dados

O exemplo de `describe-db-snapshots` a seguir recupera os detalhes de um snapshot de banco de dados para uma instância de banco de dados.

```
aws rds describe-db-snapshots \
    --db-snapshot-identifier mydbsnapshot
```

Saída:

```
{
    "DBSnapshots": [
        {
            "DBSnapshotIdentifier": "mydbsnapshot",
            "DBInstanceIdentifier": "mysqldb",
            "SnapshotCreateTime": "2018-02-08T22:28:08.598Z",
            "Engine": "mysql",
            "AllocatedStorage": 20,
            "Status": "available",
            "Port": 3306,
            "AvailabilityZone": "us-east-1f",
            "VpcId": "vpc-6594f31c",
            "InstanceCreateTime": "2018-02-08T22:24:55.973Z",
```

```
            "MasterUsername": "mysqladmin",
            "EngineVersion": "5.6.37",
            "LicenseModel": "general-public-license",
            "SnapshotType": "manual",
            "OptionGroupName": "default:mysql-5-6",
            "PercentProgress": 100,
            "StorageType": "gp2",
            "Encrypted": false,
            "DBSnapshotArn": "arn:aws:rds:us-
east-1:123456789012:snapshot:mydbsnapshot",
            "IAMDatabaseAuthenticationEnabled": false,
            "ProcessorFeatures": [],
            "DbiResourceId": "db-AKIAIOSFODNN7EXAMPLE"
        }
    ]
}
```

Para obter mais informações, consulte Criar um snapshot de banco de dados no Guia do usuário do Amazon RDS.

Exemplo 2: como encontrar o número de snapshots manuais criados

O exemplo de `describe-db-snapshots` a seguir usa o operador `length` na opção `--query` para retornar o número de snapshots manuais que foram criados em uma determinada região da AWS.

```
aws rds describe-db-snapshots \
    --snapshot-type manual \
    --query "length(*[].{DBSnapshots:SnapshotType})" \
    --region eu-central-1
```

Saída:

```
35
```

Para obter mais informações, consulte Criar um snapshot de banco de dados no Guia do usuário do Amazon RDS.

- Para obter detalhes da API, consulte DescribeDBSnapshots na Referência de comandos da AWS CLI.

Go

SDK para Go V2

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
type DbInstances struct {
 RdsClient *rds.Client
}



// GetSnapshot gets a DB instance snapshot.
func (instances *DbInstances) GetSnapshot(snapshotName string)
 (*types.DBSnapshot, error) {
 output, err := instances.RdsClient.DescribeDBSnapshots(context.TODO(),
  &rds.DescribeDBSnapshotsInput{
   DBSnapshotIdentifier: aws.String(snapshotName),
  })
 if err != nil {
  log.Printf("Couldn't get snapshot %v: %v\n", snapshotName, err)
  return nil, err
 } else {
  return &output.DBSnapshots[0], nil
 }
}
```

- Para obter detalhes da API, consulte DescribeDBSnapshots na Referência da API AWS SDK para Go.

Python

SDK para Python (Boto3).

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
class InstanceWrapper:
    """Encapsulates Amazon RDS DB instance actions."""

    def __init__(self, rds_client):
        """
        :param rds_client: A Boto3 Amazon RDS client.
        """
        self.rds_client = rds_client

    @classmethod
    def from_client(cls):
        """
        Instantiates this class from a Boto3 client.
        """
        rds_client = boto3.client("rds")
        return cls(rds_client)


    def get_snapshot(self, snapshot_id):
        """
        Gets a DB instance snapshot.

        :param snapshot_id: The ID of the snapshot to retrieve.
        :return: The retrieved snapshot.
        """
        try:
            response = self.rds_client.describe_db_snapshots(
                DBSnapshotIdentifier=snapshot_id
            )
            snapshot = response["DBSnapshots"][0]
        except ClientError as err:
            logger.error(
```

```
                "Couldn't get snapshot %s. Here's why: %s: %s",
                snapshot_id,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return snapshot
```

- Para obter detalhes da API, consulte DescribeDBSnapshots na referência da API do AWS SDK para Python (Boto3).

Ruby

SDK para Ruby

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
require "aws-sdk-rds" # v2: require 'aws-sdk'

# List all Amazon Relational Database Service (Amazon RDS) DB instance
# snapshots.
#
# @param rds_resource [Aws::RDS::Resource] An SDK for Ruby Amazon RDS resource.
# @return instance_snapshots [Array, nil] All instance snapshots, or nil if
 error.
def list_instance_snapshots(rds_resource)
  instance_snapshots = []
  rds_resource.db_snapshots.each do |s|
    instance_snapshots.append({
                                "id": s.snapshot_id,
                                "status": s.status
                              })
  end
  instance_snapshots
```

```
rescue Aws::Errors::ServiceError => e
  puts "Couldn't list instance snapshots:\n #{e.message}"
end
```

- Para ter detalhes da API, consulte DescribeDBSnapshots na Referência de API do AWS SDK para Ruby.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`DescribeOrderableDBInstanceOptions`** com o AWS SDK ou a CLI

Os exemplos de código a seguir mostram como usar o `DescribeOrderableDBInstanceOptions`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação no contexto no seguinte exemplo de código:

- Começar a usar instâncias de banco de dados

.NET

SDK para .NET

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
/// <summary>
/// Get a list of orderable DB instance options for a specific
/// engine and engine version.
/// </summary>
```

```
    /// <param name="engine">Name of the engine.</param>
    /// <param name="engineVersion">Version of the engine.</param>
    /// <returns>List of OrderableDBInstanceOptions.</returns>
    public async Task<List<OrderableDBInstanceOption>>
DescribeOrderableDBInstanceOptions(string engine, string engineVersion)
    {
        // Use a paginator to get a list of DB instance options.
        var results = new List<OrderableDBInstanceOption>();
        var paginateInstanceOptions =
_amazonRDS.Paginators.DescribeOrderableDBInstanceOptions(
            new DescribeOrderableDBInstanceOptionsRequest()
            {
                Engine = engine,
                EngineVersion = engineVersion,
            });
        // Get the entire list using the paginator.
        await foreach (var instanceOptions in
paginateInstanceOptions.OrderableDBInstanceOptions)
        {
            results.Add(instanceOptions);
        }
        return results;
    }
```

- Para obter detalhes da API, consulte DescribeOrderableDBInstanceOptions na Referência da API AWS SDK para .NET.

C++

SDK para C++

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
        Aws::Client::ClientConfiguration clientConfig;
        // Optional: Set to the AWS Region (overrides config file).
```

```
        // clientConfig.region = "us-east-1";

    Aws::RDS::RDSClient client(clientConfig);


//! Routine which gets available 'micro' DB instance classes, displays the list
//! to the user, and returns the user selection.
/*!
 \sa chooseMicroDBInstanceClass()
 \param engineName: The DB engine name.
 \param engineVersion: The DB engine version.
 \param dbInstanceClass: String for DB instance class chosen by the user.
 \param client: 'RDSClient' instance.
 \return bool: Successful completion.
 */
bool AwsDoc::RDS::chooseMicroDBInstanceClass(const Aws::String &engine,
                                             const Aws::String &engineVersion,
                                             Aws::String &dbInstanceClass,
                                             const Aws::RDS::RDSClient &client) {
    std::vector<Aws::String> instanceClasses;
    Aws::String marker;
    do {
        Aws::RDS::Model::DescribeOrderableDBInstanceOptionsRequest request;
        request.SetEngine(engine);
        request.SetEngineVersion(engineVersion);
        if (!marker.empty()) {
            request.SetMarker(marker);
        }

        Aws::RDS::Model::DescribeOrderableDBInstanceOptionsOutcome outcome =
                client.DescribeOrderableDBInstanceOptions(request);

        if (outcome.IsSuccess()) {
            const Aws::Vector<Aws::RDS::Model::OrderableDBInstanceOption>
 &options =
                    outcome.GetResult().GetOrderableDBInstanceOptions();
            for (const Aws::RDS::Model::OrderableDBInstanceOption &option:
 options) {
                const Aws::String &instanceClass = option.GetDBInstanceClass();
                if (instanceClass.find("micro") != std::string::npos) {
                    if (std::find(instanceClasses.begin(), instanceClasses.end(),
                                  instanceClass) ==
                        instanceClasses.end()) {
                        instanceClasses.push_back(instanceClass);
```

```
                    }
                }
            }
            marker = outcome.GetResult().GetMarker();
        }
        else {
            std::cerr << "Error with RDS::DescribeOrderableDBInstanceOptions. "
                      << outcome.GetError().GetMessage()
                      << std::endl;
            return false;
        }
    } while (!marker.empty());

    std::cout << "The available micro DB instance classes for your database
 engine are:"
              << std::endl;
    for (int i = 0; i < instanceClasses.size(); ++i) {
        std::cout << "   " << i + 1 << ": " << instanceClasses[i] << std::endl;
    }

    int choice = askQuestionForIntRange(
            "Which micro DB instance class do you want to use? ",
            1, static_cast<int>(instanceClasses.size()));
    dbInstanceClass = instanceClasses[choice - 1];
    return true;
}
```

- Para obter detalhes da API, consulte DescribeOrderableDBInstanceOptions na Referência da API AWS SDK para C++.

CLI

AWS CLI

Como descrever as opções de instância solicitável do banco de dados

O exemplo de `describe-orderable-db-instance-options` a seguir recupera detalhes sobre as opções que podem ser solicitadas em uma instância de banco de dados que executa o mecanismo de banco de dados do MySQL.

```
aws rds describe-orderable-db-instance-options \
```

```
    --engine mysql
```

Saída:

```
{
    "OrderableDBInstanceOptions": [
        {
            "MinStorageSize": 5,
            "ReadReplicaCapable": true,
            "MaxStorageSize": 6144,
            "AvailabilityZones": [
                {
                    "Name": "us-east-1a"
                },
                {
                    "Name": "us-east-1b"
                },
                {
                    "Name": "us-east-1c"
                },
                {
                    "Name": "us-east-1d"
                }
            ],
            "SupportsIops": false,
            "AvailableProcessorFeatures": [],
            "MultiAZCapable": true,
            "DBInstanceClass": "db.m1.large",
            "Vpc": true,
            "StorageType": "gp2",
            "LicenseModel": "general-public-license",
            "EngineVersion": "5.5.46",
            "SupportsStorageEncryption": false,
            "SupportsEnhancedMonitoring": true,
            "Engine": "mysql",
            "SupportsIAMDatabaseAuthentication": false,
            "SupportsPerformanceInsights": false
        }
    ]
    ...some output truncated...
}
```

- Para obter detalhes da API, consulte DescribeOrderableDBInstanceOptions na Referência de comandos da AWS CLI.

Go

SDK para Go V2

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
type DbInstances struct {
 RdsClient *rds.Client
}



// GetOrderableInstances uses a paginator to get DB instance options that can be
 used to create DB instances that are
// compatible with a set of specifications.
func (instances *DbInstances) GetOrderableInstances(engine string, engineVersion
 string) (
 []types.OrderableDBInstanceOption, error) {

 var output *rds.DescribeOrderableDBInstanceOptionsOutput
 var instanceOptions []types.OrderableDBInstanceOption
 var err error
 orderablePaginator :=
 rds.NewDescribeOrderableDBInstanceOptionsPaginator(instances.RdsClient,
  &rds.DescribeOrderableDBInstanceOptionsInput{
   Engine:        aws.String(engine),
   EngineVersion: aws.String(engineVersion),
  })
 for orderablePaginator.HasMorePages() {
  output, err = orderablePaginator.NextPage(context.TODO())
  if err != nil {
   log.Printf("Couldn't get orderable DB instance options: %v\n", err)
   break
```

```
  } else {
   instanceOptions = append(instanceOptions,
 output.OrderableDBInstanceOptions...)
  }
 }
 return instanceOptions, err
}
```

- Para obter detalhes da API, consulte DescribeOrderableDBInstanceOptions na Referência da API AWS SDK para Go.

Java

SDK para Java 2.x

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
   // Get a list of allowed engine versions.
   public static void getAllowedEngines(RdsClient rdsClient, String
dbParameterGroupFamily) {
       try {
           DescribeDbEngineVersionsRequest versionsRequest =
DescribeDbEngineVersionsRequest.builder()
                   .dbParameterGroupFamily(dbParameterGroupFamily)
                   .engine("mysql")
                   .build();

           DescribeDbEngineVersionsResponse response =
rdsClient.describeDBEngineVersions(versionsRequest);
           List<DBEngineVersion> dbEngines = response.dbEngineVersions();
           for (DBEngineVersion dbEngine : dbEngines) {
               System.out.println("The engine version is " +
dbEngine.engineVersion());
               System.out.println("The engine description is " +
dbEngine.dbEngineDescription());
```

```
            }

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
    }
```

- Para obter detalhes da API, consulte DescribeOrderableDBInstanceOptions na Referência da API AWS SDK for Java 2.x.

Python

SDK para Python (Boto3).

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
class InstanceWrapper:
    """Encapsulates Amazon RDS DB instance actions."""

    def __init__(self, rds_client):
        """
        :param rds_client: A Boto3 Amazon RDS client.
        """
        self.rds_client = rds_client

    @classmethod
    def from_client(cls):
        """
        Instantiates this class from a Boto3 client.
        """
        rds_client = boto3.client("rds")
        return cls(rds_client)


    def get_orderable_instances(self, db_engine, db_engine_version):
```

```
        """
        Gets DB instance options that can be used to create DB instances that are
        compatible with a set of specifications.

        :param db_engine: The database engine that must be supported by the DB
instance.
        :param db_engine_version: The engine version that must be supported by
the DB instance.
        :return: The list of DB instance options that can be used to create a
compatible DB instance.
        """
        try:
            inst_opts = []
            paginator = self.rds_client.get_paginator(
                "describe_orderable_db_instance_options"
            )
            for page in paginator.paginate(
                Engine=db_engine, EngineVersion=db_engine_version
            ):
                inst_opts += page["OrderableDBInstanceOptions"]
        except ClientError as err:
            logger.error(
                "Couldn't get orderable DB instances. Here's why: %s: %s",
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return inst_opts
```

- Para ter detalhes da API, consulte DescribeOrderableDBInstanceOptions na Referência de API do AWS SDK para Python (Boto3).

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`GenerateRDSAuthToken`** com o AWS SDK ou a CLI

O código de exemplo a seguir mostra como usar `GenerateRDSAuthToken`.

Java

SDK para Java 2.x

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

Use a classe RdsUtilities para gerar um token de autenticação.

```
public class GenerateRDSAuthToken {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                    <dbInstanceIdentifier> <masterUsername>

                Where:
                    dbInstanceIdentifier - The database instance identifier.\s
                    masterUsername - The master user name.\s
                """;

        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }

        String dbInstanceIdentifier = args[0];
        String masterUsername = args[1];
        Region region = Region.US_WEST_2;
        RdsClient rdsClient = RdsClient.builder()
                .region(region)
                .build();

        String token = getAuthToken(rdsClient, dbInstanceIdentifier,
 masterUsername);
        System.out.println("The token response is " + token);
    }

    public static String getAuthToken(RdsClient rdsClient, String
 dbInstanceIdentifier, String masterUsername) {
```

```
        RdsUtilities utilities = rdsClient.utilities();
        try {
            GenerateAuthenticationTokenRequest tokenRequest =
 GenerateAuthenticationTokenRequest.builder()
                    .credentialsProvider(ProfileCredentialsProvider.create())
                    .username(masterUsername)
                    .port(3306)
                    .hostname(dbInstanceIdentifier)
                    .build();

            return utilities.generateAuthenticationToken(tokenRequest);

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
        return "";
    }
}
```

- Para ter detalhes da API, consulte GenerateRDSAuthToken na Referência de API do AWS SDK for Java 2.x.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`ModifyDBInstance`** com o AWS SDK ou a CLI

Os exemplos de código a seguir mostram como usar o `ModifyDBInstance`.

CLI

AWS CLI

Exemplo 1: Como modificar uma instância de banco de dados

O exemplo de `modify-db-instance` a seguir associa um grupo de opção e um grupo de parâmetros a uma instância de banco de dados do Microsoft SQL Server compatível. O

parâmetro `--apply-immediately` faz com que os grupos de opção e parâmetro sejam associados imediatamente em vez de ter que esperar a próxima janela de manutenção.

```
aws rds modify-db-instance \
    --db-instance-identifier database-2 \
    --option-group-name test-se-2017 \
    --db-parameter-group-name test-sqlserver-se-2017 \
    --apply-immediately
```

Saída:

```
{
    "DBInstance": {
        "DBInstanceIdentifier": "database-2",
        "DBInstanceClass": "db.r4.large",
        "Engine": "sqlserver-se",
        "DBInstanceStatus": "available",

        ...output omitted...

        "DBParameterGroups": [
            {
                "DBParameterGroupName": "test-sqlserver-se-2017",
                "ParameterApplyStatus": "applying"
            }
        ],
        "AvailabilityZone": "us-west-2d",

        ...output omitted...

        "MultiAZ": true,
        "EngineVersion": "14.00.3281.6.v1",
        "AutoMinorVersionUpgrade": false,
        "ReadReplicaDBInstanceIdentifiers": [],
        "LicenseModel": "license-included",
        "OptionGroupMemberships": [
            {
                "OptionGroupName": "test-se-2017",
                "Status": "pending-apply"
            }
        ],
        "CharacterSetName": "SQL_Latin1_General_CP1_CI_AS",
        "SecondaryAvailabilityZone": "us-west-2c",
```

```
        "PubliclyAccessible": true,
        "StorageType": "gp2",

        ...output omitted...

        "DeletionProtection": false,
        "AssociatedRoles": [],
        "MaxAllocatedStorage": 1000
    }
}
```

Para obter mais informações, consulte Modificar uma instância de banco de dados do Amazon RDS no Guia do usuário do Amazon RDS.

Exemplo 2: Como associar um grupo de segurança a uma instância de banco de dados

O seguinte exemplo de `modify-db-instance` associa um grupo de segurança da VPC específico e remove grupos de segurança de banco de dados da instância de banco de dados:

```
aws rds modify-db-instance \
    --db-instance-identifier dbName \
    --vpc-security-group-ids sg-ID
```

Saída:

```
{
"DBInstance": {
    "DBInstanceIdentifier": "dbName",
    "DBInstanceClass": "db.t3.micro",
    "Engine": "mysql",
    "DBInstanceStatus": "available",
    "MasterUsername": "admin",
    "Endpoint": {
        "Address": "dbName.abcdefghijk.us-west-2.rds.amazonaws.com",
        "Port": 3306,
        "HostedZoneId": "ABCDEFGHIJK1234"
    },
    "AllocatedStorage": 20,
    "InstanceCreateTime": "2024-02-15T00:37:58.793000+00:00",
    "PreferredBackupWindow": "11:57-12:27",
    "BackupRetentionPeriod": 7,
    "DBSecurityGroups": [],
```

```
    "VpcSecurityGroups": [
        {
            "VpcSecurityGroupId": "sg-ID",
            "Status": "active"
        }
    ],
    ... output omitted ...
    "MultiAZ": false,
    "EngineVersion": "8.0.35",
    "AutoMinorVersionUpgrade": true,
    "ReadReplicaDBInstanceIdentifiers": [],
    "LicenseModel": "general-public-license",

    ... output ommited ...
    }
}
```

Para ter mais informações, consulte Controlar acesso com grupos de segurança no Guia do usuário do Amazon RDS.

- Para obter detalhes da API, consulte ModifyDBInstance na Referência de comandos da AWS CLI.

Java

SDK para Java 2.x

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.rds.RdsClient;
import software.amazon.awssdk.services.rds.model.ModifyDbInstanceRequest;
import software.amazon.awssdk.services.rds.model.ModifyDbInstanceResponse;
import software.amazon.awssdk.services.rds.model.RdsException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
```

```
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class ModifyDBInstance {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                    <dbInstanceIdentifier> <dbSnapshotIdentifier>\s
                Where:
                    dbInstanceIdentifier - The database instance identifier.\s
                    masterUserPassword - The updated password that corresponds to
 the master user name.\s
                """;

        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }

        String dbInstanceIdentifier = args[0];
        String masterUserPassword = args[1];
        Region region = Region.US_WEST_2;
        RdsClient rdsClient = RdsClient.builder()
                .region(region)
                .build();

        updateIntance(rdsClient, dbInstanceIdentifier, masterUserPassword);
        rdsClient.close();
    }

    public static void updateIntance(RdsClient rdsClient, String
 dbInstanceIdentifier, String masterUserPassword) {
        try {
            // For a demo - modify the DB instance by modifying the master
 password.
            ModifyDbInstanceRequest modifyDbInstanceRequest =
 ModifyDbInstanceRequest.builder()
                    .dbInstanceIdentifier(dbInstanceIdentifier)
                    .publiclyAccessible(true)
                    .masterUserPassword(masterUserPassword)
```

```
                    .build();

            ModifyDbInstanceResponse instanceResponse =
 rdsClient.modifyDBInstance(modifyDbInstanceRequest);
            System.out.print("The ARN of the modified database is: " +
 instanceResponse.dbInstance().dbInstanceArn());

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
    }
}
```

- Para obter detalhes da API, consulte ModifyDBInstance na Referência da API AWS SDK for Java 2.x.

Kotlin

SDK para Kotlin

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
suspend fun updateIntance(
    dbInstanceIdentifierVal: String?,
    masterUserPasswordVal: String?,
) {
    val request =
        ModifyDbInstanceRequest {
            dbInstanceIdentifier = dbInstanceIdentifierVal
            publiclyAccessible = true
            masterUserPassword = masterUserPasswordVal
        }

    RdsClient { region = "us-west-2" }.use { rdsClient ->
        val instanceResponse = rdsClient.modifyDbInstance(request)
```

```
        println("The ARN of the modified database is
 ${instanceResponse.dbInstance?.dbInstanceArn}")
    }
}
```

- Para ter detalhes da API, consulte ModifyDBInstance na Referência de API do AWS SDK para Kotlin.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`ModifyDBParameterGroup`** com o AWS SDK ou a CLI

Os exemplos de código a seguir mostram como usar o `ModifyDBParameterGroup`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação no contexto no seguinte exemplo de código:

- Começar a usar instâncias de banco de dados

.NET

SDK para .NET

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
    /// <summary>
    /// Update a DB parameter group. Use the action
DescribeDBParameterGroupsAsync
    /// to determine when the DB parameter group is ready to use.
    /// </summary>
    /// <param name="name">Name of the DB parameter group.</param>
```

```
    /// <param name="parameters">List of parameters. Maximum of 20 per request.</
param>
    /// <returns>The updated DB parameter group name.</returns>
    public async Task<string> ModifyDBParameterGroup(
        string name, List<Parameter> parameters)
    {
        var response = await _amazonRDS.ModifyDBParameterGroupAsync(
            new ModifyDBParameterGroupRequest()
            {
                DBParameterGroupName = name,
                Parameters = parameters,
            });
        return response.DBParameterGroupName;
    }
```

- Para obter detalhes da API, consulte ModifyDBParameterGroup na Referência da API AWS SDK para .NET.

C++

SDK para C++

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
        Aws::Client::ClientConfiguration clientConfig;
        // Optional: Set to the AWS Region (overrides config file).
        // clientConfig.region = "us-east-1";

    Aws::RDS::RDSClient client(clientConfig);

        Aws::RDS::Model::ModifyDBParameterGroupRequest request;
        request.SetDBParameterGroupName(PARAMETER_GROUP_NAME);
        request.SetParameters(updateParameters);

        Aws::RDS::Model::ModifyDBParameterGroupOutcome outcome =
```

```
                client.ModifyDBParameterGroup(request);

        if (outcome.IsSuccess()) {
            std::cout << "The DB parameter group was successfully modified."
                      << std::endl;
        }
        else {
            std::cerr << "Error with RDS::ModifyDBParameterGroup. "
                      << outcome.GetError().GetMessage()
                      << std::endl;
        }
```

- Para obter detalhes da API, consulte ModifyDBParameterGroup na Referência da API AWS SDK para C++.

CLI

AWS CLI

Como modificar um grupo de parâmetros de banco de dados

O exemplo de `modify-db-parameter-group` a seguir altera o valor do parâmetro `clr enabled` em um grupo de parâmetros de banco de dados. O parâmetro `--apply-immediately` faz com que o grupo de parâmetros de banco de dados seja modificado imediatamente em vez de ter que esperar a próxima janela de manutenção.

```
aws rds modify-db-parameter-group \
    --db-parameter-group-name test-sqlserver-se-2017 \
    --parameters "ParameterName='clr
 enabled',ParameterValue=1,ApplyMethod=immediate"
```

Saída:

```
{
    "DBParameterGroupName": "test-sqlserver-se-2017"
}
```

Para obter mais informações, consulte Modificar parâmetros em um grupo de parâmetros de banco de dados no Guia do usuário do Amazon RDS.

- Para obter detalhes da API, consulte ModifyDBParameterGroup na Referência de comandos da AWS CLI.

Go

SDK para Go V2

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
type DbInstances struct {
 RdsClient *rds.Client
}



// UpdateParameters updates parameters in a named DB parameter group.
func (instances *DbInstances) UpdateParameters(parameterGroupName string, params
 []types.Parameter) error {
 _, err := instances.RdsClient.ModifyDBParameterGroup(context.TODO(),
  &rds.ModifyDBParameterGroupInput{
   DBParameterGroupName: aws.String(parameterGroupName),
   Parameters:           params,
  })
 if err != nil {
  log.Printf("Couldn't update parameters in %v: %v\n", parameterGroupName, err)
  return err
 } else {
  return nil
 }
}
```

- Para obter detalhes da API, consulte ModifyDBParameterGroup na Referência da API AWS SDK para Go.

Java

SDK para Java 2.x

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
   // Modify auto_increment_offset and auto_increment_increment parameters.
   public static void modifyDBParas(RdsClient rdsClient, String dbGroupName) {
       try {
           Parameter parameter1 = Parameter.builder()
                   .parameterName("auto_increment_offset")
                   .applyMethod("immediate")
                   .parameterValue("5")
                   .build();

           List<Parameter> paraList = new ArrayList<>();
           paraList.add(parameter1);
           ModifyDbParameterGroupRequest groupRequest =
ModifyDbParameterGroupRequest.builder()
                   .dbParameterGroupName(dbGroupName)
                   .parameters(paraList)
                   .build();

           ModifyDbParameterGroupResponse response =
rdsClient.modifyDBParameterGroup(groupRequest);
           System.out.println("The parameter group " +
response.dbParameterGroupName() + " was successfully modified");

       } catch (RdsException e) {
           System.out.println(e.getLocalizedMessage());
           System.exit(1);
       }
   }
```

- Para obter detalhes da API, consulte ModifyDBParameterGroup na Referência da API AWS SDK for Java 2.x.

Python

SDK para Python (Boto3).

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
class InstanceWrapper:
    """Encapsulates Amazon RDS DB instance actions."""

    def __init__(self, rds_client):
        """
        :param rds_client: A Boto3 Amazon RDS client.
        """
        self.rds_client = rds_client

    @classmethod
    def from_client(cls):
        """
        Instantiates this class from a Boto3 client.
        """
        rds_client = boto3.client("rds")
        return cls(rds_client)


    def update_parameters(self, parameter_group_name, update_parameters):
        """
        Updates parameters in a custom DB parameter group.

        :param parameter_group_name: The name of the parameter group to update.
        :param update_parameters: The parameters to update in the group.
        :return: Data about the modified parameter group.
        """
        try:
            response = self.rds_client.modify_db_parameter_group(
                DBParameterGroupName=parameter_group_name,
 Parameters=update_parameters
            )
        except ClientError as err:
```

```
            logger.error(
                "Couldn't update parameters in %s. Here's why: %s: %s",
                parameter_group_name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return response
```

- Para ter detalhes da API, consulte ModifyDBParameterGroup na Referência de API do AWS SDK para Python (Boto3).

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`RebootDBInstance`** com o AWS SDK ou a CLI

Os exemplos de código a seguir mostram como usar o `RebootDBInstance`.

CLI

AWS CLI

Como reinicializar uma instância de banco de dados

O exemplo de `reboot-db-instance` a seguir inicia uma reinicialização da instância de banco de dados especificada.

```
aws rds reboot-db-instance \
    --db-instance-identifier test-mysql-instance
```

Saída:

```
{
    "DBInstance": {
        "DBInstanceIdentifier": "test-mysql-instance",
        "DBInstanceClass": "db.t3.micro",
        "Engine": "mysql",
```

```
        "DBInstanceStatus": "rebooting",
        "MasterUsername": "admin",
        "Endpoint": {
            "Address": "test-mysql-instance.############.us-
west-2.rds.amazonaws.com",
            "Port": 3306,
            "HostedZoneId": "Z1PVIF0EXAMPLE"
        },

    ... output omitted...

    }
}
```

Para obter mais informações, consulte Reinicializar uma instância de banco de dados no Guia do usuário do Amazon RDS.

- Para obter detalhes da API, consulte RebootDBInstance na Referência de comandos da AWS CLI.

Java

SDK para Java 2.x

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.rds.RdsClient;
import software.amazon.awssdk.services.rds.model.RebootDbInstanceRequest;
import software.amazon.awssdk.services.rds.model.RebootDbInstanceResponse;
import software.amazon.awssdk.services.rds.model.RdsException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
```

```
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class RebootDBInstance {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                    <dbInstanceIdentifier>\s

                Where:
                    dbInstanceIdentifier - The database instance identifier\s
                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String dbInstanceIdentifier = args[0];
        Region region = Region.US_WEST_2;
        RdsClient rdsClient = RdsClient.builder()
                .region(region)
                .build();

        rebootInstance(rdsClient, dbInstanceIdentifier);
        rdsClient.close();
    }

    public static void rebootInstance(RdsClient rdsClient, String
 dbInstanceIdentifier) {
        try {
            RebootDbInstanceRequest rebootDbInstanceRequest =
 RebootDbInstanceRequest.builder()
                    .dbInstanceIdentifier(dbInstanceIdentifier)
                    .build();

            RebootDbInstanceResponse instanceResponse =
 rdsClient.rebootDBInstance(rebootDbInstanceRequest);
            System.out.print("The database " +
 instanceResponse.dbInstance().dbInstanceArn() + " was rebooted");

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
```

```
            System.exit(1);
        }
    }
}
```

- Para ter detalhes da API, consulte RebootDBInstance na Referência de API do AWS SDK for Java 2.x.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Cenários do Amazon RDS usando AWS SDKs

Os exemplos de código a seguir mostram como implementar cenários comuns no Amazon RDS com AWS SDKs. Esses cenários mostram como realizar tarefas específicas chamando várias funções no Amazon RDS. Cada exemplo inclui um link para o GitHub, em que é possível encontrar instruções sobre como configurar e executar o código.

Exemplos

- Começar a usar instâncias de banco de dados do Amazon RDS usando um AWS SDK

Começar a usar instâncias de banco de dados do Amazon RDS usando um AWS SDK

Os exemplos de código a seguir mostram como:

- Crie um grupo de parâmetros de banco de dados e defina os valores dos parâmetros.
- Criar uma instância de banco de dados configurada para usar o grupo de parâmetros. A instância de banco de dados também contém um banco de dados.
- Criar um snapshot da instância.
- Exclua a instância e o grupo de parâmetros.

.NET

SDK para .NET

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

Execute um cenário interativo em um prompt de comando.

```
/// <summary>
/// Scenario for RDS DB instance example.
/// </summary>
public class RDSInstanceScenario
{
    /*
    Before running this .NET code example, set up your development environment,
 including your credentials.

    This .NET example performs the following tasks:
    1.  Returns a list of the available DB engine families using the
 DescribeDBEngineVersionsAsync method.
    2.  Selects an engine family and creates a custom DB parameter group using
 the CreateDBParameterGroupAsync method.
    3.  Gets the parameter groups using the DescribeDBParameterGroupsAsync
 method.
    4.  Gets parameters in the group using the DescribeDBParameters method.
    5.  Parses and displays parameters in the group.
    6.  Modifies both the auto_increment_offset and auto_increment_increment
 parameters
        using the ModifyDBParameterGroupAsync method.
    7.  Gets and displays the updated parameters using the DescribeDBParameters
 method with a source of "user".
    8.  Gets a list of allowed engine versions using the
 DescribeDBEngineVersionsAsync method.
    9.  Displays and selects from a list of micro instance classes available for
 the selected engine and version.
    10. Creates an RDS DB instance that contains a MySql database and uses the
 parameter group
        using the CreateDBInstanceAsync method.
```

```
   11. Waits for DB instance to be ready using the DescribeDBInstancesAsync
method.
   12. Prints out the connection endpoint string for the new DB instance.
   13. Creates a snapshot of the DB instance using the CreateDBSnapshotAsync
method.
   14. Waits for DB snapshot to be ready using the DescribeDBSnapshots method.
   15. Deletes the DB instance using the DeleteDBInstanceAsync method.
   16. Waits for DB instance to be deleted using the DescribeDbInstances method.
   17. Deletes the parameter group using the DeleteDBParameterGroupAsync.
   */

   private static readonly string sepBar = new('-', 80);
   private static RDSWrapper rdsWrapper = null!;
   private static ILogger logger = null!;
   private static readonly string engine = "mysql";
   static async Task Main(string[] args)
   {
       // Set up dependency injection for the Amazon RDS service.
       using var host = Host.CreateDefaultBuilder(args)
           .ConfigureLogging(logging =>
               logging.AddFilter("System", LogLevel.Debug)
                   .AddFilter<DebugLoggerProvider>("Microsoft",
LogLevel.Information)
                   .AddFilter<ConsoleLoggerProvider>("Microsoft",
LogLevel.Trace))
           .ConfigureServices((_, services) =>
               services.AddAWSService<IAmazonRDS>()
                   .AddTransient<RDSWrapper>()
           )
           .Build();

       logger = LoggerFactory.Create(builder =>
       {
           builder.AddConsole();
       }).CreateLogger<RDSInstanceScenario>();

       rdsWrapper = host.Services.GetRequiredService<RDSWrapper>();

       Console.WriteLine(sepBar);
       Console.WriteLine(
           "Welcome to the Amazon Relational Database Service (Amazon RDS) DB
instance scenario example.");
       Console.WriteLine(sepBar);
```

```
        try
        {
            var parameterGroupFamily = await ChooseParameterGroupFamily();

            var parameterGroup = await
CreateDbParameterGroup(parameterGroupFamily);

            var parameters = await
DescribeParametersInGroup(parameterGroup.DBParameterGroupName,
                new List<string> { "auto_increment_offset",
"auto_increment_increment" });

            await ModifyParameters(parameterGroup.DBParameterGroupName,
parameters);

            await
DescribeUserSourceParameters(parameterGroup.DBParameterGroupName);

            var engineVersionChoice = await
ChooseDbEngineVersion(parameterGroupFamily);

            var instanceChoice = await ChooseDbInstanceClass(engine,
engineVersionChoice.EngineVersion);

            var newInstanceIdentifier = "Example-Instance-" + DateTime.Now.Ticks;

            var newInstance = await CreateRdsNewInstance(parameterGroup, engine,
engineVersionChoice.EngineVersion,
                instanceChoice.DBInstanceClass, newInstanceIdentifier);
            if (newInstance != null)
            {
                DisplayConnectionString(newInstance);

                await CreateSnapshot(newInstance);

                await DeleteRdsInstance(newInstance);
            }

            await DeleteParameterGroup(parameterGroup);

            Console.WriteLine("Scenario complete.");
            Console.WriteLine(sepBar);
        }
        catch (Exception ex)
```

```
        {
            logger.LogError(ex, "There was a problem executing the scenario.");
        }
    }

    /// <summary>
    /// Choose the RDS DB parameter group family from a list of available
options.
    /// </summary>
    /// <returns>The selected parameter group family.</returns>
    public static async Task<string> ChooseParameterGroupFamily()
    {
        Console.WriteLine(sepBar);
        // 1. Get a list of available engines.
        var engines = await rdsWrapper.DescribeDBEngineVersions(engine);

        Console.WriteLine("1. The following is a list of available DB parameter
group families:");
        int i = 1;
        var parameterGroupFamilies = engines.GroupBy(e =>
e.DBParameterGroupFamily).ToList();
        foreach (var parameterGroupFamily in parameterGroupFamilies)
        {
            // List the available parameter group families.
            Console.WriteLine(
                $"\t{i}. Family: {parameterGroupFamily.Key}");
            i++;
        }

        var choiceNumber = 0;
        while (choiceNumber < 1 || choiceNumber > parameterGroupFamilies.Count)
        {
            Console.WriteLine("Select an available DB parameter group family by
entering a number from the list above:");
            var choice = Console.ReadLine();
            Int32.TryParse(choice, out choiceNumber);
        }
        var parameterGroupFamilyChoice = parameterGroupFamilies[choiceNumber -
1];
        Console.WriteLine(sepBar);
        return parameterGroupFamilyChoice.Key;
    }

    /// <summary>
```

```
    /// Create and get information on a DB parameter group.
    /// </summary>
    /// <param name="dbParameterGroupFamily">The DBParameterGroupFamily for the
new DB parameter group.</param>
    /// <returns>The new DBParameterGroup.</returns>
    public static async Task<DBParameterGroup> CreateDbParameterGroup(string
dbParameterGroupFamily)
    {
        Console.WriteLine(sepBar);
        Console.WriteLine($"2. Create new DB parameter group with family
{dbParameterGroupFamily}:");

        var parameterGroup = await rdsWrapper.CreateDBParameterGroup(
            "ExampleParameterGroup-" + DateTime.Now.Ticks,
            dbParameterGroupFamily, "New example parameter group");

        var groupInfo =
            await rdsWrapper.DescribeDBParameterGroups(parameterGroup
                .DBParameterGroupName);

        Console.WriteLine(
            $"3. New DB parameter group: \n\t{groupInfo[0].Description}, \n\tARN
{groupInfo[0].DBParameterGroupArn}");
        Console.WriteLine(sepBar);
        return parameterGroup;
    }

    /// <summary>
    /// Get and describe parameters from a DBParameterGroup.
    /// </summary>
    /// <param name="parameterGroupName">Name of the DBParameterGroup.</param>
    /// <param name="parameterNames">Optional specific names of parameters to
describe.</param>
    /// <returns>The list of requested parameters.</returns>
    public static async Task<List<Parameter>> DescribeParametersInGroup(string
parameterGroupName, List<string>? parameterNames = null)
    {
        Console.WriteLine(sepBar);
        Console.WriteLine("4. Get some parameters from the group.");
        Console.WriteLine(sepBar);

        var parameters =
            await rdsWrapper.DescribeDBParameters(parameterGroupName);
```

```
        var matchingParameters =
            parameters.Where(p => parameterNames == null ||
parameterNames.Contains(p.ParameterName)).ToList();

        Console.WriteLine("5. Parameter information:");
        matchingParameters.ForEach(p =>
            Console.WriteLine(
                $"\n\tParameter: {p.ParameterName}." +
                $"\n\tDescription: {p.Description}." +
                $"\n\tAllowed Values: {p.AllowedValues}." +
                $"\n\tValue: {p.ParameterValue}."));

        Console.WriteLine(sepBar);

        return matchingParameters;
    }

    /// <summary>
    /// Modify a parameter from a DBParameterGroup.
    /// </summary>
    /// <param name="parameterGroupName">Name of the DBParameterGroup.</param>
    /// <param name="parameters">The parameters to modify.</param>
    /// <returns>Async task.</returns>
    public static async Task ModifyParameters(string parameterGroupName,
List<Parameter> parameters)
    {
        Console.WriteLine(sepBar);
        Console.WriteLine("6. Modify some parameters in the group.");

        foreach (var p in parameters)
        {
            if (p.IsModifiable && p.DataType == "integer")
            {
                int newValue = 0;
                while (newValue == 0)
                {
                    Console.WriteLine(
                        $"Enter a new value for {p.ParameterName} from the
allowed values {p.AllowedValues} ");

                    var choice = Console.ReadLine();
                    Int32.TryParse(choice, out newValue);
                }
```

```
                p.ParameterValue = newValue.ToString();
            }
        }

        await rdsWrapper.ModifyDBParameterGroup(parameterGroupName, parameters);

        Console.WriteLine(sepBar);
    }

    /// <summary>
    /// Describe the user source parameters in the group.
    /// </summary>
    /// <param name="parameterGroupName">Name of the DBParameterGroup.</param>
    /// <returns>Async task.</returns>
    public static async Task DescribeUserSourceParameters(string
parameterGroupName)
    {
        Console.WriteLine(sepBar);
        Console.WriteLine("7. Describe user source parameters in the group.");

        var parameters =
            await rdsWrapper.DescribeDBParameters(parameterGroupName, "user");


        parameters.ForEach(p =>
            Console.WriteLine(
                $"\n\tParameter: {p.ParameterName}." +
                $"\n\tDescription: {p.Description}." +
                $"\n\tAllowed Values: {p.AllowedValues}." +
                $"\n\tValue: {p.ParameterValue}."));

        Console.WriteLine(sepBar);
    }


    /// <summary>
    /// Choose a DB engine version.
    /// </summary>
    /// <param name="dbParameterGroupFamily">DB parameter group family for engine
choice.</param>
    /// <returns>The selected engine version.</returns>
    public static async Task<DBEngineVersion> ChooseDbEngineVersion(string
dbParameterGroupFamily)
    {
```

```
        Console.WriteLine(sepBar);
        // Get a list of allowed engines.
        var allowedEngines =
            await rdsWrapper.DescribeDBEngineVersions(engine,
 dbParameterGroupFamily);

        Console.WriteLine($"Available DB engine versions for parameter group
 family {dbParameterGroupFamily}:");
        int i = 1;
        foreach (var version in allowedEngines)
        {
            Console.WriteLine(
                $"\t{i}. Engine: {version.Engine} Version
 {version.EngineVersion}.");
            i++;
        }

        var choiceNumber = 0;
        while (choiceNumber < 1 || choiceNumber > allowedEngines.Count)
        {
            Console.WriteLine("8. Select an available DB engine version by
 entering a number from the list above:");
            var choice = Console.ReadLine();
            Int32.TryParse(choice, out choiceNumber);
        }

        var engineChoice = allowedEngines[choiceNumber - 1];
        Console.WriteLine(sepBar);
        return engineChoice;
    }

    /// <summary>
    /// Choose a DB instance class for a particular engine and engine version.
    /// </summary>
    /// <param name="engine">DB engine for DB instance choice.</param>
    /// <param name="engineVersion">DB engine version for DB instance choice.</
param>
    /// <returns>The selected orderable DB instance option.</returns>
    public static async Task<OrderableDBInstanceOption>
 ChooseDbInstanceClass(string engine, string engineVersion)
    {
        Console.WriteLine(sepBar);
        // Get a list of allowed DB instance classes.
        var allowedInstances =
```

```
            await rdsWrapper.DescribeOrderableDBInstanceOptions(engine,
 engineVersion);

        Console.WriteLine($"8. Available micro DB instance classes for engine
 {engine} and version {engineVersion}:");
        int i = 1;

        // Filter to micro instances for this example.
        allowedInstances = allowedInstances
            .Where(i => i.DBInstanceClass.Contains("micro")).ToList();

        foreach (var instance in allowedInstances)
        {
            Console.WriteLine(
                $"\t{i}. Instance class: {instance.DBInstanceClass} (storage type
 {instance.StorageType})");
            i++;
        }

        var choiceNumber = 0;
        while (choiceNumber < 1 || choiceNumber > allowedInstances.Count)
        {
            Console.WriteLine("9. Select an available DB instance class by
 entering a number from the list above:");
            var choice = Console.ReadLine();
            Int32.TryParse(choice, out choiceNumber);
        }

        var instanceChoice = allowedInstances[choiceNumber - 1];
        Console.WriteLine(sepBar);
        return instanceChoice;
    }

    /// <summary>
    /// Create a new RDS DB instance.
    /// </summary>
    /// <param name="parameterGroup">Parameter group to use for the DB
 instance.</param>
    /// <param name="engineName">Engine to use for the DB instance.</param>
    /// <param name="engineVersion">Engine version to use for the DB instance.</
param>
    /// <param name="instanceClass">Instance class to use for the DB instance.</
param>
```

```
   /// <param name="instanceIdentifier">Instance identifier to use for the DB
instance.</param>
   /// <returns>The new DB instance.</returns>
   public static async Task<DBInstance?> CreateRdsNewInstance(DBParameterGroup
parameterGroup,
       string engineName, string engineVersion, string instanceClass, string
instanceIdentifier)
   {
       Console.WriteLine(sepBar);
       Console.WriteLine($"10. Create a new DB instance with identifier
{instanceIdentifier}.");
       bool isInstanceReady = false;
       DBInstance newInstance;
       var instances = await rdsWrapper.DescribeDBInstances();
       isInstanceReady = instances.FirstOrDefault(i =>
           i.DBInstanceIdentifier == instanceIdentifier)?.DBInstanceStatus ==
"available";

       if (isInstanceReady)
       {
           Console.WriteLine("Instance already created.");
           newInstance = instances.First(i => i.DBInstanceIdentifier ==
instanceIdentifier);
       }
       else
       {
           Console.WriteLine("Please enter an admin user name:");
           var username = Console.ReadLine();

           Console.WriteLine("Please enter an admin password:");
           var password = Console.ReadLine();

           newInstance = await rdsWrapper.CreateDBInstance(
               "ExampleInstance",
               instanceIdentifier,
               parameterGroup.DBParameterGroupName,
               engineName,
               engineVersion,
               instanceClass,
               20,
               username,
               password
           );
```

```
            // 11. Wait for the DB instance to be ready.

            Console.WriteLine("11. Waiting for DB instance to be ready...");
            while (!isInstanceReady)
            {
                instances = await
 rdsWrapper.DescribeDBInstances(instanceIdentifier);
                isInstanceReady = instances.FirstOrDefault()?.DBInstanceStatus ==
 "available";
                newInstance = instances.First();
                Thread.Sleep(30000);
            }
        }

        Console.WriteLine(sepBar);
        return newInstance;
    }

    /// <summary>
    /// Display a connection string for an RDS DB instance.
    /// </summary>
    /// <param name="instance">The DB instance to use to get a connection
 string.</param>
    public static void DisplayConnectionString(DBInstance instance)
    {
        Console.WriteLine(sepBar);
        // Display the connection string.
        Console.WriteLine("12. New DB instance connection string: ");
        Console.WriteLine(
            $"\n{engine} -h {instance.Endpoint.Address} -P
 {instance.Endpoint.Port} "
            + $"-u {instance.MasterUsername} -p [YOUR PASSWORD]\n");

        Console.WriteLine(sepBar);
    }

    /// <summary>
    /// Create a snapshot from an RDS DB instance.
    /// </summary>
    /// <param name="instance">DB instance to use when creating a snapshot.</
param>
    /// <returns>The snapshot object.</returns>
    public static async Task<DBSnapshot> CreateSnapshot(DBInstance instance)
    {
```

```
        Console.WriteLine(sepBar);
        // Create a snapshot.
        Console.WriteLine($"13. Creating snapshot from DB instance
{instance.DBInstanceIdentifier}.");
        var snapshot = await
rdsWrapper.CreateDBSnapshot(instance.DBInstanceIdentifier, "ExampleSnapshot-" +
DateTime.Now.Ticks);

        // Wait for the snapshot to be available
        bool isSnapshotReady = false;

        Console.WriteLine($"14. Waiting for snapshot to be ready...");
        while (!isSnapshotReady)
        {
            var snapshots = await
rdsWrapper.DescribeDBSnapshots(instance.DBInstanceIdentifier);
            isSnapshotReady = snapshots.FirstOrDefault()?.Status == "available";
            snapshot = snapshots.First();
            Thread.Sleep(30000);
        }

        Console.WriteLine(
            $"Snapshot {snapshot.DBSnapshotIdentifier} status is
{snapshot.Status}.");
        Console.WriteLine(sepBar);
        return snapshot;
    }

    /// <summary>
    /// Delete an RDS DB instance.
    /// </summary>
    /// <param name="instance">The DB instance to delete.</param>
    /// <returns>Async task.</returns>
    public static async Task DeleteRdsInstance(DBInstance newInstance)
    {
        Console.WriteLine(sepBar);
        // Delete the DB instance.
        Console.WriteLine($"15. Delete the DB instance
{newInstance.DBInstanceIdentifier}.");
        await rdsWrapper.DeleteDBInstance(newInstance.DBInstanceIdentifier);

        // Wait for the DB instance to delete.
        Console.WriteLine($"16. Waiting for the DB instance to delete...");
        bool isInstanceDeleted = false;
```

```
        while (!isInstanceDeleted)
        {
            var instance = await rdsWrapper.DescribeDBInstances();
            isInstanceDeleted = instance.All(i => i.DBInstanceIdentifier !=
 newInstance.DBInstanceIdentifier);
            Thread.Sleep(30000);
        }

        Console.WriteLine("DB instance deleted.");
        Console.WriteLine(sepBar);
    }

    /// <summary>
    /// Delete a DB parameter group.
    /// </summary>
    /// <param name="parameterGroup">The parameter group to delete.</param>
    /// <returns>Async task.</returns>
    public static async Task DeleteParameterGroup(DBParameterGroup
 parameterGroup)
    {
        Console.WriteLine(sepBar);
        // Delete the parameter group.
        Console.WriteLine($"17. Delete the DB parameter group
 {parameterGroup.DBParameterGroupName}.");
        await
 rdsWrapper.DeleteDBParameterGroup(parameterGroup.DBParameterGroupName);

        Console.WriteLine(sepBar);
    }
```

Os métodos de wrapper usados pelo cenário para as ações de instância de banco de dados.

```
/// <summary>
/// Wrapper methods to use Amazon Relational Database Service (Amazon RDS) with
 DB instance operations.
/// </summary>
public partial class RDSWrapper
{
    private readonly IAmazonRDS _amazonRDS;
    public RDSWrapper(IAmazonRDS amazonRDS)
```

```
    {
        _amazonRDS = amazonRDS;
    }


    /// <summary>
    /// Get a list of DB engine versions for a particular DB engine.
    /// </summary>
    /// <param name="engine">Name of the engine.</param>
    /// <param name="dbParameterGroupFamily">Optional parameter group family
name.</param>
    /// <returns>List of DBEngineVersions.</returns>
    public async Task<List<DBEngineVersion>> DescribeDBEngineVersions(string
engine,
        string dbParameterGroupFamily = null)
    {
        var response = await _amazonRDS.DescribeDBEngineVersionsAsync(
            new DescribeDBEngineVersionsRequest()
            {
                Engine = engine,
                DBParameterGroupFamily = dbParameterGroupFamily
            });
        return response.DBEngineVersions;
    }



    /// <summary>
    /// Get a list of orderable DB instance options for a specific
    /// engine and engine version.
    /// </summary>
    /// <param name="engine">Name of the engine.</param>
    /// <param name="engineVersion">Version of the engine.</param>
    /// <returns>List of OrderableDBInstanceOptions.</returns>
    public async Task<List<OrderableDBInstanceOption>>
DescribeOrderableDBInstanceOptions(string engine, string engineVersion)
    {
        // Use a paginator to get a list of DB instance options.
        var results = new List<OrderableDBInstanceOption>();
        var paginateInstanceOptions =
_amazonRDS.Paginators.DescribeOrderableDBInstanceOptions(
            new DescribeOrderableDBInstanceOptionsRequest()
            {
                Engine = engine,
```

```
                EngineVersion = engineVersion,
            });
        // Get the entire list using the paginator.
        await foreach (var instanceOptions in
paginateInstanceOptions.OrderableDBInstanceOptions)
        {
            results.Add(instanceOptions);
        }
        return results;
    }



    /// <summary>
    /// Returns a list of DB instances.
    /// </summary>
    /// <param name="dbInstanceIdentifier">Optional name of a specific DB
instance.</param>
    /// <returns>List of DB instances.</returns>
    public async Task<List<DBInstance>> DescribeDBInstances(string
dbInstanceIdentifier = null)
    {
        var results = new List<DBInstance>();
        var instancesPaginator = _amazonRDS.Paginators.DescribeDBInstances(
            new DescribeDBInstancesRequest
            {
                DBInstanceIdentifier = dbInstanceIdentifier
            });
        // Get the entire list using the paginator.
        await foreach (var instances in instancesPaginator.DBInstances)
        {
            results.Add(instances);
        }
        return results;
    }



    /// <summary>
    /// Create an RDS DB instance with a particular set of properties. Use the
action DescribeDBInstancesAsync
    /// to determine when the DB instance is ready to use.
    /// </summary>
    /// <param name="dbName">Name for the DB instance.</param>
```

```
   /// <param name="dbInstanceIdentifier">DB instance identifier.</param>
   /// <param name="parameterGroupName">DB parameter group to associate with the
instance.</param>
   /// <param name="dbEngine">The engine for the DB instance.</param>
   /// <param name="dbEngineVersion">Version for the DB instance.</param>
   /// <param name="instanceClass">Class for the DB instance.</param>
   /// <param name="allocatedStorage">The amount of storage in gibibytes (GiB)
to allocate to the DB instance.</param>
   /// <param name="adminName">Admin user name.</param>
   /// <param name="adminPassword">Admin user password.</param>
   /// <returns>DB instance object.</returns>
   public async Task<DBInstance> CreateDBInstance(string dbName, string
dbInstanceIdentifier,
       string parameterGroupName, string dbEngine, string dbEngineVersion,
       string instanceClass, int allocatedStorage, string adminName, string
adminPassword)
   {
       var response = await _amazonRDS.CreateDBInstanceAsync(
           new CreateDBInstanceRequest()
           {
               DBName = dbName,
               DBInstanceIdentifier = dbInstanceIdentifier,
               DBParameterGroupName = parameterGroupName,
               Engine = dbEngine,
               EngineVersion = dbEngineVersion,
               DBInstanceClass = instanceClass,
               AllocatedStorage = allocatedStorage,
               MasterUsername = adminName,
               MasterUserPassword = adminPassword
           });

       return response.DBInstance;
   }



   /// <summary>
   /// Delete a particular DB instance.
   /// </summary>
   /// <param name="dbInstanceIdentifier">DB instance identifier.</param>
   /// <returns>DB instance object.</returns>
   public async Task<DBInstance> DeleteDBInstance(string dbInstanceIdentifier)
   {
       var response = await _amazonRDS.DeleteDBInstanceAsync(
```

```
            new DeleteDBInstanceRequest()
            {
                DBInstanceIdentifier = dbInstanceIdentifier,
                SkipFinalSnapshot = true,
                DeleteAutomatedBackups = true
            });

        return response.DBInstance;
    }
```

Os métodos de wrapper usados pelo cenário para o grupos de parâmetros de banco de dados.

```
/// <summary>
/// Wrapper methods to use Amazon Relational Database Service (Amazon RDS) with
 parameter groups.
/// </summary>
public partial class RDSWrapper
{

    /// <summary>
    /// Get descriptions of DB parameter groups.
    /// </summary>
    /// <param name="name">Optional name of the DB parameter group to describe.</
param>
    /// <returns>The list of DB parameter group descriptions.</returns>
    public async Task<List<DBParameterGroup>> DescribeDBParameterGroups(string
 name = null)
    {
        var response = await _amazonRDS.DescribeDBParameterGroupsAsync(
            new DescribeDBParameterGroupsRequest()
            {
                DBParameterGroupName = name
            });
        return response.DBParameterGroups;
    }



    /// <summary>
```

```
    /// Create a new DB parameter group. Use the action
 DescribeDBParameterGroupsAsync
    /// to determine when the DB parameter group is ready to use.
    /// </summary>
    /// <param name="name">Name of the DB parameter group.</param>
    /// <param name="family">Family of the DB parameter group.</param>
    /// <param name="description">Description of the DB parameter group.</param>
    /// <returns>The new DB parameter group.</returns>
    public async Task<DBParameterGroup> CreateDBParameterGroup(
        string name, string family, string description)
    {
        var response = await _amazonRDS.CreateDBParameterGroupAsync(
            new CreateDBParameterGroupRequest()
            {
                DBParameterGroupName = name,
                DBParameterGroupFamily = family,
                Description = description
            });
        return response.DBParameterGroup;
    }



    /// <summary>
    /// Update a DB parameter group. Use the action
 DescribeDBParameterGroupsAsync
    /// to determine when the DB parameter group is ready to use.
    /// </summary>
    /// <param name="name">Name of the DB parameter group.</param>
    /// <param name="parameters">List of parameters. Maximum of 20 per request.</
param>
    /// <returns>The updated DB parameter group name.</returns>
    public async Task<string> ModifyDBParameterGroup(
        string name, List<Parameter> parameters)
    {
        var response = await _amazonRDS.ModifyDBParameterGroupAsync(
            new ModifyDBParameterGroupRequest()
            {
                DBParameterGroupName = name,
                Parameters = parameters,
            });
        return response.DBParameterGroupName;
    }
```

```
    /// <summary>
    /// Delete a DB parameter group. The group cannot be a default DB parameter
group
    /// or be associated with any DB instances.
    /// </summary>
    /// <param name="name">Name of the DB parameter group.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> DeleteDBParameterGroup(string name)
    {
        var response = await _amazonRDS.DeleteDBParameterGroupAsync(
            new DeleteDBParameterGroupRequest()
            {
                DBParameterGroupName = name,
            });
        return response.HttpStatusCode == HttpStatusCode.OK;
    }



    /// <summary>
    /// Get a list of DB parameters from a specific parameter group.
    /// </summary>
    /// <param name="dbParameterGroupName">Name of a specific DB parameter
group.</param>
    /// <param name="source">Optional source for selecting parameters.</param>
    /// <returns>List of parameter values.</returns>
    public async Task<List<Parameter>> DescribeDBParameters(string
dbParameterGroupName, string source = null)
    {
        var results = new List<Parameter>();
        var paginateParameters = _amazonRDS.Paginators.DescribeDBParameters(
            new DescribeDBParametersRequest()
            {
                DBParameterGroupName = dbParameterGroupName,
                Source = source
            });
        // Get the entire list using the paginator.
        await foreach (var parameters in paginateParameters.Parameters)
        {
            results.Add(parameters);
        }
        return results;
```

```
    }
```

Os métodos de wrapper usados pelo cenário para as ações de snapshot de banco de dados.

```
/// <summary>
/// Wrapper methods to use Amazon Relational Database Service (Amazon RDS) with
 snapshots.
/// </summary>
public partial class RDSWrapper
{

    /// <summary>
    /// Create a snapshot of a DB instance.
    /// </summary>
    /// <param name="dbInstanceIdentifier">DB instance identifier.</param>
    /// <param name="snapshotIdentifier">Identifier for the snapshot.</param>
    /// <returns>DB snapshot object.</returns>
    public async Task<DBSnapshot> CreateDBSnapshot(string dbInstanceIdentifier,
 string snapshotIdentifier)
    {
        var response = await _amazonRDS.CreateDBSnapshotAsync(
            new CreateDBSnapshotRequest()
            {
                DBSnapshotIdentifier = snapshotIdentifier,
                DBInstanceIdentifier = dbInstanceIdentifier
            });

        return response.DBSnapshot;
    }



    /// <summary>
    /// Return a list of DB snapshots for a particular DB instance.
    /// </summary>
    /// <param name="dbInstanceIdentifier">DB instance identifier.</param>
    /// <returns>List of DB snapshots.</returns>
    public async Task<List<DBSnapshot>> DescribeDBSnapshots(string
 dbInstanceIdentifier)
    {
```

```
        var results = new List<DBSnapshot>();
        var snapshotsPaginator = _amazonRDS.Paginators.DescribeDBSnapshots(
            new DescribeDBSnapshotsRequest()
            {
                DBInstanceIdentifier = dbInstanceIdentifier
            });

        // Get the entire list using the paginator.
        await foreach (var snapshots in snapshotsPaginator.DBSnapshots)
        {
            results.Add(snapshots);
        }
        return results;
    }
```

- Para obter detalhes da API, consulte os tópicos a seguir na Referência da API AWS SDK para .NET.
  - CreateDBInstance
  - CreateDBParameterGroup
  - CreateDBSnapshot
  - DeleteDBInstance
  - DeleteDBParameterGroup
  - DescribeDBEngineVersions
  - DescribeDBInstances
  - DescribeDBParameterGroups
  - DescribeDBParameters
  - DescribeDBSnapshots
  - DescribeOrderableDBInstanceOptions
  - ModifyDBParameterGroup

C++

SDK para C++

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
        Aws::Client::ClientConfiguration clientConfig;
        // Optional: Set to the AWS Region (overrides config file).
        // clientConfig.region = "us-east-1";

//! Routine which creates an Amazon RDS instance and demonstrates several
 operations
//! on that instance.
/*!
 \sa gettingStartedWithDBInstances()
 \param clientConfiguration: AWS client configuration.
 \return bool: Successful completion.
 */
bool AwsDoc::RDS::gettingStartedWithDBInstances(
        const Aws::Client::ClientConfiguration &clientConfig) {
    Aws::RDS::RDSClient client(clientConfig);

    printAsterisksLine();
    std::cout << "Welcome to the Amazon Relational Database Service (Amazon RDS)"
              << std::endl;
    std::cout << "get started with DB instances demo." << std::endl;
    printAsterisksLine();

    std::cout << "Checking for an existing DB parameter group named '" <<
              PARAMETER_GROUP_NAME << "'." << std::endl;
    Aws::String dbParameterGroupFamily("Undefined");
    bool parameterGroupFound = true;
    {
        // 1. Check if the DB parameter group already exists.
        Aws::RDS::Model::DescribeDBParameterGroupsRequest request;
        request.SetDBParameterGroupName(PARAMETER_GROUP_NAME);

        Aws::RDS::Model::DescribeDBParameterGroupsOutcome outcome =
```

```
                client.DescribeDBParameterGroups(request);

        if (outcome.IsSuccess()) {
            std::cout << "DB parameter group named '" <<
                      PARAMETER_GROUP_NAME << "' already exists." << std::endl;
            dbParameterGroupFamily = outcome.GetResult().GetDBParameterGroups()
[0].GetDBParameterGroupFamily();
        }
        else if (outcome.GetError().GetErrorType() ==
                 Aws::RDS::RDSErrors::D_B_PARAMETER_GROUP_NOT_FOUND_FAULT) {
            std::cout << "DB parameter group named '" <<
                      PARAMETER_GROUP_NAME << "' does not exist." << std::endl;
            parameterGroupFound = false;
        }
        else {
            std::cerr << "Error with RDS::DescribeDBParameterGroups. "
                      << outcome.GetError().GetMessage()
                      << std::endl;
            return false;
        }
    }

    if (!parameterGroupFound) {
        Aws::Vector<Aws::RDS::Model::DBEngineVersion> engineVersions;

        // 2. Get available engine versions for the specified engine.
        if (!getDBEngineVersions(DB_ENGINE, NO_PARAMETER_GROUP_FAMILY,
                                 engineVersions, client)) {
            return false;
        }

        std::cout << "Getting available database engine versions for " <<
 DB_ENGINE
                  << "."
                  << std::endl;
        std::vector<Aws::String> families;
        for (const Aws::RDS::Model::DBEngineVersion &version: engineVersions) {
            Aws::String family = version.GetDBParameterGroupFamily();
            if (std::find(families.begin(), families.end(), family) ==
                families.end()) {
                families.push_back(family);
                std::cout << "  " << families.size() << ": " << family <<
 std::endl;
            }
```

```
        }

        int choice = askQuestionForIntRange("Which family do you want to use? ",
1,
                                            static_cast<int>(families.size()));
        dbParameterGroupFamily = families[choice - 1];
    }
    if (!parameterGroupFound) {
        // 3.  Create a DB parameter group.
        Aws::RDS::Model::CreateDBParameterGroupRequest request;
        request.SetDBParameterGroupName(PARAMETER_GROUP_NAME);
        request.SetDBParameterGroupFamily(dbParameterGroupFamily);
        request.SetDescription("Example parameter group.");

        Aws::RDS::Model::CreateDBParameterGroupOutcome outcome =
                client.CreateDBParameterGroup(request);

        if (outcome.IsSuccess()) {
            std::cout << "The DB parameter group was successfully created."
                      << std::endl;
        }
        else {
            std::cerr << "Error with RDS::CreateDBParameterGroup. "
                      << outcome.GetError().GetMessage()
                      << std::endl;
            return false;
        }
    }

    printAsterisksLine();
    std::cout << "Let's set some parameter values in your parameter group."
              << std::endl;

    Aws::String marker;
    Aws::Vector<Aws::RDS::Model::Parameter> autoIncrementParameters;
    // 4.  Get the parameters in the DB parameter group.
    if (!getDBParameters(PARAMETER_GROUP_NAME, AUTO_INCREMENT_PREFIX, NO_SOURCE,
                         autoIncrementParameters,
                         client)) {
        cleanUpResources(PARAMETER_GROUP_NAME, "", client);
        return false;
    }

    Aws::Vector<Aws::RDS::Model::Parameter> updateParameters;
```

```
    for (Aws::RDS::Model::Parameter &autoIncParameter: autoIncrementParameters) {
        if (autoIncParameter.GetIsModifiable() &&
            (autoIncParameter.GetDataType() == "integer")) {
            std::cout << "The " << autoIncParameter.GetParameterName()
                      << " is described as: " <<
                      autoIncParameter.GetDescription() << "." << std::endl;
            if (autoIncParameter.ParameterValueHasBeenSet()) {
                std::cout << "The current value is "
                          << autoIncParameter.GetParameterValue()
                          << "." << std::endl;
            }
            std::vector<int> splitValues = splitToInts(
                    autoIncParameter.GetAllowedValues(), '-');
            if (splitValues.size() == 2) {
                int newValue = askQuestionForIntRange(
                        Aws::String("Enter a new value in the range ") +
                        autoIncParameter.GetAllowedValues() + ": ",
                        splitValues[0], splitValues[1]);
                autoIncParameter.SetParameterValue(std::to_string(newValue));
                updateParameters.push_back(autoIncParameter);

            }
            else {
                std::cerr << "Error parsing " <<
autoIncParameter.GetAllowedValues()
                          << std::endl;
            }
        }
    }

    {
        // 5.  Modify the auto increment parameters in the group.
        Aws::RDS::Model::ModifyDBParameterGroupRequest request;
        request.SetDBParameterGroupName(PARAMETER_GROUP_NAME);
        request.SetParameters(updateParameters);

        Aws::RDS::Model::ModifyDBParameterGroupOutcome outcome =
                client.ModifyDBParameterGroup(request);

        if (outcome.IsSuccess()) {
            std::cout << "The DB parameter group was successfully modified."
                      << std::endl;
        }
```

```
        else {
            std::cerr << "Error with RDS::ModifyDBParameterGroup. "
                      << outcome.GetError().GetMessage()
                      << std::endl;
        }
    }

    std::cout
            << "You can get a list of parameters you've set by specifying a
source of 'user'."
            << std::endl;

    Aws::Vector<Aws::RDS::Model::Parameter> userParameters;
    // 6.  Display the modified parameters in the group.
    if (!getDBParameters(PARAMETER_GROUP_NAME, NO_NAME_PREFIX, "user",
userParameters,
                         client)) {
        cleanUpResources(PARAMETER_GROUP_NAME, "", client);
        return false;
    }

    for (const auto &userParameter: userParameters) {
        std::cout << "  " << userParameter.GetParameterName() << ", " <<
                  userParameter.GetDescription() << ", parameter value - "
                  << userParameter.GetParameterValue() << std::endl;
    }

    printAsterisksLine();
    std::cout << "Checking for an existing DB instance." << std::endl;

    Aws::RDS::Model::DBInstance dbInstance;
    // 7.  Check if the DB instance already exists.
    if (!describeDBInstance(DB_INSTANCE_IDENTIFIER, dbInstance, client)) {
        cleanUpResources(PARAMETER_GROUP_NAME, "", client);
        return false;
    }

    if (dbInstance.DbInstancePortHasBeenSet()) {
        std::cout << "The DB instance already exists." << std::endl;
    }
    else {
        std::cout << "Let's create a DB instance." << std::endl;
        const Aws::String administratorName = askQuestion(
                "Enter an administrator username for the database: ");
```

```
        const Aws::String administratorPassword = askQuestion(
                "Enter a password for the administrator (at least 8 characters):
");
        Aws::Vector<Aws::RDS::Model::DBEngineVersion> engineVersions;

        // 8.  Get a list of available engine versions.
        if (!getDBEngineVersions(DB_ENGINE, dbParameterGroupFamily,
engineVersions,
                                 client)) {
            cleanUpResources(PARAMETER_GROUP_NAME, "", client);
            return false;
        }

        std::cout << "The available engines for your parameter group are:" <<
std::endl;

        int index = 1;
        for (const Aws::RDS::Model::DBEngineVersion &engineVersion:
engineVersions) {
            std::cout << "  " << index << ": " <<
engineVersion.GetEngineVersion()
                      << std::endl;
            ++index;
        }
        int choice = askQuestionForIntRange("Which engine do you want to use? ",
1,

static_cast<int>(engineVersions.size()));
        const Aws::RDS::Model::DBEngineVersion engineVersion =
engineVersions[choice -
                                                                              1];

        Aws::String dbInstanceClass;
        // 9.  Get a list of micro instance classes.
        if (!chooseMicroDBInstanceClass(engineVersion.GetEngine(),
                                        engineVersion.GetEngineVersion(),
                                        dbInstanceClass,
                                        client)) {
            cleanUpResources(PARAMETER_GROUP_NAME, "", client);
            return false;
        }

        std::cout << "Creating a DB instance named '" << DB_INSTANCE_IDENTIFIER
                  << "' and database '" << DB_NAME << "'.\n"
```

```
                << "The DB instance is configured to use your custom parameter
group '"
                << PARAMETER_GROUP_NAME << "',\n"
                << "selected engine version " <<
engineVersion.GetEngineVersion()
                << ",\n"
                << "selected DB instance class '" << dbInstanceClass << "',"
                << " and " << DB_ALLOCATED_STORAGE << " GiB of " <<
DB_STORAGE_TYPE
                << " storage.\nThis typically takes several minutes." <<
std::endl;

       Aws::RDS::Model::CreateDBInstanceRequest request;
       request.SetDBName(DB_NAME);
       request.SetDBInstanceIdentifier(DB_INSTANCE_IDENTIFIER);
       request.SetDBParameterGroupName(PARAMETER_GROUP_NAME);
       request.SetEngine(engineVersion.GetEngine());
       request.SetEngineVersion(engineVersion.GetEngineVersion());
       request.SetDBInstanceClass(dbInstanceClass);
       request.SetStorageType(DB_STORAGE_TYPE);
       request.SetAllocatedStorage(DB_ALLOCATED_STORAGE);
       request.SetMasterUsername(administratorName);
       request.SetMasterUserPassword(administratorPassword);

       Aws::RDS::Model::CreateDBInstanceOutcome outcome =
               client.CreateDBInstance(request);

       if (outcome.IsSuccess()) {
           std::cout << "The DB instance creation has started."
                     << std::endl;
       }
       else {
           std::cerr << "Error with RDS::CreateDBInstance. "
                     << outcome.GetError().GetMessage()
                     << std::endl;
           cleanUpResources(PARAMETER_GROUP_NAME, "", client);
           return false;
       }
   }

   std::cout << "Waiting for the DB instance to become available." << std::endl;

   int counter = 0;
   // 11. Wait for the DB instance to become available.
```

```
   do {
       std::this_thread::sleep_for(std::chrono::seconds(1));
       ++counter;
       if (counter > 900) {
           std::cerr << "Wait for instance to become available timed out ofter "
                     << counter
                     << " seconds." << std::endl;
           cleanUpResources(PARAMETER_GROUP_NAME, DB_INSTANCE_IDENTIFIER,
client);
           return false;
       }

       dbInstance = Aws::RDS::Model::DBInstance();
       if (!describeDBInstance(DB_INSTANCE_IDENTIFIER, dbInstance, client)) {
           cleanUpResources(PARAMETER_GROUP_NAME, DB_INSTANCE_IDENTIFIER,
client);
           return false;
       }

       if ((counter % 20) == 0) {
           std::cout << "Current DB instance status is '"
                     << dbInstance.GetDBInstanceStatus()
                     << "' after " << counter << " seconds." << std::endl;
       }
   } while (dbInstance.GetDBInstanceStatus() != "available");

   if (dbInstance.GetDBInstanceStatus() == "available") {
       std::cout << "The DB instance has been created." << std::endl;
   }

   printAsterisksLine();

   // 12. Display the connection string that can be used to connect a 'mysql'
shell to the database.
   displayConnection(dbInstance);

   printAsterisksLine();

   if (askYesNoQuestion(
           "Do you want to create a snapshot of your DB instance (y/n)? ")) {
       Aws::String snapshotID(DB_INSTANCE_IDENTIFIER + "-" +
                              Aws::String(Aws::Utils::UUID::RandomUUID()));
       {
```

```
            std::cout << "Creating a snapshot named " << snapshotID << "." <<
std::endl;
            std::cout << "This typically takes a few minutes." << std::endl;

            // 13. Create a snapshot of the DB instance.
            Aws::RDS::Model::CreateDBSnapshotRequest request;
            request.SetDBInstanceIdentifier(DB_INSTANCE_IDENTIFIER);
            request.SetDBSnapshotIdentifier(snapshotID);

            Aws::RDS::Model::CreateDBSnapshotOutcome outcome =
                    client.CreateDBSnapshot(request);

            if (outcome.IsSuccess()) {
                std::cout << "Snapshot creation has started."
                          << std::endl;
            }
            else {
                std::cerr << "Error with RDS::CreateDBSnapshot. "
                          << outcome.GetError().GetMessage()
                          << std::endl;
                cleanUpResources(PARAMETER_GROUP_NAME, DB_INSTANCE_IDENTIFIER,
client);
                return false;
            }
        }

        std::cout << "Waiting for snapshot to become available." << std::endl;

        Aws::RDS::Model::DBSnapshot snapshot;
        counter = 0;
        do {
            std::this_thread::sleep_for(std::chrono::seconds(1));
            ++counter;
            if (counter > 600) {
                std::cerr << "Wait for snapshot to be available timed out ofter "
                          << counter
                          << " seconds." << std::endl;
                cleanUpResources(PARAMETER_GROUP_NAME, DB_INSTANCE_IDENTIFIER,
client);
                return false;
            }

            // 14. Wait for the snapshot to become available.
            Aws::RDS::Model::DescribeDBSnapshotsRequest request;
```

```
            request.SetDBSnapshotIdentifier(snapshotID);

            Aws::RDS::Model::DescribeDBSnapshotsOutcome outcome =
                    client.DescribeDBSnapshots(request);

            if (outcome.IsSuccess()) {
                snapshot = outcome.GetResult().GetDBSnapshots()[0];
            }
            else {
                std::cerr << "Error with RDS::DescribeDBSnapshots. "
                          << outcome.GetError().GetMessage()
                          << std::endl;
                cleanUpResources(PARAMETER_GROUP_NAME, DB_INSTANCE_IDENTIFIER,
 client);
                return false;
            }

            if ((counter % 20) == 0) {
                std::cout << "Current snapshot status is '"
                          << snapshot.GetStatus()
                          << "' after " << counter << " seconds." << std::endl;
            }
        } while (snapshot.GetStatus() != "available");

        if (snapshot.GetStatus() != "available") {
            std::cout << "A snapshot has been created." << std::endl;
        }
    }

    printAsterisksLine();

    bool result = true;
    if (askYesNoQuestion(
            "Do you want to delete the DB instance and parameter group (y/n)? "))
 {
        result = cleanUpResources(PARAMETER_GROUP_NAME, DB_INSTANCE_IDENTIFIER,
 client);
    }

    return result;
}


//! Routine which gets DB parameters using the 'DescribeDBParameters' api.
```

```
/*!
 \sa getDBParameters()
 \param parameterGroupName: The name of the parameter group.
 \param namePrefix: Prefix string to filter results by parameter name.
 \param source: A source such as 'user', ignored if empty.
 \param parametersResult: Vector of 'Parameter' objects returned by the routine.
 \param client: 'RDSClient' instance.
 \return bool: Successful completion.
 */
bool AwsDoc::RDS::getDBParameters(const Aws::String &parameterGroupName,
                                  const Aws::String &namePrefix,
                                  const Aws::String &source,
                                  Aws::Vector<Aws::RDS::Model::Parameter>
 &parametersResult,
                                  const Aws::RDS::RDSClient &client) {
    Aws::String marker;
    do {
        Aws::RDS::Model::DescribeDBParametersRequest request;
        request.SetDBParameterGroupName(PARAMETER_GROUP_NAME);
        if (!marker.empty()) {
            request.SetMarker(marker);
        }
        if (!source.empty()) {
            request.SetSource(source);
        }

        Aws::RDS::Model::DescribeDBParametersOutcome outcome =
                client.DescribeDBParameters(request);

        if (outcome.IsSuccess()) {
            const Aws::Vector<Aws::RDS::Model::Parameter> &parameters =
                    outcome.GetResult().GetParameters();
            for (const Aws::RDS::Model::Parameter &parameter: parameters) {
                if (!namePrefix.empty()) {
                    if (parameter.GetParameterName().find(namePrefix) == 0) {
                        parametersResult.push_back(parameter);
                    }
                }
                else {
                    parametersResult.push_back(parameter);
                }
            }

            marker = outcome.GetResult().GetMarker();
```

```
        }
        else {
            std::cerr << "Error with RDS::DescribeDBParameters. "
                      << outcome.GetError().GetMessage()
                      << std::endl;
            return false;
        }
    } while (!marker.empty());

    return true;
}


//! Routine which gets available DB engine versions for an engine name and
//! an optional parameter group family.
/*!
 \sa getDBEngineVersions()
 \param engineName: A DB engine name.
 \param parameterGroupFamily: A parameter group family name, ignored if empty.
 \param engineVersionsResult: Vector of 'DBEngineVersion' objects returned by the
 routine.
 \param client: 'RDSClient' instance.
 \return bool: Successful completion.
 */
bool AwsDoc::RDS::getDBEngineVersions(const Aws::String &engineName,
                                      const Aws::String &parameterGroupFamily,

 Aws::Vector<Aws::RDS::Model::DBEngineVersion> &engineVersionsResult,
                                      const Aws::RDS::RDSClient &client) {
    Aws::RDS::Model::DescribeDBEngineVersionsRequest request;
    request.SetEngine(engineName);
    if (!parameterGroupFamily.empty()) {
        request.SetDBParameterGroupFamily(parameterGroupFamily);
    }

    engineVersionsResult.clear();
    Aws::String marker; // Used for pagination.

    do {
        if (!marker.empty()) {
            request.SetMarker(marker);
        }
```

```
        Aws::RDS::Model::DescribeDBEngineVersionsOutcome outcome =
                client.DescribeDBEngineVersions(request);

        if (outcome.IsSuccess()) {
            auto &engineVersions = outcome.GetResult().GetDBEngineVersions();
            engineVersionsResult.insert(engineVersionsResult.end(),
 engineVersions.begin(),
                                        engineVersions.end());
            marker = outcome.GetResult().GetMarker();
        }
        else {
            std::cerr << "Error with RDS::DescribeDBEngineVersionsRequest. "
                      << outcome.GetError().GetMessage()
                      << std::endl;
            return false;
        }

    } while (!marker.empty());


    return true;
}


//! Routine which gets a DB instance description.
/*!
 \sa describeDBInstance()
 \param dbInstanceIdentifier: A DB instance identifier.
 \param instanceResult: The 'DBInstance' object containing the description.
 \param client: 'RDSClient' instance.
 \return bool: Successful completion.
 */
bool AwsDoc::RDS::describeDBInstance(const Aws::String &dbInstanceIdentifier,
                                     Aws::RDS::Model::DBInstance &instanceResult,
                                     const Aws::RDS::RDSClient &client) {
    Aws::RDS::Model::DescribeDBInstancesRequest request;
    request.SetDBInstanceIdentifier(dbInstanceIdentifier);

    Aws::RDS::Model::DescribeDBInstancesOutcome outcome =
            client.DescribeDBInstances(request);

    bool result = true;
    if (outcome.IsSuccess()) {
        instanceResult = outcome.GetResult().GetDBInstances()[0];
```

```
    }
    else if (outcome.GetError().GetErrorType() !=
             Aws::RDS::RDSErrors::D_B_INSTANCE_NOT_FOUND_FAULT) {
        result = false;
        std::cerr << "Error with RDS::DescribeDBInstances. "
                  << outcome.GetError().GetMessage()
                  << std::endl;
    }
        // This example does not log an error if the DB instance does not exist.
        // Instead, instanceResult is set to empty.
    else {
        instanceResult = Aws::RDS::Model::DBInstance();
    }

    return result;
}


//! Routine which gets available 'micro' DB instance classes, displays the list
//! to the user, and returns the user selection.
/*!
 \sa chooseMicroDBInstanceClass()
 \param engineName: The DB engine name.
 \param engineVersion: The DB engine version.
 \param dbInstanceClass: String for DB instance class chosen by the user.
 \param client: 'RDSClient' instance.
 \return bool: Successful completion.
 */
bool AwsDoc::RDS::chooseMicroDBInstanceClass(const Aws::String &engine,
                                             const Aws::String &engineVersion,
                                             Aws::String &dbInstanceClass,
                                             const Aws::RDS::RDSClient &client) {
    std::vector<Aws::String> instanceClasses;
    Aws::String marker;
    do {
        Aws::RDS::Model::DescribeOrderableDBInstanceOptionsRequest request;
        request.SetEngine(engine);
        request.SetEngineVersion(engineVersion);
        if (!marker.empty()) {
            request.SetMarker(marker);
        }

        Aws::RDS::Model::DescribeOrderableDBInstanceOptionsOutcome outcome =
                client.DescribeOrderableDBInstanceOptions(request);
```

```
        if (outcome.IsSuccess()) {
            const Aws::Vector<Aws::RDS::Model::OrderableDBInstanceOption>
 &options =
                    outcome.GetResult().GetOrderableDBInstanceOptions();
            for (const Aws::RDS::Model::OrderableDBInstanceOption &option:
 options) {
                const Aws::String &instanceClass = option.GetDBInstanceClass();
                if (instanceClass.find("micro") != std::string::npos) {
                    if (std::find(instanceClasses.begin(), instanceClasses.end(),
                                  instanceClass) ==
                        instanceClasses.end()) {
                        instanceClasses.push_back(instanceClass);
                    }
                }
            }
            marker = outcome.GetResult().GetMarker();
        }
        else {
            std::cerr << "Error with RDS::DescribeOrderableDBInstanceOptions. "
                      << outcome.GetError().GetMessage()
                      << std::endl;
            return false;
        }
    } while (!marker.empty());

    std::cout << "The available micro DB instance classes for your database
 engine are:"
              << std::endl;
    for (int i = 0; i < instanceClasses.size(); ++i) {
        std::cout << "   " << i + 1 << ": " << instanceClasses[i] << std::endl;
    }

    int choice = askQuestionForIntRange(
            "Which micro DB instance class do you want to use? ",
            1, static_cast<int>(instanceClasses.size()));
    dbInstanceClass = instanceClasses[choice - 1];
    return true;
}

//! Routine which deletes resources created by the scenario.
/*!
\sa cleanUpResources()
\param parameterGroupName: A parameter group name, this may be empty.
```

```
\param dbInstanceIdentifier: A DB instance identifier, this may be empty.
\param client: 'RDSClient' instance.
\return bool: Successful completion.
*/
bool AwsDoc::RDS::cleanUpResources(const Aws::String &parameterGroupName,
                                   const Aws::String &dbInstanceIdentifier,
                                   const Aws::RDS::RDSClient &client) {
    bool result = true;
    if (!dbInstanceIdentifier.empty()) {
        {
            // 15. Delete the DB instance.
            Aws::RDS::Model::DeleteDBInstanceRequest request;
            request.SetDBInstanceIdentifier(dbInstanceIdentifier);
            request.SetSkipFinalSnapshot(true);
            request.SetDeleteAutomatedBackups(true);

            Aws::RDS::Model::DeleteDBInstanceOutcome outcome =
                    client.DeleteDBInstance(request);

            if (outcome.IsSuccess()) {
                std::cout << "DB instance deletion has started."
                          << std::endl;
            }
            else {
                std::cerr << "Error with RDS::DeleteDBInstance. "
                          << outcome.GetError().GetMessage()
                          << std::endl;
                result = false;
            }
        }

        std::cout
                << "Waiting for DB instance to delete before deleting the
 parameter group."
                << std::endl;
        std::cout << "This may take a while." << std::endl;

        int counter = 0;
        Aws::RDS::Model::DBInstance dbInstance;
        do {
            std::this_thread::sleep_for(std::chrono::seconds(1));
            ++counter;
            if (counter > 800) {
```

```
                std::cerr << "Wait for instance to delete timed out ofter " <<
 counter
                          << " seconds." << std::endl;
                return false;
            }

            dbInstance = Aws::RDS::Model::DBInstance();
            // 16. Wait for the DB instance to be deleted.
            if (!describeDBInstance(dbInstanceIdentifier, dbInstance, client)) {
                return false;
            }

            if (dbInstance.DBInstanceIdentifierHasBeenSet() && (counter % 20) ==
 0) {
                std::cout << "Current DB instance status is '"
                          << dbInstance.GetDBInstanceStatus()
                          << "' after " << counter << " seconds." << std::endl;
            }
        } while (dbInstance.DBInstanceIdentifierHasBeenSet());
    }

    if (!parameterGroupName.empty()) {
        // 17. Delete the parameter group.
        Aws::RDS::Model::DeleteDBParameterGroupRequest request;
        request.SetDBParameterGroupName(parameterGroupName);

        Aws::RDS::Model::DeleteDBParameterGroupOutcome outcome =
                client.DeleteDBParameterGroup(request);

        if (outcome.IsSuccess()) {
            std::cout << "The DB parameter group was successfully deleted."
                      << std::endl;
        }
        else {
            std::cerr << "Error with RDS::DeleteDBParameterGroup. "
                      << outcome.GetError().GetMessage()
                      << std::endl;
            result = false;
        }
    }

    return result;
}
```

- Para obter detalhes da API, consulte os tópicos a seguir na Referência da API AWS SDK para C++.
  - CreateDBInstance
  - CreateDBParameterGroup
  - CreateDBSnapshot
  - DeleteDBInstance
  - DeleteDBParameterGroup
  - DescribeDBEngineVersions
  - DescribeDBInstances
  - DescribeDBParameterGroups
  - DescribeDBParameters
  - DescribeDBSnapshots
  - DescribeOrderableDBInstanceOptions
  - ModifyDBParameterGroup

Go

SDK para Go V2

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

Execute um cenário interativo em um prompt de comando.

```
// GetStartedInstances is an interactive example that shows you how to use the
 AWS SDK for Go
// with Amazon Relation Database Service (Amazon RDS) to do the following:
//
//  1. Create a custom DB parameter group and set parameter values.
```

```
//  2. Create a DB instance that is configured to use the parameter group. The DB
 instance
//     also contains a database.
//  3. Take a snapshot of the DB instance.
//  4. Delete the DB instance and parameter group.
type GetStartedInstances struct {
 sdkConfig  aws.Config
 instances  actions.DbInstances
 questioner demotools.IQuestioner
 helper     IScenarioHelper
 isTestRun  bool
}

// NewGetStartedInstances constructs a GetStartedInstances instance from a
 configuration.
// It uses the specified config to get an Amazon RDS
// client and create wrappers for the actions used in the scenario.
func NewGetStartedInstances(sdkConfig aws.Config, questioner
 demotools.IQuestioner,
 helper IScenarioHelper) GetStartedInstances {
 rdsClient := rds.NewFromConfig(sdkConfig)
 return GetStartedInstances{
  sdkConfig:  sdkConfig,
  instances:  actions.DbInstances{RdsClient: rdsClient},
  questioner: questioner,
  helper:     helper,
 }
}

// Run runs the interactive scenario.
func (scenario GetStartedInstances) Run(dbEngine string, parameterGroupName
 string,
 instanceName string, dbName string) {
 defer func() {
  if r := recover(); r != nil {
   log.Println("Something went wrong with the demo.")
  }
 }()

 log.Println(strings.Repeat("-", 88))
 log.Println("Welcome to the Amazon Relational Database Service (Amazon RDS) DB
 Instance demo.")
 log.Println(strings.Repeat("-", 88))
```

```
 parameterGroup := scenario.CreateParameterGroup(dbEngine, parameterGroupName)
 scenario.SetUserParameters(parameterGroupName)
 instance := scenario.CreateInstance(instanceName, dbEngine, dbName,
 parameterGroup)
 scenario.DisplayConnection(instance)
 scenario.CreateSnapshot(instance)
 scenario.Cleanup(instance, parameterGroup)

 log.Println(strings.Repeat("-", 88))
 log.Println("Thanks for watching!")
 log.Println(strings.Repeat("-", 88))
}

// CreateParameterGroup shows how to get available engine versions for a
 specified
// database engine and create a DB parameter group that is compatible with a
// selected engine family.
func (scenario GetStartedInstances) CreateParameterGroup(dbEngine string,
 parameterGroupName string) *types.DBParameterGroup {

 log.Printf("Checking for an existing DB parameter group named %v.\n",
  parameterGroupName)
 parameterGroup, err := scenario.instances.GetParameterGroup(parameterGroupName)
 if err != nil {
  panic(err)
 }
 if parameterGroup == nil {
  log.Printf("Getting available database engine versions for %v.\n", dbEngine)
  engineVersions, err := scenario.instances.GetEngineVersions(dbEngine, "")
  if err != nil {
   panic(err)
  }

  familySet := map[string]struct{}{}
  for _, family := range engineVersions {
   familySet[*family.DBParameterGroupFamily] = struct{}{}
  }
  var families []string
  for family := range familySet {
   families = append(families, family)
  }
  sort.Strings(families)
  familyIndex := scenario.questioner.AskChoice("Which family do you want to use?
\n", families)
```

```
  log.Println("Creating a DB parameter group.")
  _, err = scenario.instances.CreateParameterGroup(
   parameterGroupName, families[familyIndex], "Example parameter group.")
  if err != nil {
   panic(err)
  }
  parameterGroup, err = scenario.instances.GetParameterGroup(parameterGroupName)
  if err != nil {
   panic(err)
  }
 }
 log.Printf("Parameter group %v:\n", *parameterGroup.DBParameterGroupFamily)
 log.Printf("\tName: %v\n", *parameterGroup.DBParameterGroupName)
 log.Printf("\tARN: %v\n", *parameterGroup.DBParameterGroupArn)
 log.Printf("\tFamily: %v\n", *parameterGroup.DBParameterGroupFamily)
 log.Printf("\tDescription: %v\n", *parameterGroup.Description)
 log.Println(strings.Repeat("-", 88))
 return parameterGroup
}

// SetUserParameters shows how to get the parameters contained in a custom
 parameter
// group and update some of the parameter values in the group.
func (scenario GetStartedInstances) SetUserParameters(parameterGroupName string)
 {
 log.Println("Let's set some parameter values in your parameter group.")
 dbParameters, err := scenario.instances.GetParameters(parameterGroupName, "")
 if err != nil {
  panic(err)
 }
 var updateParams []types.Parameter
 for _, dbParam := range dbParameters {
  if strings.HasPrefix(*dbParam.ParameterName, "auto_increment") &&
   dbParam.IsModifiable && *dbParam.DataType == "integer" {
   log.Printf("The %v parameter is described as:\n\t%v",
    *dbParam.ParameterName, *dbParam.Description)
   rangeSplit := strings.Split(*dbParam.AllowedValues, "-")
   lower, _ := strconv.Atoi(rangeSplit[0])
   upper, _ := strconv.Atoi(rangeSplit[1])
   newValue := scenario.questioner.AskInt(
    fmt.Sprintf("Enter a value between %v and %v:", lower, upper),
    demotools.InIntRange{Lower: lower, Upper: upper})
   dbParam.ParameterValue = aws.String(strconv.Itoa(newValue))
   updateParams = append(updateParams, dbParam)
```

```
  }
 }
 err = scenario.instances.UpdateParameters(parameterGroupName, updateParams)
 if err != nil {
  panic(err)
 }
 log.Println("To get a list of parameters that you set previously, specify a
 source of 'user'.")
 userParameters, err := scenario.instances.GetParameters(parameterGroupName,
 "user")
 if err != nil {
  panic(err)
 }
 log.Println("Here are the parameters you set:")
 for _, param := range userParameters {
  log.Printf("\t%v: %v\n", *param.ParameterName, *param.ParameterValue)
 }
 log.Println(strings.Repeat("-", 88))
}

// CreateInstance shows how to create a DB instance that contains a database of a
// specified type. The database is also configured to use a custom DB parameter
 group.
func (scenario GetStartedInstances) CreateInstance(instanceName string, dbEngine
 string,
 dbName string, parameterGroup *types.DBParameterGroup) *types.DBInstance {

 log.Println("Checking for an existing DB instance.")
 instance, err := scenario.instances.GetInstance(instanceName)
 if err != nil {
  panic(err)
 }
 if instance == nil {
  adminUsername := scenario.questioner.Ask(
   "Enter an administrator username for the database: ", demotools.NotEmpty{})
  adminPassword := scenario.questioner.AskPassword(
   "Enter a password for the administrator (at least 8 characters): ", 7)
  engineVersions, err := scenario.instances.GetEngineVersions(dbEngine,
   *parameterGroup.DBParameterGroupFamily)
  if err != nil {
   panic(err)
  }
  var engineChoices []string
  for _, engine := range engineVersions {
```

```
   engineChoices = append(engineChoices, *engine.EngineVersion)
  }
  engineIndex := scenario.questioner.AskChoice(
   "The available engines for your parameter group are:\n", engineChoices)
  engineSelection := engineVersions[engineIndex]
  instOpts, err :=
 scenario.instances.GetOrderableInstances(*engineSelection.Engine,
   *engineSelection.EngineVersion)
  if err != nil {
   panic(err)
  }
  optSet := map[string]struct{}{}
  for _, opt := range instOpts {
   if strings.Contains(*opt.DBInstanceClass, "micro") {
    optSet[*opt.DBInstanceClass] = struct{}{}
   }
  }
  var optChoices []string
  for opt := range optSet {
   optChoices = append(optChoices, opt)
  }
  sort.Strings(optChoices)
  optIndex := scenario.questioner.AskChoice(
   "The available micro DB instance classes for your database engine are:\n",
 optChoices)
  storageType := "standard"
  allocatedStorage := int32(5)
  log.Printf("Creating a DB instance named %v and database %v.\n"+
   "The DB instance is configured to use your custom parameter group %v,\n"+
   "selected engine %v,\n"+
   "selected DB instance class %v,"+
   "and %v GiB of %v storage.\n"+
   "This typically takes several minutes.",
   instanceName, dbName, *parameterGroup.DBParameterGroupName,
 *engineSelection.EngineVersion,
   optChoices[optIndex], allocatedStorage, storageType)
  instance, err = scenario.instances.CreateInstance(
   instanceName, dbName, *engineSelection.Engine, *engineSelection.EngineVersion,
   *parameterGroup.DBParameterGroupName, optChoices[optIndex], storageType,
   allocatedStorage, adminUsername, adminPassword)
  if err != nil {
   panic(err)
  }
  for *instance.DBInstanceStatus != "available" {
```

```
   scenario.helper.Pause(30)
   instance, err = scenario.instances.GetInstance(instanceName)
   if err != nil {
    panic(err)
   }
  }
  log.Println("Instance created and available.")
 }
 log.Println("Instance data:")
 log.Printf("\tDBInstanceIdentifier: %v\n", *instance.DBInstanceIdentifier)
 log.Printf("\tARN: %v\n", *instance.DBInstanceArn)
 log.Printf("\tStatus: %v\n", *instance.DBInstanceStatus)
 log.Printf("\tEngine: %v\n", *instance.Engine)
 log.Printf("\tEngine version: %v\n", *instance.EngineVersion)
 log.Println(strings.Repeat("-", 88))
 return instance
}

// DisplayConnection displays connection information about a DB instance and tips
// on how to connect to it.
func (scenario GetStartedInstances) DisplayConnection(instance *types.DBInstance)
 {
 log.Println(
  "You can now connect to your database by using your favorite MySQL client.\n" +
   "One way to connect is by using the 'mysql' shell on an Amazon EC2 instance\n"
 +
   "that is running in the same VPC as your DB instance. Pass the endpoint,\n" +
   "port, and administrator username to 'mysql'. Then, enter your password\n" +
   "when prompted:")
 log.Printf("\n\tmysql -h %v -P %v -u %v -p\n",
  *instance.Endpoint.Address, instance.Endpoint.Port, *instance.MasterUsername)
 log.Println("For more information, see the User Guide for RDS:\n" +
  "\thttps://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/
CHAP_GettingStarted.CreatingConnecting.MySQL.html#CHAP_GettingStarted.Connecting.MySQL")
 log.Println(strings.Repeat("-", 88))
}

// CreateSnapshot shows how to create a DB instance snapshot and wait until it's
 available.
func (scenario GetStartedInstances) CreateSnapshot(instance *types.DBInstance) {
 if scenario.questioner.AskBool(
  "Do you want to create a snapshot of your DB instance (y/n)? ", "y") {
  snapshotId := fmt.Sprintf("%v-%v", *instance.DBInstanceIdentifier,
 scenario.helper.UniqueId())
```

```
  log.Printf("Creating a snapshot named %v. This typically takes a few minutes.
\n", snapshotId)
  snapshot, err :=
 scenario.instances.CreateSnapshot(*instance.DBInstanceIdentifier, snapshotId)
  if err != nil {
   panic(err)
  }
  for *snapshot.Status != "available" {
   scenario.helper.Pause(30)
   snapshot, err = scenario.instances.GetSnapshot(snapshotId)
   if err != nil {
    panic(err)
   }
  }
  log.Println("Snapshot data:")
  log.Printf("\tDBSnapshotIdentifier: %v\n", *snapshot.DBSnapshotIdentifier)
  log.Printf("\tARN: %v\n", *snapshot.DBSnapshotArn)
  log.Printf("\tStatus: %v\n", *snapshot.Status)
  log.Printf("\tEngine: %v\n", *snapshot.Engine)
  log.Printf("\tEngine version: %v\n", *snapshot.EngineVersion)
  log.Printf("\tDBInstanceIdentifier: %v\n", *snapshot.DBInstanceIdentifier)
  log.Printf("\tSnapshotCreateTime: %v\n", *snapshot.SnapshotCreateTime)
  log.Println(strings.Repeat("-", 88))
 }
}

// Cleanup shows how to clean up a DB instance and DB parameter group.
// Before the DB parameter group can be deleted, all associated DB instances must
 first be deleted.
func (scenario GetStartedInstances) Cleanup(
 instance *types.DBInstance, parameterGroup *types.DBParameterGroup) {

 if scenario.questioner.AskBool(
  "\nDo you want to delete the database instance and parameter group (y/n)? ",
 "y") {
  log.Printf("Deleting database instance %v.\n", *instance.DBInstanceIdentifier)
  err := scenario.instances.DeleteInstance(*instance.DBInstanceIdentifier)
  if err != nil {
   panic(err)
  }
  log.Println(
   "Waiting for the DB instance to delete. This typically takes several
 minutes.")
  for instance != nil {
```

```
   scenario.helper.Pause(30)
   instance, err = scenario.instances.GetInstance(*instance.DBInstanceIdentifier)
   if err != nil {
    panic(err)
   }
  }
  log.Printf("Deleting parameter group %v.",
 *parameterGroup.DBParameterGroupName)
  err =
 scenario.instances.DeleteParameterGroup(*parameterGroup.DBParameterGroupName)
  if err != nil {
   panic(err)
  }
 }
}
```

Defina as funções que são chamadas pelo cenário para gerenciar as ações do Amazon RDS.

```
type DbInstances struct {
 RdsClient *rds.Client
}


// GetParameterGroup gets a DB parameter group by name.
func (instances *DbInstances) GetParameterGroup(parameterGroupName string) (
 *types.DBParameterGroup, error) {
 output, err := instances.RdsClient.DescribeDBParameterGroups(
  context.TODO(), &rds.DescribeDBParameterGroupsInput{
   DBParameterGroupName: aws.String(parameterGroupName),
  })
 if err != nil {
  var notFoundError *types.DBParameterGroupNotFoundFault
  if errors.As(err, &notFoundError) {
   log.Printf("Parameter group %v does not exist.\n", parameterGroupName)
   err = nil
  } else {
   log.Printf("Error getting parameter group %v: %v\n", parameterGroupName, err)
  }
  return nil, err
 } else {
```

```
  return &output.DBParameterGroups[0], err
 }
}



// CreateParameterGroup creates a DB parameter group that is based on the
 specified
// parameter group family.
func (instances *DbInstances) CreateParameterGroup(
 parameterGroupName string, parameterGroupFamily string, description string) (
 *types.DBParameterGroup, error) {

 output, err := instances.RdsClient.CreateDBParameterGroup(context.TODO(),
  &rds.CreateDBParameterGroupInput{
   DBParameterGroupName:   aws.String(parameterGroupName),
   DBParameterGroupFamily: aws.String(parameterGroupFamily),
   Description:            aws.String(description),
  })
 if err != nil {
  log.Printf("Couldn't create parameter group %v: %v\n", parameterGroupName, err)
  return nil, err
 } else {
  return output.DBParameterGroup, err
 }
}



// DeleteParameterGroup deletes the named DB parameter group.
func (instances *DbInstances) DeleteParameterGroup(parameterGroupName string)
 error {
 _, err := instances.RdsClient.DeleteDBParameterGroup(context.TODO(),
  &rds.DeleteDBParameterGroupInput{
   DBParameterGroupName: aws.String(parameterGroupName),
  })
 if err != nil {
  log.Printf("Couldn't delete parameter group %v: %v\n", parameterGroupName, err)
  return err
 } else {
  return nil
 }
}
```

```
// GetParameters gets the parameters that are contained in a DB parameter group.
func (instances *DbInstances) GetParameters(parameterGroupName string, source
 string) (
 []types.Parameter, error) {

 var output *rds.DescribeDBParametersOutput
 var params []types.Parameter
 var err error
 parameterPaginator := rds.NewDescribeDBParametersPaginator(instances.RdsClient,
  &rds.DescribeDBParametersInput{
   DBParameterGroupName: aws.String(parameterGroupName),
   Source:               aws.String(source),
  })
 for parameterPaginator.HasMorePages() {
  output, err = parameterPaginator.NextPage(context.TODO())
  if err != nil {
   log.Printf("Couldn't get parameters for %v: %v\n", parameterGroupName, err)
   break
  } else {
   params = append(params, output.Parameters...)
  }
 }
 return params, err
}



// UpdateParameters updates parameters in a named DB parameter group.
func (instances *DbInstances) UpdateParameters(parameterGroupName string, params
 []types.Parameter) error {
 _, err := instances.RdsClient.ModifyDBParameterGroup(context.TODO(),
  &rds.ModifyDBParameterGroupInput{
   DBParameterGroupName: aws.String(parameterGroupName),
   Parameters:           params,
  })
 if err != nil {
  log.Printf("Couldn't update parameters in %v: %v\n", parameterGroupName, err)
  return err
 } else {
  return nil
 }
}
```

```
// CreateSnapshot creates a snapshot of a DB instance.
func (instances *DbInstances) CreateSnapshot(instanceName string, snapshotName
 string) (
 *types.DBSnapshot, error) {
 output, err := instances.RdsClient.CreateDBSnapshot(context.TODO(),
 &rds.CreateDBSnapshotInput{
  DBInstanceIdentifier: aws.String(instanceName),
  DBSnapshotIdentifier: aws.String(snapshotName),
 })
 if err != nil {
  log.Printf("Couldn't create snapshot %v: %v\n", snapshotName, err)
  return nil, err
 } else {
  return output.DBSnapshot, nil
 }
}



// GetSnapshot gets a DB instance snapshot.
func (instances *DbInstances) GetSnapshot(snapshotName string)
 (*types.DBSnapshot, error) {
 output, err := instances.RdsClient.DescribeDBSnapshots(context.TODO(),
  &rds.DescribeDBSnapshotsInput{
   DBSnapshotIdentifier: aws.String(snapshotName),
  })
 if err != nil {
  log.Printf("Couldn't get snapshot %v: %v\n", snapshotName, err)
  return nil, err
 } else {
  return &output.DBSnapshots[0], nil
 }
}



// CreateInstance creates a DB instance.
func (instances *DbInstances) CreateInstance(instanceName string, dbName string,
 dbEngine string, dbEngineVersion string, parameterGroupName string,
 dbInstanceClass string,
```

```
 storageType string, allocatedStorage int32, adminName string, adminPassword
 string) (
 *types.DBInstance, error) {
 output, err := instances.RdsClient.CreateDBInstance(context.TODO(),
 &rds.CreateDBInstanceInput{
  DBInstanceIdentifier: aws.String(instanceName),
  DBName:               aws.String(dbName),
  DBParameterGroupName: aws.String(parameterGroupName),
  Engine:               aws.String(dbEngine),
  EngineVersion:        aws.String(dbEngineVersion),
  DBInstanceClass:      aws.String(dbInstanceClass),
  StorageType:          aws.String(storageType),
  AllocatedStorage:     aws.Int32(allocatedStorage),
  MasterUsername:       aws.String(adminName),
  MasterUserPassword:   aws.String(adminPassword),
 })
 if err != nil {
  log.Printf("Couldn't create instance %v: %v\n", instanceName, err)
  return nil, err
 } else {
  return output.DBInstance, nil
 }
}



// GetInstance gets data about a DB instance.
func (instances *DbInstances) GetInstance(instanceName string) (
 *types.DBInstance, error) {
 output, err := instances.RdsClient.DescribeDBInstances(context.TODO(),
  &rds.DescribeDBInstancesInput{
   DBInstanceIdentifier: aws.String(instanceName),
  })
 if err != nil {
  var notFoundError *types.DBInstanceNotFoundFault
  if errors.As(err, &notFoundError) {
   log.Printf("DB instance %v does not exist.\n", instanceName)
   err = nil
  } else {
   log.Printf("Couldn't get instance %v: %v\n", instanceName, err)
  }
  return nil, err
 } else {
  return &output.DBInstances[0], nil
```

```
 }
}



// DeleteInstance deletes a DB instance.
func (instances *DbInstances) DeleteInstance(instanceName string) error {
 _, err := instances.RdsClient.DeleteDBInstance(context.TODO(),
 &rds.DeleteDBInstanceInput{
  DBInstanceIdentifier:   aws.String(instanceName),
  SkipFinalSnapshot:      true,
  DeleteAutomatedBackups: aws.Bool(true),
 })
 if err != nil {
  log.Printf("Couldn't delete instance %v: %v\n", instanceName, err)
  return err
 } else {
  return nil
 }
}



// GetEngineVersions gets database engine versions that are available for the
 specified engine
// and parameter group family.
func (instances *DbInstances) GetEngineVersions(engine string,
 parameterGroupFamily string) (
 []types.DBEngineVersion, error) {
 output, err := instances.RdsClient.DescribeDBEngineVersions(context.TODO(),
  &rds.DescribeDBEngineVersionsInput{
   Engine:                 aws.String(engine),
   DBParameterGroupFamily: aws.String(parameterGroupFamily),
  })
 if err != nil {
  log.Printf("Couldn't get engine versions for %v: %v\n", engine, err)
  return nil, err
 } else {
  return output.DBEngineVersions, nil
 }
}
```

```
// GetOrderableInstances uses a paginator to get DB instance options that can be
 used to create DB instances that are
// compatible with a set of specifications.
func (instances *DbInstances) GetOrderableInstances(engine string, engineVersion
 string) (
 []types.OrderableDBInstanceOption, error) {

 var output *rds.DescribeOrderableDBInstanceOptionsOutput
 var instanceOptions []types.OrderableDBInstanceOption
 var err error
 orderablePaginator :=
 rds.NewDescribeOrderableDBInstanceOptionsPaginator(instances.RdsClient,
  &rds.DescribeOrderableDBInstanceOptionsInput{
   Engine:        aws.String(engine),
   EngineVersion: aws.String(engineVersion),
  })
 for orderablePaginator.HasMorePages() {
  output, err = orderablePaginator.NextPage(context.TODO())
  if err != nil {
   log.Printf("Couldn't get orderable DB instance options: %v\n", err)
   break
  } else {
   instanceOptions = append(instanceOptions,
 output.OrderableDBInstanceOptions...)
  }
 }
 return instanceOptions, err
}
```

- Para obter detalhes da API, consulte os tópicos a seguir na Referência da API AWS SDK para Go.
  - CreateDBInstance
  - CreateDBParameterGroup
  - CreateDBSnapshot
  - DeleteDBInstance
  - DeleteDBParameterGroup
  - DescribeDBEngineVersions
  - DescribeDBInstances

- DescribeDBParameterGroups
- DescribeDBParameters
- DescribeDBSnapshots
- DescribeOrderableDBInstanceOptions
- ModifyDBParameterGroup

Java

SDK para Java 2.x

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

Execute várias operações.

```
import com.google.gson.Gson;
import
 software.amazon.awssdk.auth.credentials.EnvironmentVariableCredentialsProvider;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.rds.RdsClient;
import software.amazon.awssdk.services.rds.model.CreateDbInstanceRequest;
import software.amazon.awssdk.services.rds.model.CreateDbInstanceResponse;
import software.amazon.awssdk.services.rds.model.CreateDbParameterGroupResponse;
import software.amazon.awssdk.services.rds.model.CreateDbSnapshotRequest;
import software.amazon.awssdk.services.rds.model.CreateDbSnapshotResponse;
import software.amazon.awssdk.services.rds.model.DBEngineVersion;
import software.amazon.awssdk.services.rds.model.DBInstance;
import software.amazon.awssdk.services.rds.model.DBParameterGroup;
import software.amazon.awssdk.services.rds.model.DBSnapshot;
import software.amazon.awssdk.services.rds.model.DeleteDbInstanceRequest;
import software.amazon.awssdk.services.rds.model.DeleteDbInstanceResponse;
import software.amazon.awssdk.services.rds.model.DescribeDbEngineVersionsRequest;
import
 software.amazon.awssdk.services.rds.model.DescribeDbEngineVersionsResponse;
import software.amazon.awssdk.services.rds.model.DescribeDbInstancesRequest;
import software.amazon.awssdk.services.rds.model.DescribeDbInstancesResponse;
```

```
import
 software.amazon.awssdk.services.rds.model.DescribeDbParameterGroupsResponse;
import software.amazon.awssdk.services.rds.model.DescribeDbParametersResponse;
import software.amazon.awssdk.services.rds.model.DescribeDbSnapshotsRequest;
import software.amazon.awssdk.services.rds.model.DescribeDbSnapshotsResponse;
import
 software.amazon.awssdk.services.rds.model.DescribeOrderableDbInstanceOptionsResponse;
import software.amazon.awssdk.services.rds.model.ModifyDbParameterGroupResponse;
import software.amazon.awssdk.services.rds.model.OrderableDBInstanceOption;
import software.amazon.awssdk.services.rds.model.Parameter;
import software.amazon.awssdk.services.rds.model.RdsException;
import software.amazon.awssdk.services.rds.model.CreateDbParameterGroupRequest;
import
 software.amazon.awssdk.services.rds.model.DescribeDbParameterGroupsRequest;
import software.amazon.awssdk.services.rds.model.DescribeDbParametersRequest;
import software.amazon.awssdk.services.rds.model.ModifyDbParameterGroupRequest;
import
 software.amazon.awssdk.services.rds.model.DescribeOrderableDbInstanceOptionsRequest;
import software.amazon.awssdk.services.rds.model.DeleteDbParameterGroupRequest;
import software.amazon.awssdk.services.secretsmanager.SecretsManagerClient;
import
 software.amazon.awssdk.services.secretsmanager.model.GetSecretValueRequest;
import
 software.amazon.awssdk.services.secretsmanager.model.GetSecretValueResponse;
import java.util.ArrayList;
import java.util.List;

/**
 * Before running this Java (v2) code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 *
 * This example requires an AWS Secrets Manager secret that contains the
 * database credentials. If you do not create a
 * secret, this example will not work. For details, see:
 *
 * https://docs.aws.amazon.com/secretsmanager/latest/userguide/integrating_how-
services-use-secrets_RS.html
 *
 * This Java example performs these tasks:
```

```
 *
 * 1. Returns a list of the available DB engines.
 * 2. Selects an engine family and create a custom DB parameter group.
 * 3. Gets the parameter groups.
 * 4. Gets parameters in the group.
 * 5. Modifies the auto_increment_offset parameter.
 * 6. Gets and displays the updated parameters.
 * 7. Gets a list of allowed engine versions.
 * 8. Gets a list of micro instance classes available for the selected engine.
 * 9. Creates an RDS database instance that contains a MySql database and uses
 * the parameter group.
 * 10. Waits for the DB instance to be ready and prints out the connection
 * endpoint value.
 * 11. Creates a snapshot of the DB instance.
 * 12. Waits for an RDS DB snapshot to be ready.
 * 13. Deletes the RDS DB instance.
 * 14. Deletes the parameter group.
 */
public class RDSScenario {
    public static long sleepTime = 20;
    public static final String DASHES = new String(new char[80]).replace("\0",
 "-");

    public static void main(String[] args) throws InterruptedException {
        final String usage = """

                Usage:
                    <dbGroupName> <dbParameterGroupFamily> <dbInstanceIdentifier>
 <dbName> <dbSnapshotIdentifier> <secretName>

                Where:
                    dbGroupName - The database group name.\s
                    dbParameterGroupFamily - The database parameter group name
 (for example, mysql8.0).
                    dbInstanceIdentifier - The database instance identifier\s
                    dbName - The database name.\s
                    dbSnapshotIdentifier - The snapshot identifier.\s
                    secretName - The name of the AWS Secrets Manager secret that
 contains the database credentials"
                """;

        if (args.length != 6) {
            System.out.println(usage);
            System.exit(1);
```

```
        }

        String dbGroupName = args[0];
        String dbParameterGroupFamily = args[1];
        String dbInstanceIdentifier = args[2];
        String dbName = args[3];
        String dbSnapshotIdentifier = args[4];
        String secretName = args[5];

        Gson gson = new Gson();
        User user = gson.fromJson(String.valueOf(getSecretValues(secretName)),
User.class);
        String masterUsername = user.getUsername();
        String masterUserPassword = user.getPassword();

        Region region = Region.US_WEST_2;
        RdsClient rdsClient = RdsClient.builder()
                .region(region)
                .build();
        System.out.println(DASHES);
        System.out.println("Welcome to the Amazon RDS example scenario.");
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("1. Return a list of the available DB engines");
        describeDBEngines(rdsClient);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("2. Create a custom parameter group");
        createDBParameterGroup(rdsClient, dbGroupName, dbParameterGroupFamily);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("3. Get the parameter group");
        describeDbParameterGroups(rdsClient, dbGroupName);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("4. Get the parameters in the group");
        describeDbParameters(rdsClient, dbGroupName, 0);
        System.out.println(DASHES);

        System.out.println(DASHES);
```

```
        System.out.println("5. Modify the auto_increment_offset parameter");
        modifyDBParas(rdsClient, dbGroupName);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("6. Display the updated value");
        describeDbParameters(rdsClient, dbGroupName, -1);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("7. Get a list of allowed engine versions");
        getAllowedEngines(rdsClient, dbParameterGroupFamily);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("8. Get a list of micro instance classes available for
the selected engine");
        getMicroInstances(rdsClient);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println(
                "9. Create an RDS database instance that contains a MySql
database and uses the parameter group");
        String dbARN = createDatabaseInstance(rdsClient, dbGroupName,
dbInstanceIdentifier, dbName, masterUsername,
                masterUserPassword);
        System.out.println("The ARN of the new database is " + dbARN);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("10. Wait for DB instance to be ready");
        waitForInstanceReady(rdsClient, dbInstanceIdentifier);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("11. Create a snapshot of the DB instance");
        createSnapshot(rdsClient, dbInstanceIdentifier, dbSnapshotIdentifier);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("12. Wait for DB snapshot to be ready");
        waitForSnapshotReady(rdsClient, dbInstanceIdentifier,
dbSnapshotIdentifier);
```

```
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("13. Delete the DB instance");
        deleteDatabaseInstance(rdsClient, dbInstanceIdentifier);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("14. Delete the parameter group");
        deleteParaGroup(rdsClient, dbGroupName, dbARN);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("The Scenario has successfully completed.");
        System.out.println(DASHES);

        rdsClient.close();
    }

    private static SecretsManagerClient getSecretClient() {
        Region region = Region.US_WEST_2;
        return SecretsManagerClient.builder()
                .region(region)

.credentialsProvider(EnvironmentVariableCredentialsProvider.create())
                .build();
    }

    public static String getSecretValues(String secretName) {
        SecretsManagerClient secretClient = getSecretClient();
        GetSecretValueRequest valueRequest = GetSecretValueRequest.builder()
                .secretId(secretName)
                .build();

        GetSecretValueResponse valueResponse =
secretClient.getSecretValue(valueRequest);
        return valueResponse.secretString();
    }

    // Delete the parameter group after database has been deleted.
    // An exception is thrown if you attempt to delete the para group while
database
    // exists.
```

```
   public static void deleteParaGroup(RdsClient rdsClient, String dbGroupName,
String dbARN)
           throws InterruptedException {
       try {
           boolean isDataDel = false;
           boolean didFind;
           String instanceARN;

           // Make sure that the database has been deleted.
           while (!isDataDel) {
               DescribeDbInstancesResponse response =
rdsClient.describeDBInstances();
               List<DBInstance> instanceList = response.dbInstances();
               int listSize = instanceList.size();
               didFind = false;
               int index = 1;
               for (DBInstance instance : instanceList) {
                   instanceARN = instance.dbInstanceArn();
                   if (instanceARN.compareTo(dbARN) == 0) {
                       System.out.println(dbARN + " still exists");
                       didFind = true;
                   }
                   if ((index == listSize) && (!didFind)) {
                       // Went through the entire list and did not find the
database ARN.
                       isDataDel = true;
                   }
                   Thread.sleep(sleepTime * 1000);
                   index++;
               }
           }

           // Delete the para group.
           DeleteDbParameterGroupRequest parameterGroupRequest =
DeleteDbParameterGroupRequest.builder()
                   .dbParameterGroupName(dbGroupName)
                   .build();

           rdsClient.deleteDBParameterGroup(parameterGroupRequest);
           System.out.println(dbGroupName + " was deleted.");

       } catch (RdsException e) {
           System.out.println(e.getLocalizedMessage());
           System.exit(1);
```

```
        }
    }

    // Delete the DB instance.
    public static void deleteDatabaseInstance(RdsClient rdsClient, String
dbInstanceIdentifier) {
        try {
            DeleteDbInstanceRequest deleteDbInstanceRequest =
DeleteDbInstanceRequest.builder()
                    .dbInstanceIdentifier(dbInstanceIdentifier)
                    .deleteAutomatedBackups(true)
                    .skipFinalSnapshot(true)
                    .build();

            DeleteDbInstanceResponse response =
rdsClient.deleteDBInstance(deleteDbInstanceRequest);
            System.out.print("The status of the database is " +
response.dbInstance().dbInstanceStatus());

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
    }

    // Waits until the snapshot instance is available.
    public static void waitForSnapshotReady(RdsClient rdsClient, String
dbInstanceIdentifier,
            String dbSnapshotIdentifier) {
        try {
            boolean snapshotReady = false;
            String snapshotReadyStr;
            System.out.println("Waiting for the snapshot to become available.");

            DescribeDbSnapshotsRequest snapshotsRequest =
DescribeDbSnapshotsRequest.builder()
                    .dbSnapshotIdentifier(dbSnapshotIdentifier)
                    .dbInstanceIdentifier(dbInstanceIdentifier)
                    .build();

            while (!snapshotReady) {
                DescribeDbSnapshotsResponse response =
rdsClient.describeDBSnapshots(snapshotsRequest);
                List<DBSnapshot> snapshotList = response.dbSnapshots();
```

```
                for (DBSnapshot snapshot : snapshotList) {
                    snapshotReadyStr = snapshot.status();
                    if (snapshotReadyStr.contains("available")) {
                        snapshotReady = true;
                    } else {
                        System.out.print(".");
                        Thread.sleep(sleepTime * 1000);
                    }
                }
            }

            System.out.println("The Snapshot is available!");
        } catch (RdsException | InterruptedException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
    }

    // Create an Amazon RDS snapshot.
    public static void createSnapshot(RdsClient rdsClient, String
dbInstanceIdentifier, String dbSnapshotIdentifier) {
        try {
            CreateDbSnapshotRequest snapshotRequest =
CreateDbSnapshotRequest.builder()
                    .dbInstanceIdentifier(dbInstanceIdentifier)
                    .dbSnapshotIdentifier(dbSnapshotIdentifier)
                    .build();

            CreateDbSnapshotResponse response =
rdsClient.createDBSnapshot(snapshotRequest);
            System.out.println("The Snapshot id is " +
response.dbSnapshot().dbiResourceId());

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
    }

    // Waits until the database instance is available.
    public static void waitForInstanceReady(RdsClient rdsClient, String
dbInstanceIdentifier) {
        boolean instanceReady = false;
        String instanceReadyStr;
```

```
        System.out.println("Waiting for instance to become available.");
        try {
            DescribeDbInstancesRequest instanceRequest =
DescribeDbInstancesRequest.builder()
                    .dbInstanceIdentifier(dbInstanceIdentifier)
                    .build();

            String endpoint = "";
            while (!instanceReady) {
                DescribeDbInstancesResponse response =
rdsClient.describeDBInstances(instanceRequest);
                List<DBInstance> instanceList = response.dbInstances();
                for (DBInstance instance : instanceList) {
                    instanceReadyStr = instance.dbInstanceStatus();
                    if (instanceReadyStr.contains("available")) {
                        endpoint = instance.endpoint().address();
                        instanceReady = true;
                    } else {
                        System.out.print(".");
                        Thread.sleep(sleepTime * 1000);
                    }
                }
            }
            System.out.println("Database instance is available! The connection
endpoint is " + endpoint);

        } catch (RdsException | InterruptedException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }

    // Create a database instance and return the ARN of the database.
    public static String createDatabaseInstance(RdsClient rdsClient,
            String dbGroupName,
            String dbInstanceIdentifier,
            String dbName,
            String masterUsername,
            String masterUserPassword) {

        try {
            CreateDbInstanceRequest instanceRequest =
CreateDbInstanceRequest.builder()
                    .dbInstanceIdentifier(dbInstanceIdentifier)
```

```
                    .allocatedStorage(100)
                    .dbName(dbName)
                    .dbParameterGroupName(dbGroupName)
                    .engine("mysql")
                    .dbInstanceClass("db.m4.large")
                    .engineVersion("8.0")
                    .storageType("standard")
                    .masterUsername(masterUsername)
                    .masterUserPassword(masterUserPassword)
                    .build();

            CreateDbInstanceResponse response =
rdsClient.createDBInstance(instanceRequest);
            System.out.print("The status is " +
response.dbInstance().dbInstanceStatus());
            return response.dbInstance().dbInstanceArn();

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }

        return "";
    }

    // Get a list of micro instances.
    public static void getMicroInstances(RdsClient rdsClient) {
        try {
            DescribeOrderableDbInstanceOptionsRequest dbInstanceOptionsRequest =
DescribeOrderableDbInstanceOptionsRequest
                    .builder()
                    .engine("mysql")
                    .build();

            DescribeOrderableDbInstanceOptionsResponse response = rdsClient

.describeOrderableDBInstanceOptions(dbInstanceOptionsRequest);
            List<OrderableDBInstanceOption> orderableDBInstances =
response.orderableDBInstanceOptions();
            for (OrderableDBInstanceOption dbInstanceOption :
orderableDBInstances) {
                System.out.println("The engine version is " +
dbInstanceOption.engineVersion());
```

```
                System.out.println("The engine description is " +
dbInstanceOption.engine());
            }

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
    }

    // Get a list of allowed engine versions.
    public static void getAllowedEngines(RdsClient rdsClient, String
dbParameterGroupFamily) {
        try {
            DescribeDbEngineVersionsRequest versionsRequest =
DescribeDbEngineVersionsRequest.builder()
                    .dbParameterGroupFamily(dbParameterGroupFamily)
                    .engine("mysql")
                    .build();

            DescribeDbEngineVersionsResponse response =
rdsClient.describeDBEngineVersions(versionsRequest);
            List<DBEngineVersion> dbEngines = response.dbEngineVersions();
            for (DBEngineVersion dbEngine : dbEngines) {
                System.out.println("The engine version is " +
dbEngine.engineVersion());
                System.out.println("The engine description is " +
dbEngine.dbEngineDescription());
            }

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
    }

    // Modify auto_increment_offset and auto_increment_increment parameters.
    public static void modifyDBParas(RdsClient rdsClient, String dbGroupName) {
        try {
            Parameter parameter1 = Parameter.builder()
                    .parameterName("auto_increment_offset")
                    .applyMethod("immediate")
                    .parameterValue("5")
                    .build();
```

```
            List<Parameter> paraList = new ArrayList<>();
            paraList.add(parameter1);
            ModifyDbParameterGroupRequest groupRequest =
ModifyDbParameterGroupRequest.builder()
                    .dbParameterGroupName(dbGroupName)
                    .parameters(paraList)
                    .build();

            ModifyDbParameterGroupResponse response =
rdsClient.modifyDBParameterGroup(groupRequest);
            System.out.println("The parameter group " +
response.dbParameterGroupName() + " was successfully modified");

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
    }

    // Retrieve parameters in the group.
    public static void describeDbParameters(RdsClient rdsClient, String
dbGroupName, int flag) {
        try {
            DescribeDbParametersRequest dbParameterGroupsRequest;
            if (flag == 0) {
                dbParameterGroupsRequest = DescribeDbParametersRequest.builder()
                        .dbParameterGroupName(dbGroupName)
                        .build();
            } else {
                dbParameterGroupsRequest = DescribeDbParametersRequest.builder()
                        .dbParameterGroupName(dbGroupName)
                        .source("user")
                        .build();
            }

            DescribeDbParametersResponse response =
rdsClient.describeDBParameters(dbParameterGroupsRequest);
            List<Parameter> dbParameters = response.parameters();
            String paraName;
            for (Parameter para : dbParameters) {
                // Only print out information about either auto_increment_offset
or
                // auto_increment_increment.
```

```
                paraName = para.parameterName();
                if ((paraName.compareTo("auto_increment_offset") == 0)
                        || (paraName.compareTo("auto_increment_increment ") ==
0)) {
                    System.out.println("*** The parameter name is  " + paraName);
                    System.out.println("*** The parameter value is  " +
para.parameterValue());
                    System.out.println("*** The parameter data type is " +
para.dataType());
                    System.out.println("*** The parameter description is " +
para.description());
                    System.out.println("*** The parameter allowed values  is " +
para.allowedValues());
                }
            }

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
    }

    public static void describeDbParameterGroups(RdsClient rdsClient, String
dbGroupName) {
        try {
            DescribeDbParameterGroupsRequest groupsRequest =
DescribeDbParameterGroupsRequest.builder()
                    .dbParameterGroupName(dbGroupName)
                    .maxRecords(20)
                    .build();

            DescribeDbParameterGroupsResponse response =
rdsClient.describeDBParameterGroups(groupsRequest);
            List<DBParameterGroup> groups = response.dbParameterGroups();
            for (DBParameterGroup group : groups) {
                System.out.println("The group name is " +
group.dbParameterGroupName());
                System.out.println("The group description is " +
group.description());
            }

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
```

```
        }
    }

    public static void createDBParameterGroup(RdsClient rdsClient, String
dbGroupName, String dbParameterGroupFamily) {
        try {
            CreateDbParameterGroupRequest groupRequest =
CreateDbParameterGroupRequest.builder()
                    .dbParameterGroupName(dbGroupName)
                    .dbParameterGroupFamily(dbParameterGroupFamily)
                    .description("Created by using the AWS SDK for Java")
                    .build();

            CreateDbParameterGroupResponse response =
rdsClient.createDBParameterGroup(groupRequest);
            System.out.println("The group name is " +
response.dbParameterGroup().dbParameterGroupName());

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
    }

    public static void describeDBEngines(RdsClient rdsClient) {
        try {
            DescribeDbEngineVersionsRequest engineVersionsRequest =
DescribeDbEngineVersionsRequest.builder()
                    .defaultOnly(true)
                    .engine("mysql")
                    .maxRecords(20)
                    .build();

            DescribeDbEngineVersionsResponse response =
rdsClient.describeDBEngineVersions(engineVersionsRequest);
            List<DBEngineVersion> engines = response.dbEngineVersions();

            // Get all DBEngineVersion objects.
            for (DBEngineVersion engineOb : engines) {
                System.out.println("The name of the DB parameter group family for
the database engine is "
                        + engineOb.dbParameterGroupFamily());
                System.out.println("The name of the database engine " +
engineOb.engine());
```

```
                System.out.println("The version number of the database engine " +
 engineOb.engineVersion());
            }

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
    }
}
```

- Para obter detalhes da API, consulte os tópicos a seguir na Referência da API AWS SDK for Java 2.x.
  - CreateDBInstance
  - CreateDBParameterGroup
  - CreateDBSnapshot
  - DeleteDBInstance
  - DeleteDBParameterGroup
  - DescribeDBEngineVersions
  - DescribeDBInstances
  - DescribeDBParameterGroups
  - DescribeDBParameters
  - DescribeDBSnapshots
  - DescribeOrderableDBInstanceOptions
  - ModifyDBParameterGroup

Kotlin

SDK para Kotlin

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
/**
Before running this code example, set up your development environment, including
 your credentials.

For more information, see the following documentation topic:

https://docs.aws.amazon.com/sdk-for-kotlin/latest/developer-guide/setup.html

This example requires an AWS Secrets Manager secret that contains the database
 credentials. If you do not create a
secret, this example will not work. For more details, see:

https://docs.aws.amazon.com/secretsmanager/latest/userguide/integrating_how-
services-use-secrets_RS.html

This example performs the following tasks:

1. Returns a list of the available DB engines by invoking the
 DescribeDbEngineVersions method.
2. Selects an engine family and create a custom DB parameter group by invoking
 the createDBParameterGroup method.
3. Gets the parameter groups by invoking the DescribeDbParameterGroups method.
4. Gets parameters in the group by invoking the DescribeDbParameters method.
5. Modifies both the auto_increment_offset and auto_increment_increment
 parameters by invoking the modifyDbParameterGroup method.
6. Gets and displays the updated parameters.
7. Gets a list of allowed engine versions by invoking the
 describeDbEngineVersions method.
8. Gets a list of micro instance classes available for the selected engine.
9. Creates an Amazon Relational Database Service (Amazon RDS) database instance
 that contains a MySQL database and uses the parameter group.
10. Waits for DB instance to be ready and prints out the connection endpoint
 value.
11. Creates a snapshot of the DB instance.
12. Waits for the DB snapshot to be ready.
13. Deletes the DB instance.
14. Deletes the parameter group.
 */

var sleepTime: Long = 20

suspend fun main(args: Array<String>) {
```

```
   val usage = """
       Usage:
           <dbGroupName> <dbParameterGroupFamily> <dbInstanceIdentifier>
<dbName> <dbSnapshotIdentifier><secretName>

       Where:
           dbGroupName - The database group name.
           dbParameterGroupFamily - The database parameter group name.
           dbInstanceIdentifier - The database instance identifier.
           dbName -  The database name.
           dbSnapshotIdentifier - The snapshot identifier.
           secretName - The name of the AWS Secrets Manager secret that contains
the database credentials.
   """

   if (args.size != 6) {
       println(usage)
       exitProcess(1)
   }

   val dbGroupName = args[0]
   val dbParameterGroupFamily = args[1]
   val dbInstanceIdentifier = args[2]
   val dbName = args[3]
   val dbSnapshotIdentifier = args[4]
   val secretName = args[5]

   val gson = Gson()
   val user = gson.fromJson(getSecretValues(secretName).toString(),
User::class.java)
   val username = user.username
   val userPassword = user.password

   println("1. Return a list of the available DB engines")
   describeDBEngines()

   println("2. Create a custom parameter group")
   createDBParameterGroup(dbGroupName, dbParameterGroupFamily)

   println("3. Get the parameter groups")
   describeDbParameterGroups(dbGroupName)

   println("4. Get the parameters in the group")
   describeDbParameters(dbGroupName, 0)
```

```
    println("5. Modify the auto_increment_offset parameter")
    modifyDBParas(dbGroupName)

    println("6. Display the updated value")
    describeDbParameters(dbGroupName, -1)

    println("7. Get a list of allowed engine versions")
    getAllowedEngines(dbParameterGroupFamily)

    println("8. Get a list of micro instance classes available for the selected
 engine")
    getMicroInstances()

    println("9. Create an RDS database instance that contains a MySql database
 and uses the parameter group")
    val dbARN = createDatabaseInstance(dbGroupName, dbInstanceIdentifier, dbName,
 username, userPassword)
    println("The ARN of the new database is $dbARN")

    println("10. Wait for DB instance to be ready")
    waitForDbInstanceReady(dbInstanceIdentifier)

    println("11. Create a snapshot of the DB instance")
    createDbSnapshot(dbInstanceIdentifier, dbSnapshotIdentifier)

    println("12. Wait for DB snapshot to be ready")
    waitForSnapshotReady(dbInstanceIdentifier, dbSnapshotIdentifier)

    println("13. Delete the DB instance")
    deleteDbInstance(dbInstanceIdentifier)

    println("14. Delete the parameter group")
    if (dbARN != null) {
        deleteParaGroup(dbGroupName, dbARN)
    }

    println("The Scenario has successfully completed.")
}

suspend fun deleteParaGroup(
    dbGroupName: String,
    dbARN: String,
) {
```

```
    var isDataDel = false
    var didFind: Boolean
    var instanceARN: String

    RdsClient { region = "us-west-2" }.use { rdsClient ->
        // Make sure that the database has been deleted.
        while (!isDataDel) {
            val response = rdsClient.describeDbInstances()
            val instanceList = response.dbInstances
            val listSize = instanceList?.size
            isDataDel = false // Reset this value.
            didFind = false // Reset this value.
            var index = 1
            if (instanceList != null) {
                for (instance in instanceList) {
                    instanceARN = instance.dbInstanceArn.toString()
                    if (instanceARN.compareTo(dbARN) == 0) {
                        println("$dbARN still exists")
                        didFind = true
                    }
                    if (index == listSize && !didFind) {
                        // Went through the entire list and did not find the
 database name.
                        isDataDel = true
                    }
                    index++
                }
            }
        }

        // Delete the para group.
        val parameterGroupRequest =
            DeleteDbParameterGroupRequest {
                dbParameterGroupName = dbGroupName
            }
        rdsClient.deleteDbParameterGroup(parameterGroupRequest)
        println("$dbGroupName was deleted.")
    }
}

suspend fun deleteDbInstance(dbInstanceIdentifierVal: String) {
    val deleteDbInstanceRequest =
        DeleteDbInstanceRequest {
            dbInstanceIdentifier = dbInstanceIdentifierVal
```

```
            deleteAutomatedBackups = true
            skipFinalSnapshot = true
        }

    RdsClient { region = "us-west-2" }.use { rdsClient ->
        val response = rdsClient.deleteDbInstance(deleteDbInstanceRequest)
        print("The status of the database is
 ${response.dbInstance?.dbInstanceStatus}")
    }
}

// Waits until the snapshot instance is available.
suspend fun waitForSnapshotReady(
    dbInstanceIdentifierVal: String?,
    dbSnapshotIdentifierVal: String?,
) {
    var snapshotReady = false
    var snapshotReadyStr: String
    println("Waiting for the snapshot to become available.")

    val snapshotsRequest =
        DescribeDbSnapshotsRequest {
            dbSnapshotIdentifier = dbSnapshotIdentifierVal
            dbInstanceIdentifier = dbInstanceIdentifierVal
        }

    while (!snapshotReady) {
        RdsClient { region = "us-west-2" }.use { rdsClient ->
            val response = rdsClient.describeDbSnapshots(snapshotsRequest)
            val snapshotList: List<DbSnapshot>? = response.dbSnapshots
            if (snapshotList != null) {
                for (snapshot in snapshotList) {
                    snapshotReadyStr = snapshot.status.toString()
                    if (snapshotReadyStr.contains("available")) {
                        snapshotReady = true
                    } else {
                        print(".")
                        delay(sleepTime * 1000)
                    }
                }
            }
        }
    }
    println("The Snapshot is available!")
```

```
}

// Create an Amazon RDS snapshot.
suspend fun createDbSnapshot(
    dbInstanceIdentifierVal: String?,
    dbSnapshotIdentifierVal: String?,
) {
    val snapshotRequest =
        CreateDbSnapshotRequest {
            dbInstanceIdentifier = dbInstanceIdentifierVal
            dbSnapshotIdentifier = dbSnapshotIdentifierVal
        }

    RdsClient { region = "us-west-2" }.use { rdsClient ->
        val response = rdsClient.createDbSnapshot(snapshotRequest)
        print("The Snapshot id is ${response.dbSnapshot?.dbiResourceId}")
    }
}

// Waits until the database instance is available.
suspend fun waitForDbInstanceReady(dbInstanceIdentifierVal: String?) {
    var instanceReady = false
    var instanceReadyStr: String
    println("Waiting for instance to become available.")

    val instanceRequest =
        DescribeDbInstancesRequest {
            dbInstanceIdentifier = dbInstanceIdentifierVal
        }
    var endpoint = ""
    while (!instanceReady) {
        RdsClient { region = "us-west-2" }.use { rdsClient ->
            val response = rdsClient.describeDbInstances(instanceRequest)
            val instanceList = response.dbInstances
            if (instanceList != null) {
                for (instance in instanceList) {
                    instanceReadyStr = instance.dbInstanceStatus.toString()
                    if (instanceReadyStr.contains("available")) {
                        endpoint = instance.endpoint?.address.toString()
                        instanceReady = true
                    } else {
                        print(".")
                        delay(sleepTime * 1000)
                    }
```

```
                }
            }
        }
    }
    println("Database instance is available! The connection endpoint is
 $endpoint")
}

// Create a database instance and return the ARN of the database.
suspend fun createDatabaseInstance(
    dbGroupNameVal: String?,
    dbInstanceIdentifierVal: String?,
    dbNameVal: String?,
    masterUsernameVal: String?,
    masterUserPasswordVal: String?,
): String? {
    val instanceRequest =
        CreateDbInstanceRequest {
            dbInstanceIdentifier = dbInstanceIdentifierVal
            allocatedStorage = 100
            dbName = dbNameVal
            dbParameterGroupName = dbGroupNameVal
            engine = "mysql"
            dbInstanceClass = "db.m4.large"
            engineVersion = "8.0"
            storageType = "standard"
            masterUsername = masterUsernameVal
            masterUserPassword = masterUserPasswordVal
        }

    RdsClient { region = "us-west-2" }.use { rdsClient ->
        val response = rdsClient.createDbInstance(instanceRequest)
        print("The status is ${response.dbInstance?.dbInstanceStatus}")
        return response.dbInstance?.dbInstanceArn
    }
}

// Get a list of micro instances.
suspend fun getMicroInstances() {
    val dbInstanceOptionsRequest =
        DescribeOrderableDbInstanceOptionsRequest {
            engine = "mysql"
        }
    RdsClient { region = "us-west-2" }.use { rdsClient ->
```

```
        val response =
 rdsClient.describeOrderableDbInstanceOptions(dbInstanceOptionsRequest)
        val orderableDBInstances = response.orderableDbInstanceOptions
        if (orderableDBInstances != null) {
            for (dbInstanceOption in orderableDBInstances) {
                println("The engine version is
 ${dbInstanceOption.engineVersion}")
                println("The engine description is ${dbInstanceOption.engine}")
            }
        }
    }
}

// Get a list of allowed engine versions.
suspend fun getAllowedEngines(dbParameterGroupFamilyVal: String?) {
    val versionsRequest =
        DescribeDbEngineVersionsRequest {
            dbParameterGroupFamily = dbParameterGroupFamilyVal
            engine = "mysql"
        }
    RdsClient { region = "us-west-2" }.use { rdsClient ->
        val response = rdsClient.describeDbEngineVersions(versionsRequest)
        val dbEngines: List<DbEngineVersion>? = response.dbEngineVersions
        if (dbEngines != null) {
            for (dbEngine in dbEngines) {
                println("The engine version is ${dbEngine.engineVersion}")
                println("The engine description is
 ${dbEngine.dbEngineDescription}")
            }
        }
    }
}

// Modify the auto_increment_offset parameter.
suspend fun modifyDBParas(dbGroupName: String) {
    val parameter1 =
        Parameter {
            parameterName = "auto_increment_offset"
            applyMethod = ApplyMethod.Immediate
            parameterValue = "5"
        }

    val paraList: ArrayList<Parameter> = ArrayList()
    paraList.add(parameter1)
```

```
    val groupRequest =
        ModifyDbParameterGroupRequest {
            dbParameterGroupName = dbGroupName
            parameters = paraList
        }

    RdsClient { region = "us-west-2" }.use { rdsClient ->
        val response = rdsClient.modifyDbParameterGroup(groupRequest)
        println("The parameter group ${response.dbParameterGroupName} was
 successfully modified")
    }
}

// Retrieve parameters in the group.
suspend fun describeDbParameters(
    dbGroupName: String?,
    flag: Int,
) {
    val dbParameterGroupsRequest: DescribeDbParametersRequest
    dbParameterGroupsRequest =
        if (flag == 0) {
            DescribeDbParametersRequest {
                dbParameterGroupName = dbGroupName
            }
        } else {
            DescribeDbParametersRequest {
                dbParameterGroupName = dbGroupName
                source = "user"
            }
        }
    RdsClient { region = "us-west-2" }.use { rdsClient ->
        val response = rdsClient.describeDbParameters(dbParameterGroupsRequest)
        val dbParameters: List<Parameter>? = response.parameters
        var paraName: String
        if (dbParameters != null) {
            for (para in dbParameters) {
                // Only print out information about either auto_increment_offset
 or auto_increment_increment.
                paraName = para.parameterName.toString()
                if (paraName.compareTo("auto_increment_offset") == 0 ||
 paraName.compareTo("auto_increment_increment ") == 0) {
                    println("*** The parameter name is  $paraName")
                    System.out.println("*** The parameter value is
 ${para.parameterValue}")
```

```
                    System.out.println("*** The parameter data type is
 ${para.dataType}")
                    System.out.println("*** The parameter description is
 ${para.description}")
                    System.out.println("*** The parameter allowed values  is
 ${para.allowedValues}")
                }
            }
        }
    }
}

suspend fun describeDbParameterGroups(dbGroupName: String?) {
    val groupsRequest =
        DescribeDbParameterGroupsRequest {
            dbParameterGroupName = dbGroupName
            maxRecords = 20
        }
    RdsClient { region = "us-west-2" }.use { rdsClient ->
        val response = rdsClient.describeDbParameterGroups(groupsRequest)
        val groups = response.dbParameterGroups
        if (groups != null) {
            for (group in groups) {
                println("The group name is ${group.dbParameterGroupName}")
                println("The group description is ${group.description}")
            }
        }
    }
}

// Create a parameter group.
suspend fun createDBParameterGroup(
    dbGroupName: String?,
    dbParameterGroupFamilyVal: String?,
) {
    val groupRequest =
        CreateDbParameterGroupRequest {
            dbParameterGroupName = dbGroupName
            dbParameterGroupFamily = dbParameterGroupFamilyVal
            description = "Created by using the AWS SDK for Kotlin"
        }

    RdsClient { region = "us-west-2" }.use { rdsClient ->
        val response = rdsClient.createDbParameterGroup(groupRequest)
```

```
        println("The group name is
 ${response.dbParameterGroup?.dbParameterGroupName}")
    }
}

// Returns a list of the available DB engines.
suspend fun describeDBEngines() {
    val engineVersionsRequest =
        DescribeDbEngineVersionsRequest {
            defaultOnly = true
            engine = "mysql"
            maxRecords = 20
        }

    RdsClient { region = "us-west-2" }.use { rdsClient ->
        val response = rdsClient.describeDbEngineVersions(engineVersionsRequest)
        val engines: List<DbEngineVersion>? = response.dbEngineVersions

        // Get all DbEngineVersion objects.
        if (engines != null) {
            for (engineOb in engines) {
                println("The name of the DB parameter group family for the
 database engine is ${engineOb.dbParameterGroupFamily}.")
                println("The name of the database engine ${engineOb.engine}.")
                println("The version number of the database engine
 ${engineOb.engineVersion}")
            }
        }
    }
}

suspend fun getSecretValues(secretName: String?): String? {
    val valueRequest =
        GetSecretValueRequest {
            secretId = secretName
        }

    SecretsManagerClient { region = "us-west-2" }.use { secretsClient ->
        val valueResponse = secretsClient.getSecretValue(valueRequest)
        return valueResponse.secretString
    }
}
```

- Para obter detalhes da API, consulte os tópicos a seguir na Referência da API AWS SDK para Kotlin.
  - CreateDBInstance
  - CreateDBParameterGroup
  - CreateDBSnapshot
  - DeleteDBInstance
  - DeleteDBParameterGroup
  - DescribeDBEngineVersions
  - DescribeDBInstances
  - DescribeDBParameterGroups
  - DescribeDBParameters
  - DescribeDBSnapshots
  - DescribeOrderableDBInstanceOptions
  - ModifyDBParameterGroup

Python

SDK para Python (Boto3).

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

Execute um cenário interativo em um prompt de comando.

```
class RdsInstanceScenario:
    """Runs a scenario that shows how to get started using Amazon RDS DB
 instances."""

    def __init__(self, instance_wrapper):
        """
        :param instance_wrapper: An object that wraps Amazon RDS DB instance
 actions.
        """
```

```
        self.instance_wrapper = instance_wrapper

    def create_parameter_group(self, parameter_group_name, db_engine):
        """
        Shows how to get available engine versions for a specified database
engine and
        create a DB parameter group that is compatible with a selected engine
family.

        :param parameter_group_name: The name given to the newly created
parameter group.
        :param db_engine: The database engine to use as a basis.
        :return: The newly created parameter group.
        """
        print(
            f"Checking for an existing DB instance parameter group named
{parameter_group_name}."
        )
        parameter_group = self.instance_wrapper.get_parameter_group(
            parameter_group_name
        )
        if parameter_group is None:
            print(f"Getting available database engine versions for {db_engine}.")
            engine_versions =
self.instance_wrapper.get_engine_versions(db_engine)
            families = list({ver["DBParameterGroupFamily"] for ver in
engine_versions})
            family_index = q.choose("Which family do you want to use? ",
families)
            print(f"Creating a parameter group.")
            self.instance_wrapper.create_parameter_group(
                parameter_group_name, families[family_index], "Example parameter
group."
            )
            parameter_group = self.instance_wrapper.get_parameter_group(
                parameter_group_name
            )
        print(f"Parameter group {parameter_group['DBParameterGroupName']}:")
        pp(parameter_group)
        print("-" * 88)
        return parameter_group

    def update_parameters(self, parameter_group_name):
        """
```

```
        Shows how to get the parameters contained in a custom parameter group and
        update some of the parameter values in the group.

        :param parameter_group_name: The name of the parameter group to query and
modify.
        """
        print("Let's set some parameter values in your parameter group.")
        auto_inc_parameters = self.instance_wrapper.get_parameters(
            parameter_group_name, name_prefix="auto_increment"
        )
        update_params = []
        for auto_inc in auto_inc_parameters:
            if auto_inc["IsModifiable"] and auto_inc["DataType"] == "integer":
                print(f"The {auto_inc['ParameterName']} parameter is described
as:")
                print(f"\t{auto_inc['Description']}")
                param_range = auto_inc["AllowedValues"].split("-")
                auto_inc["ParameterValue"] = str(
                    q.ask(
                        f"Enter a value between {param_range[0]} and
{param_range[1]}: ",
                        q.is_int,
                        q.in_range(int(param_range[0]), int(param_range[1])),
                    )
                )
                update_params.append(auto_inc)
        self.instance_wrapper.update_parameters(parameter_group_name,
update_params)
        print(
            "You can get a list of parameters you've set by specifying a source
of 'user'."
        )
        user_parameters = self.instance_wrapper.get_parameters(
            parameter_group_name, source="user"
        )
        pp(user_parameters)
        print("-" * 88)

    def create_instance(self, instance_name, db_name, db_engine,
parameter_group):
        """
        Shows how to create a DB instance that contains a database of a specified
        type and is configured to use a custom DB parameter group.
```

```
        :param instance_name: The name given to the newly created DB instance.
        :param db_name: The name given to the created database.
        :param db_engine: The engine of the created database.
        :param parameter_group: The parameter group that is associated with the
DB instance.
        :return: The newly created DB instance.
        """
        print("Checking for an existing DB instance.")
        db_inst = self.instance_wrapper.get_db_instance(instance_name)
        if db_inst is None:
            print("Let's create a DB instance.")
            admin_username = q.ask(
                "Enter an administrator user name for the database: ",
q.non_empty
            )
            admin_password = q.ask(
                "Enter a password for the administrator (at least 8 characters):
",
                q.non_empty,
            )
            engine_versions = self.instance_wrapper.get_engine_versions(
                db_engine, parameter_group["DBParameterGroupFamily"]
            )
            engine_choices = [ver["EngineVersion"] for ver in engine_versions]
            print("The available engines for your parameter group are:")
            engine_index = q.choose("Which engine do you want to use? ",
engine_choices)
            engine_selection = engine_versions[engine_index]
            print(
                "The available micro DB instance classes for your database engine
are:"
            )
            inst_opts = self.instance_wrapper.get_orderable_instances(
                engine_selection["Engine"], engine_selection["EngineVersion"]
            )
            inst_choices = list(
                {
                    opt["DBInstanceClass"]
                    for opt in inst_opts
                    if "micro" in opt["DBInstanceClass"]
                }
            )
            inst_index = q.choose(
```

```
                "Which micro DB instance class do you want to use? ",
inst_choices
            )
            group_name = parameter_group["DBParameterGroupName"]
            storage_type = "standard"
            allocated_storage = 5
            print(
                f"Creating a DB instance named {instance_name} and database
{db_name}.\n"
                f"The DB instance is configured to use your custom parameter
group {group_name},\n"
                f"selected engine {engine_selection['EngineVersion']},\n"
                f"selected DB instance class {inst_choices[inst_index]},"
                f"and {allocated_storage} GiB of {storage_type} storage.\n"
                f"This typically takes several minutes."
            )
            db_inst = self.instance_wrapper.create_db_instance(
                db_name,
                instance_name,
                group_name,
                engine_selection["Engine"],
                engine_selection["EngineVersion"],
                inst_choices[inst_index],
                storage_type,
                allocated_storage,
                admin_username,
                admin_password,
            )
            while db_inst.get("DBInstanceStatus") != "available":
                wait(10)
                db_inst = self.instance_wrapper.get_db_instance(instance_name)
        print("Instance data:")
        pp(db_inst)
        print("-" * 88)
        return db_inst

    @staticmethod
    def display_connection(db_inst):
        """
        Displays connection information about a DB instance and tips on how to
        connect to it.

        :param db_inst: The DB instance to display.
        """
```

```
        print(
            "You can now connect to your database using your favorite MySql
 client.\n"
            "One way to connect is by using the 'mysql' shell on an Amazon EC2
 instance\n"
            "that is running in the same VPC as your DB instance. Pass the
 endpoint,\n"
            "port, and administrator user name to 'mysql' and enter your password
\n"
            "when prompted:\n"
        )
        print(
            f"\n\tmysql -h {db_inst['Endpoint']['Address']} -P
 {db_inst['Endpoint']['Port']} "
            f"-u {db_inst['MasterUsername']} -p\n"
        )
        print(
            "For more information, see the User Guide for Amazon RDS:\n"
            "\thttps://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/
CHAP_GettingStarted.CreatingConnecting.MySQL.html#CHAP_GettingStarted.Connecting.MySQL"
        )
        print("-" * 88)

    def create_snapshot(self, instance_name):
        """
        Shows how to create a DB instance snapshot and wait until it's available.

        :param instance_name: The name of a DB instance to snapshot.
        """
        if q.ask(
            "Do you want to create a snapshot of your DB instance (y/n)? ",
 q.is_yesno
        ):
            snapshot_id = f"{instance_name}-{uuid.uuid4()}"
            print(
                f"Creating a snapshot named {snapshot_id}. This typically takes a
 few minutes."
            )
            snapshot = self.instance_wrapper.create_snapshot(snapshot_id,
 instance_name)
            while snapshot.get("Status") != "available":
                wait(10)
                snapshot = self.instance_wrapper.get_snapshot(snapshot_id)
            pp(snapshot)
```

```
            print("-" * 88)

    def cleanup(self, db_inst, parameter_group_name):
        """
        Shows how to clean up a DB instance and parameter group.
        Before the parameter group can be deleted, all associated DB instances
must first
        be deleted.

        :param db_inst: The DB instance to delete.
        :param parameter_group_name: The DB parameter group to delete.
        """
        if q.ask(
            "\nDo you want to delete the DB instance and parameter group (y/n)?
",
            q.is_yesno,
        ):
            print(f"Deleting DB instance {db_inst['DBInstanceIdentifier']}.")

self.instance_wrapper.delete_db_instance(db_inst["DBInstanceIdentifier"])
            print(
                "Waiting for the DB instance to delete. This typically takes
several minutes."
            )
            while db_inst is not None:
                wait(10)
                db_inst = self.instance_wrapper.get_db_instance(
                    db_inst["DBInstanceIdentifier"]
                )
            print(f"Deleting parameter group {parameter_group_name}.")
            self.instance_wrapper.delete_parameter_group(parameter_group_name)

    def run_scenario(self, db_engine, parameter_group_name, instance_name,
db_name):
        logging.basicConfig(level=logging.INFO, format="%(levelname)s:
%(message)s")

        print("-" * 88)
        print(
            "Welcome to the Amazon Relational Database Service (Amazon RDS)\n"
            "get started with DB instances demo."
        )
        print("-" * 88)
```

```
        parameter_group = self.create_parameter_group(parameter_group_name,
 db_engine)
        self.update_parameters(parameter_group_name)
        db_inst = self.create_instance(
            instance_name, db_name, db_engine, parameter_group
        )
        self.display_connection(db_inst)
        self.create_snapshot(instance_name)
        self.cleanup(db_inst, parameter_group_name)

        print("\nThanks for watching!")
        print("-" * 88)


if __name__ == "__main__":
    try:
        scenario = RdsInstanceScenario(InstanceWrapper.from_client())
        scenario.run_scenario(
            "mysql",
            "doc-example-parameter-group",
            "doc-example-instance",
            "docexampledb",
        )
    except Exception:
        logging.exception("Something went wrong with the demo.")
```

Defina as funções que são chamadas pelo cenário para gerenciar as ações do Amazon RDS.

```
class InstanceWrapper:
    """Encapsulates Amazon RDS DB instance actions."""

    def __init__(self, rds_client):
        """
        :param rds_client: A Boto3 Amazon RDS client.
        """
        self.rds_client = rds_client

    @classmethod
    def from_client(cls):
        """
        Instantiates this class from a Boto3 client.
        """
```

```
        rds_client = boto3.client("rds")
        return cls(rds_client)


    def get_parameter_group(self, parameter_group_name):
        """
        Gets a DB parameter group.

        :param parameter_group_name: The name of the parameter group to retrieve.
        :return: The parameter group.
        """
        try:
            response = self.rds_client.describe_db_parameter_groups(
                DBParameterGroupName=parameter_group_name
            )
            parameter_group = response["DBParameterGroups"][0]
        except ClientError as err:
            if err.response["Error"]["Code"] == "DBParameterGroupNotFound":
                logger.info("Parameter group %s does not exist.",
parameter_group_name)
            else:
                logger.error(
                    "Couldn't get parameter group %s. Here's why: %s: %s",
                    parameter_group_name,
                    err.response["Error"]["Code"],
                    err.response["Error"]["Message"],
                )
                raise
        else:
            return parameter_group


    def create_parameter_group(
        self, parameter_group_name, parameter_group_family, description
    ):
        """
        Creates a DB parameter group that is based on the specified parameter
group
        family.

        :param parameter_group_name: The name of the newly created parameter
group.
        :param parameter_group_family: The family that is used as the basis of
the new
```

```
                                        parameter group.
        :param description: A description given to the parameter group.
        :return: Data about the newly created parameter group.
        """
        try:
            response = self.rds_client.create_db_parameter_group(
                DBParameterGroupName=parameter_group_name,
                DBParameterGroupFamily=parameter_group_family,
                Description=description,
            )
        except ClientError as err:
            logger.error(
                "Couldn't create parameter group %s. Here's why: %s: %s",
                parameter_group_name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return response


    def delete_parameter_group(self, parameter_group_name):
        """
        Deletes a DB parameter group.

        :param parameter_group_name: The name of the parameter group to delete.
        :return: Data about the parameter group.
        """
        try:
            self.rds_client.delete_db_parameter_group(
                DBParameterGroupName=parameter_group_name
            )
        except ClientError as err:
            logger.error(
                "Couldn't delete parameter group %s. Here's why: %s: %s",
                parameter_group_name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise


    def get_parameters(self, parameter_group_name, name_prefix="", source=None):
```

```
        """
        Gets the parameters that are contained in a DB parameter group.

        :param parameter_group_name: The name of the parameter group to query.
        :param name_prefix: When specified, the retrieved list of parameters is
filtered
                            to contain only parameters that start with this
prefix.
        :param source: When specified, only parameters from this source are
retrieved.
                       For example, a source of 'user' retrieves only parameters
that
                       were set by a user.
        :return: The list of requested parameters.
        """
        try:
            kwargs = {"DBParameterGroupName": parameter_group_name}
            if source is not None:
                kwargs["Source"] = source
            parameters = []
            paginator = self.rds_client.get_paginator("describe_db_parameters")
            for page in paginator.paginate(**kwargs):
                parameters += [
                    p
                    for p in page["Parameters"]
                    if p["ParameterName"].startswith(name_prefix)
                ]
        except ClientError as err:
            logger.error(
                "Couldn't get parameters for %s. Here's why: %s: %s",
                parameter_group_name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return parameters


    def update_parameters(self, parameter_group_name, update_parameters):
        """
        Updates parameters in a custom DB parameter group.

        :param parameter_group_name: The name of the parameter group to update.
```

```
        :param update_parameters: The parameters to update in the group.
        :return: Data about the modified parameter group.
        """
        try:
            response = self.rds_client.modify_db_parameter_group(
                DBParameterGroupName=parameter_group_name,
Parameters=update_parameters
            )
        except ClientError as err:
            logger.error(
                "Couldn't update parameters in %s. Here's why: %s: %s",
                parameter_group_name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return response


    def create_snapshot(self, snapshot_id, instance_id):
        """
        Creates a snapshot of a DB instance.

        :param snapshot_id: The ID to give the created snapshot.
        :param instance_id: The ID of the DB instance to snapshot.
        :return: Data about the newly created snapshot.
        """
        try:
            response = self.rds_client.create_db_snapshot(
                DBSnapshotIdentifier=snapshot_id,
DBInstanceIdentifier=instance_id
            )
            snapshot = response["DBSnapshot"]
        except ClientError as err:
            logger.error(
                "Couldn't create snapshot of %s. Here's why: %s: %s",
                instance_id,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return snapshot
```

```
    def get_snapshot(self, snapshot_id):
        """
        Gets a DB instance snapshot.

        :param snapshot_id: The ID of the snapshot to retrieve.
        :return: The retrieved snapshot.
        """
        try:
            response = self.rds_client.describe_db_snapshots(
                DBSnapshotIdentifier=snapshot_id
            )
            snapshot = response["DBSnapshots"][0]
        except ClientError as err:
            logger.error(
                "Couldn't get snapshot %s. Here's why: %s: %s",
                snapshot_id,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return snapshot


    def get_engine_versions(self, engine, parameter_group_family=None):
        """
        Gets database engine versions that are available for the specified engine
        and parameter group family.

        :param engine: The database engine to look up.
        :param parameter_group_family: When specified, restricts the returned
list of
                                       engine versions to those that are
compatible with
                                       this parameter group family.
        :return: The list of database engine versions.
        """
        try:
            kwargs = {"Engine": engine}
            if parameter_group_family is not None:
                kwargs["DBParameterGroupFamily"] = parameter_group_family
            response = self.rds_client.describe_db_engine_versions(**kwargs)
```

```
            versions = response["DBEngineVersions"]
        except ClientError as err:
            logger.error(
                "Couldn't get engine versions for %s. Here's why: %s: %s",
                engine,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return versions


    def get_orderable_instances(self, db_engine, db_engine_version):
        """
        Gets DB instance options that can be used to create DB instances that are
        compatible with a set of specifications.

        :param db_engine: The database engine that must be supported by the DB
instance.
        :param db_engine_version: The engine version that must be supported by
the DB instance.
        :return: The list of DB instance options that can be used to create a
compatible DB instance.
        """
        try:
            inst_opts = []
            paginator = self.rds_client.get_paginator(
                "describe_orderable_db_instance_options"
            )
            for page in paginator.paginate(
                Engine=db_engine, EngineVersion=db_engine_version
            ):
                inst_opts += page["OrderableDBInstanceOptions"]
        except ClientError as err:
            logger.error(
                "Couldn't get orderable DB instances. Here's why: %s: %s",
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return inst_opts
```

```
    def get_db_instance(self, instance_id):
        """
        Gets data about a DB instance.

        :param instance_id: The ID of the DB instance to retrieve.
        :return: The retrieved DB instance.
        """
        try:
            response = self.rds_client.describe_db_instances(
                DBInstanceIdentifier=instance_id
            )
            db_inst = response["DBInstances"][0]
        except ClientError as err:
            if err.response["Error"]["Code"] == "DBInstanceNotFound":
                logger.info("Instance %s does not exist.", instance_id)
            else:
                logger.error(
                    "Couldn't get DB instance %s. Here's why: %s: %s",
                    instance_id,
                    err.response["Error"]["Code"],
                    err.response["Error"]["Message"],
                )
                raise
        else:
            return db_inst


    def create_db_instance(
        self,
        db_name,
        instance_id,
        parameter_group_name,
        db_engine,
        db_engine_version,
        instance_class,
        storage_type,
        allocated_storage,
        admin_name,
        admin_password,
    ):
        """
        Creates a DB instance.
```

```
        :param db_name: The name of the database that is created in the DB
instance.
        :param instance_id: The ID to give the newly created DB instance.
        :param parameter_group_name: A parameter group to associate with the DB
instance.
        :param db_engine: The database engine of a database to create in the DB
instance.
        :param db_engine_version: The engine version for the created database.
        :param instance_class: The DB instance class for the newly created DB
instance.
        :param storage_type: The storage type of the DB instance.
        :param allocated_storage: The amount of storage allocated on the DB
instance, in GiBs.
        :param admin_name: The name of the admin user for the created database.
        :param admin_password: The admin password for the created database.
        :return: Data about the newly created DB instance.
        """
        try:
            response = self.rds_client.create_db_instance(
                DBName=db_name,
                DBInstanceIdentifier=instance_id,
                DBParameterGroupName=parameter_group_name,
                Engine=db_engine,
                EngineVersion=db_engine_version,
                DBInstanceClass=instance_class,
                StorageType=storage_type,
                AllocatedStorage=allocated_storage,
                MasterUsername=admin_name,
                MasterUserPassword=admin_password,
            )
            db_inst = response["DBInstance"]
        except ClientError as err:
            logger.error(
                "Couldn't create DB instance %s. Here's why: %s: %s",
                instance_id,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return db_inst


    def delete_db_instance(self, instance_id):
```

```
        """
        Deletes a DB instance.

        :param instance_id: The ID of the DB instance to delete.
        :return: Data about the deleted DB instance.
        """
        try:
            response = self.rds_client.delete_db_instance(
                DBInstanceIdentifier=instance_id,
                SkipFinalSnapshot=True,
                DeleteAutomatedBackups=True,
            )
            db_inst = response["DBInstance"]
        except ClientError as err:
            logger.error(
                "Couldn't delete DB instance %s. Here's why: %s: %s",
                instance_id,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return db_inst
```

- Para obter detalhes da API, consulte os tópicos a seguir na Referência da API AWS SDK para Python (Boto3).
  - CreateDBInstance
  - CreateDBParameterGroup
  - CreateDBSnapshot
  - DeleteDBInstance
  - DeleteDBParameterGroup
  - DescribeDBEngineVersions
  - DescribeDBInstances
  - DescribeDBParameterGroups
  - DescribeDBParameters

- DescribeDBSnapshots
- DescribeOrderableDBInstanceOptions
- ModifyDBParameterGroup

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Exemplos de tecnologia servidor para o Amazon RDS usando SDKs da AWS

Os exemplos de código a seguir mostram como usar o Amazon RDS com SDKs da AWS.

Exemplos

- Como se conectar a um banco de dados do Amazon RDS em uma função do Lambda

Como se conectar a um banco de dados do Amazon RDS em uma função do Lambda

Os exemplos de código a seguir mostram como implementar uma função do Lambda que se conecte a um banco de dados do RDS. A função faz uma solicitação simples ao banco de dados e exibe o resultado.

Go

SDK para Go V2

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no repositório dos Exemplos sem servidor.

Conectar-se a um banco de dados do Amazon RDS em uma função do Lambda usando Go.

```
// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
```

```
// SPDX-License-Identifier: Apache-2.0
/*
Golang v2 code here.
*/

package main

import (
 "context"
 "database/sql"
 "encoding/json"
 "fmt"
 "os"

 "github.com/aws/aws-lambda-go/lambda"
 "github.com/aws/aws-sdk-go-v2/config"
 "github.com/aws/aws-sdk-go-v2/feature/rds/auth"
 _ "github.com/go-sql-driver/mysql"
)

type MyEvent struct {
 Name string `json:"name"`
}

func HandleRequest(event *MyEvent) (map[string]interface{}, error) {

 var dbName string = os.Getenv("DatabaseName")
 var dbUser string = os.Getenv("DatabaseUser")
 var dbHost string = os.Getenv("DBHost") // Add hostname without https
 var dbPort int = os.Getenv("Port")      // Add port number
 var dbEndpoint string = fmt.Sprintf("%s:%d", dbHost, dbPort)
 var region string = os.Getenv("AWS_REGION")

 cfg, err := config.LoadDefaultConfig(context.TODO())
 if err != nil {
  panic("configuration error: " + err.Error())
 }

 authenticationToken, err := auth.BuildAuthToken(
  context.TODO(), dbEndpoint, region, dbUser, cfg.Credentials)
 if err != nil {
  panic("failed to create authentication token: " + err.Error())
 }
```

```
 dsn := fmt.Sprintf("%s:%s@tcp(%s)/%s?tls=true&allowCleartextPasswords=true",
  dbUser, authenticationToken, dbEndpoint, dbName,
 )

 db, err := sql.Open("mysql", dsn)
 if err != nil {
  panic(err)
 }

 defer db.Close()

 var sum int
 err = db.QueryRow("SELECT ?+? AS sum", 3, 2).Scan(&sum)
 if err != nil {
  panic(err)
 }
 s := fmt.Sprint(sum)
 message := fmt.Sprintf("The selected sum is: %s", s)

 messageBytes, err := json.Marshal(message)
 if err != nil {
  return nil, err
 }

 messageString := string(messageBytes)
 return map[string]interface{}{
  "statusCode": 200,
  "headers":    map[string]string{"Content-Type": "application/json"},
  "body":       messageString,
 }, nil
}

func main() {
 lambda.Start(HandleRequest)
}
```

Java

SDK para Java 2.x

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no repositório dos Exemplos sem servidor.

Conectar-se a um banco de dados do Amazon RDS em uma função do Lambda usando Java.

```
import com.amazonaws.services.lambda.runtime.Context;
import com.amazonaws.services.lambda.runtime.RequestHandler;
import com.amazonaws.services.lambda.runtime.events.APIGatewayProxyRequestEvent;
import com.amazonaws.services.lambda.runtime.events.APIGatewayProxyResponseEvent;
import software.amazon.awssdk.auth.credentials.DefaultCredentialsProvider;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.rdsdata.RdsDataClient;
import software.amazon.awssdk.services.rdsdata.model.ExecuteStatementRequest;
import software.amazon.awssdk.services.rdsdata.model.ExecuteStatementResponse;
import software.amazon.awssdk.services.rdsdata.model.Field;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;

public class RdsLambdaHandler implements
 RequestHandler<APIGatewayProxyRequestEvent, APIGatewayProxyResponseEvent> {

    @Override
    public APIGatewayProxyResponseEvent handleRequest(APIGatewayProxyRequestEvent
 event, Context context) {
        APIGatewayProxyResponseEvent response = new
 APIGatewayProxyResponseEvent();

        try {
            // Obtain auth token
            String token = createAuthToken();

            // Define connection configuration
```

```
            String connectionString = String.format("jdbc:mysql://%s:%s/%s?
useSSL=true&requireSSL=true",
                    System.getenv("ProxyHostName"),
                    System.getenv("Port"),
                    System.getenv("DBName"));

            // Establish a connection to the database
            try (Connection connection =
 DriverManager.getConnection(connectionString, System.getenv("DBUserName"),
 token);
                 PreparedStatement statement =
 connection.prepareStatement("SELECT ? + ? AS sum")) {

                statement.setInt(1, 3);
                statement.setInt(2, 2);

                try (ResultSet resultSet = statement.executeQuery()) {
                    if (resultSet.next()) {
                        int sum = resultSet.getInt("sum");
                        response.setStatusCode(200);
                        response.setBody("The selected sum is: " + sum);
                    }
                }
            }

        } catch (Exception e) {
            response.setStatusCode(500);
            response.setBody("Error: " + e.getMessage());
        }

        return response;
    }

    private String createAuthToken() {
        // Create RDS Data Service client
        RdsDataClient rdsDataClient = RdsDataClient.builder()
                .region(Region.of(System.getenv("AWS_REGION")))
                .credentialsProvider(DefaultCredentialsProvider.create())
                .build();

        // Define authentication request
        ExecuteStatementRequest request = ExecuteStatementRequest.builder()
                .resourceArn(System.getenv("ProxyHostName"))
                .secretArn(System.getenv("DBUserName"))
```

```
                .database(System.getenv("DBName"))
                .sql("SELECT 'RDS IAM Authentication'")
                .build();

        // Execute request and obtain authentication token
        ExecuteStatementResponse response =
 rdsDataClient.executeStatement(request);
        Field tokenField = response.records().get(0).get(0);

        return tokenField.stringValue();
    }
}
```

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no repositório dos Exemplos sem servidor.

Conectar-se a um banco de dados do Amazon RDS em uma função do Lambda usando JavaScript.

```
// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: Apache-2.0
/*
Node.js code here.
*/
// ES6+ example
import { Signer } from "@aws-sdk/rds-signer";
import mysql from 'mysql2/promise';

async function createAuthToken() {
  // Define connection authentication parameters
  const dbinfo = {

    hostname: process.env.ProxyHostName,
```

```
    port: process.env.Port,
    username: process.env.DBUserName,
    region: process.env.AWS_REGION,

  }

  // Create RDS Signer object
  const signer = new Signer(dbinfo);

  // Request authorization token from RDS, specifying the username
  const token = await signer.getAuthToken();
  return token;
}

async function dbOps() {

  // Obtain auth token
  const token = await createAuthToken();
  // Define connection configuration
  let connectionConfig = {
    host: process.env.ProxyHostName,
    user: process.env.DBUserName,
    password: token,
    database: process.env.DBName,
    ssl: 'Amazon RDS'
  }
  // Create the connection to the DB
  const conn = await mysql.createConnection(connectionConfig);
  // Obtain the result of the query
  const [res,] = await conn.execute('select ?+? as sum', [3, 2]);
  return res;

}

export const handler = async (event) => {
  // Execute database flow
  const result = await dbOps();
  // Return result
  return {
    statusCode: 200,
    body: JSON.stringify("The selected sum is: " + result[0].sum)
  }
};
```

Conectar-se a um banco de dados do Amazon RDS em uma função do Lambda usando TypeScript.

```
// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: Apache-2.0

import { Signer } from "@aws-sdk/rds-signer";
import mysql from 'mysql2/promise';

// RDS settings
// Using '!' (non-null assertion operator) to tell the TypeScript compiler that
 the DB settings are not null or undefined,
const proxy_host_name = process.env.PROXY_HOST_NAME!
const port = parseInt(process.env.PORT!)
const db_name = process.env.DB_NAME!
const db_user_name = process.env.DB_USER_NAME!
const aws_region = process.env.AWS_REGION!


async function createAuthToken(): Promise<string> {

    // Create RDS Signer object
    const signer = new Signer({
        hostname: proxy_host_name,
        port: port,
        region: aws_region,
        username: db_user_name
    });

    // Request authorization token from RDS, specifying the username
    const token = await signer.getAuthToken();
    return token;
}

async function dbOps(): Promise<mysql.QueryResult | undefined> {
    try {
        // Obtain auth token
        const token = await createAuthToken();
        const conn = await mysql.createConnection({
            host: proxy_host_name,
            user: db_user_name,
```

```
            password: token,
            database: db_name,
            ssl: 'Amazon RDS' // Ensure you have the CA bundle for SSL connection
        });
        const [rows, fields] = await conn.execute('SELECT ? + ? AS sum', [3, 2]);
        console.log('result:', rows);
        return rows;
    }
    catch (err) {
        console.log(err);
    }
}

export const lambdaHandler = async (event: any): Promise<{ statusCode: number;
 body: string }> => {
    // Execute database flow
    const result = await dbOps();

    // Return error is result is undefined
    if (result == undefined)
        return {
            statusCode: 500,
            body: JSON.stringify(`Error with connection to DB host`)
        }

    // Return result
    return {
        statusCode: 200,
        body: JSON.stringify(`The selected sum is: ${result[0].sum}`)
    };
};
```

PHP

SDK para PHP

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no repositório dos Exemplos sem servidor.

Conectar-se a um banco de dados do Amazon RDS em uma função do Lambda usando PHP.

```
<?php
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0

# using bref/bref and bref/logger for simplicity

use Bref\Context\Context;
use Bref\Event\Handler as StdHandler;
use Bref\Logger\StderrLogger;
use Aws\Rds\AuthTokenGenerator;
use Aws\Credentials\CredentialProvider;

require __DIR__ . '/vendor/autoload.php';

class Handler implements StdHandler
{
    private StderrLogger $logger;
    public function __construct(StderrLogger $logger)
    {
        $this->logger = $logger;
    }


    private function getAuthToken(): string {
        // Define connection authentication parameters
        $dbConnection = [
            'hostname' => getenv('DB_HOSTNAME'),
            'port' => getenv('DB_PORT'),
            'username' => getenv('DB_USERNAME'),
            'region' => getenv('AWS_REGION'),
        ];

        // Create RDS AuthTokenGenerator object
        $generator = new
 AuthTokenGenerator(CredentialProvider::defaultProvider());

        // Request authorization token from RDS, specifying the username
        return $generator->createToken(
            $dbConnection['hostname'] . ':' . $dbConnection['port'],
            $dbConnection['region'],
            $dbConnection['username']
        );
```

```
    }

    private function getQueryResults() {
        // Obtain auth token
        $token = $this->getAuthToken();

        // Define connection configuration
        $connectionConfig = [
            'host' => getenv('DB_HOSTNAME'),
            'user' => getenv('DB_USERNAME'),
            'password' => $token,
            'database' => getenv('DB_NAME'),
        ];

        // Create the connection to the DB
        $conn = new PDO(

"mysql:host={$connectionConfig['host']};dbname={$connectionConfig['database']}",
            $connectionConfig['user'],
            $connectionConfig['password'],
            [
                PDO::MYSQL_ATTR_SSL_CA => '/path/to/rds-ca-2019-root.pem',
                PDO::MYSQL_ATTR_SSL_VERIFY_SERVER_CERT => true,
            ]
        );

        // Obtain the result of the query
        $stmt = $conn->prepare('SELECT ?+? AS sum');
        $stmt->execute([3, 2]);

        return $stmt->fetch(PDO::FETCH_ASSOC);
    }

    /**
     * @param mixed $event
     * @param Context $context
     * @return array
     */
    public function handle(mixed $event, Context $context): array
    {
        $this->logger->info("Processing query");

        // Execute database flow
        $result = $this->getQueryResults();
```

```
        return [
            'sum' => $result['sum']
        ];
    }
}

$logger = new StderrLogger();
return new Handler($logger);
```

Python

SDK para Python (Boto3).

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no repositório dos Exemplos sem servidor.

Conectar-se a um banco de dados do Amazon RDS em uma função do Lambda usando Python.

```
import json
import os
import boto3
import pymysql

# RDS settings
proxy_host_name = os.environ['PROXY_HOST_NAME']
port = int(os.environ['PORT'])
db_name = os.environ['DB_NAME']
db_user_name = os.environ['DB_USER_NAME']
aws_region = os.environ['AWS_REGION']


# Fetch RDS Auth Token
def get_auth_token():
    client = boto3.client('rds')
    token = client.generate_db_auth_token(
        DBHostname=proxy_host_name,
```

```
        Port=port
        DBUsername=db_user_name
        Region=aws_region
    )
    return token

def lambda_handler(event, context):
    token = get_auth_token()
    try:
        connection = pymysql.connect(
            host=proxy_host_name,
            user=db_user_name,
            password=token,
            db=db_name,
            port=port,
            ssl={'ca': 'Amazon RDS'}  # Ensure you have the CA bundle for SSL
 connection
        )

        with connection.cursor() as cursor:
            cursor.execute('SELECT %s + %s AS sum', (3, 2))
            result = cursor.fetchone()

        return result

    except Exception as e:
        return (f"Error: {str(e)}")  # Return an error message if an exception
 occurs
```

Ruby

SDK para Ruby

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no repositório dos Exemplos sem servidor.

Conectar-se a um banco de dados do Amazon RDS em uma função do Lambda usando Ruby.

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
# Ruby code here.

require 'aws-sdk-rds'
require 'json'
require 'mysql2'

def lambda_handler(event:, context:)
  endpoint = ENV['DBEndpoint'] # Add the endpoint without https"
  port = ENV['Port']           # 3306
  user = ENV['DBUser']
  region = ENV['DBRegion']     # 'us-east-1'
  db_name = ENV['DBName']

  credentials = Aws::Credentials.new(
    ENV['AWS_ACCESS_KEY_ID'],
    ENV['AWS_SECRET_ACCESS_KEY'],
    ENV['AWS_SESSION_TOKEN']
  )
  rds_client = Aws::RDS::AuthTokenGenerator.new(
    region: region,
    credentials: credentials
  )

  token = rds_client.auth_token(
    endpoint: endpoint+ ':' + port,
    user_name: user,
    region: region
  )

  begin
    conn = Mysql2::Client.new(
      host: endpoint,
      username: user,
      password: token,
      port: port,
      database: db_name,
      sslca: '/var/task/global-bundle.pem',
      sslverify: true,
      enable_cleartext_plugin: true
    )
    a = 3
```

```
    b = 2
    result = conn.query("SELECT #{a} + #{b} AS sum").first['sum']
    puts result
    conn.close
    {
      statusCode: 200,
      body: result.to_json
    }
  rescue => e
    puts "Database connection failed due to #{e}"
  end
end
```

Rust

SDK para Rust

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no repositório dos Exemplos sem servidor.

Conectar-se a um banco de dados do Amazon RDS em uma função do Lambda usando Rust.

```
use aws_config::BehaviorVersion;
use aws_credential_types::provider::ProvideCredentials;
use aws_sigv4::{
    http_request::{sign, SignableBody, SignableRequest, SigningSettings},
    sign::v4,
};
use lambda_runtime::{run, service_fn, Error, LambdaEvent};
use serde_json::{json, Value};
use sqlx::postgres::PgConnectOptions;
use std::env;
use std::time::{Duration, SystemTime};

const RDS_CERTS: &[u8] = include_bytes!("global-bundle.pem");

async fn generate_rds_iam_token(
    db_hostname: &str,
    port: u16,
```

```
    db_username: &str,
) -> Result<String, Error> {
    let config = aws_config::load_defaults(BehaviorVersion::v2024_03_28()).await;

    let credentials = config
        .credentials_provider()
        .expect("no credentials provider found")
        .provide_credentials()
        .await
        .expect("unable to load credentials");
    let identity = credentials.into();
    let region = config.region().unwrap().to_string();

    let mut signing_settings = SigningSettings::default();
    signing_settings.expires_in = Some(Duration::from_secs(900));
    signing_settings.signature_location =
 aws_sigv4::http_request::SignatureLocation::QueryParams;

    let signing_params = v4::SigningParams::builder()
        .identity(&identity)
        .region(&region)
        .name("rds-db")
        .time(SystemTime::now())
        .settings(signing_settings)
        .build()?;

    let url = format!(
        "https://{db_hostname}:{port}/?Action=connect&DBUser={db_user}",
        db_hostname = db_hostname,
        port = port,
        db_user = db_username
    );

    let signable_request =
        SignableRequest::new("GET", &url, std::iter::empty(),
 SignableBody::Bytes(&[]))
            .expect("signable request");

    let (signing_instructions, _signature) =
        sign(signable_request, &signing_params.into())?.into_parts();

    let mut url = url::Url::parse(&url).unwrap();
    for (name, value) in signing_instructions.params() {
        url.query_pairs_mut().append_pair(name, &value);
```

```
    }

    let response = url.to_string().split_off("https://".len());

    Ok(response)
}

#[tokio::main]
async fn main() -> Result<(), Error> {
    run(service_fn(handler)).await
}

async fn handler(_event: LambdaEvent<Value>) -> Result<Value, Error> {
    let db_host = env::var("DB_HOSTNAME").expect("DB_HOSTNAME must be set");
    let db_port = env::var("DB_PORT")
        .expect("DB_PORT must be set")
        .parse::<u16>()
        .expect("PORT must be a valid number");
    let db_name = env::var("DB_NAME").expect("DB_NAME must be set");
    let db_user_name = env::var("DB_USERNAME").expect("DB_USERNAME must be set");

    let token = generate_rds_iam_token(&db_host, db_port, &db_user_name).await?;

    let opts = PgConnectOptions::new()
        .host(&db_host)
        .port(db_port)
        .username(&db_user_name)
        .password(&token)
        .database(&db_name)
        .ssl_root_cert_from_pem(RDS_CERTS.to_vec())
        .ssl_mode(sqlx::postgres::PgSslMode::Require);

    let pool = sqlx::postgres::PgPoolOptions::new()
        .connect_with(opts)
        .await?;

    let result: i32 = sqlx::query_scalar("SELECT $1 + $2")
        .bind(3)
        .bind(2)
        .fetch_one(&pool)
        .await?;

    println!("Result: {:?}", result);
```

```
    Ok(json!({
        "statusCode": 200,
        "content-type": "text/plain",
        "body": format!("The selected sum is: {result}")
    }))
}
```

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Exemplos do Amazon RDS entre serviços usando AWS SDKs

Os exemplos de aplicações a seguir usam AWS SDKs para combinar o Amazon RDS com outros Serviços da AWS. Cada exemplo inclui um link para o GitHub, em que é possível encontrar instruções sobre como configurar e executar a aplicação.

Exemplos

- Crie um rastreador de itens de trabalho do Aurora Sem Servidor

Crie um rastreador de itens de trabalho do Aurora Sem Servidor

Os exemplos de código a seguir mostram como criar uma aplicação Web que rastreia os itens de trabalho em um banco de dados do Amazon Aurora Sem Servidor e usa o Amazon Simple Email Service (Amazon SES) para enviar relatórios.

.NET

SDK para .NET

Mostra como usar o AWS SDK para .NET para desenvolver uma aplicação Web que monitora itens de trabalho no banco de dados do Amazon Aurora e envia relatórios por e-mail usando o Amazon Simple Email Service (Amazon SES). Este exemplo usa um front-end criado com React.js para interagir com um back-end .NET RESTful.

- Integre uma aplicação Web do React com os serviços da AWS.
- Liste, adicione, atualize e exclua itens em uma tabela do Aurora.

- Envie um relatório por e-mail dos itens de trabalho filtrados usando o Amazon SES.
- Implante e gerencie recursos de exemplo com o script do AWS CloudFormation incluído.

Para obter o código-fonte completo e instruções sobre como configurar e executar o exemplo, consulte o exemplo completo no GitHub.

Serviços utilizados neste exemplo

- Aurora
- Amazon RDS
- Serviços de dados do Amazon RDS
- Amazon SES

C++

SDK para C++

Mostra como criar uma aplicação Web que rastreia e gera relatórios sobre itens de trabalho armazenados em um banco de dados do Amazon Aurora Sem Servidor.

Para obter o código-fonte completo e instruções sobre como configurar a API REST de C++ que consulta os dados do Amazon Aurora Sem Servidor e para uso por uma aplicação React, consulte o exemplo completo no GitHub.

Serviços utilizados neste exemplo

- Aurora
- Amazon RDS
- Serviços de dados do Amazon RDS
- Amazon SES

Java

SDK para Java 2.x

Mostra como construir uma aplicação Web que monitora e gera relatórios sobre itens de trabalho armazenados em um banco de dados do Amazon RDS.

Para obter o código-fonte completo e instruções sobre como configurar a API Spring REST que consulta os dados do Amazon Aurora Sem Servidor e para uso por uma aplicação React, consulte o exemplo completo no GitHub.

Para obter o código-fonte completo e instruções sobre como configurar e executar o exemplo que utiliza a API JDBC, consulte o exemplo completo no GitHub.

Serviços utilizados neste exemplo

- Aurora
- Amazon RDS
- Serviços de dados do Amazon RDS
- Amazon SES

JavaScript

SDK para JavaScript (v3)

Mostra como usar o AWS SDK para JavaScript (v3) para criar uma aplicação Web que rastreia itens de trabalho em um banco de dados do Amazon Aurora e envia relatórios por e-mail usando o Amazon Simple Email Service (Amazon SES). Este exemplo usa um front-end criado com React.js para interagir com um back-end Node.js Express.

- Integre uma aplicação Web React.js com Serviços da AWS.
- Liste, adicione e atualize itens em uma tabela do Aurora.
- Use o Amazon SES para enviar um relatório por e-mail dos itens de trabalho filtrados.
- Implante e gerencie recursos de exemplo com o script do AWS CloudFormation incluído.

Para obter o código-fonte completo e instruções sobre como configurar e executar o exemplo, consulte o exemplo completo no GitHub.

Serviços utilizados neste exemplo

- Aurora
- Amazon RDS
- Serviços de dados do Amazon RDS
- Amazon SES

Kotlin

SDK para Kotlin

Mostra como construir uma aplicação Web que monitora e gera relatórios sobre itens de trabalho armazenados em um banco de dados do Amazon RDS.

Para obter o código-fonte completo e instruções sobre como configurar a API Spring REST que consulta os dados do Amazon Aurora Sem Servidor e para uso por uma aplicação React, consulte o exemplo completo no GitHub.

Serviços utilizados neste exemplo

- Aurora
- Amazon RDS
- Serviços de dados do Amazon RDS
- Amazon SES

PHP

SDK para PHP

Mostra como usar o AWS SDK para PHP para construir uma aplicação Web que monitora itens de trabalho no banco de dados do Amazon RDS e envia relatórios por e-mail usando o Amazon Simple Email Service (Amazon SES). Este exemplo usa um front-end construído com React.js para interagir com um back-end PHP RESTful.

- Integre uma aplicação Web React.js com os serviços da AWS.
- Liste, adicione, atualize e exclua itens em uma tabela do Amazon RDS.
- Envie um relatório por e-mail dos itens de trabalho filtrados usando o Amazon SES.
- Implante e gerencie recursos de exemplo com o script do AWS CloudFormation incluído.

Para obter o código-fonte completo e instruções sobre como configurar e executar o exemplo, consulte o exemplo completo no GitHub.

Serviços utilizados neste exemplo

- Aurora
- Amazon RDS
- Serviços de dados do Amazon RDS

- Amazon SES

Python

SDK para Python (Boto3).

Mostra como usar o AWS SDK para Python (Boto3) para criar um serviço REST que rastreia itens de trabalho no banco de dados do Amazon Aurora Sem Servidor e envia relatórios por e-mail usando o Amazon Simple Email Service (Amazon SES). Este exemplo usa o framework da Web do Flask para lidar com o roteamento HTTP e se integra a uma página da Web do React para apresentar uma aplicação Web totalmente funcional.

- Crie um serviço REST do Flask que se integre a Serviços da AWS.
- Leia, grave e atualize itens de trabalho armazenados em um banco de dados do Aurora Sem Servidor.
- Crie um segredo do AWS Secrets Manager que contenha credenciais do banco de dados e use-o a fim de autenticar chamadas para o banco de dados.
- Use o Amazon SES para enviar relatórios por e-mail de itens de trabalho.

Para obter o código-fonte completo e instruções sobre como configurar e executar o exemplo, consulte o exemplo completo no GitHub.

Serviços utilizados neste exemplo

- Aurora
- Amazon RDS
- Serviços de dados do Amazon RDS
- Amazon SES

Para obter uma lista completa dos Guias do desenvolvedor do AWS SDK e exemplos de código, consulte Usar este serviço com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Segurança no Amazon RDS

A segurança na nuvem da AWS é nossa maior prioridade. Como cliente da AWS, você se beneficiará de um data center e de uma arquitetura de rede criados para atender aos requisitos das empresas com as maiores exigências de segurança.

A segurança é uma responsabilidade compartilhada entre a AWS e você. O modelo de responsabilidade compartilhada descreve isto como segurança da nuvem e segurança na nuvem:

- Segurança da nuvem: a AWS é responsável pela proteção da infraestrutura que executa produtos da AWS na Nuvem AWS. A AWS também fornece serviços que podem ser usados com segurança. Auditores de terceiros testam e verificam regularmente a eficácia da nossa segurança como parte dos programas de conformidade da AWS. Para saber mais sobre os programas de compatibilidade que se aplicam ao Amazon RDS, consulte Serviços da AWS no escopo pelo programa de conformidade.
- Segurança na nuvem: sua responsabilidade é determinada pelo serviço da AWS que você usa. Você também é responsável por outros fatores, inclusive a confidencialidade dos dados, os requisitos da organização, as leis e as regulamentações vigentes.

Essa documentação ajuda a entender como aplicar o Modelo de Responsabilidade Compartilhada ao usar o Amazon RDS. Os tópicos a seguir mostram como configurar o Amazon RDS para atender aos seus objetivos de segurança e conformidade. Saiba também como usar outros serviços da AWS que ajudam a monitorar e proteger os recursos do Amazon RDS.

Você pode gerenciar o acesso a seus recursos do Amazon RDS e a seus bancos de dados em instância de banco de dados. O método usado para gerenciar o acesso depende do tipo de tarefa que o usuário precisa realizar com o Amazon RDS:

- Execute a instância de banco de dados em uma nuvem privada virtual (VPC) baseada no serviço da Amazon VPC para obter o maior controle possível de acesso à rede. Para ter mais informações sobre como criar uma instância de banco de dados em uma VPC, consulte Amazon VPC e Amazon RDS.
- Use políticas do AWS Identity and Access Management (IAM) para atribuir permissões que determinam quem tem permissão para gerenciar os recursos do Amazon RDS. Por exemplo, você pode usar o IAM para determinar quem tem permissão para criar, descrever, modificar e excluir instâncias de banco de dados, marcar recursos ou modificar grupos de segurança.

- Use grupos de segurança para controlar quais endereços IP ou instâncias do Amazon EC2 podem se conectar aos seus bancos de dados em uma instância de banco de dados. Quando você cria uma instância de banco de dados, seu firewall impede qualquer acesso ao banco de dados, exceto por meio de regras especificadas por um grupo de segurança associado.
- Use conexões Secure Socket Layer (SSL) ou Transport Layer Security (TLS) com instâncias de banco de dados que executam os seguintes mecanismos de banco de dados: Db2, MySQL, MariaDB, PostgreSQL, Oracle ou Microsoft SQL Server. Para ter mais informações sobre como usar o SSL/TLS com uma instância de banco de dados, consulte Usar SSL/TLS para criptografar uma conexão com uma instância ou um cluster de banco de dados.
- Use a criptografia do Amazon RDS para proteger suas instâncias de banco de dados e snapshots em repouso. A criptografia do Amazon RDS usa o algoritmo de criptografia AES-256 padrão do setor para criptografar seus dados no servidor que hospeda a instância de banco de dados. Para obter mais informações, consulte Criptografar recursos do Amazon RDS.
- Use a criptografia de rede e a criptografia de dados transparente com instâncias de bancos de dados Oracle. Para obter mais informações, consulte Oracle Native Network Encryption e Oracle Transparent Data Encryption
- Use os recursos de segurança do seu mecanismo de banco de dados para controlar quem pode fazer login nos bancos de dados em uma instância de banco de dados. Esses recursos funcionam como se o banco de dados estivesse em sua rede local.

> **Note**
>
> Basta configurar a segurança para os casos de uso. Não é necessário configurar o acesso de segurança para os processos que o Amazon RDS gerencia. Isso inclui a criação de backups, a replicação de dados entre uma instância do banco de dados primário e uma réplica de leitura e outros processos.

Para ter mais informações sobre como gerenciar o acesso a recursos do Amazon RDS e os bancos de dados de uma instância de banco de dados, consulte os tópicos a seguir.

Tópicos

- Autenticação do banco de dados com Amazon RDS
- Gerenciamento de senhas com Amazon RDS e AWS Secrets Manager
- Proteção de dados no Amazon RDS

- Gerenciamento de identidade e acesso no Amazon RDS
- Registrar em log e monitorar no Amazon RDS
- Validação de conformidade do Amazon RDS
- Resiliência no Amazon RDS
- Segurança da infraestrutura no Amazon RDS
- API do Amazon RDS e endpoints da VPC de interface (AWS PrivateLink)
- Práticas recomendadas de segurança do Amazon RDS
- Controlar acesso com grupos de segurança
- Privilégios da conta de usuário mestre
- Usar funções vinculadas ao serviço do Amazon RDS
- Amazon VPC e Amazon RDS

Autenticação do banco de dados com Amazon RDS

O Amazon RDS permite autenticar usuários do banco de dados de várias maneiras.

As autenticações de banco de dados do IAM, do Kerberos e por senha usam diferentes métodos de autenticação no banco de dados. Portanto, um usuário específico pode fazer login em um banco de dados usando apenas um método de autenticação.

No PostgreSQL, use apenas uma das seguintes configurações de função para um usuário de um banco de dados específico:

- Para usar a autenticação de banco de dados do IAM, atribua a função `rds_iam` ao usuário.
- Para usar a autenticação do Kerberos, atribua a função `rds_ad` ao usuário.
- Para usar a autenticação por senha, não atribua as funções `rds_iam` ou `rds_ad` ao usuário.

Não atribua ambas as funções `rds_iam` e `rds_ad` a um usuário de um banco de dados PostgreSQL direta ou indiretamente por acesso de concessão aninhado. Se a função `rds_iam` for adicionada ao usuário mestre, a autenticação do IAM terá precedência sobre a autenticação por senha, então o usuário mestre terá que fazer login como um usuário do IAM.

> **Important**
>
> É altamente recomendável não usar o usuário mestre diretamente nas aplicações. Em vez disso, siga as práticas recomendadas de usar um usuário do banco de dados criado com os privilégios mínimos obrigatórios para a aplicação.

Tópicos

- Autenticação com senha
- Autenticação do banco de dados do IAM
- Autenticação de Kerberos

Autenticação com senha

Com a autenticação com senha, seu banco de dados executa toda a administração de contas de usuário. Crie usuários com instruções SQL, como `CREATE USER`, usando a cláusula apropriada exigida pelo mecanismo de banco de dados para especificar senhas. Por exemplo, no MySQL, a instrução é `CREATE USER` *`name`* `IDENTIFIED BY` *`password`*, enquanto, no PostgreSQL, ela é `CREATE USER` *`name`* `WITH PASSWORD` *`password`*.

Com a autenticação com senha, seu banco de dados controla e autentica contas de usuário. Se um mecanismo de banco de dados tiver recursos de gerenciamento de senhas fortes, ele poderá aumentar a segurança. A autenticação de banco de dados pode ser mais fácil de administrar usando autenticação com senha quando você tem pequenas comunidades de usuários. Como as senhas de texto não criptografado são geradas nesse caso, a integração com o AWS Secrets Manager pode aumentar a segurança.

Para ter informações sobre como usar o Secrets Manager com o Amazon RDS, consulte Creating a basic secret e Rotating secrets for supported Amazon RDS databases no Guia do usuário do AWS Secrets Manager. Para obter informações sobre como recuperar os segredos de forma programática nas aplicações personalizadas, consulte Recuperar o valor do segredo no Guia do usuário do AWS Secrets Manager.

Autenticação do banco de dados do IAM

Você pode se autenticar na instânciao de banco de dados usando a autenticação de banco de dados do AWS Identity and Access Management (IAM). Com esse método de autenticação, você não

precisa usar uma senha ao conectar-se a uma instância de banco de dados. Em vez disso, você usa um token de autenticação.

Para ter mais informações sobre autenticação de banco de dados do IAM, incluindo informações sobre disponibilidade de mecanismos de banco de dados específicos, consulte Autenticação do banco de dados do IAM para MariaDB, MySQL e PostgreSQL.

Autenticação de Kerberos

O Amazon RDS oferece suporte à autenticação externa de usuários de banco de dados usando o Kerberos e o Microsoft Active Directory. O Kerberos é um protocolo de autenticação de rede que usa tíquetes e criptografia de chave simétrica para eliminar a necessidade de transmitir senhas pela rede. O Kerberos foi integrado ao Active Directory e foi projetado para autenticar usuários em recursos de rede, como bancos de dados.

O suporte do Amazon RDS ao Kerberos e ao Active Directory oferece os benefícios de autenticação única e autenticação centralizada dos usuários do banco de dados. Você pode manter suas credenciais de usuário no Active Directory. O Active Directory oferece um lugar centralizado para armazenar e gerenciar credenciais para várias instâncias de banco de dados.

Para usar as credenciais do Active Directory autogerenciado, é necessário configurar uma relação de confiança com o AWS Directory Service para Microsoft Active Directory ao qual a instância de banco de dados está associada.

O RDS para PostgreSQL e o RDS para MySQL comportam relacionamentos de confiança florestal unidirecionais e bidirecionais com autenticação em toda a floresta ou autenticação seletiva.

Em alguns casos, é possível configurar a autenticação Kerberos em uma relação de confiança externa. Isso exige que o Active Directory autogerenciado tenha configurações adicionais. Isso inclui, entre outros, o Kerberos Forest Search Order.

As instâncias de banco de dados do Microsoft SQL Server e do PostgreSQL são compatíveis com relações de confiança de floresta unidirecionais e bidirecionais. As instâncias de banco de dados Oracle comportam relações de confiança externas e de floresta unidirecionais e bidirecionais. Para ter mais informações, consulte Quando criar uma relação de confiança no Guia de administração do AWS Directory Service.

Para obter informações sobre a autenticação Kerberos com um mecanismo de banco de dados específico, consulte o seguinte:

- Trabalhar com o AWS Managed Active Directory com o RDS para SQL Server
- Uso da autenticação do Kerberos para o Amazon RDS para MySQL
- Configurar a autenticação Kerberos do Amazon RDS para Oracle
- Usar a autenticação Kerberos com o Amazon RDS para PostgreSQL
- Usar a autenticação do Kerberos para o Amazon RDS para Db2 .

> **Note**
>
> Atualmente, a autenticação Kerberos não tem suporte para instâncias de banco de dados MariaDB.

Gerenciamento de senhas com Amazon RDS e AWS Secrets Manager

O Amazon RDS integra-se ao Secrets Manager para gerenciar senhas do usuário principal para suas instâncias e clusters de banco de dados multi-AZ.

Tópicos

- Limitações da integração do Secrets Manager com o Amazon RDS
- Visão geral do gerenciamento de senhas do usuário principal com AWS Secrets Manager
- Benefícios do gerenciamento de senhas do usuário principal com o Secrets Manager
- Permissões necessárias para a integração do Secrets Manager
- Impor o gerenciamento da senha do usuário principal pelo RDS no AWS Secrets Manager
- Gerenciar a senha do usuário principal para uma instância de banco de dados com o Secrets Manager
- Gerenciar a senha do usuário principal para um banco de dados de locatário do RDS para Oracle com o Secrets Manager
- Gerenciar a senha do usuário principal para um cluster de banco de dados multi-AZ com o Secrets Manager
- Alternar o segredo da senha do usuário principal para uma instância de banco de dados
- Alternar o segredo de uma senha principal do usuário para um cluster de banco de dados multi-AZ
- Visualizar os detalhes sobre um segredo para uma instância de banco de dados
- Visualizar os detalhes sobre um segredo para um cluster de banco de dados multi-AZ
- Visualizar detalhes sobre um segredo para um banco de dados de locatários
- Disponibilidade de regiões e versões

Limitações da integração do Secrets Manager com o Amazon RDS

O gerenciamento de senhas do usuário principal com o Secrets Manager não é compatível com os seguintes recursos:

- Criação de réplicas de leitura quando o banco de dados ou o cluster de banco de dados de origem gerencia credenciais com o Secrets Manager. Isso se aplica a todos os mecanismos de banco de dados, exceto RDS para SQL Server.

- Implantações azul/verde do Amazon RDS
- Amazon RDS Custom
- Alternância do Oracle Data Guard

Visão geral do gerenciamento de senhas do usuário principal com AWS Secrets Manager

Com o AWS Secrets Manager, é possível substituir credenciais codificadas em seu código, inclusive senhas de banco de dados, por uma chamada de API ao Secrets Manager para recuperar o segredo de forma programática. Para ter mais informações sobre o Secrets Manager, consulte o Guia do usuário do AWS Secrets Manager.

Quando você armazena segredos de banco de dados no Secrets Manager, sua Conta da AWS incorre em cobranças. Para obter mais informações sobre definição de preços, consulte Definição de preço do AWS Secrets Manager.

Você pode especificar que o RDS gerencie a senha de usuário principal no Secrets Manager para uma instância de banco de dados do Amazon RDS ou cluster de banco de dados multi-AZ ao realizar uma das seguintes operações:

- Criar uma instância de banco de dados
- Criar um cluster de banco de dados multi-AZ
- Criar um banco de dados de locatário em um CDB do RDS para Oracle
- Modificar uma instância de banco de dados
- Modificar um cluster de banco de dados multi-AZ
- Modificar um banco de dados de locatário (somente RDS para Oracle)
- Restaurar uma instância de banco de dados do Amazon S3
- Restaurar uma instância de banco de dados com base em um snapshot ou para um ponto no tempo (somente RDS para Oracle)

Quando você especifica que o RDS gerencie a senha do usuário principal no Secrets Manager, o RDS gera a senha e a armazena no Secrets Manager. Você pode interagir diretamente com o segredo para recuperar as credenciais do usuário principal. Você também pode especificar uma chave gerenciada pelo cliente para criptografar o segredo ou usar a chave do KMS fornecida pelo Secrets Manager.

O RDS gerencia as configurações do segredo e o alterna a cada sete dias por padrão. Você pode modificar algumas configurações, como o cronograma de alternância. Se você excluir uma instância de banco de dados que gerencie um segredo no Secrets Manager, o segredo e seus metadados associados também serão excluídos.

Para conectar-se a uma instância de banco de dados ou a um cluster de banco de dados multi-AZ com as credenciais em um segredo, você pode recuperar o segredo do Secrets Manager. Para ter mais informações, consulte Recuperar segredos do AWS Secrets Manager e Conecte-se a um banco de dados SQL com credenciais em um segredo do AWS Secrets Manager no Guia do usuário do AWS Secrets Manager.

Benefícios do gerenciamento de senhas do usuário principal com o Secrets Manager

O gerenciamento de senhas do usuário principal pelo RDS com o Secrets Manager oferece os seguintes benefícios:

- O RDS gera automaticamente credenciais de banco de dados.
- O RDS armazena e gerencia automaticamente as credenciais do banco de dados no AWS Secrets Manager.
- O RDS alterna as credenciais do banco de dados regularmente, sem exigir alterações na aplicação.
- O Secrets Manager protege as credenciais do banco de dados do acesso humano e da visualização de texto sem formatação.
- O Secrets Manager permite a recuperação de credenciais do banco de dados em segredos para conexões de banco de dados.
- O Secrets Manager permite um controle refinado do acesso às credenciais do banco de dados em segredos com o uso do IAM.
- Você também pode separar a criptografia do banco de dados da criptografia de credenciais com chaves do KMS diferentes.
- É possível eliminar o gerenciamento e a alternância manuais das credenciais do banco de dados.
- Você pode monitorar facilmente as credenciais do banco de dados com o AWS CloudTrail e o Amazon CloudWatch.

Para obter mais informações sobre os benefícios do Secrets Manager, consulte o Guia do usuário do AWS Secrets Manager.

Permissões necessárias para a integração do Secrets Manager

Os usuários devem ter as permissões necessárias para realizar operações relacionadas à integração do Secrets Manager. É possível criar políticas do IAM que concedam permissões para realizar operações de API específicas nos recursos especificados necessários. Depois, você pode anexar essas políticas aos conjuntos de permissões do IAM ou às funções que exigem essas permissões. Para ter mais informações, consulte Gerenciamento de identidade e acesso no Amazon RDS.

Para operações de criação, modificação ou restauração, o usuário que especifica que o Amazon RDS gerencie a senha do usuário principal no Secrets Manager deve ter permissões para realizar as seguintes operações:

- `kms:DescribeKey`
- `secretsmanager:CreateSecret`
- `secretsmanager:TagResource`

A permissão `kms:DescribeKey` é necessária para acessar sua chave gerenciada pelo cliente para o `MasterUserSecretKmsKeyId` e para descrever `aws/secretsmanager`.

Para operações de criação, modificação ou restauração, o usuário que especifica a senha do usuário principal para criptografar o segredo no Secrets Manager deve ter permissões para realizar as seguintes operações:

- `kms:Decrypt`
- `kms:GenerateDataKey`
- `kms:CreateGrant`

Para operações de modificação, o usuário que alterna a senha de usuário principal no Secrets Manager deve ter permissões para realizar a seguinte operação:

- `secretsmanager:RotateSecret`

Impor o gerenciamento da senha do usuário principal pelo RDS no AWS Secrets Manager

Você pode usar as chaves de condição do IAM para impor o gerenciamento pelo RDS da senha do usuário principal no AWS Secrets Manager. A política a seguir não permite que os usuários criem nem restaurem instâncias de banco de dados ou clusters de banco de dados ou criem ou modifiquem bancos de dados de locatário a menos que a senha do usuário principal seja gerenciada pelo RDS no Secrets Manager.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": ["rds:CreateDBInstance", "rds:CreateDBCluster",
 "rds:RestoreDBInstanceFromS3", "rds:RestoreDBClusterFromS3",
                       "rds:RestoreDBInstanceFromDBSnapshot",
 "rds:RestoreDBInstanceToPointInTime", "rds:CreateTenantDatabase",
                       "rds:ModifyTenantDatabase"],
            "Resource": "*",
            "Condition": {
                "Bool": {
                    "rds:ManageMasterUserPassword": false
                }
            }
        }
    ]
}
```

> **Note**
>
> Esta política impõe o gerenciamento de senhas no AWS Secrets Manager no momento da criação. No entanto, você ainda pode desativar a integração do Secrets Manager e definir manualmente uma senha principal modificando a instância.
> Para evitar isso, inclua `rds:ModifyDBInstance` e `rds:ModifyDBCluster` no bloco "Ação" da política. Esteja ciente de que isso impede que o usuário aplique quaisquer

modificações adicionais às instâncias existentes que não têm a integração com o Secrets Manager habilitada.

Para ter mais informações sobre como usar as chaves de condição em políticas do IAM, consulte Chaves de condição de políticas do Amazon RDS e Políticas de exemplo: usar chaves de condição.

Gerenciar a senha do usuário principal para uma instância de banco de dados com o Secrets Manager

Você pode configurar o gerenciamento pelo RDS da senha do usuário principal no Secrets Manager ao realizar as seguintes ações:

- Criar uma instância de banco de dados do Amazon RDS
- Modificar uma instância de banco de dados do Amazon RDS
- Restauração de um backup em uma instância de banco de dados MySQL
- Restaurar uma instância de banco de dados (somente RDS para Oracle)
- Restaurar uma instância de banco de dados para um momento especificado no Amazon RDS (somente RDS para Oracle)

Você pode realizar as operações anteriores usando o console do RDS, a AWS CLI ou a API do RDS.

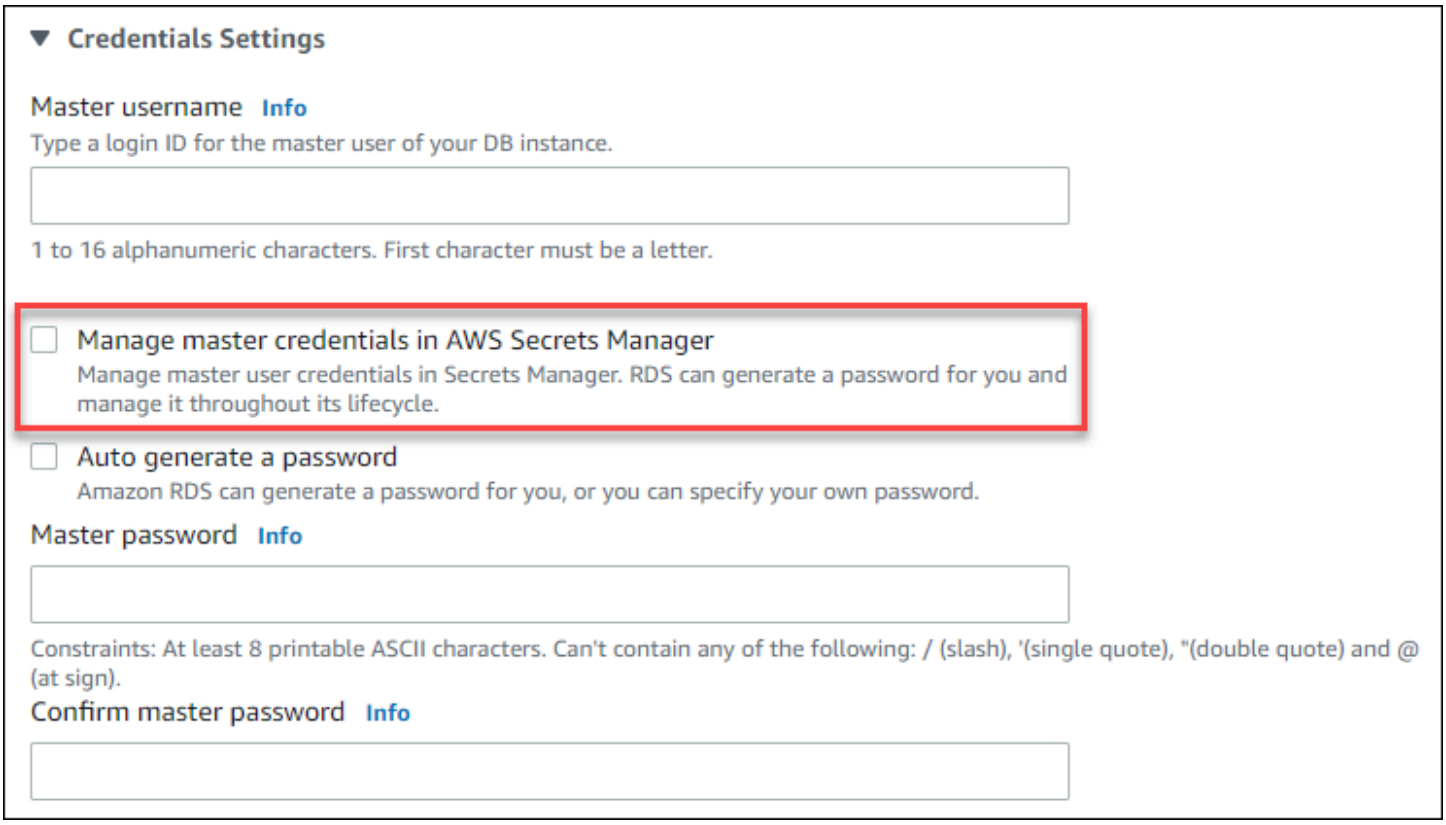
Console

Siga as instruções para criar ou modificar uma instância de banco de dados com o console do RDS:

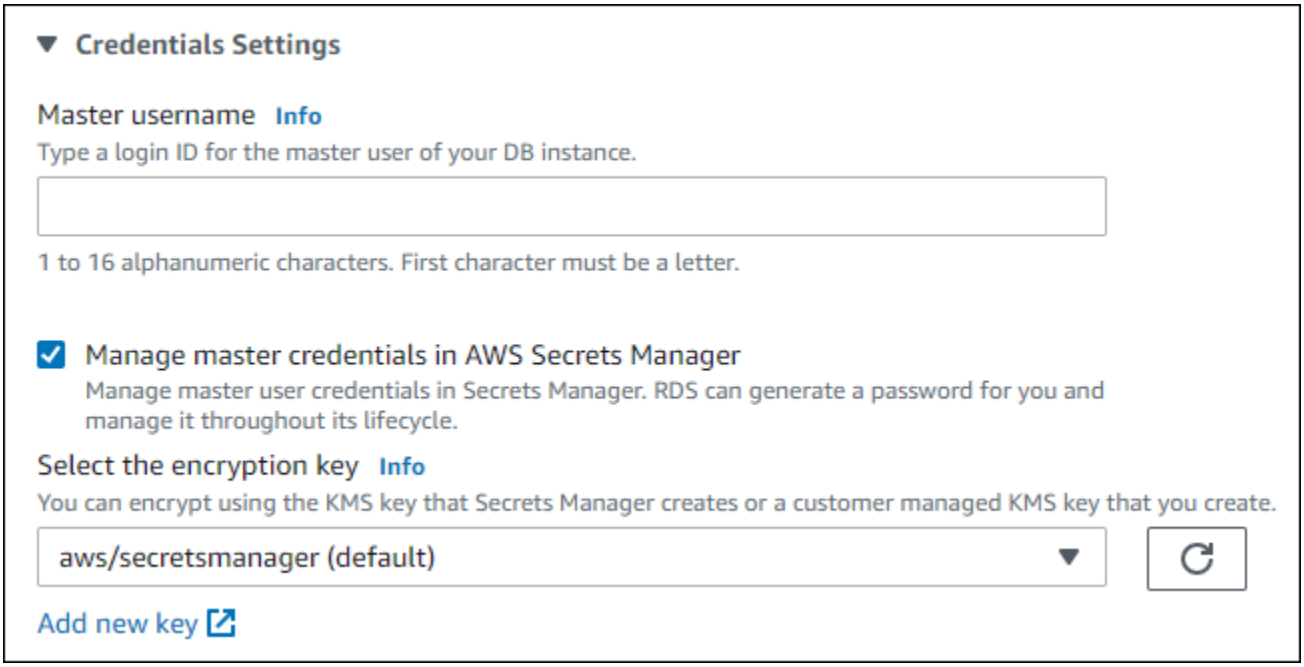
- Criar uma instância de banco de dados
- Modificar uma instância de banco de dados do Amazon RDS
- Importar dados do Amazon S3 para uma nova instância de banco de dados MySQL

Ao usar o console do RDS para realizar uma dessas operações, você pode especificar que a senha do usuário principal seja gerenciada pelo RDS no Secrets Manager. Ao criar ou restaurar uma instância de banco de dados, selecione Gerenciar credenciais principais no AWS Secrets Manager em Configurações de credenciais. Quando estiver modificando uma instância de banco de dados, selecione Gerenciar credenciais principais no AWS Secrets Manager em Configurações.

A imagem a seguir é um exemplo da configuração Gerenciar credenciais principais no AWS Secrets Manager quando você está criando ou restaurando uma instância de banco de dados.



Quando você seleciona essa opção, o RDS gera a senha do usuário principal e a gerencia durante todo o ciclo de vida no Secrets Manager.



Você pode optar por criptografar o segredo com uma chave do KMS fornecida pelo Secrets Manager ou com uma chave gerenciada pelo cliente criada por você. Depois que o RDS estiver gerenciando as credenciais de banco de dados de uma instância de banco de dados, não será possível alterar a chave do KMS usada para criptografar o segredo.

Você pode escolher outras configurações para atender às suas necessidades. Para ter mais informações sobre as configurações disponíveis ao criar uma instância de banco de dados, consulte Configurações para instâncias de banco de dados. Para ter mais informações sobre as configurações disponíveis ao modificar uma instância de banco de dados, consulte Configurações para instâncias de banco de dados.

AWS CLI

Para gerenciar a senha do usuário principal com o RDS no Secrets Manager, especifique a opção `--manage-master-user-password` em um dos seguintes comandos da AWS CLI:

- create-db-instance
- modify-db-instance
- restore-db-instance-from-s3
- restore-db-instance-from-db-snapshot (somente RDS para Oracle)
- restore-db-instance-to-point-in-time (somente RDS para Oracle)

Quando você especifica a opção `--manage-master-user-password` nesses comandos, o RDS gera a senha do usuário principal e a gerencia durante todo o ciclo de vida no Secrets Manager.

Para criptografar o segredo, você pode especificar uma chave gerenciada pelo cliente ou usar a chave do KMS fornecida pelo Secrets Manager. Use a opção `--master-user-secret-kms-key-id` para especificar uma chave gerenciada pelo cliente. O identificador de chave do AWS KMS é o ARN da chave, o ID da chave, o ARN do alias ou o nome do alias da chave do KMS. Para usar uma chave do KMS em outra Conta da AWS, é necessário usar o ARN da chave ou o ARN do alias. Depois que o RDS estiver gerenciando as credenciais de banco de dados de uma instância de banco de dados, não será possível alterar a chave do KMS usada para criptografar o segredo.

Você pode escolher outras configurações para atender às suas necessidades. Para ter mais informações sobre as configurações disponíveis ao criar uma instância de banco de dados, consulte Configurações para instâncias de banco de dados. Para ter mais informações sobre as configurações disponíveis ao modificar uma instância de banco de dados, consulte Configurações para instâncias de banco de dados.

O exemplo a seguir cria uma instância de banco de dados e especifica que o RDS gerencie a senha do usuário principal no Secrets Manager. O segredo é criptografado usando a chave do KMS fornecida pelo Secrets Manager.

Example

Para Linux, macOS ou Unix:

```
aws rds create-db-instance \
    --db-instance-identifier mydbinstance \
    --engine mysql \
    --engine-version 8.0.39 \
    --db-instance-class db.r5b.large \
    --allocated-storage 200 \
    --master-username testUser \
    --manage-master-user-password
```

Para Windows:

```
aws rds create-db-instance ^
    --db-instance-identifier mydbinstance ^
    --engine mysql ^
    --engine-version 8.0.39 ^
    --db-instance-class db.r5b.large ^
    --allocated-storage 200 ^
    --master-username testUser ^
    --manage-master-user-password
```

API do RDS

Para especificar que o RDS gerencie a senha do usuário principal no Secrets Manager, defina o parâmetro `ManageMasterUserPassword` como `true` em uma das seguintes operações da API do RDS:

- CreateDBInstance
- ModifyDBInstance
- RestoreDBInstanceFromS3
- RestoreDBInstanceFromSnapshot (somente RDS para Oracle)
- RestoreDBInstanceToPointInTime (somente RDS para Oracle)

Quando você define o parâmetro `ManageMasterUserPassword` como `true` em uma dessas operações, o RDS gera a senha do usuário principal e a gerencia durante todo o ciclo de vida no Secrets Manager.

Para criptografar o segredo, você pode especificar uma chave gerenciada pelo cliente ou usar a chave do KMS fornecida pelo Secrets Manager. Use o parâmetro `MasterUserSecretKmsKeyId` para especificar uma chave gerenciada pelo cliente. O identificador de chave do AWS KMS é o ARN da chave, o ID da chave, o ARN do alias ou o nome do alias da chave do KMS. Para usar uma chave do KMS em outra Conta da AWS, é necessário usar o ARN da chave ou o ARN do alias. Depois que o RDS estiver gerenciando as credenciais de banco de dados de uma instância de banco de dados, não será possível alterar a chave do KMS usada para criptografar o segredo.

Gerenciar a senha do usuário principal para um banco de dados de locatário do RDS para Oracle com o Secrets Manager

Você pode configurar o gerenciamento pelo RDS da senha do usuário principal no Secrets Manager ao realizar as seguintes ações:

- Adicionar um banco de dados de locatários do RDS for Oracle à sua instância de CDB
- Modificando um banco de dados de locatários do RDS para Oracle

Você pode usar o console do RDS, a AWS CLI ou a API do RDS para realizar as ações anteriores.

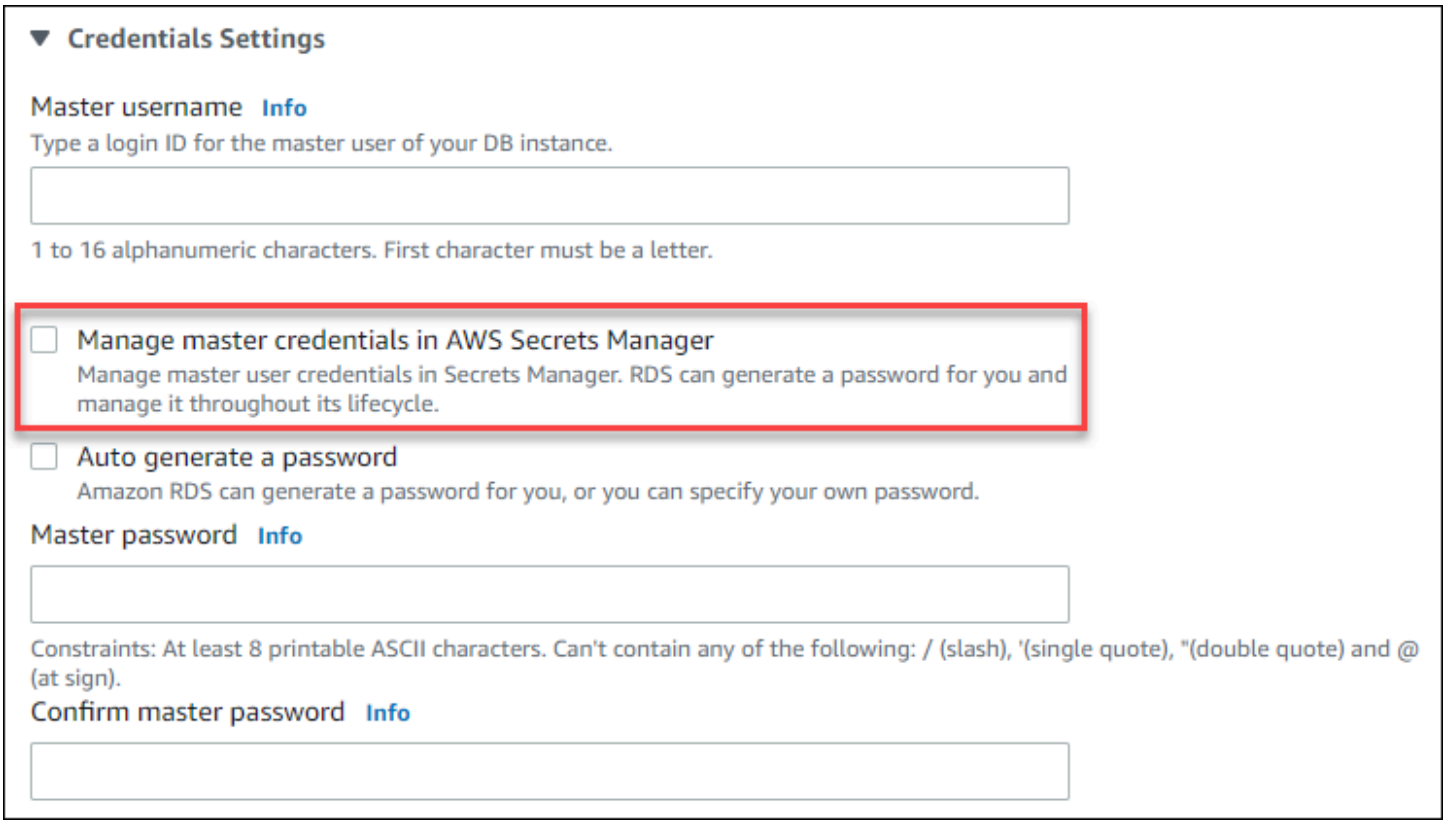
Console

Siga as instruções para criar ou modificar um banco de dados de locatário do RDS para Oracle com o console do RDS:

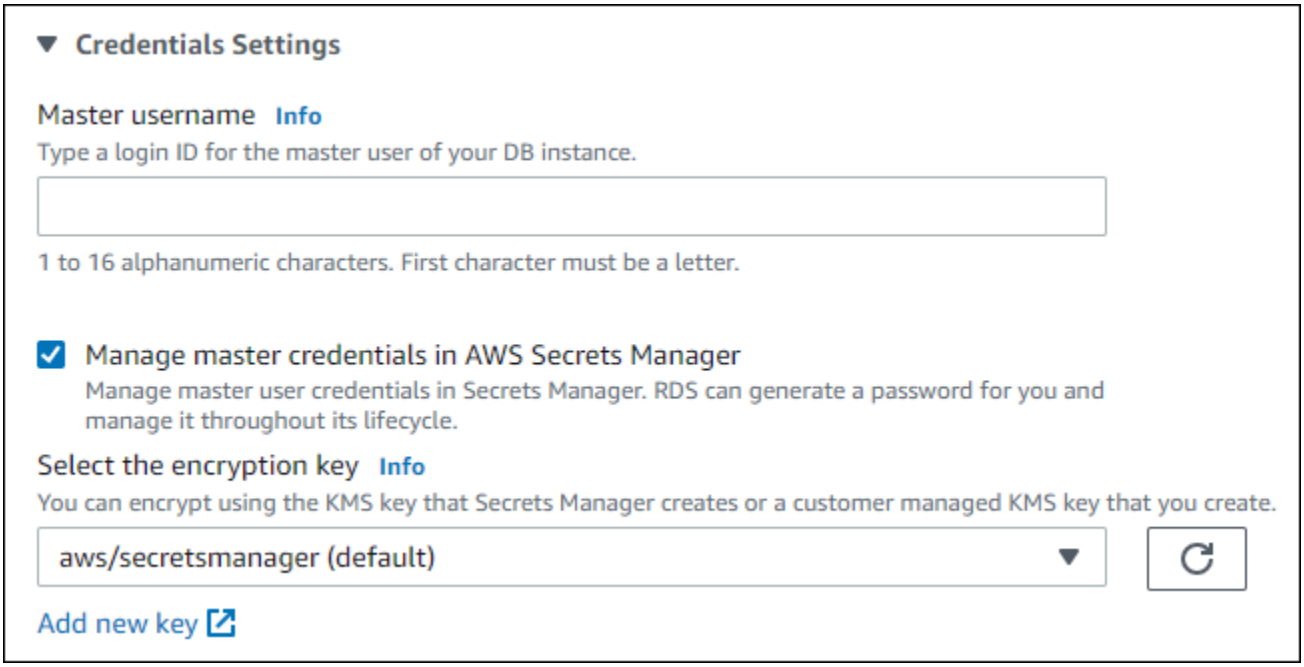
- Adicionar um banco de dados de locatários do RDS for Oracle à sua instância de CDB
- Modificando um banco de dados de locatários do RDS para Oracle

Ao usar o console do RDS para realizar uma das operações anteriores, você pode especificar que o RDS gerencie a senha principal no Secrets Manager. Ao criar um banco de dados de locatário, selecione Gerenciar credenciais principais no AWS Secrets Manager em Configurações de credenciais. Ao modificar um de banco de dados de locatário, selecione Gerenciar credenciais principais no AWS Secrets Manager em Configurações.

A imagem a seguir é um exemplo da configuração Gerenciar credenciais principais no AWS Secrets Manager quando você está criando um banco de dados de locatário.



Quando você seleciona essa opção, o RDS gera a senha do usuário principal e a gerencia durante todo o ciclo de vida no Secrets Manager.



Você pode optar por criptografar o segredo com uma chave do KMS fornecida pelo Secrets Manager ou com uma chave gerenciada pelo cliente criada por você. Depois que o RDS estiver gerenciando as credenciais de banco de dados para um banco de dados, não será possível alterar a chave do KMS usada para criptografar o segredo.

Você pode escolher outras configurações para atender às suas necessidades. Para ter mais informações sobre as configurações disponíveis ao criar um banco de dados de locatário, consulte Configurações para instâncias de banco de dados. Para ter mais informações sobre as configurações disponíveis ao modificar um banco de dados de locatário, consulte Configurações para instâncias de banco de dados.

AWS CLI

Para gerenciar a senha do usuário principal com o RDS no Secrets Manager, especifique a opção `--manage-master-user-password` em um dos seguintes comandos da AWS CLI:

- create-tenant-database
- modify-tenant-database

Quando você especifica a opção `--manage-master-user-password` no comando anterior, o RDS gera a senha do usuário principal e a gerencia durante todo o ciclo de vida no Secrets Manager.

Para criptografar o segredo, você pode especificar uma chave gerenciada pelo cliente ou usar a chave do KMS fornecida pelo Secrets Manager. Use a opção `--master-user-secret-kms-key-id` para especificar uma chave gerenciada pelo cliente. O identificador de chave do AWS KMS é o ARN da chave, o ID da chave, o ARN do alias ou o nome do alias da chave do KMS. Para usar uma chave do KMS em outra Conta da AWS, é necessário usar o ARN da chave ou o ARN do alias. Depois que o RDS estiver gerenciando as credenciais de banco de dados para um banco de dados, não será possível alterar a chave do KMS usada para criptografar o segredo.

Você pode escolher outras configurações para atender às suas necessidades. Para ter mais informações sobre as configurações disponíveis ao criar um banco de dados de locatário, consulte create-tenant-database. Para ter mais informações sobre as configurações disponíveis ao modificar um banco de dados de locatário, consulte modify-tenant-database.

O exemplo a seguir cria um banco de dados de locatário do RDS para Oracle e especifica que o RDS gerencie a senha do usuário principal no Secrets Manager. O segredo é criptografado usando a chave do KMS fornecida pelo Secrets Manager.

Example

Para Linux, macOS ou Unix:

```
aws rds create-tenant-database --region us-east-1 \
    --db-instance-identifier my-cdb-inst \
    --tenant-db-name mypdb2 \
    --master-username mypdb2-admin \
    --character-set-name UTF-16 \
    --manage-master-user-password
```

Para Windows:

```
aws rds create-tenant-database --region us-east-1 ^
    --db-instance-identifier my-cdb-inst ^
    --tenant-db-name mypdb2 ^
    --master-username mypdb2-admin ^
    --character-set-name UTF-16 ^
    --manage-master-user-password
```

API do RDS

Para especificar que o RDS gerencie a senha do usuário principal no Secrets Manager, defina o parâmetro `ManageMasterUserPassword` como `true` em uma das seguintes operações da API do RDS:

- CreateTenantDatabase
- ModifyTenantDatabase

Quando você define o parâmetro `ManageMasterUserPassword` como `true` em uma dessas operações, o RDS gera a senha do usuário principal e a gerencia durante todo o ciclo de vida no Secrets Manager.

Para criptografar o segredo, você pode especificar uma chave gerenciada pelo cliente ou usar a chave do KMS fornecida pelo Secrets Manager. Use o parâmetro `MasterUserSecretKmsKeyId` para especificar uma chave gerenciada pelo cliente. O identificador de chave do AWS KMS é o ARN da chave, o ID da chave, o ARN do alias ou o nome do alias da chave do KMS. Para usar uma chave do KMS em outra Conta da AWS, é necessário usar o ARN da chave ou o ARN do alias. Depois que o RDS estiver gerenciando as credenciais de banco de dados para um banco de dados, não será possível alterar a chave do KMS usada para criptografar o segredo.

Gerenciar a senha do usuário principal para um cluster de banco de dados multi-AZ com o Secrets Manager

Você pode configurar o gerenciamento pelo RDS da senha do usuário principal no Secrets Manager ao realizar as seguintes ações:

- Criar um cluster de banco de dados multi-AZ para o Amazon RDS
- Modificar um cluster de banco de dados multi-AZ para o Amazon RDS

Você pode usar o console do RDS, a AWS CLI ou a API do RDS para realizar essas ações.

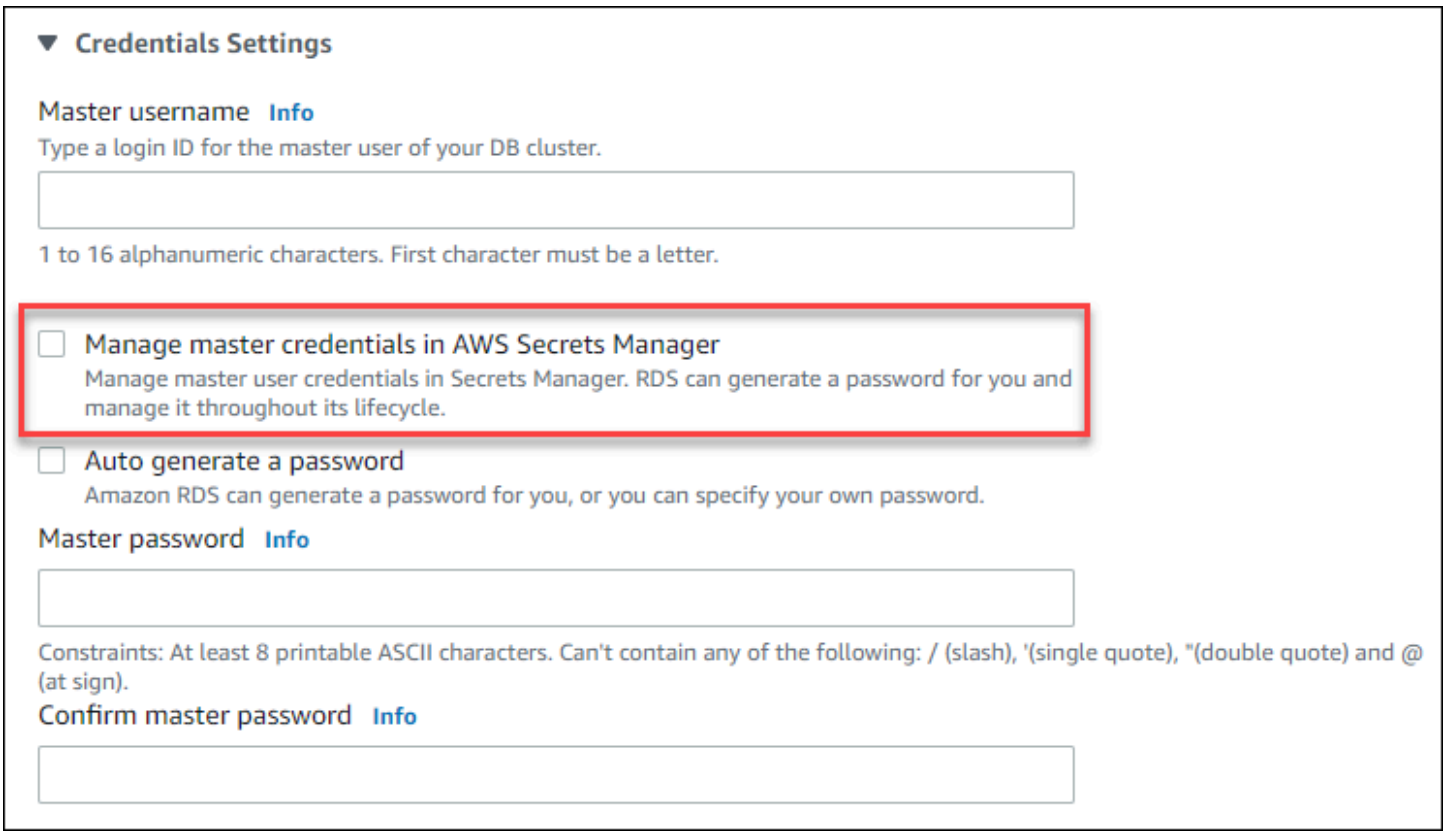
Console

Siga as instruções para criar ou modificar um cluster de banco de dados multi-AZ com o console do RDS:

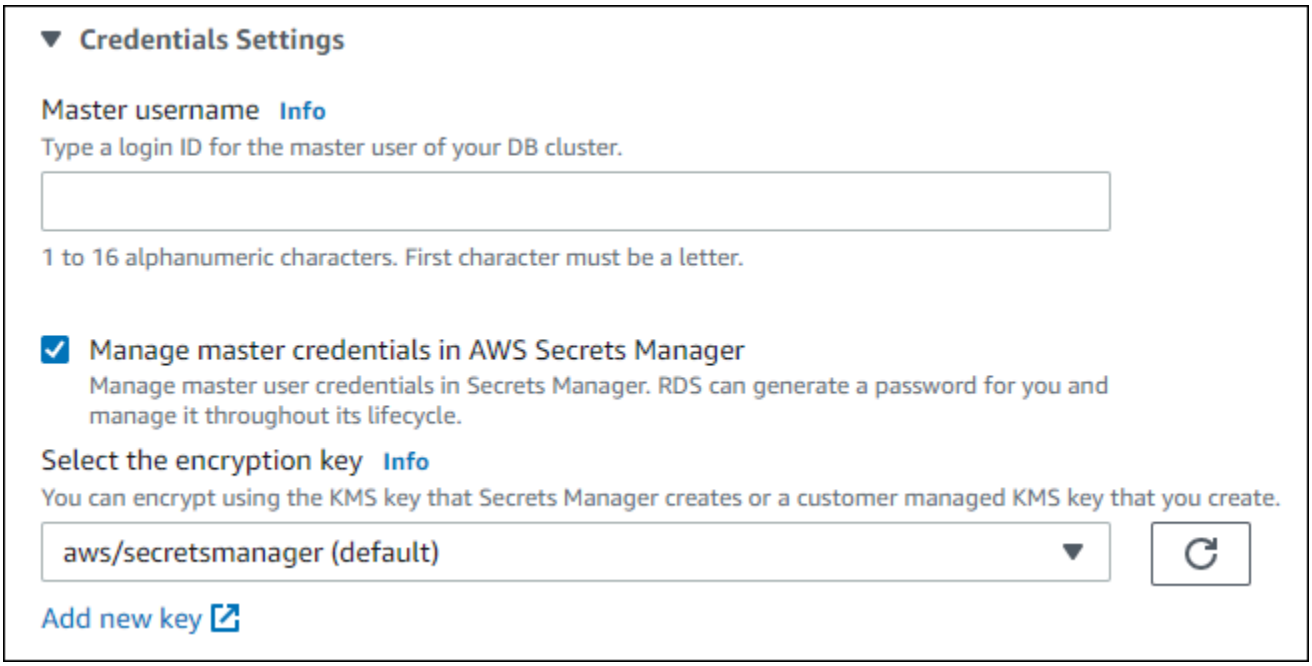
- Criar um cluster de banco de dados
- Modificar um cluster de banco de dados multi-AZ para o Amazon RDS

Ao usar o console do RDS para realizar uma dessas operações, você pode especificar que a senha do usuário principal seja gerenciada pelo RDS no Secrets Manager. Para fazer isso ao criar um cluster de banco de dados, selecione Gerenciar credenciais principais no AWS Secrets Manager em Configurações de credenciais. Quando estiver modificando um cluster de banco de dados, selecione Gerenciar credenciais principais no AWS Secrets Manager em Configurações.

A imagem a seguir é um exemplo da configuração Gerenciar credenciais principais no AWS Secrets Manager quando você está criando um cluster de banco de dados.



Quando você seleciona essa opção, o RDS gera a senha do usuário principal e a gerencia durante todo o ciclo de vida no Secrets Manager.



Você pode optar por criptografar o segredo com uma chave do KMS fornecida pelo Secrets Manager ou com uma chave gerenciada pelo cliente criada por você. Depois que o RDS estiver gerenciando

as credenciais de banco de dados de um cluster de banco de dados, não será possível alterar a chave do KMS usada para criptografar o segredo.

Você pode escolher outras configurações para atender às suas necessidades.

Para ter mais informações sobre as configurações disponíveis ao criar um cluster de banco de dados multi-AZ, consulte Configurações para criar clusters de banco de dados multi-AZ. Para ter mais informações sobre as configurações disponíveis ao modificar um cluster de banco de dados multi-AZ, consulte Configurações para modificar clusters de banco de dados multi-AZ.

AWS CLI

Para especificar que o RDS gerencie a senha do usuário principal no Secrets Manager, especifique a opção `--manage-master-user-password` em um dos seguintes comandos:

- create-db-cluster
- modify-db-cluster

Quando você especifica a opção `--manage-master-user-password` nesses comandos, o RDS gera a senha do usuário principal e a gerencia durante todo o ciclo de vida no Secrets Manager.

Para criptografar o segredo, você pode especificar uma chave gerenciada pelo cliente ou usar a chave do KMS fornecida pelo Secrets Manager. Use a opção `--master-user-secret-kms-key-id` para especificar uma chave gerenciada pelo cliente. O identificador de chave do AWS KMS é o ARN da chave, o ID da chave, o ARN do alias ou o nome do alias da chave do KMS. Para usar uma chave do KMS em outra Conta da AWS, é necessário usar o ARN da chave ou o ARN do alias. Depois que o RDS estiver gerenciando as credenciais de banco de dados de um cluster de banco de dados, não será possível alterar a chave do KMS usada para criptografar o segredo.

Você pode escolher outras configurações para atender às suas necessidades.

Para ter mais informações sobre as configurações disponíveis ao criar um cluster de banco de dados multi-AZ, consulte Configurações para criar clusters de banco de dados multi-AZ. Para ter mais informações sobre as configurações disponíveis ao modificar um cluster de banco de dados multi-AZ, consulte Configurações para modificar clusters de banco de dados multi-AZ.

Este exemplo cria um cluster de banco de dados multi-AZ e especifica que o RDS gerencie a senha no Secrets Manager. O segredo é criptografado usando a chave do KMS fornecida pelo Secrets Manager.

Example

Para Linux, macOS ou Unix:

```
aws rds create-db-cluster \
   --db-cluster-identifier mysql-multi-az-db-cluster \
   --engine mysql \
   --engine-version 8.0.39  \
   --backup-retention-period 1  \
   --allocated-storage 4000 \
   --storage-type io1 \
   --iops 10000 \
   --db-cluster-instance-class db.r6gd.xlarge \
   --master-username testUser \
   --manage-master-user-password
```

Para Windows:

```
aws rds create-db-cluster ^
   --db-cluster-identifier mysql-multi-az-db-cluster ^
   --engine mysql ^
   --engine-version 8.0.39 ^
   --backup-retention-period 1 ^
   --allocated-storage 4000 ^
   --storage-type io1 ^
   --iops 10000 ^
   --db-cluster-instance-class db.r6gd.xlarge ^
   --master-username testUser ^
   --manage-master-user-password
```

API do RDS

Para especificar que o RDS gerencie a senha do usuário principal no Secrets Manager, defina o parâmetro `ManageMasterUserPassword` como `true` em uma das seguintes operações:

- CreateDBCluster
- ModifyDBCluster

Quando você define o parâmetro `ManageMasterUserPassword` como `true` em uma dessas operações, o RDS gera a senha do usuário principal e a gerencia durante todo o ciclo de vida no Secrets Manager.

Para criptografar o segredo, você pode especificar uma chave gerenciada pelo cliente ou usar a chave do KMS fornecida pelo Secrets Manager. Use o parâmetro `MasterUserSecretKmsKeyId` para especificar uma chave gerenciada pelo cliente. O identificador de chave do AWS KMS é o ARN da chave, o ID da chave, o ARN do alias ou o nome do alias da chave do KMS. Para usar uma chave do KMS em outra Conta da AWS, é necessário usar o ARN da chave ou o ARN do alias. Depois que o RDS estiver gerenciando as credenciais de banco de dados de um cluster de banco de dados, não será possível alterar a chave do KMS usada para criptografar o segredo.

Alternar o segredo da senha do usuário principal para uma instância de banco de dados

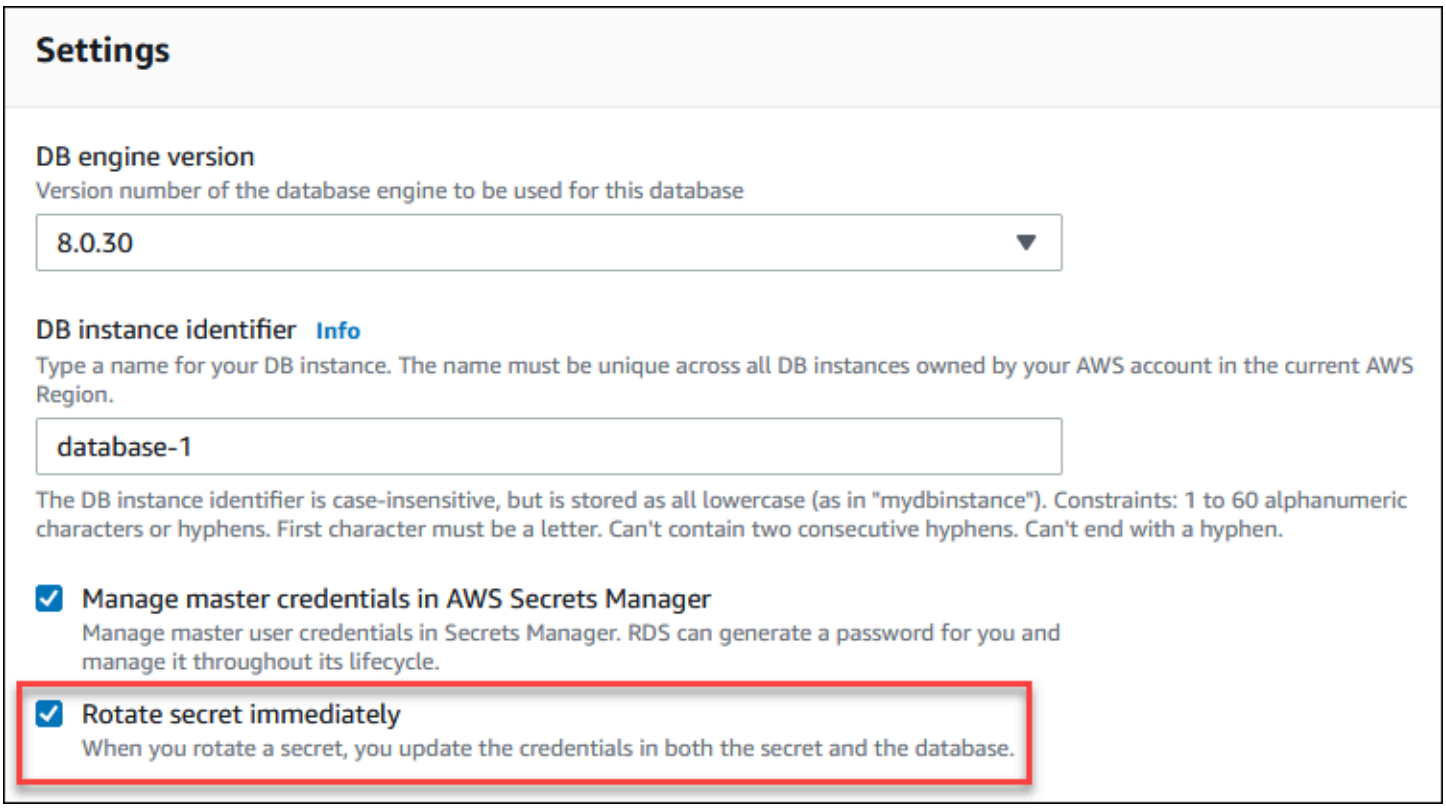
Quando o RDS alterna um segredo de senha do usuário principal, o Secrets Manager gera uma nova versão secreta para o segredo existente. A nova versão do segredo contém a nova senha do usuário principal. O Amazon RDS altera a senha do usuário principal da instância de banco de dados para corresponder à nova versão do segredo.

Você pode alternar um segredo imediatamente em vez de esperar por uma alternância programada. Para alternar o segredo de uma senha do usuário principal no Secrets Manager, modifique a instância de banco de dados. Para ter mais informações sobre como modificar uma instância de banco de dados , consulte Modificar uma instância de banco de dados do Amazon RDS.

Você pode alternar o segredo de uma senha do usuário principal imediatamente com o console do RDS, a AWS CLI ou a API do RDS. A nova senha tem sempre 28 caracteres e contém pelo menos um caractere maiúsculo e um minúsculo, um número e um sinal de pontuação.

Console

Para alternar o segredo de uma senha do usuário principal usando o console do RDS, modifique a instância de banco de dados e selecione Rotate secret immediately (Alternar segredo imediatamente) em Settings (Configurações).



Siga as instruções para modificar uma instância de banco de dados com o console em Modificar uma instância de banco de dados do Amazon RDS. Você deve escolher Apply immediately (Aplicar imediatamente) na página de confirmação.

AWS CLI

Para alternar o segredo de uma senha do usuário principal usando a AWS CLI, utilize o comando modify-db-instance e especifique a opção `--rotate-master-user-password`. Você deve especificar a opção `--apply-immediately` ao alternar a senha principal.

Este exemplo alterna o segredo de uma senha do usuário principal.

Example

Para Linux, macOS ou Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --rotate-master-user-password \
    --apply-immediately
```

Para Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --rotate-master-user-password ^
    --apply-immediately
```

API do RDS

Você pode alternar o segredo de uma senha do usuário principal utilizando a operação ModifyDBInstance e definindo o parâmetro `RotateMasterUserPassword` como `true`. Você deve definir o parâmetro `ApplyImmediately` como `true` ao alternar a senha principal.

Alternar o segredo de uma senha principal do usuário para um cluster de banco de dados multi-AZ

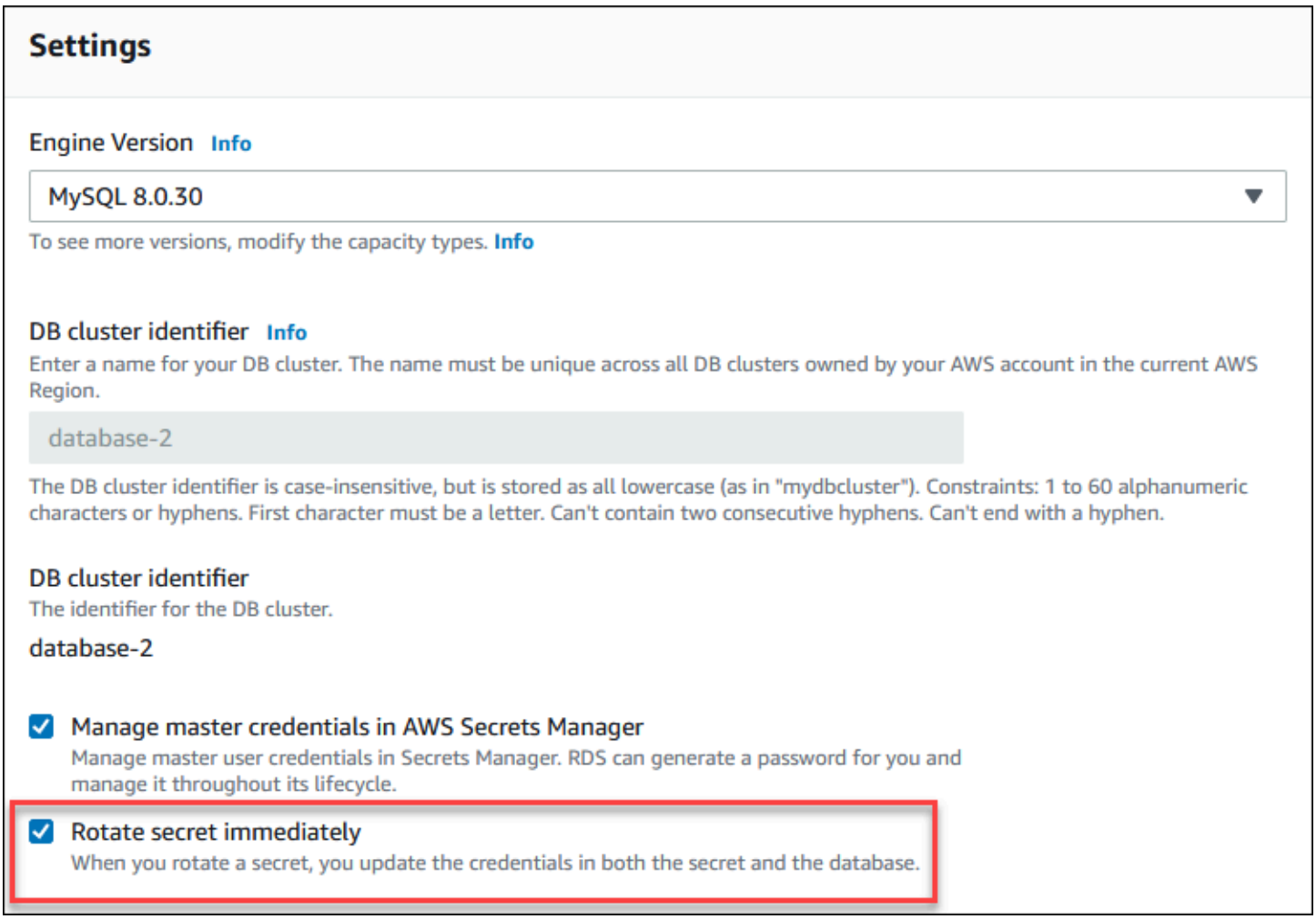
Quando o RDS alterna um segredo de senha do usuário principal, o Secrets Manager gera uma nova versão para o segredo existente. A nova versão do segredo contém a nova senha do usuário principal. O Amazon RDS altera a senha do usuário principal do cluster de banco de dados multi-AZ para corresponder à nova versão do segredo.

Você pode alternar um segredo imediatamente em vez de esperar por uma alternância programada. Para alternar o segredo de uma senha do usuário principal no Secrets Manager, modifique o cluster de banco de dados multi-AZ. Para obter informações sobre como modificar clusters de banco de dados multi-AZ, consulte Modificar um cluster de banco de dados multi-AZ para o Amazon RDS.

Você pode alternar o segredo de uma senha do usuário principal imediatamente com o console do RDS, a AWS CLI ou a API do RDS. A nova senha tem sempre 28 caracteres e contém pelo menos um caractere maiúsculo e um minúsculo, um número e um sinal de pontuação.

Console

Para alternar o segredo de uma senha do usuário principal usando o console do RDS, modifique o cluster de banco de dados multi-AZ e selecione Rotate secret immediately (Alternar segredo imediatamente) em Settings (Configurações).



Siga as instruções para modificar um cluster de banco de dados multi-AZ com o console do RDS em Modificar um cluster de banco de dados multi-AZ para o Amazon RDS. Você deve escolher Apply immediately (Aplicar imediatamente) na página de confirmação.

AWS CLI

Para alternar o segredo de uma senha do usuário principal usando a AWS CLI, utilize o comando modify-db-cluster e especifique a opção `--rotate-master-user-password`. Você deve especificar a opção `--apply-immediately` ao alternar a senha principal.

Este exemplo alterna o segredo de uma senha do usuário principal.

Example

Para Linux, macOS ou Unix:

```
aws rds modify-db-cluster \
```

```
    --db-cluster-identifier mydbcluster \
    --rotate-master-user-password \
    --apply-immediately
```

Para Windows:

```
aws rds modify-db-cluster ^
    --db-cluster-identifier mydbcluster ^
    --rotate-master-user-password ^
    --apply-immediately
```

API do RDS

Você pode alternar o segredo de uma senha do usuário principal usando a operação ModifyDBCluster e definindo o parâmetro `RotateMasterUserPassword` como `true`. Você deve definir o parâmetro `ApplyImmediately` como `true` ao alternar a senha principal.

Visualizar os detalhes sobre um segredo para uma instância de banco de dados

É possível recuperar seus segredos usando o console (https://console.aws.amazon.com/secretsmanager/) ou a AWS CLI (comando get-secret-value do Secrets Manager).

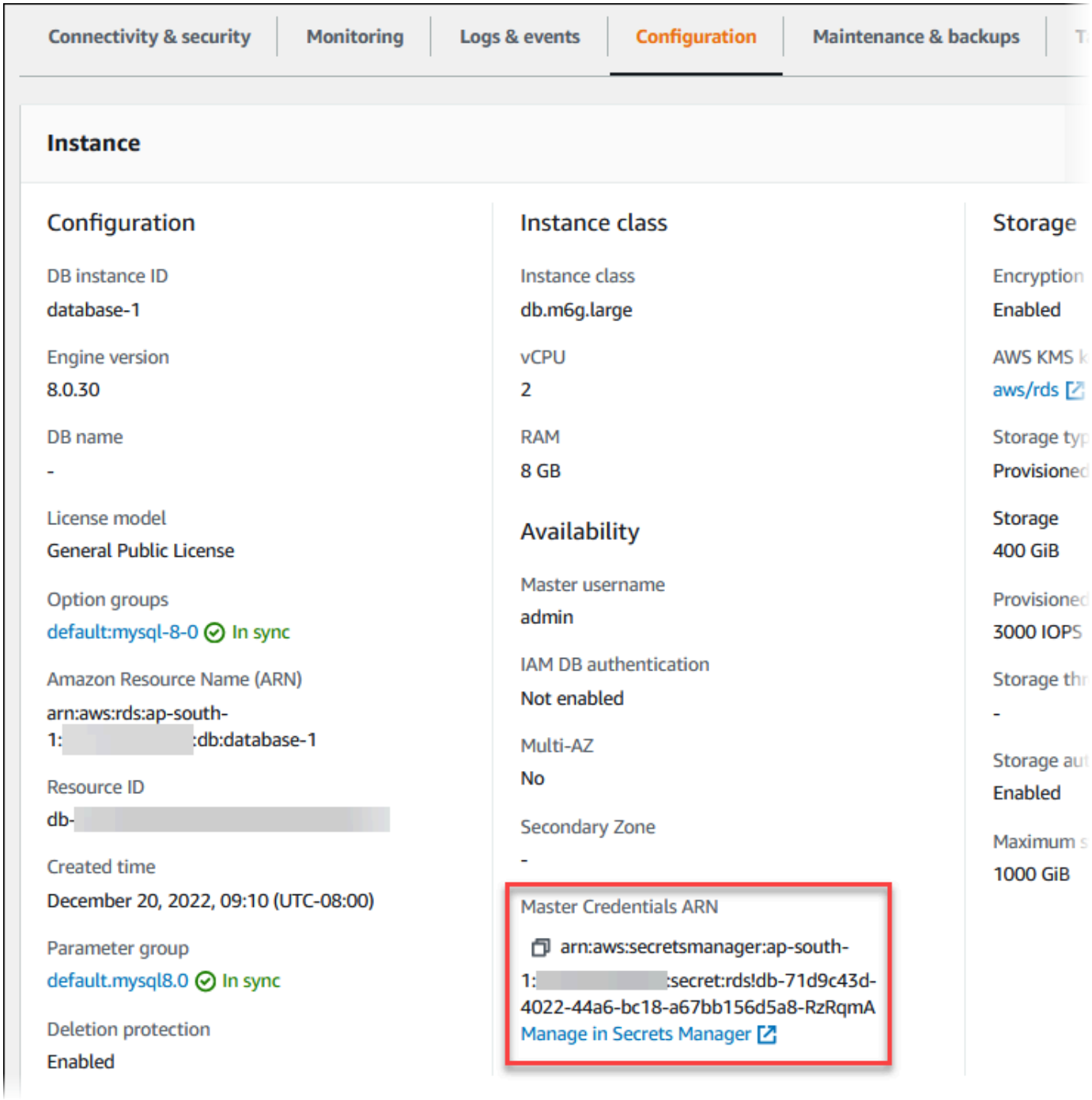
Você pode encontrar o nome do recurso da Amazon (ARN) de um segredo gerenciado pelo RDS no Secrets Manager com o console do RDS, a AWS CLI o ou a API do RDS.

Console

Como ver os detalhes sobre um segredo gerenciado pelo RDS no Secrets Manager

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Bancos de dados.
3. Escolha o nome da instância de banco de dados para mostrar os detalhes.
4. Escolha a guia Configuração.

   Em Master Credentials ARN (ARN das credenciais principais), você pode ver o ARN do segredo.



Você pode seguir o link Manage in Secrets Manager (Gerenciar no Secrets Manager) para visualizar e gerenciar o segredo no console do Secrets Manager.

AWS CLI

Você pode usar o comando da CLI do RDS describe-db-instances para encontrar as seguintes informações sobre um segredo gerenciado pelo RDS no Secrets Manager:

- `SecretArn`: o ARN do segredo
- `SecretStatus`: o status do segredo

  Os valores de status possíveis são os seguintes:

  - `creating`: o segredo está sendo criado.
  - `active`: o segredo está disponível para uso e alternância normais.
  - `rotating`: o segredo está sendo alternado.
  - `impaired`: o segredo pode ser usado para acessar as credenciais do banco de dados, mas não pode ser alternado. Um segredo pode ter esse status se, por exemplo, as permissões forem alteradas para que o RDS não possa mais acessar o segredo nem a chave do KMS do segredo.

    Quando um segredo tem esse status, você pode corrigir a condição que o causou. Se você corrigir a condição que causou o status, ele permanecerá `impaired` até a próxima alternância. Como alternativa, você pode modificar a instância de banco de dados para desativar o gerenciamento automático das credenciais do banco de dados e, depois, modificar a instância de banco de dados novamente para ativar o gerenciamento automático das credenciais do banco de dados. Para modificar a instância de banco de dados use a opção `--manage-master-user-password` no comando modify-db-instance.
- `KmsKeyId`: o ARN da chave do KMS usada para criptografar o segredo

Especifique a opção `--db-instance-identifier` para mostrar a saída de uma instância de banco de dados específica. Este exemplo mostra a saída de um segredo usado por uma instância de banco de dados.

Example

```
aws rds describe-db-instances --db-instance-identifier mydbinstance
```

Veja a seguir a saída de exemplo de um segredo:

```
"MasterUserSecret": {
                "SecretArn": "arn:aws:secretsmanager:eu-west-1:123456789012:secret:rds!
db-033d7456-2c96-450d-9d48-f5de3025e51c-xmJRDx",
                "SecretStatus": "active",
                "KmsKeyId": "arn:aws:kms:eu-
west-1:123456789012:key/0987dcba-09fe-87dc-65ba-ab0987654321"
            }
```

Quando você tiver o ARN do segredo, poderá visualizar detalhes sobre o segredo usando o comando da CLI get-secret-value do Secrets Manager.

Este exemplo mostra os detalhes do segredo na saída de exemplo anterior.

Example

Para Linux, macOS ou Unix:

```
aws secretsmanager get-secret-value \
    --secret-id 'arn:aws:secretsmanager:eu-west-1:123456789012:secret:rds!
db-033d7456-2c96-450d-9d48-f5de3025e51c-xmJRDx'
```

Para Windows:

```
aws secretsmanager get-secret-value ^
    --secret-id 'arn:aws:secretsmanager:eu-west-1:123456789012:secret:rds!
db-033d7456-2c96-450d-9d48-f5de3025e51c-xmJRDx'
```

API do RDS

Você pode visualizar o ARN, o status e a chave do KMS de um segredo gerenciado pelo RDS no Secrets Manager usando a operação DescribeDBInstances e definindo o parâmetro `DBInstanceIdentifier` como um identificador de instância de banco de dados. Detalhes sobre o segredo estão incluídos na saída.

Quando você tiver o ARN do segredo, poderá visualizar detalhes sobre o segredo usando o comando da CLI GetSecretValue do Secrets Manager.

Visualizar os detalhes sobre um segredo para um cluster de banco de dados multi-AZ

É possível recuperar seus segredos usando o console (https://console.aws.amazon.com/secretsmanager/) ou a AWS CLI (comando get-secret-value do Secrets Manager).

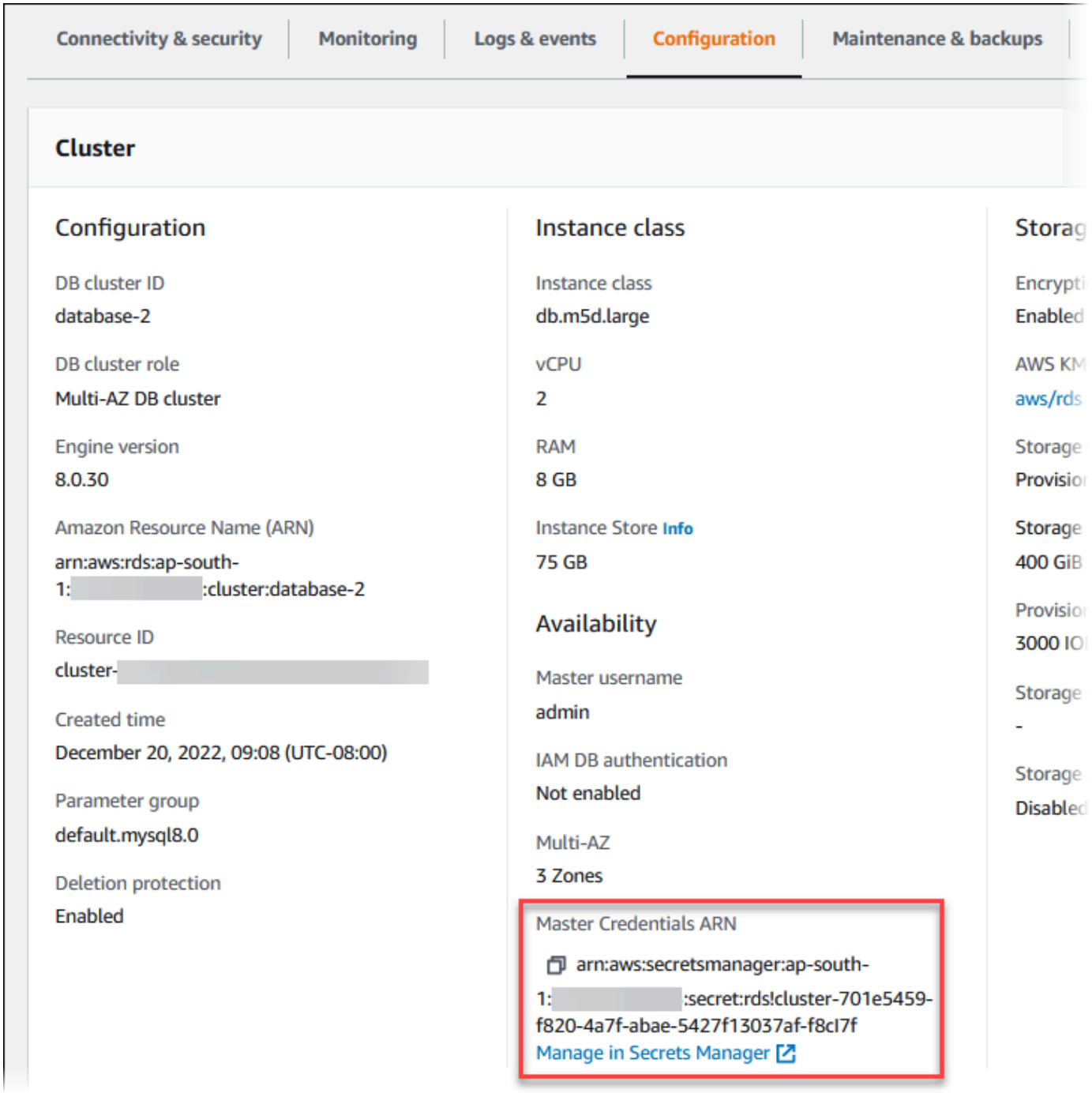
Você pode encontrar o nome do recurso da Amazon (ARN) de um segredo gerenciado pelo RDS no Secrets Manager com o console do RDS, a AWS CLI o ou a API do RDS.

Console

Como ver os detalhes sobre um segredo gerenciado pelo RDS no Secrets Manager

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Bancos de dados.
3. Selecione o nome do cluster de banco de dados multi-AZ para mostrar os detalhes.
4. Escolha a guia Configuração.

   Em Master Credentials ARN (ARN das credenciais principais), você pode ver o ARN do segredo.



Você pode seguir o link Manage in Secrets Manager (Gerenciar no Secrets Manager) para visualizar e gerenciar o segredo no console do Secrets Manager.

AWS CLI

Você pode usar o comando describe-db-clusters da AWS CLI do RDS para encontrar as seguintes informações sobre um segredo gerenciado pelo RDS no Secrets Manager:

- `SecretArn`: o ARN do segredo
- `SecretStatus`: o status do segredo

  Os valores de status possíveis são os seguintes:

  - `creating`: o segredo está sendo criado.
  - `active`: o segredo está disponível para uso e alternância normais.
  - `rotating`: o segredo está sendo alternado.
  - `impaired`: o segredo pode ser usado para acessar as credenciais do banco de dados, mas não pode ser alternado. Um segredo pode ter esse status se, por exemplo, as permissões forem alteradas para que o RDS não possa mais acessar o segredo nem a chave do KMS do segredo.

    Quando um segredo tem esse status, você pode corrigir a condição que o causou. Se você corrigir a condição que causou o status, ele permanecerá `impaired` até a próxima alternância. Como alternativa, você pode modificar o cluster de banco de dados para desativar o gerenciamento automático das credenciais do banco de dados e, depois, modificar o cluster de banco de dados novamente para ativar o gerenciamento automático das credenciais do banco de dados. Para modificar o cluster de banco de dados, use a opção `--manage-master-user-password` no comando modify-db-cluster.
- `KmsKeyId`: o ARN da chave do KMS usada para criptografar o segredo

Especifique a opção `--db-cluster-identifier` para mostrar a saída de um cluster de banco de dados específico. Este exemplo mostra a saída de um segredo usado por um cluster de banco de dados.

Example

```
aws rds describe-db-clusters --db-cluster-identifier mydbcluster
```

O exemplo a seguir mostra a saída de um segredo:

```
"MasterUserSecret": {
                "SecretArn": "arn:aws:secretsmanager:eu-west-1:123456789012:secret:rds!
cluster-033d7456-2c96-450d-9d48-f5de3025e51c-xmJRDx",
                "SecretStatus": "active",
                "KmsKeyId": "arn:aws:kms:eu-
west-1:123456789012:key/0987dcba-09fe-87dc-65ba-ab0987654321"
            }
```

Quando você tiver o ARN do segredo, poderá visualizar detalhes sobre o segredo usando o comando da CLI get-secret-value do Secrets Manager.

Este exemplo mostra os detalhes do segredo na saída de exemplo anterior.

Example

Para Linux, macOS ou Unix:

```
aws secretsmanager get-secret-value \
    --secret-id 'arn:aws:secretsmanager:eu-west-1:123456789012:secret:rds!
cluster-033d7456-2c96-450d-9d48-f5de3025e51c-xmJRDx'
```

Para Windows:

```
aws secretsmanager get-secret-value ^
    --secret-id 'arn:aws:secretsmanager:eu-west-1:123456789012:secret:rds!
cluster-033d7456-2c96-450d-9d48-f5de3025e51c-xmJRDx'
```

API do RDS

Você pode visualizar o ARN, o status e a chave do KMS de um segredo gerenciado pelo RDS no Secrets Manager usando a operação DescribeDBClusters do RDS e definindo o parâmetro `DBClusterIdentifier` como um identificador de cluster de banco de dados. Detalhes sobre o segredo estão incluídos na saída.

Quando você tiver o ARN do segredo, poderá visualizar detalhes sobre o segredo usando o comando da CLI GetSecretValue do Secrets Manager.

Visualizar detalhes sobre um segredo para um banco de dados de locatários

É possível recuperar seus segredos usando o console (https://console.aws.amazon.com/secretsmanager/) ou a AWS CLI (comando get-secret-value do Secrets Manager).

É possível encontrar o nome do recurso da Amazon (ARN) de um segredo gerenciado pelo Amazon RDS no AWS Secrets Manager com o console do Amazon RDS, a AWS CLI o ou a API do Amazon RDS.

Console

Como visualizar os detalhes sobre um segredo gerenciado pelo Amazon RDS no AWS Secrets Manager para um banco de dados de locatários

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Bancos de dados.
3. Escolha o nome da instância de banco de dados que contém o banco de dados de locatários para mostrar os respectivos detalhes.
4. Escolha a guia Configuração.

   Na seção Bancos de dados de locatários, localize o banco de dados de locatários e visualize o ARN de credenciais principais.

   Você pode seguir o link Manage in Secrets Manager (Gerenciar no Secrets Manager) para visualizar e gerenciar o segredo no console do Secrets Manager.

AWS CLI

Você pode usar o comando describe-tenant-databases da AWS CLI do Amazon RDS para encontrar as seguintes informações sobre um segredo gerenciado pelo Amazon RDS no AWS Secrets Manager para um banco de dados de locatários:

- `SecretArn`: o ARN do segredo
- `SecretStatus`: o status do segredo

  Os valores de status possíveis são os seguintes:

  - `creating`: o segredo está sendo criado.
  - `active`: o segredo está disponível para uso e alternância normais.
  - `rotating`: o segredo está sendo alternado.
  - `impaired`: o segredo pode ser usado para acessar as credenciais do banco de dados, mas não pode ser alternado. Um segredo pode ter esse status se, por exemplo, as permissões forem alteradas para que o Amazon RDS não possa mais acessar o segredo nem a chave do KMS do segredo.

Quando um segredo tem esse status, você pode corrigir a condição que o causou. Se você corrigir a condição que causou o status, ele permanecerá `impaired` até a próxima alternância. Você também pode modificar o banco de dados de locatários para desativar o gerenciamento automático das credenciais do banco de dados e, depois, modificar novamente o banco de dados de locatários para ativar o gerenciamento automático das credenciais do banco de dados. Para modificar o banco de dados de locatários, use a opção `--manage-master-user-password` no comando modify-tenant-database.

- `KmsKeyId`: o ARN da chave do KMS usada para criptografar o segredo

Especifique a opção `--db-instance-identifier` para mostrar a saída para bancos de dados de locatários em uma instância de banco de dados específica. Você também pode especificar a opção `--tenant-db-name` para mostrar a saída para um banco de dados de locatários específico. Este exemplo mostra a saída para um segredo usado por um banco de dados de locatários.

Example

```
aws rds describe-tenant-databases \
    --db-instance-identifier database-3 \
    --query "TenantDatabases[0].MasterUserSecret"
```

Veja a seguir a saída de exemplo de um segredo:

```
{
    "SecretArn": "arn:aws:secretsmanager:us-east-2:123456789012:secret:rds!db-ABC123",
    "SecretStatus": "active",
    "KmsKeyId": "arn:aws:kms:us-east-2:123456789012:key/aa11bb22-####-####-####-
fedcba123456"
}
```

Quando você tiver o ARN do segredo, poderá visualizar detalhes sobre o segredo usando o comando get-secret-value da AWS CLI do Secrets Manager.

Este exemplo mostra os detalhes do segredo na saída de exemplo anterior.

Example

Para Linux, macOS ou Unix:

```
aws secretsmanager get-secret-value \
```

```
    --secret-id 'arn:aws:secretsmanager:us-east-2:123456789012:secret:rds!db-ABC123'
```

Para Windows:

```
aws secretsmanager get-secret-value ^
    --secret-id 'arn:aws:secretsmanager:us-east-2:123456789012:secret:rds!db-ABC123'
```

API do Amazon RDS

Você pode visualizar o ARN, o status e a chave do KMS de um segredo gerenciado pelo Amazon RDS no AWS Secrets Manager usando a operação DescribeTenantDatabases e definindo o parâmetro `DBInstanceIdentifier` como um identificador de instância de banco de dados. Você também pode definir o parâmetro `TenantDBName` com um nome específico de banco de dados de locatários. Detalhes sobre o segredo estão incluídos na saída.

Quando você tiver o ARN do segredo, poderá visualizar detalhes sobre o segredo usando o comando da CLI GetSecretValue do Secrets Manager.

Disponibilidade de regiões e versões

A disponibilidade e a compatibilidade de recursos variam entre versões específicas de cada mecanismo de banco de dados e entre Regiões da AWS. Para ter mais informações sobre a disponibilidade de versões e regiões com a integração do Secrets Manager com o Amazon RDS, consulte Regiões e mecanismos de banco de dados compatíveis com a integração do Secrets Manager com o Amazon RDS.

Proteção de dados no Amazon RDS

O modelo de responsabilidade compartilhada da AWS aplica-se à proteção de dados no Amazon Relational Database Service. Conforme descrito nesse modelo, a AWS é responsável por proteger a infraestrutura global que executa toda a Nuvem AWS. Você é responsável por manter o controle sobre o conteúdo hospedado nessa infraestrutura. Você também é responsável pelas tarefas de configuração e gerenciamento de segurança dos Serviços da AWS que usa. Para obter mais informações sobre a privacidade de dados, consulte as Data Privacy FAQ. Para obter mais informações sobre a proteção de dados na Europa, consulte a postagem do blog AWS Shared Responsibility Model and RGPD no Blog de segurança da AWS.

Para fins de proteção de dados, recomendamos que você proteja as credenciais da Conta da AWS e configure as contas de usuário individuais com AWS IAM Identity Center ou AWS Identity and Access

Management (IAM). Dessa maneira, cada usuário receberá apenas as permissões necessárias para cumprir suas obrigações de trabalho. Recomendamos também que você proteja seus dados das seguintes formas:

- Use uma autenticação multifator (MFA) com cada conta.
- Use SSL/TLS para se comunicar com os recursos da AWS. Exigimos TLS 1.2 e recomendamos TLS 1.3.
- Configure os logs de API e atividade do usuário com AWS CloudTrail. Para obter informações sobre como usar as trilhas do CloudTrail para capturar atividades da AWS, consulte Working with CloudTrail trails no Guia do usuário do AWS CloudTrail.
- Use as soluções de criptografia AWS, juntamente com todos os controles de segurança padrão em Serviços da AWS.
- Use serviços gerenciados de segurança avançada, como o Amazon Macie, que ajuda a localizar e proteger dados sigilosos armazenados no Amazon S3.
- Se você precisar de módulos criptográficos validados pelo FIPS 140-3 ao acessar a AWS por meio de uma interface de linha de comandos ou de uma API, use um endpoint do FIPS. Para obter mais informações sobre os endpoints FIPS disponíveis, consulte Federal Information Processing Standard (FIPS) 140-3.

É altamente recomendável que nunca sejam colocadas informações confidenciais ou sigilosas, como endereços de e-mail de clientes, em tags ou campos de formato livre, como um campo Nome. Isso também vale para o uso do Amazon RDS ou de outros Serviços da AWS com o console, a API, a AWS CLI ou os AWS SDKs. Quaisquer dados inseridos em tags ou em campos de texto de formato livre usados para nomes podem ser usados para logs de faturamento ou de diagnóstico. Se você fornecer um URL para um servidor externo, é fortemente recomendável que não sejam incluídas informações de credenciais no URL para validar a solicitação nesse servidor.

Tópicos

- Proteção de dados usando criptografia
- Privacidade do tráfego entre redes

Proteção de dados usando criptografia

É possível habilitar a criptografia para recursos do banco de dados. Você também pode criptografar conexões com instâncias de banco de dados.

Tópicos

- Criptografar recursos do Amazon RDS
- Gerenciamento de AWS KMS key
- Usar SSL/TLS para criptografar uma conexão com uma instância ou um cluster de banco de dados
- Alternar o certificado SSL/TLS

Criptografar recursos do Amazon RDS

O Amazon RDS pode criptografar instânciasde banco de dados do Amazon RDS. Os dados que são criptografados em repouso incluem o armazenamento subjacente para instâncias de banco de dados, seus backups automatizados, réplicas de leitura e snapshots.

Os clusters de de banco de dados Amazon RDS usam o algoritmo de criptografia AES-256 padrão do setor para criptografar seus dados no servidor que hospeda os clusters de de banco de dados Amazon RDS. Após a criptografia dos seus dados, o Amazon RDS lida com a autenticação do acesso e a decodificação dos seus dados de forma transparente com um mínimo impacto sobre a performance. Você não precisa modificar suas aplicações cliente de banco de dados para usar a criptografia.

> **Note**
>
> Para instâncias, os dados em trânsito entre a origem e as réplicas de leitura são criptografados, até mesmo quando a replicação ocorre entre regiões da AWS.

Tópicos

- Visão geral da criptografia de recursos do Amazon RDS
- Criptografar uma instância de banco de dados
- Determinar se a criptografia está habilitada para uma instância de banco de dados
- Disponibilidade da criptografia do Amazon RDS
- Criptografia em trânsito
- Limitações das instâncias de banco de dados criptografados do Amazon RDS

Visão geral da criptografia de recursos do Amazon RDS

As instâncias de banco de dados criptografadas do Amazon RDS fornecem uma camada adicional de proteção de dados, protegendo seus dados contra o acesso não autorizado ao armazenamento subjacente. Use a criptografia do Amazon RDS para aumentar a proteção de dados nas aplicações implantadas na nuvem e cumprir os requisitos de conformidade para criptografia em repouso.
Em uma instância de banco de dados criptografada do Amazon RDS, todos os logs, backups e snapshots são criptografados. Consulte mais informações sobre a disponibilidade e as limitações da criptografia em Disponibilidade da criptografia do Amazon RDS e Limitações das instâncias de banco de dados criptografados do Amazon RDS.

O Amazon RDS usa uma chave do AWS Key Management Service para criptografar esses recursos. O AWS KMS combina hardware e software seguros e altamente disponíveis para fornecer um sistema de gerenciamento de chaves escalado para a nuvem. É possível usar uma Chave gerenciada pela AWS ou criar chaves gerenciadas pelo cliente.

Ao criar uma instância de banco de dados criptografada, você pode escolher uma chave gerenciada pelo cliente ou a Chave gerenciada pela AWS para que o Amazon RDS criptografe a sua instância de banco de dados. Se você não especificar o identificador de chave para uma chave gerenciada pelo cliente, o Amazon RDS usará a Chave gerenciada pela AWS para a sua nova instância de banco de dados. O Amazon RDS cria uma Chave gerenciada pela AWS para o Amazon RDS para sua conta da AWS. A sua conta da AWS tem uma Chave gerenciada pela AWS diferente para o Amazon RDS para cada região da AWS.

Para gerenciar as chaves gerenciadas pelo cliente usadas para criptografia e descriptografia de recursos do Amazon RDS, use o AWS Key Management Service (AWS KMS).

Com o AWS KMS, é possível criar chaves gerenciadas pelo cliente e definir as políticas que controlam o uso dessas chaves. O AWS KMS é compatível com o CloudTrail, o que permite auditar o uso da chave do KMS para verificar se as chaves gerenciadas pelo cliente estão sendo usadas adequadamente. Você pode usar as chaves gerenciadas com o Amazon Aurora e serviços compatíveis da AWS, como o Amazon S3, Amazon EBS e Amazon Redshift. Para obter uma lista de serviços integrados ao AWS KMS, consulte Integração de serviços da AWS. Algumas considerações sobre o uso de chaves do KMS:

- Depois de criar uma instância de banco de dados criptografada, não será possível alterar a chave do KMS usada por essa instância de banco de dados. Portanto, determine os requisitos da chave do KMS antes de criar a instância de banco de dados criptografada.

Se você precisar alterar a chave de criptografia da instância de banco de dados, crie um snapshot manual da instância e habilite a criptografia ao copiar o snapshot. Consulte mais informações neste artigo do Centro de Conhecimento do re:Post.

- Se você copiar um snapshot criptografado, poderá usar uma chave do KMS para criptografar o snapshot de destino diferente da usada para criptografar o snapshot de origem.
- Uma réplica de leitura de uma instância criptografada do Amazon RDS deve ser criptografada usando a mesma chave do KMS que a instância do banco de dados primário quando ambas estiverem na mesma região da AWS.
- Se a instância do banco de dados primário e a réplica de leitura estiverem em regiões da AWS distintas, criptografe a réplica de leitura usando a chave do KMS daquela região da AWS.
- Não é possível compartilhar um snapshot que foi criptografado usando a Chave gerenciada pela AWS da conta da AWS que o compartilhou.
- O Amazon RDS também oferece suporte para criptografar uma instância de banco de dados Oracle ou SQL Server com o Transparent Data Encryption (TDE). O TDE pode ser usado com a criptografia do RDS em repouso, embora o uso do TDE e da criptografia do RDS em repouso simultaneamente possa afetar ligeiramente a performance de seu banco de dados. Você deve gerenciar diferentes chaves para cada método de criptografia. Para ter mais informações sobre o TDE, consulte Oracle Transparent Data Encryption ou Suporte para criptografia de dados transparente no SQL Server.

> **Important**
>
> O Amazon RDS perde o acesso à chave do KMS para uma instância de banco de dados quando você desabilita a chave do KMS. Se você perder o acesso a uma chave KMS, a instância de banco de dados criptografada entrará no estado `inaccessible-encryption-credentials-recoverable` duas horas após a detecção nas instâncias em que os backups estão habilitados. A instância de banco de dados permanece nesse estado por sete dias, durante os quais a instância é interrompida. As chamadas de API feitas para a instância de banco de dados durante esse período podem não ser bem-sucedidas. Para recuperar a instância de banco de dados, habilite a chave do KMS e reinicie a instância. Habilite a chave KMS no AWS Management Console, na AWS CLI ou na API do RDS. Reinicie a instância de banco de dados utilizando o comando start-db-instance da AWS CLI ou o AWS Management Console.

O estado `inaccessible-encryption-credentials-recoverable` só se aplica a instâncias de banco de dados que podem ser interrompidas. Esse estado recuperável não se aplica a instâncias que não podem ser interrompidas, como réplicas de leitura e instâncias com réplicas de leitura. Para obter mais informações, consulte Limitações decorrentes da interrupção de uma instância de banco de dados.

Se a instância de banco de dados não for recuperada em até sete dias, ela entrará no estado `inaccessible-encryption-credentials` do terminal. Nesse estado, a instância de banco de dados não pode mais ser usada e só é possível restaurá-la por um backup. Recomendamos que você sempre habilite backups para instâncias de banco de dados criptografadas a fim de se proteger contra a perda de dados criptografados nos bancos de dados.

Durante a criação de uma instância de banco de dados, o Amazon RDS verifica se a entidade principal da chamada tem acesso à chave do KMS e gera uma concessão da chave do KMS, que é usada durante toda a vida útil da instância de banco de dados. A revogação do acesso da entidade principal da chamada à chave do KMS não afeta um banco de dados em execução. Ao usar chaves do KMS em cenários entre contas, como copiar um snapshot para outra conta, a chave do KMS precisa ser compartilhada com a outra conta. Se você criar uma instância de banco de dados com base no snapshot sem especificar uma chave do KMS diferente, a nova instância usará a chave do KMS da conta de origem. A revogação do acesso à chave após a criação da instância de banco de dados não afeta a instância. No entanto, a desabilitação da chave afeta todas as instâncias de banco de dados criptografadas com essa chave. Para evitar isso, especifique uma chave diferente durante a operação de cópia do snapshot.

As instâncias de banco de dados com backups desabilitados permanecem disponíveis até que os volumes sejam desanexados do host durante uma modificação ou recuperação da instância. O RDS move as instâncias para o estado `inaccessible-encryption-credentials-recoverable` ou `inaccessible-encryption-credentials`, conforme aplicável.

Para ter mais informações sobre as chaves do KMS, consulte AWS KMS keys no Guia do desenvolvedor do AWS Key Management Service.

Criptografar uma instância de banco de dados

Para criptografar uma nova instância de banco de dados, escolha Enable encryption (Habilitar criptografia) no console do Amazon RDS. Para obter informações sobre como criar uma instância de banco de dados, consulte Criar uma instância de banco de dados do Amazon RDS.

Se você usar o comando create-db-instance da AWS CLI para criar uma instância de banco de dados criptografada, defina o parâmetro `--storage-encrypted`. Se você usar a operação da API CreateDBInstance, defina o parâmetro `StorageEncrypted` como true.

Se você usar o comando AWS CLI da `create-db-instance` para criar uma instância de banco de dados criptografada com uma chave gerenciada pelo cliente, defina o parâmetro `--kms-key-id` para qualquer identificador de chave do KMS. Se você usar a operação `CreateDBInstance` da API do Amazon RDS, defina o parâmetro `KmsKeyId` para qualquer identificador de chave do KMS. Para usar uma chave gerenciada pelo cliente em outra conta da AWS, especifique o ARN da chave ou o ARN do alias.

Determinar se a criptografia está habilitada para uma instância de banco de dados

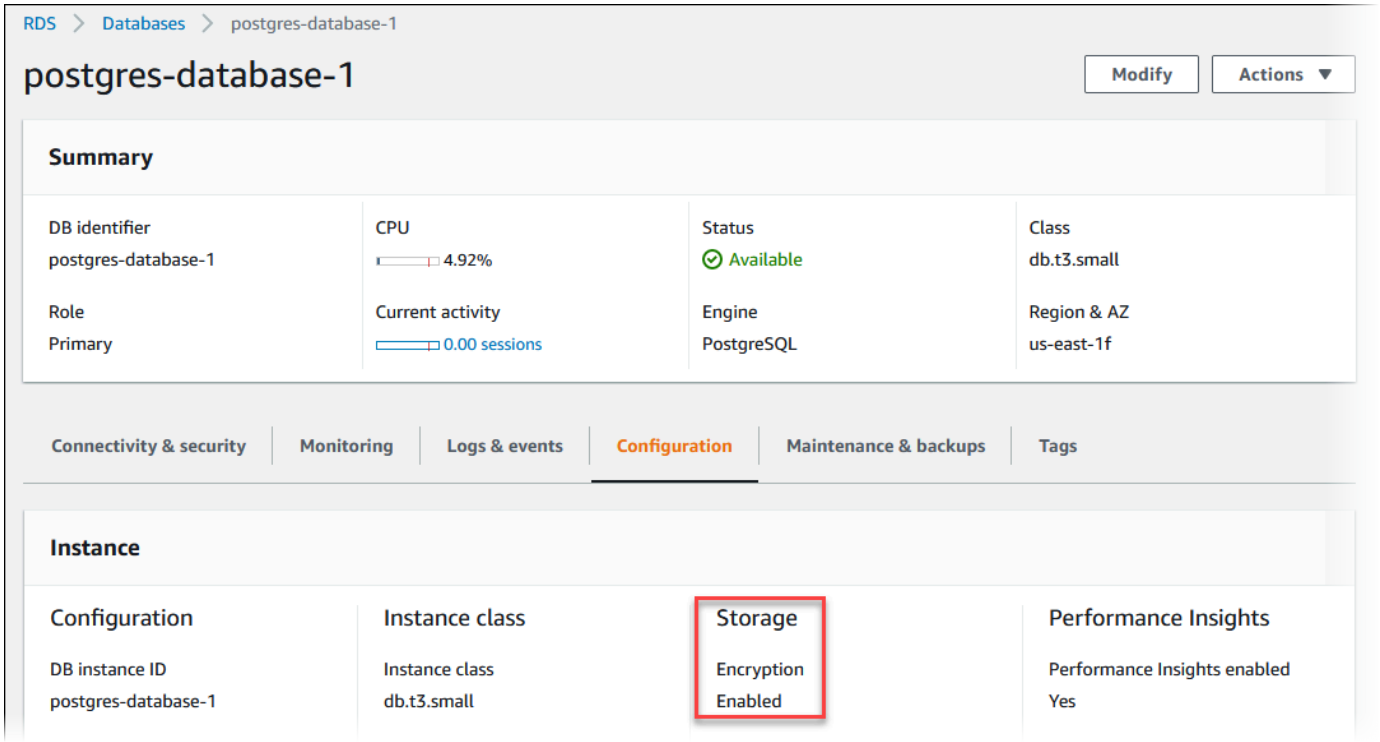
É possível utilizar o AWS Management Console, a AWS CLI ou a API do RDS para verificar se a criptografia em repouso está habilitada para uma instância de banco de dados.

Console

Para verificar se a criptografia em repouso está habilitada para uma instância de banco de dados

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Bancos de dados.
3. Escolha no nome da instância de banco de dados que você deseja verificar para mostrar os detalhes.
4. Escolha a guia Configuration (Configuração) e verifique o valor Encryption (Criptografia) abaixo de Storage (Armazenamento).

   Ele mostra Enabled (Habilitado) ou Not enabled (Não habilitado).



AWS CLI

Para verificar se a criptografia em repouso está habilitada para uma instância de banco de dados usando a AWS CLI, chame o comando describe-db-instances comando com a seguinte opção:

- `--db-instance-identifier`: o nome da instância de banco de dados.

O exemplo a seguir utiliza uma consulta para retornar `TRUE` ou `FALSE` referente à criptografia em repouso para a instância de banco de dados `mydb`.

Example

```
aws rds describe-db-instances --db-instance-identifier mydb --query "*[].
{StorageEncrypted:StorageEncrypted}" --output text
```

API do RDS

Para verificar se a criptografia em repouso está habilitada para uma instância de banco de dados usando a API do Amazon RDS, chame a operação DescribeDBInstances com este parâmetro:

- `DBInstanceIdentifier`: o nome da instância de banco de dados.

Disponibilidade da criptografia do Amazon RDS

No momento, a criptografia do Amazon RDS está disponível para todos os mecanismos de banco de dados e tipos de armazenamento.

A criptografia do Amazon RDS está disponível para a maioria das classes de instância de banco de dados. A tabela a seguir lista as classes de instância de bancos de dados que não são compatíveis com a criptografia do Amazon RDS:

Tipo de instância	Classe de instância
Uso geral (M1)	db.m1.small db.m1.medium db.m1.large db.m1.xlarge
Otimizado para memória (M2)	db.m2.xlarge db.m2.2xlarge db.m2.4xlarge
Expansível (T2)	db.t2.micro

Criptografia em trânsito

Criptografia na camada física

Todos os dados que fluem pelas Regiões da AWS por meio da rede global da AWS são automaticamente criptografados na camada física antes de saírem das instalações seguras da AWS. Todo o tráfego entre AZs é criptografado. Camadas adicionais de criptografia, inclusive as listadas nesta seção, podem oferecer mais proteções.

Criptografia fornecida pelo emparelhamento da Amazon VPC e do Transit Gateway entre regiões

Todo o tráfego entre regiões que usa o emparelhamento da Amazon VPC e do Transit Gateway é automaticamente criptografado em massa ao sair de uma região. Uma camada adicional de

criptografia é fornecida automaticamente à camada física para todo tráfego antes que ele saia das instalações seguras da AWS.

Criptografia entre instâncias

A AWS fornece conectividade privada e segura entre instâncias de banco de dados de todos os tipos. Além disso, alguns tipos de instância usam os recursos de descarregamento do hardware subjacente Nitro System para criptografar automaticamente o tráfego em trânsito entre instâncias. Essa criptografia usa algoritmos de criptografia autenticada com dados associados (AEAD) com criptografia de 256 bits. Não há impacto na performance da rede. Para oferecer suporte a essa criptografia adicional de tráfego em trânsito entre instâncias, os seguintes requisitos devem ser atendidos:

- As instâncias utilizam os seguintes tipos de instância:
  - Uso geral: M6i, M6id, M6in, M6idn, M7g
  - Otimizadas para memória: R6i, R6id, R6in, R6idn, R7g, X2idn, X2iedn, X2iezn
- As instâncias estão na mesma Região da AWS.
- As instâncias estão na mesma VPC ou VPCs emparelhadas, e o tráfego não passa por um dispositivo ou serviço de rede virtual, como um balanceador de carga ou um gateway de trânsito.

Limitações das instâncias de banco de dados criptografados do Amazon RDS

As seguintes limitações existem para instâncias criptografadas de banco de dados do Amazon RDS:

- Você só pode criptografar uma instância de banco de dados do Amazon RDS ao criá-la, e não após a criação.

  Entretanto, como é possível criptografar uma cópia de um snapshot não criptografado, é possível efetivamente adicionar criptografia a uma instância de banco de dados não criptografada. Ou seja, é possível criar um snapshot da sua instância de banco de dados e depois criar uma cópia criptografada desse snapshot. Em seguida, você pode restaurar uma instância de banco de dados a partir do snapshot criptografado, logo, você terá uma cópia criptografada da sua instância de banco de dados original. Para ter mais informações, consulte Copiar um snapshot de banco de dados para o Amazon RDS.
- Não é possível desativar a criptografia em uma instância de banco de dados criptografada.
- Não é possível criar um snapshot criptografado de uma instância de banco de dados não criptografada.

- Um snapshot de um cluster de banco de dados criptografo deve ser criptografado usando a mesma chave do KMS que o cluster de banco de dados.
- Não é possível ter uma réplica de leitura criptografada de uma instância de banco de dados não criptografada nem uma réplica de leitura não criptografada de uma instância de banco de dados criptografada.
- As réplicas de leitura criptografadas deverão ser criptografadas com a mesma chave do KMS que a instância de banco de dados de origem quando as duas estiverem na mesma região da AWS.
- Não é possível restaurar um backup ou um snapshot não criptografado em uma instância de banco de dados criptografada.
- Para copiar um snapshot criptografado de uma região da AWS para outra, é necessário especificar a chave do KMS na região da AWS de destino. Isso ocorre porque as chaves do KMS são específicas da região da AWS em que são criadas.

  O snapshot de origem permanece criptografado ao longo do processo de cópia. O Amazon RDS usa criptografia de envelope para proteger os dados durante o processo de cópia. Para ter mais informações sobre a criptografia de envelope, consulte Criptografia de envelope no Guia do desenvolvedor do AWS Key Management Service.
- Não é possível descriptografar uma instância de banco de dados criptografada. No entanto, é possível exportar dados de uma instância de banco de dados criptografada e importar os dados para uma instância de banco de dados não criptografada.

Gerenciamento de AWS KMS key

O Amazon RDS integra-se automaticamente ao AWS Key Management Service (AWS KMS) para gerenciar chaves. O Amazon RDS usa criptografia envelopada. Para ter mais informações sobre a criptografia de envelope, consulte Criptografia de envelope no Guia do desenvolvedor do AWS Key Management Service.

Você pode usar dois tipos de chave do AWS KMS para criptografar instâncias de banco de dados.

- Se você quiser o controle total de uma chave do KMS, precisará criar uma chave gerenciada pelo cliente. Para obter mais informações sobre chaves gerenciadas pelo cliente, consulte Chaves gerenciadas pelo cliente no Guia do desenvolvedor do AWS Key Management Service.
- Chaves gerenciadas pela AWS são chaves do KMS em sua conta que são criadas, gerenciadas e usadas em seu nome por um produto da AWS integrado ao AWS KMS. Por padrão, a Chave gerenciada pela AWS do RDS (`aws/rds`) é usada para criptografia. Você não pode gerenciar,

nem rotacionar, nem excluir a Chave gerenciada pela AWS do RDS. Para obter mais informações sobre as Chaves gerenciadas pela AWS, consulte Chaves gerenciadas pela AWS no Guia do desenvolvedor do AWS Key Management Service.

Para gerenciar as chaves do KMS usadas para instâncias de banco de dados criptografadas do Amazon RDS, use o AWS Key Management Service (AWS KMS) no console do AWS KMS, a AWS CLI ou a API do AWS KMS. Para visualizar logs de auditoria de cada ação realizada com uma chave gerenciada pela AWS ou pelo cliente, use o AWS CloudTrail. Para obter mais informações sobre a alternância de chaves, consulte Alternância de chaves do AWS KMS.

Como autorizar o uso de uma chave gerenciada pelo cliente

Quando o RDS utiliza uma chave gerenciada pelo cliente em operações criptográficas, ele atua em nome do usuário que está criando ou alterando o recurso do RDS.

Para criar um recurso do RDS usando uma chave gerenciada pelo cliente, um usuário deve ter permissões para acionar as seguintes operações na chave gerenciada pelo cliente:

- `kms:CreateGrant`
- `kms:DescribeKey`

Você pode especificar essas permissões em uma política de chaves ou em uma política do IAM, se a política de chaves permitir.

Important

Ao usar instruções de negação explícitas para todos os recursos (*) nas políticas de chave do AWS KMS com serviços gerenciados como o Amazon RDS, você deve especificar uma condição para dar permissão à conta proprietária do recurso. As operações podem falhar sem essa condição, mesmo que a regra de negação inclua exceções para seu usuário do IAM.

Tip

Para seguir o princípio de menor privilégio, não permita acesso total a `kms:CreateGrant`. Em vez disso, use a chave de condição kms:ViaService para permitir que o usuário crie

concessões na chave do KMS somente quando a concessão for criada em nome do usuário por um serviço da AWS.

É possível tornar a política do IAM mais rígida de várias maneiras. Por exemplo, se você quiser permitir que a chave gerenciada pelo cliente seja usada somente para solicitações provenientes do RDS, poderá utilizar a chave de condição kms:ViaService com o valor `rds.<region>.amazonaws.com`. Também é possível usar as chaves ou valores em Contexto de criptografia do Amazon RDS como uma condição para usar a chave gerenciada pelo cliente para criptografia.

Para obter mais informações, consulte Permitir que usuários de outras contas usem uma chave do KMS no Guia do desenvolvedor do AWS Key Management Service e em Políticas de chave no AWS KMS.

Contexto de criptografia do Amazon RDS

Quando o RDS usa sua chave do KMS ou quando o Amazon EBS usa a chave do KMS em nome do RDS, o serviço especifica um contexto de criptografia. O contexto de criptografia é dados autenticados adicionados (AAD) que o AWS KMS usa para garantir a integridade dos dados. Quando um contexto de criptografia é especificado para uma operação de criptografia, o serviço deve especificar esse mesmo contexto para a operação de descriptografia. Caso contrário, ocorrerá uma falha na descriptografia. O contexto de criptografia é também gravado nos logs do AWS CloudTrail para ajudar você a compreender por que uma determinada chave do KMS foi usada. Os logs do CloudTrail podem conter muitas entradas que descrevem o uso de uma chave do KMS, mas o contexto de criptografia em cada entrada de log pode ajudar a determinar o motivo desse uso específico.

No mínimo, o Amazon RDS sempre usa o ID da instância de banco de dados para o contexto de criptografia, como no seguinte exemplo em formato JSON:

```
{ "aws:rds:db-id": "db-CQYSMDPBRZ7BPMH7Y3RTDG5QY" }
```

Esse contexto de criptografia pode ajudar a identificar para qual instância de banco de dados a chave do KMS foi usada.

Quando a chave do KMS é usada em determinada instância de banco de dados e em um volume do Amazon EBS específico, o ID da instância de banco de dados e o ID do volume do Amazon EBS são usados no contexto de criptografia, como no seguinte exemplo em formato JSON:

```
{
  "aws:rds:db-id": "db-BRG7VYS3SVIFQW7234EJQOM5RQ",
  "aws:ebs:id": "vol-ad8c6542"
}
```

Usar SSL/TLS para criptografar uma conexão com uma instância ou um cluster de banco de dados

É possível usar o Security Socket Layer (SSL) ou o Transport Layer Security (TLS) na aplicação para criptografar uma conexão com um banco de dados que executa Db2, MariaDB, Microsoft SQL Server, MySQL, Oracle ou PostgreSQL.

As conexões SSL/TLS fornecem uma camada de segurança criptografando dados que se movem entre o cliente e a instância ou o cluster de banco de dados. Opcionalmente, sua conexão SSL/TLS pode realizar a verificação da identidade do servidor validando o certificado do servidor instalado no banco de dados. Para exigir a verificação da identidade do servidor, siga este processo geral:

1. Escolha a Autoridade de certificação (CA) que assina ocertificado de servidor de banco de dados para seu banco de dados. Para obter mais informações sobre autoridades de certificação, consulte Autoridades certificadoras.
2. Baixe um pacote de certificados para usar quando você estiver se conectando com o banco de dados. Para baixar um pacote de certificados, consulte Pacote de certificados por Região da AWS..

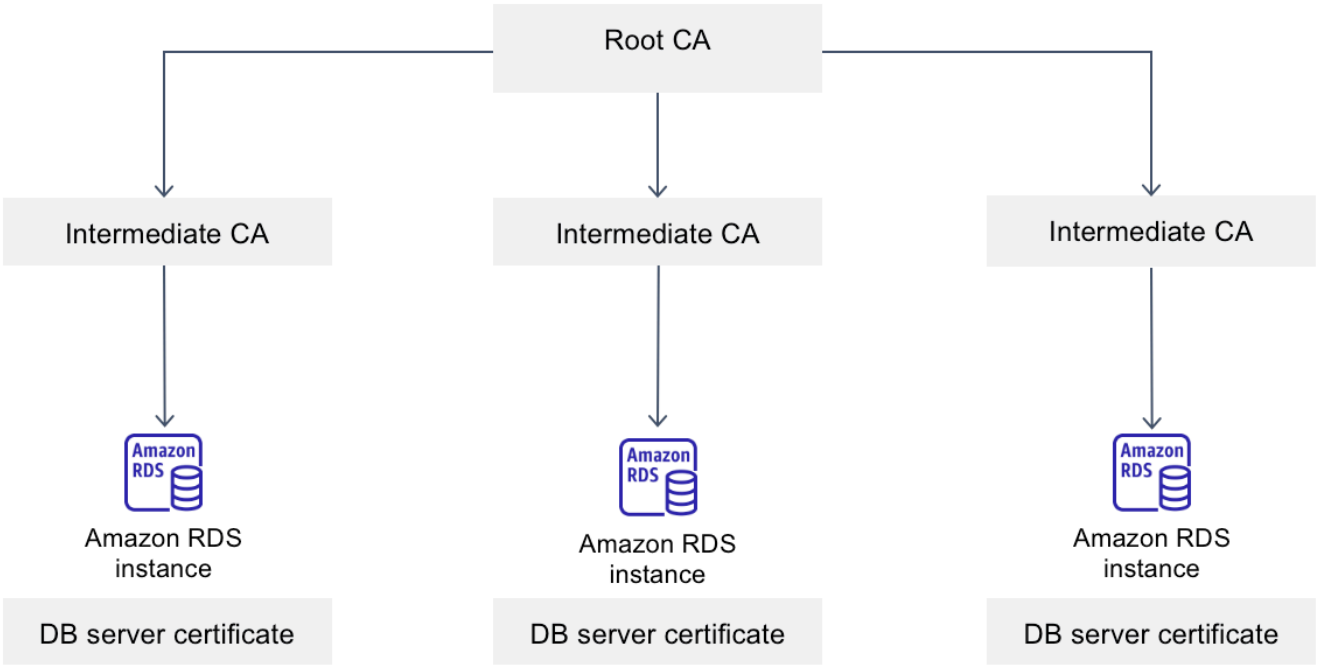
   > **Note**
   >
   > Todos os certificados somente estão disponíveis para download usando conexões SSL/TLS.

3. Conecte-se ao banco de dados usando o processo do mecanismo de banco de dados para implementar conexões SSL/TLS. Cada mecanismo de banco de dados tem seu próprio processo de implementação do SSL/TLS. Para saber como implementar o SSL/TLS para o banco de dados, siga o link correspondente ao seu mecanismo de banco de dados:
   - Usar SSL/TLS com uma instância de banco de dados do Amazon RDS para Db2
   - Suporte de SSL/TLS para instâncias de banco de dados do MariaDB no Amazon RDS
   - ???
   - Suporte do SSL/TLS para instâncias de banco de dados do MySQL no Amazon RDS

- Usar SSL com uma instância de banco de dados Oracle
- Usar o SSL com uma instância de banco de dados PostgreSQL

Autoridades certificadoras

A autoridade de certificação (CA) é o certificado que identifica a CA raiz no início da cadeia de certificados. A CA assina o certificado do servidor de banco de dados, que é instalado em cada instância de banco de dados. O certificado do servidor de banco de dados identifica a instância de banco de dados como um servidor confiável.



O Amazon RDS fornece as CAs a seguir para assinar o certificado do servidor de banco de dados para um banco de dados.

Autoridade certificadora (CA)	Descrição	Nome comum (CN)
rds-ca-2019	Utiliza uma autoridade de certificação com algoritmo de chave privada RSA 2048 e algoritmo de assinatura SHA256. Essa CA expira em 2024 e não é compatível com a alternância automática de certificados do servidor. Se você	*Identificador de região* do Amazon RDS CA 2019

Autoridade certificadora (CA)	Descrição	Nome comum (CN)
	estiver usando essa CA e quiser manter o mesmo padrão, recomendamos alternar para a CA rds-ca-rsa2048-g1.	
rds-ca-rsa2048-g1	Utiliza uma autoridade de certificação com algoritmo de chave privada RSA 2048 e algoritmo de assinatura SHA256 na maioria das Regiões da AWS. Nas AWS GovCloud (US) Regions, essa CA utiliza uma autoridade de certifica ção com algoritmo de chave privada RSA 2048 e algoritmo de assinatura SHA384. Essa CA permanece válida por mais tempo do que a CA rds-ca-2019. Essa CA é compatível com a alternância automátic a de certificados do servidor.	*Identificador de região* do Amazon RDS RSA2048 G1
rds-ca-rsa4096-g1	Utiliza uma autoridade de certificação com algoritmo de chave privada RSA 4096 e algoritmo de assinatura SHA384. Essa CA é compatível com a alternância automática de certificados do servidor.	*Identificador de região* do Amazon RSA4096 G1
rds-ca-ecc384-g1	Utiliza uma autoridade de certificação com algoritmo de chave privada ECC 384 e algoritmo de assinatura SHA384. Essa CA é compatível com a alternância automática de certificados do servidor.	*Identificador de região* do Amazon RDS ECC384 G1

> **Note**
>
> Se você estiver usando a AWS CLI, poderá ver as validades das autoridades de certificação listadas acima usando describe-certificates.

Esses certificados CA estão incluídos no pacote de certificados regionais e globais. Quando você usa a CA rds-ca-rsa2048-g1, rds-ca-rsa4096-g1 ou rds-ca-ecc384-g1 com um banco de dados, o RDS gerencia o certificado do servidor no banco de dados. O RDS alterna automaticamente o certificado do servidor de banco de dados antes que ele expire.

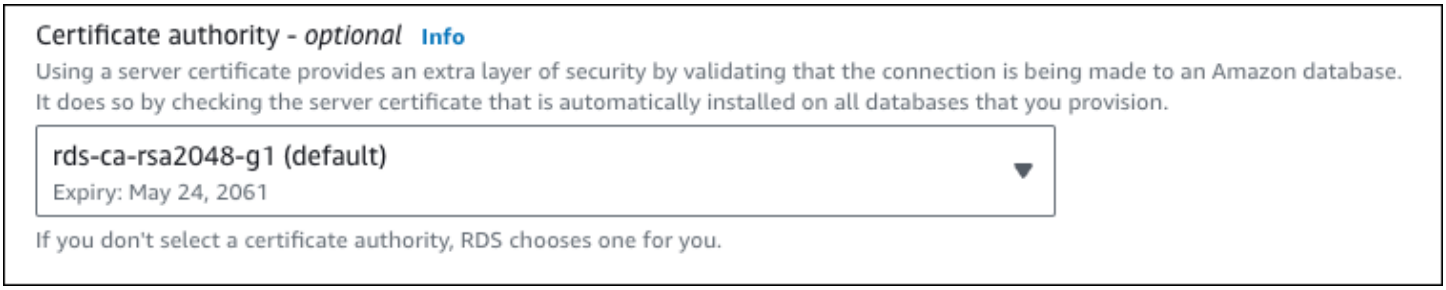
Configurar a CA do banco de dados

Você pode definir a CA para um banco de dados ao realizar as seguintes tarefas:

- Crie uma instância de banco de dados ou um cluster de banco de dados multi-AZ: é possível definir a CA ao criar uma instância ou um cluster de banco de dados. Para obter instruções, consulte the section called “Criar uma instância de banco de dados” ou the section called “Criar um cluster de banco de dados multi-AZ”.
- Modificar uma instância de banco de dados ou um cluster de banco de dados multi-AZ: é possível definir a CA para uma instância ou um cluster de banco de dados modificando-a. Para obter instruções, consulte the section called “Modificação de uma instância de banco de dados” ou the section called “Modificar um cluster de banco de dados multi-AZ”.

> **Note**
>
> A CA padrão é definida como rds-ca-rsa2048-g1. Você pode substituir a CA padrão para sua Conta da AWS usando o comando modify-certificates.

As CAs disponíveis dependem do mecanismo de banco de dados e da versão do mecanismo de banco de dados. Ao usar o AWS Management Console, você pode selecionar a CA usando a configuração Certificate authority (Autoridade de certificação), conforme mostrado na imagem a seguir.



O console mostra apenas as CAs que estão disponíveis para o mecanismo de banco de dados e a versão do mecanismo de banco de dados. Se estiver usando a AWS CLI, você poderá definir a CA

para uma instância de banco de dados usando o comando create-db-instance ou modify-db-instance. É possível definir a CA para um cluster de banco de dados multi-AZ usando o comando create-db-cluster ou modify-db-cluster.

Se estiver usando a AWS CLI, você poderá ver as CAs disponíveis para sua conta usando o comando describe-certificates. Esse comando também mostra a data de expiração de cada CA em `ValidTill` na saída. Você pode encontrar as CAs que estão disponíveis para uma versão específica do mecanismo de banco de dados e do mecanismo de banco de dados usando o comando describe-db-engine-versions.

O exemplo a seguir mostra as CAs disponíveis para a versão padrão do mecanismo de banco de dados do RDS para PostgreSQL.

```
aws rds describe-db-engine-versions --default-only --engine postgres
```

A saída é semelhante à seguinte. As CAs disponíveis estão listadas em `SupportedCACertificateIdentifiers`. A saída também mostra se a versão do mecanismo de banco de dados é compatível com a alternância do certificado sem reiniciar em `SupportsCertificateRotationWithoutRestart`.

```
{
    "DBEngineVersions": [
        {
            "Engine": "postgres",
            "MajorEngineVersion": "13",
            "EngineVersion": "13.4",
            "DBParameterGroupFamily": "postgres13",
            "DBEngineDescription": "PostgreSQL",
            "DBEngineVersionDescription": "PostgreSQL 13.4-R1",
            "ValidUpgradeTarget": [],
            "SupportsLogExportsToCloudwatchLogs": false,
            "SupportsReadReplica": true,
            "SupportedFeatureNames": [
                "Lambda"
            ],
            "Status": "available",
            "SupportsParallelQuery": false,
            "SupportsGlobalDatabases": false,
            "SupportsBabelfish": false,
            "SupportsCertificateRotationWithoutRestart": true,
            "SupportedCACertificateIdentifiers": [
```

```
                "rds-ca-2019",
                "rds-ca-rsa2048-g1",
                "rds-ca-ecc384-g1",
                "rds-ca-rsa4096-g1"
            ]
        }
    ]
}
```

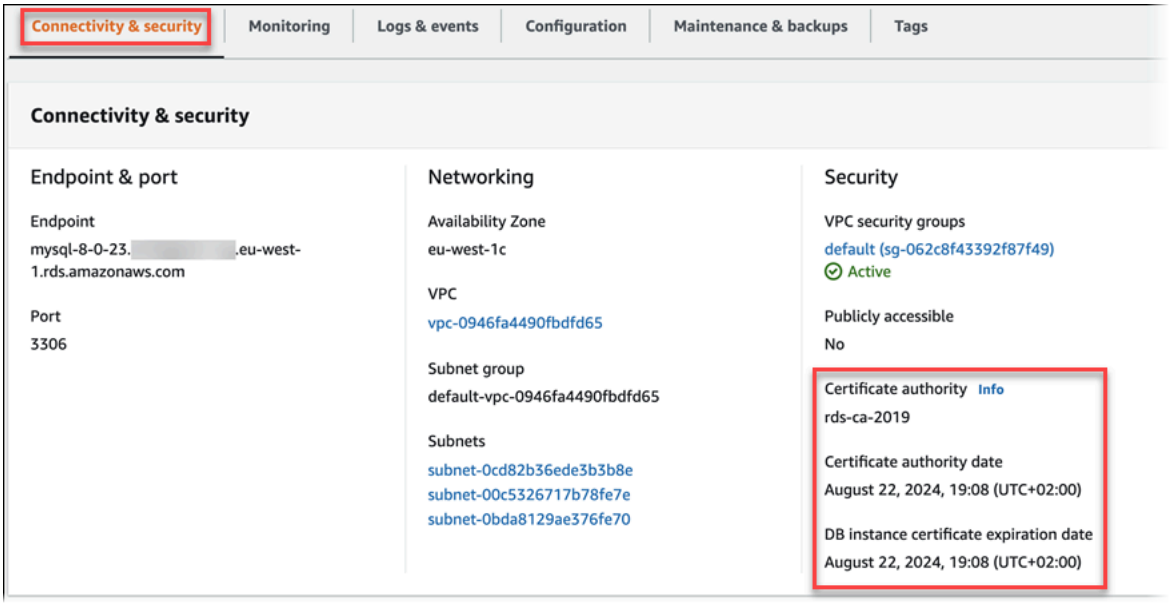
Validades do certificado do servidor de banco de dados

A validade do certificado do servidor de banco de dados depende do mecanismo de banco de dados e da versão do respectivo mecanismo. Se a versão do mecanismo comportar a alternância de certificado sem reinicialização, a validade do certificado do mecanismo será de um ano. Caso contrário, será de três anos.

Para obter mais informações sobre alternância de certificados de servidor de banco de dados, consulte Alternância automática de certificados do servidor.

Visualizar a CA da instância de banco de dados

É possível visualizar os detalhes sobre a CA para um banco de dados na guia Conectividade e segurança no console, como na imagem a seguir.



Se estiver usando a AWS CLI, você poderá visualizar os detalhes da CA de uma instância de banco de dados usando o comando describe-db-instances. É possível visualizar os detalhes sobre a CA para um cluster de banco de dados multi-AZ usando o comando describe-db-clusters.

Baixar pacotes de certificados para Amazon RDS

Quando você se conecta ao banco de dados com SSL ou TLS, a instância do banco de dados exige um certificado de confiança do Amazon RDS. Selecione o link apropriado na tabela a seguir para baixar o pacote que corresponde à Região da AWS onde você hospeda o banco de dados.

Pacote de certificados por Região da AWS

Os pacotes de certificados para todas as Regiões da AWS e GovCloud (EUA) contêm os seguintes certificados:

- Certificados intermediários `rds-ca-2019` e raiz.
- Certificados CA raiz `rds-ca-rsa2048-g1`, `rds-ca-rsa4096-g1` e `rds-ca-ecc384-g1`. O repositório confiável da aplicação só precisa registrar o certificado CA raiz.

> **Note**
>
> O proxy do Amazon RDS usa certificados do AWS Certificate Manager (ACM). Se você estiver usando o RDS Proxy, não será necessário baixar os certificados do Amazon RDS nem atualizar as aplicações que usam conexões do RDS Proxy. Para ter mais informações, consulte Usar TLS/SSL com o RDS Proxy.

Para baixar um pacote de certificados para uma Região da AWS, selecione o link da Região da AWS que hospeda o banco de dados na tabela a seguir.

Região da AWS	Pacote de certificados (PEM)	Pacote de certificados (PKCS7)
Qualquer Região da AWS comercial.	global-bundle.pem	global-bundle.p7b
Leste dos EUA (Norte da Virgínia)	us-east-1-bundle.pem	us-east-1-bundle.p7b
Leste dos EUA (Ohio)	us-east-2-bundle.pem	us-east-2-bundle.p7b
Oeste dos EUA (N. da Califórnia)	us-west-1-bundle.pem	us-west-1-bundle.p7b

Região da AWS	Pacote de certificados (PEM)	Pacote de certificados (PKCS7)
Oeste dos EUA (Oregon)	us-west-2-bundle.pem	us-west-2-bundle.p7b
África (Cidade do Cabo)	af-south-1-bundle.pem	af-south-1-bundle.p7b
Ásia-Pacífico (Hong Kong)	ap-east-1-bundle.pem	ap-east-1-bundle.p7b
Ásia-Pacífico (Hyderabad)	ap-south-2-bundle.pem	ap-south-2-bundle.p7b
Ásia-Pacífico (Jacarta)	ap-southeast-3-bundle.pem	ap-southeast-3-bundle.p7b
Ásia-Pacífico (Melbourne)	ap-southeast-4-bundle.pem	ap-southeast-4-bundle.p7b
Ásia-Pacífico (Mumbai)	ap-south-1-bundle.pem	ap-south-1-bundle.p7b
Asia Pacific (Osaka)	ap-northeast-3-bundle.pem	ap-northeast-3-bundle.p7b
Ásia-Pacífico (Tóquio)	ap-northeast-1-bundle.pem	ap-northeast-1-bundle.p7b
Ásia-Pacífico (Seul)	ap-northeast-2-bundle.pem	ap-northeast-2-bundle.p7b
Ásia-Pacífico (Singapura)	ap-southeast-1-bundle.pem	ap-southeast-1-bundle.p7b
Ásia-Pacífico (Sydney)	ap-southeast-2-bundle.pem	ap-southeast-2-bundle.p7b
Canadá (Central)	ca-central-1-bundle.pem	ca-central-1-bundle.p7b
Oeste do Canadá (Calgary)	ca-west-1-bundle.pem	ca-west-1-bundle.p7b
Europa (Frankfurt)	eu-central-1-bundle.pem	eu-central-1-bundle.p7b
Europa (Irlanda)	eu-west-1-bundle.pem	eu-west-1-bundle.p7b
Europa (Londres)	eu-west-2-bundle.pem	eu-west-2-bundle.p7b
Europa (Milão)	eu-south-1-bundle.pem	eu-south-1-bundle.p7b
Europa (Paris)	eu-west-3-bundle.pem	eu-west-3-bundle.p7b
Europa (Espanha)	eu-south-2-bundle.pem	eu-south-2-bundle.p7b

Região da AWS	Pacote de certificados (PEM)	Pacote de certificados (PKCS7)
Europa (Estocolmo)	eu-north-1-bundle.pem	eu-north-1-bundle.p7b
Europa (Zurique)	eu-central-2-bundle.pem	eu-central-2-bundle.p7b
Israel (Tel Aviv)	il-central-1-bundle.pem	il-central-1-bundle.p7b
Oriente Médio (Barém)	me-south-1-bundle.pem	me-south-1-bundle.p7b
Oriente Médio (Emirados Árabes Unidos)	me-central-1-bundle.pem	me-central-1-bundle.p7b
América do Sul (São Paulo)	sa-east-1-bundle.pem	sa-east-1-bundle.p7b
Quaisquer AWS GovCloud (US) Regions.	global-bundle.pem	global-bundle.p7b
AWS GovCloud (Leste dos EUA)	us-gov-east-1-bundle.pem	us-gov-east-1-bundle.p7b
AWS GovCloud (Oeste dos EUA)	us-gov-west-1-bundle.pem	us-gov-west-1-bundle.p7b

Visualizar o conteúdo do certificado CA

Para verificar o conteúdo do pacote de certificados da CA, use o seguinte comando:

```
keytool -printcert -v -file global-bundle.pem
```

Alternar o certificado SSL/TLS

Os certificados rds-ca-2019 da autoridade de certificação do Amazon RDS expiraram em agosto de 2024. Se você usa ou planeja usar Secure Sockets Layer (SSL) ou Transport Layer Security (TLS) com verificação de certificado para se conectar às instâncias de banco de dados do RDS ou clusters de banco de dados multi-AZ,considere usar um dos novos certificados CA rds-ca-rsa2048-g1, rds-ca-rsa4096-g1 ou rds-ca-ecc384-g1. Se você não usa SSL/TLS com verificação de certificado no momento, é possível que ainda tenha algum certificado de CA expirado e precise atualizá-lo para um

novo certificado de CA se planeja usar SSL/TLS com verificação de certificado para se conectar aos bancos de dados do RDS.

O Amazon RDS fornece novos certificados CA como uma prática recomendada de segurança da AWS. Para ter informações sobre os novos certificados e as regiões da AWS compatíveis, consulte Usar SSL/TLS para criptografar uma conexão com uma instância ou um cluster de banco de dados.

Para atualizar o certificado CA do seu banco de dados, use os seguintes métodos:

- Atualizar o certificado CA modificando a instância ou o cluster de banco de dados
- Atualizar seu certificado CA aplicando manutenção

Antes de atualizar as instâncias de banco de dados ou clusters de banco de dados multi-AZ para usar o novo certificado CA, atualize os clientes ou as aplicações que se conectam aos bancos de dados do RDS.

Considerações sobre a troca de certificados

Pense nas seguintes situações antes de trocar seu certificado:

- O Amazon RDS Proxy usa certificados do AWS Certificate Manager (ACM). Se estiver usando o RDS Proxy, ao trocar o certificado SSL/TLS, não será necessário atualizar as aplicações que usam conexões do RDS Proxy. Para obter mais informações, consulteUsar TLS/SSL com o RDS Proxy
- Se estiver usando uma aplicação Go versão 1.15 com uma instância de banco de dados ou um cluster de banco de dados multi-AZ criada ou atualizada para o certificado rds-ca-2019 antes de 28 de julho de 2020, você deverá atualizar o certificado novamente. Atualize o certificado para rds-ca-rsa2048-g1, rds-ca-rsa4096-g1 ou rds-ca-ecc384-g1, dependendo do seu mecanismo .

  Use o comando `modify-db-instance` para uma instância de banco de dados, ou o comando `modify-db-cluster` para um cluster de banco de dados multi-AZ usando o novo identificador de certificado CA. Você pode encontrar as CAs que estão disponíveis para uma versão específica do mecanismo de banco de dados e do mecanismo de banco de dados usando o comando `describe-db-engine-versions`.

  Caso você tenha criado a instância de banco de dados ou atualizado o certificado dela após 28 de julho de 2020, nenhuma ação será necessária. Para obter mais informações, consulte Go GitHub issue #39568.

Atualizar o certificado CA modificando a instância ou o cluster de banco de dados

O exemplo a seguir atualiza o certificado de CA de rds-ca-2019 para rds-ca-rsa2048-g1. Você pode escolher um certificado diferente. Para ter mais informações, consulte Autoridades certificadoras..

Atualize seu repositório confiável de aplicações para reduzir o tempo de inatividade associado à atualização do certificado de CA. Consulte mais informações sobre reinicializações associadas à alternância de certificados de CA em Alternância automática de certificados do servidor.

Como atualizar o certificado CA modificando a instância ou o cluster de banco de dados

1. Baixe o novo certificado SSL/TLS conforme descrito em Usar SSL/TLS para criptografar uma conexão com uma instância ou um cluster de banco de dados.
2. Atualize as aplicações para usarem o novo certificado SSL/TLS.

   Os métodos para atualizar aplicações para novos certificados SSL/TLS dependem de suas aplicações específicas. Trabalhe com os desenvolvedores de aplicações para atualizar os certificados SSL/TLS para suas aplicações.

   Para obter informações sobre como verificar conexões SSL/TLS e atualizar aplicações para cada mecanismo de banco de dados, consulte os seguintes tópicos:

   - Atualizar aplicações para conexão com instâncias do MariaDB usando novos certificados SSL/TLS
   - Atualizar aplicações para conexão com instâncias de banco de dados Microsoft SQL Server usando novos certificados SSL/TLS
   - Atualizar aplicações para conexão com instâncias de banco de dados MySQL usando novos certificados SSL/TLS
   - Atualizar aplicações para conexão com instâncias de banco de dados Oracle usando novos certificados SSL/TLS
   - Atualizar aplicações para conexão com instâncias de banco de dados PostgreSQL usando novos certificados SSL/TLS

   Para conhecer um script de exemplo que atualiza um armazenamento confiável para um sistema operacional Linux, consulte Script de exemplo para importar certificados para o seu armazenamento confiável.

> **Note**
>
> O pacote de certificados contém certificados tanto para a CA antiga como para a nova, portanto, é possível atualizar a aplicação de maneira segura e manter a conectividade durante o período de transição. Se você estiver usando o AWS Database Migration Service a fim de migrar um banco de dados para uma instância ou um cluster de banco de dados, recomendamos o uso do pacote de certificados para garantir a conectividade durante a migração.

3. Modifique a instância de banco de dados ou cluster de banco de dados multi-AZ para alterar a CA de rds-ca-2019 para rds-ca-rsa2048-g1. Para verificar se o banco de dados requer reinicialização para atualizar os certificados de CA, use o comando descrebe-db-engine-versions e verifique o sinalizador `SupportsCertificateRotationWithoutRestart`.

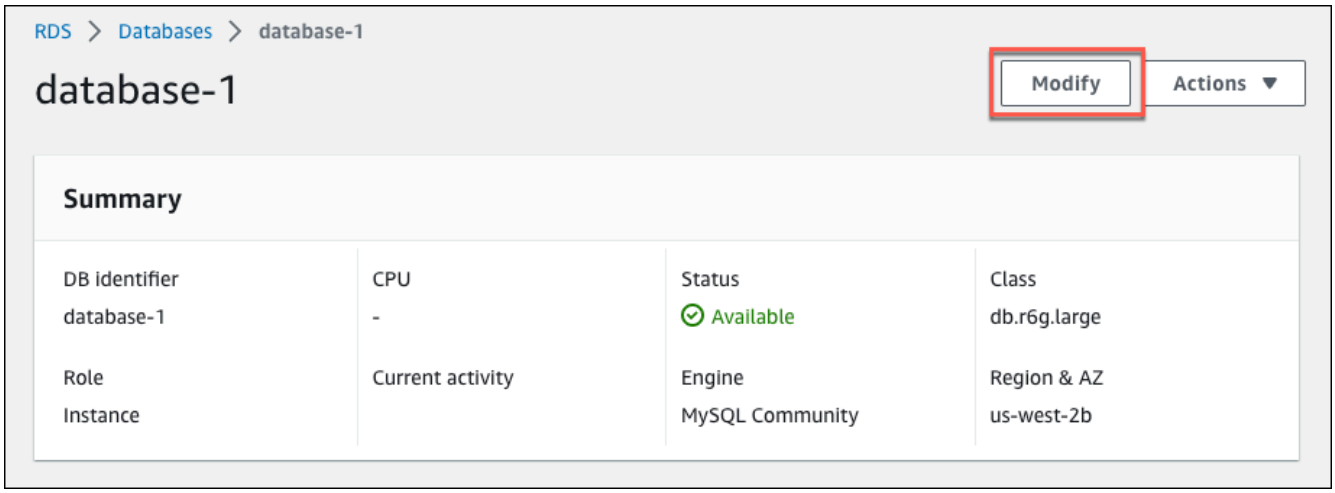
> **Important**
>
> Se você estiver enfrentando problemas de conectividade após a expiração do certificado, especifique a opção Apply immediately (Aplicar imediatamente) no console ou a opção `--apply-immediately` usando a AWS CLI. Por padrão, essa operação é programada para ser executada durante a próxima janela de manutenção.
> Para instâncias de banco de dados do RDS para Oracle, recomendamos que você reinicie o banco de dados do Oracle para evitar erros de conexão.
> Para instâncias multi-AZ do RDS para SQL Server com a opção “Sempre ativa” ou “Espelhamento” habilitada, é esperado um failover quando a instância é reiniciada após a alternância do certificado.
> Para definir uma substituição de CA da instância que é diferente da CA padrão do RDS, use o comando modify-certificates da CLI.

É possível usa o AWS Management Console ou a AWS CLI para alterar o certificado CA de rds-ca-2019 para rds-ca-rsa2048-g1 para uma instância de banco de dados ou cluster de banco de dados multi-AZ.

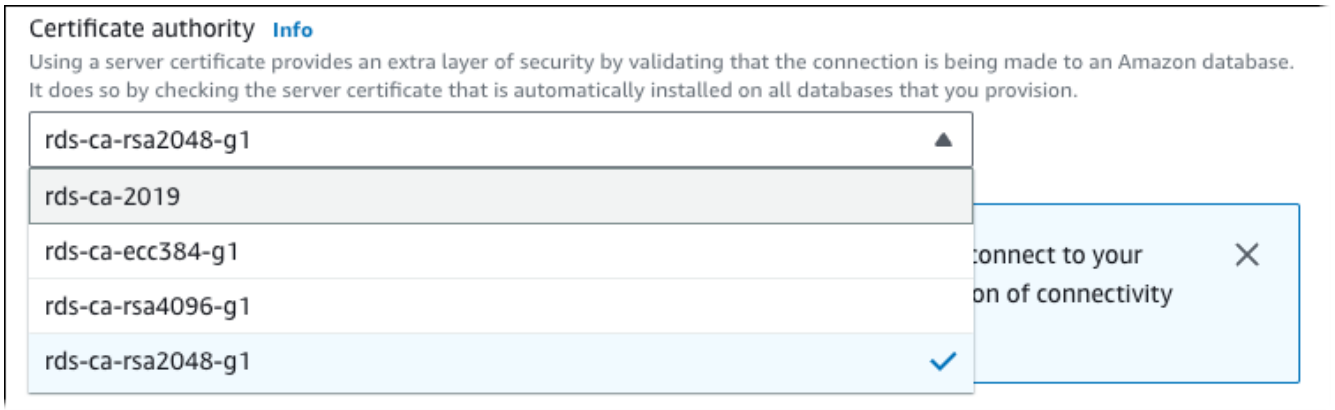
Console

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.

2. No painel de navegação, escolha Bancos de dados e selecione a instância de banco de dados ou o cluster de banco de dados multi-AZ que você deseja modificar.
3. Escolha Modificar.



4. Na seção Conectividade, escolha rds-ca-rsa2048-g1.



5. Escolha Continue (Continuar) e verifique o resumo de modificações.
6. Para aplicar as alterações imediatamente, escolha Apply immediately.
7. Na página de confirmação, revise suas alterações. Se estiverem corretas, escolha Modificar instância de banco de dados ou Modificar cluster para salvar as alterações.

> **Important**
>
> Ao programar essa operação, certifique-se de ter atualizado o armazenamento de confiança do lado do cliente com antecedência.

Ou escolha Back (Voltar) para editar as alterações ou Cancel (Cancelar) para cancelar as alterações.

AWS CLI

Para usar a AWS CLI a fim de alterar a CA de rds-ca-2019 para rds-ca-rsa2048-g1 para uma instância de banco de dados ou cluster de banco de dados multi-AZ, chame o comando modify-db-instance ou modify-db-cluster. Especifique o identificador da instância ou do cluster de banco de dados e a opção `--ca-certificate-identifier`.

Use o parâmetro `--apply-immediately` para aplicar a atualização imediatamente. Por padrão, essa operação é programada para ser executada durante a próxima janela de manutenção.

> **Important**
>
> Ao programar essa operação, certifique-se de ter atualizado o armazenamento de confiança do lado do cliente com antecedência.

Example

instância de banco de dados

O exemplo a seguir modifica `mydbinstance` definindo o certificado CA como `rds-ca-rsa2048-g1`.

Para Linux, macOS ou Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --ca-certificate-identifier rds-ca-rsa2048-g1
```

Para Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --ca-certificate-identifier rds-ca-rsa2048-g1
```

> **Note**
>
> Se a instância exigir reinicialização, você poderá usar o comando modify-db-instance da CLI e especificar a opção `--no-certificate-rotation-restart`.

Example

Cluster de banco de dados multi-AZ

O exemplo a seguir modifica `mydbcluster` definindo o certificado CA como `rds-ca-rsa2048-g1`.

Para Linux, macOS ou Unix:

```
aws rds modify-db-cluster \
    --db-cluster-identifier mydbcluster \
    --ca-certificate-identifier rds-ca-rsa2048-g1
```

Para Windows:

```
aws rds modify-db-cluster ^
    --db-cluster-identifier mydbcluster ^
    --ca-certificate-identifier rds-ca-rsa2048-g1
```

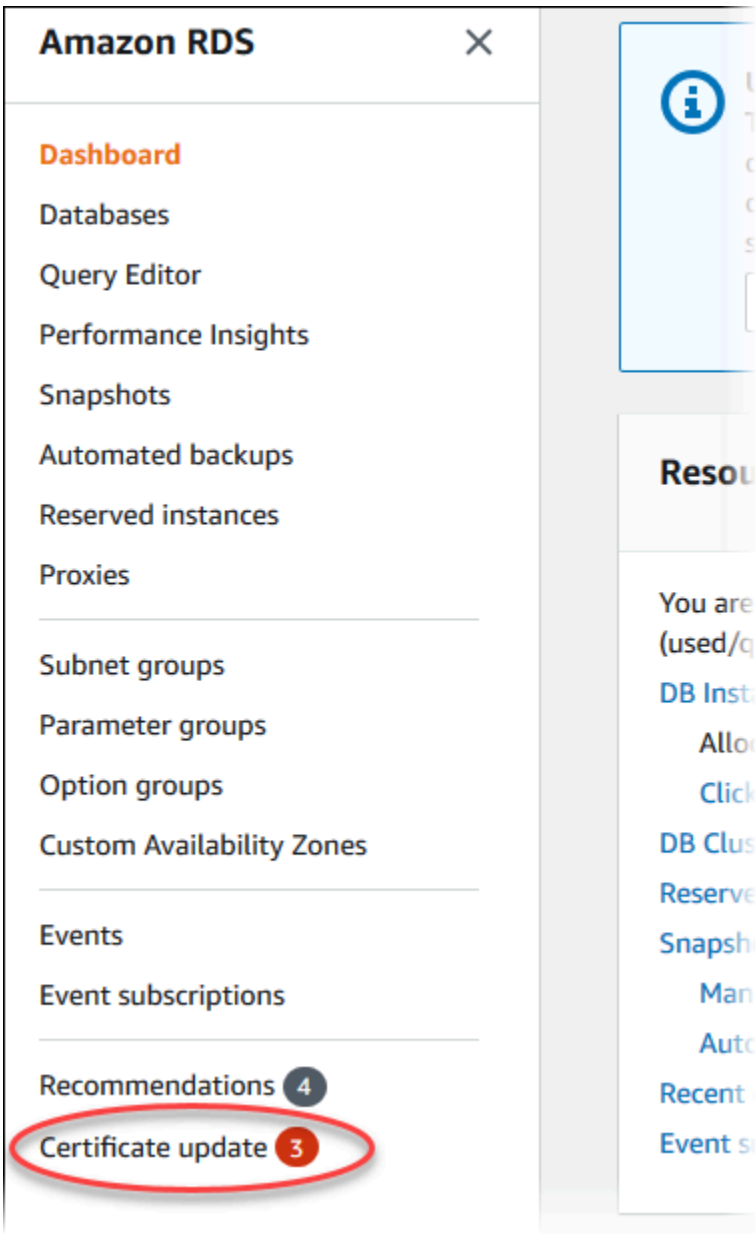
Atualizar seu certificado CA aplicando manutenção

Siga as etapas a seguir para atualizar o certificado CA aplicando a manutenção.

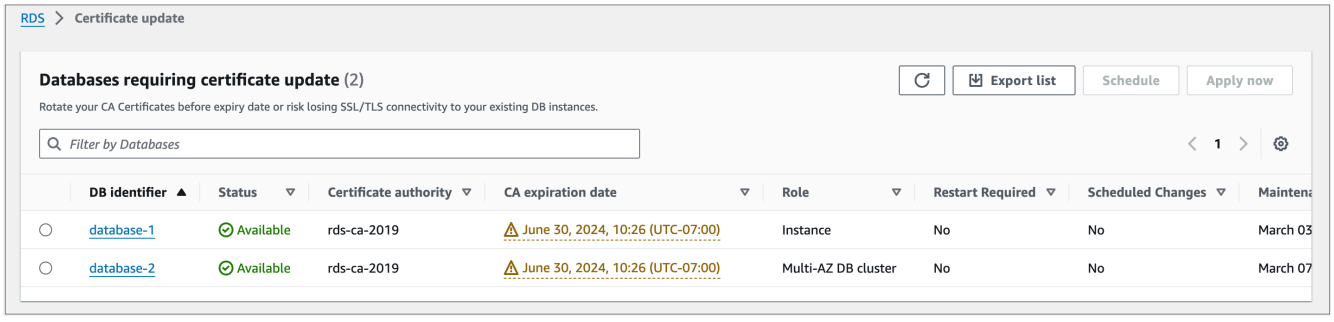
Console

Como atualizar o certificado CA aplicando a manutenção

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, selecione Atualização de certificado.



A página Bancos de dados que precisam de atualização de certificado é exibida.



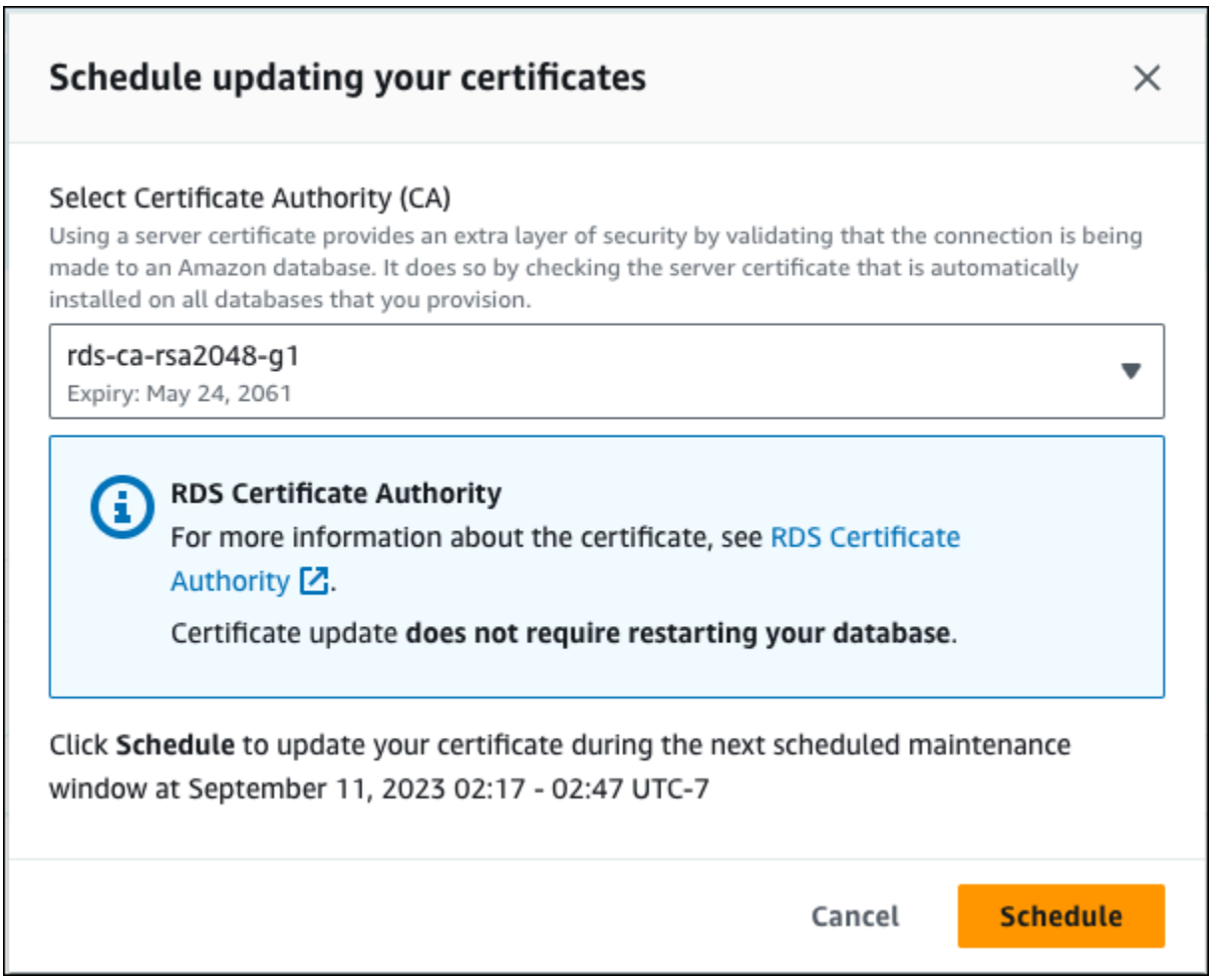
> **Note**
>
> Essa página mostra apenas as instâncias e os clusters de banco de dados na Região da AWS atual. Se você tiver bancos de dados em mais de uma Região da AWS, confira essa página em cada Região da AWS para ver todas as instâncias de banco de dados com certificados SSL/TLS antigos.

3. Escolha a instância de banco de dados ou o cluster de banco de dados multi-AZ que você deseja atualizar.

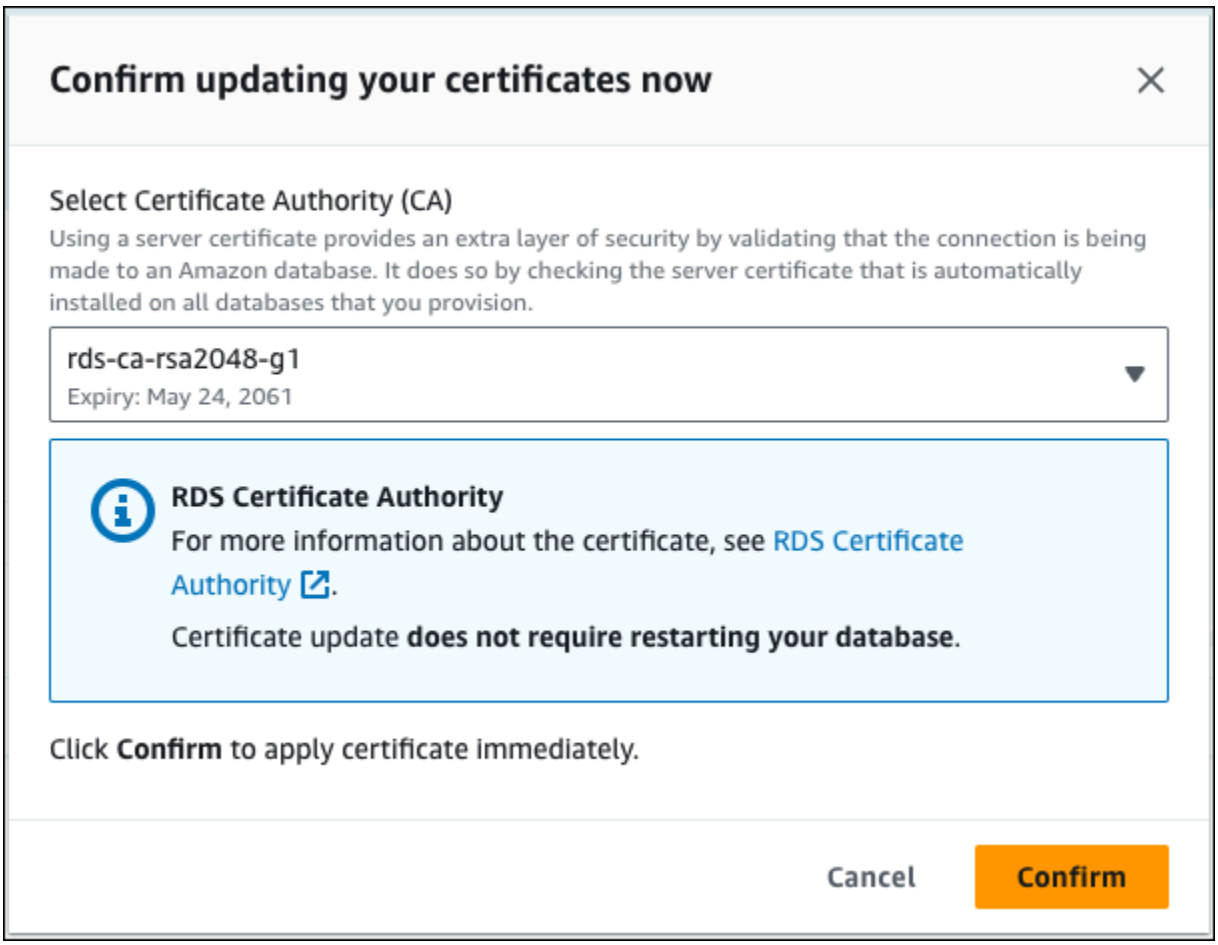
   Você pode programar a alternância de certificado para sua próxima janela de manutenção escolhendo Programar. Aplique a mudança imediatamente escolhendo Aplicar agora.

   > **Important**
   >
   > Se você tiver problemas de conectividade após a expiração do certificado, use a opção Aplicar agora.

4. a. Se você escolher Programar, precisará confirmar a alternância do certificado de CA. Essa solicitação de confirmação também indica a janela agendada para sua atualização.



b. Se você escolher Aplicar agora, precisará confirmar a alternância do certificado de CA.



> **Important**
>
> Antes de programar a rotação do certificado CA no banco de dados, atualize todas as aplicações cliente que usam SSL/TLS e o certificado do servidor para se conectar. Essas atualizações são específicas ao seu mecanismo de banco de dados. Depois de atualizar essas aplicações cliente, você pode confirmar a rotação do certificado CA.

Para continuar, escolha a caixa de seleção e escolha Confirm (Confirmar).

5. Repita as etapas 3 e 4 para cada instância e cluster de banco de dados instância que você deseja atualizar.

Alternância automática de certificados do servidor

Se a CA raiz comportar a troca automática de certificados de servidor, o RDS gerenciará automaticamente a troca do certificado do servidor de banco de dados. Como o RDS usa a mesma CA raiz para essa alternância automática, então você não precisa baixar um novo pacote de CA. Consulte Autoridades certificadoras.

A alternância e a validade do certificado do servidor de banco de dados dependem do mecanismo de banco de dados:

- Se o mecanismo de banco de dados comportar a alternância sem reinicialização, o RDS alternará automaticamente o certificado do servidor de banco de dados sem exigir nenhuma ação de sua parte. O RDS tenta alternar o certificado do servidor de banco de dados em sua janela de manutenção preferida na meia-vida do respectivo certificado. O novo certificado do servidor de banco de dados é válido por 12 meses.
- Se o mecanismo de banco de dados não comportar a alternância sem reinicialização, o RDS notificará você sobre um evento de manutenção pelo menos seis meses antes da expiração do certificado do servidor de banco de dados. O novo certificado do servidor de banco de dados é válido por 12 meses.

Use o comando describe-db-engine-versions e inspecione o sinalizador `SupportsCertificateRotationWithoutRestart` para identificar se a versão do mecanismo de banco de dados é compatível com a alternância de certificado sem reinicialização. Para obter mais informações, consulteConfigurar a CA do banco de dados

Script de exemplo para importar certificados para o seu armazenamento confiável

Veja os exemplos de scripts do shell que importam o pacote de certificados para um armazenamento de confiança.

Cada script de shell de amostra usa o keytool, que faz parte do Java Development Kit (JDK). Para obter mais informações sobre como instalar o JDK, consulte o Guia de instalação do JDK.

Linux

Veja a seguir um exemplo de script shell que importa o pacote de certificados para um armazenamento confiável em um sistema operacional Linux.

```
mydir=tmp/certs
```

```
if [ ! -e "${mydir}" ]
then
mkdir -p "${mydir}"
fi truststore=${mydir}/rds-truststore.jks storepassword=changeit

curl -sS "https://truststore.pki.rds.amazonaws.com/global/global-bundle.pem">
 ${mydir}/global-bundle.pem
awk 'split_after == 1 {n++;split_after=0} /-----END CERTIFICATE-----/
 {split_after=1}{print > "rds-ca-" n+1 ".pem"}' < ${mydir}/global-bundle.pem

for CERT in rds-ca-*; do alias=$(openssl x509 -noout -text -in $CERT | perl -ne
 'next unless /Subject:/; s/.*(CN=|CN = )//; print')
  echo "Importing $alias"
  keytool -import -file ${CERT} -alias "${alias}" -storepass ${storepassword} -
keystore ${truststore} -noprompt
  rm $CERT
done

rm ${mydir}/global-bundle.pem

echo "Trust store content is: "

keytool -list -v -keystore "$truststore" -storepass ${storepassword} | grep Alias |
 cut -d " " -f3- | while read alias
do expiry=`keytool -list -v -keystore "$truststore" -storepass ${storepassword} -
alias "${alias}" | grep Valid | perl -ne 'if(/until: (.*?)\n/) { print "$1\n"; }'`
   echo " Certificate ${alias} expires in '$expiry'"
done
```

macOS

Veja a seguir um exemplo de script do shell que importa o pacote de certificados em um armazenamento de confiança no macOS.

```
mydir=tmp/certs
if [ ! -e "${mydir}" ]
then
mkdir -p "${mydir}"
fi truststore=${mydir}/rds-truststore.jks storepassword=changeit

curl -sS "https://truststore.pki.rds.amazonaws.com/global/global-bundle.pem">
 ${mydir}/global-bundle.pem
```

```
split -p "-----BEGIN CERTIFICATE-----" ${mydir}/global-bundle.pem rds-ca-

for CERT in rds-ca-*; do alias=$(openssl x509 -noout -text -in $CERT | perl -ne
 'next unless /Subject:/; s/.*(CN=|CN = )//; print')
  echo "Importing $alias"
  keytool -import -file ${CERT} -alias "${alias}" -storepass ${storepassword} -
keystore ${truststore} -noprompt
  rm $CERT
done

rm ${mydir}/global-bundle.pem

echo "Trust store content is: "

keytool -list -v -keystore "$truststore" -storepass ${storepassword} | grep Alias |
 cut -d " " -f3- | while read alias
do expiry=`keytool -list -v -keystore "$truststore" -storepass ${storepassword} -
alias "${alias}" | grep Valid | perl -ne 'if(/until: (.*?)\n/) { print "$1\n"; }'`
   echo " Certificate ${alias} expires in '$expiry'"
done
```

Privacidade do tráfego entre redes

As conexões são protegidas entre o Amazon RDS e as aplicações on-premises e entre o Amazon RDS e outros recursos da AWS na mesma região da AWS.

Tráfego entre clientes de serviço e on-premises e as aplicações

Você tem duas opções de conectividade entre sua rede privada e a AWS:

- Uma conexão AWS Site-to-Site VPN. Para ter mais informações, consulte O que é o AWS Site-to-Site VPN?
- Uma conexão AWS Direct Connect. Para ter mais informações, consulte O que é o AWS Direct Connect?

Você obtém acesso ao Amazon RDS pela rede usando operações de API publicadas pela AWS. Os clientes devem oferecer compatibilidade com:

- Transport Layer Security (TLS). Exigimos TLS 1.2 e recomendamos TLS 1.3.

- Conjuntos de criptografia com perfect forward secrecy (PFS) como DHE (Ephemeral Diffie-Hellman) ou ECDHE (Ephemeral Elliptic Curve Diffie-Hellman). A maioria dos sistemas modernos, como Java 7 e versões posteriores, comporta esses modos.

Além disso, as solicitações devem ser assinadas usando um ID da chave de acesso e uma chave de acesso secreta associada a uma entidade principal do IAM. Ou você pode usar o AWS Security Token Service (AWS STS) para gerar credenciais de segurança temporárias para assinar solicitações.

Gerenciamento de identidade e acesso no Amazon RDS

O AWS Identity and Access Management (IAM) é um serviço da AWS service (Serviço da AWS) que ajuda um administrador a controlar com segurança o acesso aos recursos da AWS. Os administradores do IAM controlam quem pode ser autenticado (conectado) e autorizado (ter permissões) para utilizar os recursos do Amazon RDS. O IAM é um AWS service (Serviço da AWS) que pode ser usado sem custo adicional.

Tópicos

- Público
- Autenticar com identidades
- Gerenciamento do acesso usando políticas
- Como o Amazon RDS funciona com o IAM
- Exemplos de políticas baseadas em identidade do Amazon RDS
- Políticas gerenciadas pela AWS para o Amazon RDS
- Atualizações do Amazon RDS para políticas gerenciadas pela AWS
- Prevenção do problema do substituto confuso entre serviços
- Autenticação do banco de dados do IAM para MariaDB, MySQL e PostgreSQL
- Solução de problemas de identidade e acesso do Amazon RDS

Público

A forma de usar o AWS Identity and Access Management (IAM) varia em função do trabalho realizado no Amazon RDS.

Usuário do serviço: se você usar o serviço Amazon RDS para fazer sua tarefa, o administrador fornecerá as credenciais e as permissões de que você precisa. À medida que usar mais recursos do Amazon RDS para fazer seu trabalho, você poderá precisar de permissões adicionais. Entender como o acesso é gerenciado pode ajudar você a solicitar as permissões corretas ao seu administrador. Se não for possível acessar um recurso no Amazon RDS, consulte Solução de problemas de identidade e acesso do Amazon RDS.

Administrador do serviço: se você for o responsável pelos recursos do Amazon RDS em sua empresa, você provavelmente terá acesso total ao Amazon RDS. Seu trabalho é determinar quais

recursos do Amazon RDS seus funcionários devem acessar. Assim, é necessário enviar solicitações ao administrador do para alterar as permissões dos usuários de seu serviço. Revise as informações nesta página para compreender os conceitos básicos do IAM. Para saber mais sobre como sua empresa pode usar o IAM com o Amazon RDS, consulte Como o Amazon RDS funciona com o IAM.

Administrador: se você é um administrador, talvez queira saber detalhes sobre como pode escrever políticas para gerenciar o acesso ao Amazon RDS. Para visualizar exemplos de políticas baseadas em identidade do Amazon RDS que podem ser usadas no IAM, consulte Exemplos de políticas baseadas em identidade do Amazon RDS.

Autenticar com identidades

A autenticação é a forma como fazer login na AWS usando suas credenciais de identidade. É necessário ser autenticado (fazer login na AWS) como Usuário raiz da conta da AWS, como usuário do IAM, ou assumindo um perfil do IAM.

É possível fazer login na AWS como uma identidade federada usando credenciais fornecidas por uma fonte de identidades. AWS IAM Identity Center (Centro de Identidade do IAM), a autenticação única da empresa e as suas credenciais do Google ou do Facebook são exemplos de identidades federadas. Quando você faz login como identidade federada, o administrador já configurou anteriormente a federação de identidades usando perfis do IAM. Quando você acessa a AWS usando a federação, está indiretamente assumindo um perfil.

A depender do tipo de usuário, você pode fazer login no AWS Management Console ou no portal de acesso AWS. Para obter mais informações sobre como fazer login na AWS, consulte Como fazer login na contaConta da AWS no Início de Sessão da AWS Guia do usuário.

Se você acessar a AWS de forma programática, a AWS fornecerá um kit de desenvolvimento de software (SDK) e uma interface de linha de comandos (CLI) para você assinar de forma criptográfica as solicitações usando as suas credenciais. Se você não utilizar as ferramentas da AWS, deverá assinar as solicitações por conta própria. Para obter mais informações sobre como usar o método recomendado para assinar solicitações por conta própria, consulte Signature Version 4 da AWS para solicitações de API no Guia de usuário do IAM.

Independente do método de autenticação usado, também pode ser necessário fornecer informações adicionais de segurança. Por exemplo, a AWS recomenda o uso da autenticação multifator (MFA) para aumentar a segurança de sua conta. Para saber mais, consulte Autenticação multifator no Guia do Usuário do AWS IAM Identity Center e Código da autenticação multifator no IAM da AWS no Guia do Usuário do IAM.

Usuário raiz de conta da AWS

Ao criar uma Conta da AWS, você começa com uma identidade de login com acesso completo a todos os Serviços da AWS e recursos na conta. Essa identidade, chamada usuário-raiz da Conta da AWS, é acessada por login com o endereço de e-mail e a senha usada para criar a conta. É altamente recomendável não usar o usuário-raiz para tarefas diárias. Proteja as credenciais do usuário-raiz e use-as para executar as tarefas que somente ele puder executar. Para obter a lista completa das tarefas que exigem login como usuário-raiz, consulte Tarefas que exigem credenciais de usuário-raiz no Guia do Usuário do IAM.

Identidade federada

Como prática recomendada, exija que os usuários, inclusive os que precisam de acesso de administrador, usem a federação com um provedor de identidades para acessar os Serviços da AWS usando credenciais temporárias.

Uma identidade federada é um usuário de seu diretório de usuários corporativos, um provedor de identidades da web, o AWS Directory Service, o diretório do Centro de Identidade ou qualquer usuário que acesse os Serviços da AWS usando credenciais fornecidas por meio de uma fonte de identidade. Quando as identidades federadas acessam Contas da AWS, elas assumem perfis que fornecem credenciais temporárias.

Para o gerenciamento de acesso centralizado, é recomendável usar o AWS IAM Identity Center. É possível criar usuários e grupos no IAM Identity Center ou conectar-se e sincronizar com um conjunto de usuários e grupos em sua própria fonte de identidade para uso em todas as suas Contas da AWS e aplicações. Para obter mais informações sobre o Centro de Identidade do IAM, consulte O que é o Centro de Identidade do IAM? no Guia do Usuário do AWS IAM Identity Center.

Usuários e grupos do IAM

Um usuário do IAM é uma identidade dentro da Conta da AWS que tem permissões específicas para uma única pessoa ou aplicação. Sempre que possível, é recomendável contar com credenciais temporárias em vez de criar usuários do IAM com credenciais de longo prazo, como senhas e chaves de acesso. No entanto, se você tiver casos de uso específicos que exijam credenciais de longo prazo com usuários do IAM, é recomendável alternar as chaves de acesso. Para obter mais informações, consulte Alternar as chaves de acesso regularmente para casos de uso que exijam credenciais de longo prazo no Guia do Usuário do IAM.

Um grupo do IAM é uma identidade que especifica uma coleção de usuários do IAM. Não é possível fazer login como um grupo. É possível usar grupos para especificar permissões para vários usuários

de uma vez. Os grupos facilitam o gerenciamento de permissões para grandes conjuntos de usuários. Por exemplo, você pode ter um grupo chamado IAMAdmins e conceder a esse grupo permissões para administrar recursos do IAM.

Usuários são diferentes de perfis. Um usuário é exclusivamente associado a uma pessoa ou a uma aplicação, mas um perfil pode ser assumido por qualquer pessoa que precisar dele. Os usuários têm credenciais permanentes de longo prazo, mas os perfis fornecem credenciais temporárias. Para saber mais, consulte Casos de uso de usuários do IAM no Guia do usuário do IAM.

Você pode se autenticar na instância de banco de dados usando a autenticação de banco de dados do IAM.

A autenticação do banco de dados do IAM funciona com os seguintes mecanismos de banco de dados:

- RDS para MariaDB
- RDS para MySQL
- RDS para PostgreSQL.

Para obter mais informações sobre a autenticação na instância usando o IAM, consulte Autenticação do banco de dados do IAM para MariaDB, MySQL e PostgreSQL.

Perfis do IAM

Um perfil do IAM é uma identidade na Conta da AWS que tem permissões específicas. Ela é semelhante a um usuário, mas não está associada a uma pessoa específica. É possível presumir temporariamente um perfil do IAM no AWS Management Console alternando perfis. É possível presumir um perfil chamando uma operação de API da AWS CLI ou da AWS, ou usando um URL personalizado. Para obter mais informações sobre métodos para o uso de perfis, consulte Utilizar perfis do IAM no Guia do usuário do IAM.

Funções do IAM com credenciais temporárias são úteis nas seguintes situações:

- Permissões temporárias para usuários: um usuário pode assumir um perfil do IAM para obter temporariamente permissões diferentes para uma tarefa específica.
- Acesso de usuário federado: para atribuir permissões a identidades federadas, é possível criar um perfil e definir permissões para ele. Quando uma identidade federada é autenticada, essa identidade é associada ao perfil e recebe as permissões definidas por ele. Para ter mais

informações sobre perfis para federação, consulte Criar um perfil para um provedor de identidade de terceiros (federação) no Guia do usuário do IAM. Se usar o Centro de Identidade do IAM, configure um conjunto de permissões. Para controlar o que suas identidades podem acessar após a autenticação, o Centro de Identidade do IAM correlaciona o conjunto de permissões a um perfil no IAM. Para obter informações sobre conjuntos de permissões, consulte Conjuntos de Permissões no Guia do Usuário do AWS IAM Identity Center.

- Acesso entre contas: é possível usar um perfil do IAM para permitir que alguém (uma entidade principal confiável) em outra conta acesse recursos em sua conta. Os perfis são a principal forma de conceder acesso entre contas. No entanto, alguns Serviços da AWS permitem que você anexe uma política diretamente a um recurso (em vez de usar um perfil como proxy). Para saber a diferença entre perfis e políticas baseadas em recurso para acesso entre contas, consulte Como os perfis do IAM diferem das políticas baseadas em recurso no Guia do usuário do IAM.
- Acesso entre serviços: alguns Serviços da AWS usam atributos em outros Serviços da AWS. Por exemplo, quando você faz uma chamada em um serviço, é comum que esse serviço execute aplicações no Amazon EC2 ou armazene objetos no Amazon S3. Um serviço pode fazer isso usando as permissões da entidade principal da chamada, usando um perfil de serviço ou um perfil vinculado ao serviço.
  - Sessões de acesso direto: ao usar um usuário ou perfil do IAM para realizar ações na AWS, você é considerado uma entidade principal. Ao usar alguns serviços, você pode executar uma ação que inicia outra ação em um serviço diferente. O recurso FAS usa as permissões da entidade principal chamando um AWS service (Serviço da AWS), bem como o AWS service (Serviço da AWS) solicitante, para fazer solicitações para serviços subsequentes. As solicitações de FAS são feitas somente quando um serviço recebe uma solicitação que exige interações com outros Serviços da AWS ou com recursos para serem concluídas. Nesse caso, você precisa ter permissões para executar ambas as ações. Para obter detalhes da política ao fazer solicitações de FAS, consulte Sessões de acesso direto.
  - Perfil de serviço: um perfil de serviço é um perfil do IAM que um serviço assume para executar ações em seu nome. Um administrador do IAM pode criar, modificar e excluir um perfil de serviço do IAM. Para obter mais informações, consulte Criar um perfil para delegar permissões a um AWS service (Serviço da AWS) no Guia do Usuário do IAM.
  - Perfil vinculado a serviço: um perfil vinculado a serviço é um tipo de perfil de serviço vinculado a um AWS service (Serviço da AWS). O serviço pode presumir o perfil para executar uma ação em seu nome. Perfis vinculadas ao serviço aparecem em sua Conta da AWS e são de propriedade do serviço. Um administrador do IAM pode visualizar, mas não editar as permissões para perfis vinculados a serviço.

- Aplicações em execução no Amazon EC2: é possível usar um perfil do IAM para gerenciar credenciais temporárias para aplicações em execução em uma instância do EC2 e fazer solicitações da AWS CLI ou da API da AWS. É preferível fazer isso a armazenar chaves de acesso na instância do EC2. Para atribuir um perfil da AWS a uma instância do EC2 e disponibilizá-la para todas as suas aplicações, crie um perfil de instância que esteja anexado a ela. Um perfil de instância contém o perfil e permite que os programas em execução na instância do EC2 obtenham credenciais temporárias. Para mais informações, consulte Utilizar um perfil do IAM para conceder permissões a aplicações em execução nas instâncias do Amazon EC2 no Guia do usuário do IAM.

Para saber se você deve usar perfis do IAM, consulte Quando criar um perfil do IAM (em vez de um usuário) no Guia do usuário do IAM.

Gerenciamento do acesso usando políticas

Você controla o acesso na AWS criando e anexando políticas às identidades do IAM ou aos recursos da AWS. Uma política é um objeto na AWS que, quando associada a uma identidade ou a um recurso, define suas permissões. A AWS avalia essas políticas quando uma entidade (usuário raiz, usuário ou perfil do IAM) faz uma solicitação. As permissões nas políticas determinam se a solicitação será permitida ou negada. A maioria das políticas é armazenada na AWS como documentos JSON. Para obter mais informações sobre a estrutura e o conteúdo de documentos de políticas JSON, consulte Visão geral das políticas JSON no Guia do usuário do IAM.

Um administrador pode usar políticas para especificar quem tem acesso aos recursos da AWS e quais ações essas pessoas podem executar nesses recursos. Cada entidade do IAM (conjunto de permissões ou perfil) começa sem permissões. Em outras palavras, por padrão, os usuários não podem fazer nada, nem mesmo alterar sua própria senha. Para dar permissão a um usuário para fazer algo, um administrador deve anexar uma política de permissões ao usuário. Ou o administrador pode adicionar o usuário a um grupo que tenha as permissões pretendidas. Quando um administrador concede permissões a um grupo, todos os usuários desse grupo recebem essas permissões.

As políticas do IAM definem permissões para uma ação, independentemente do método usado para executar a operação. Por exemplo, suponha que você tenha uma política que permite a ação `iam:GetRole`. Um usuário com essa política pode obter informações de perfis do AWS Management Console, da AWS CLI ou da API AWS.

Políticas baseadas em identidade

As políticas baseadas em identidade são documentos JSON de políticas de permissões que você pode anexar a uma identidade, como um conjunto de permissões ou um perfil. Essas políticas controlam quais ações cada identidade pode realizar, em quais recursos e em que condições. Para saber como criar uma política baseada em identidade, consulte Criar políticas do IAM no Guia do usuário do IAM.

As políticas baseadas em identidade podem ser categorizadas ainda adicionalmente como políticas em linha ou políticas gerenciadas. As políticas em linha são anexadas diretamente a um único conjunto de permissões ou perfil. As políticas gerenciadas são políticas independentes que podem ser anexadas a vários conjuntos de permissões e perfis em sua conta da AWS. As políticas gerenciadas incluem políticas gerenciadas pela AWS e políticas gerenciadas pelo cliente. Para saber como escolher entre uma política gerenciada ou uma política em linha, consulte Escolher entre políticas gerenciadas e políticas em linha no Guia do Usuário do IAM.

Para obter informações sobre políticas gerenciadas pela AWS específicas do Amazon RDS, consulte Políticas gerenciadas pela AWS para o Amazon RDS.

Outros tipos de política

A AWS oferece compatibilidade com tipos de política menos comuns. Esses tipos de política podem definir o máximo de permissões concedidas a você pelos tipos de política mais comuns.

- Limites de permissões: um limite de permissões é um recurso avançado no qual você define o máximo de permissões que uma política baseada em identidade pode conceder a uma entidade do IAM (conjunto de permissões ou perfil). É possível definir um limite de permissões para uma entidade. As permissões resultantes são a interseção das políticas baseadas em identidade da entidade e seus limites de permissões. As políticas baseadas em recurso que especificam o conjunto de permissões ou o perfil no campo `Principal` não são limitadas pelo limite de permissões. Uma negação explícita em qualquer uma dessas políticas substitui a permissão. Para obter mais informações sobre limites de permissões, consulte Limites de permissões para identidades do IAM no Guia do usuário do IAM.
- Políticas de controle de serviço (SCPs): SCPs são políticas JSON que especificam as permissões máximas para uma organização ou unidade organizacional (UO) no AWS Organizations. O AWS Organizations é um serviço para agrupar e gerenciar centralmente várias contas da AWS pertencentes à sua empresa. Se você habilitar todos os recursos em uma organização, poderá aplicar políticas de controle de serviço (SCPs) a qualquer uma ou a todas as contas. O SCP

limita as permissões para entidades em contas membro, o que inclui cada Usuário raiz da conta da AWS. Para obter mais informações sobre o Organizações e SCPs, consulte Como os SCPs Funcionam no Manual do Usuário do AWS Organizations.

- Políticas de sessão: são políticas avançadas que você transmite como um parâmetro quando cria de forma programática uma sessão temporária para um perfil ou um usuário federado. As permissões da sessão resultante são a interseção entre as políticas baseadas em identidade dos conjuntos de permissões ou do perfil e as políticas de sessão. As permissões também podem ser provenientes de uma política baseada em recursos. Uma negação explícita em qualquer uma dessas políticas substitui a permissão. Para obter mais informações, consulte Políticas de sessão no Guia do usuário do IAM.

Vários tipos de política

Quando vários tipos de política são aplicáveis a uma solicitação, é mais complicado compreender as permissões resultantes. Para saber como a AWS determina se deve permitir ou não uma solicitação quando há vários tipos de política envolvidos, consulte Lógica da avaliação de políticas no Guia do usuário do IAM.

Como o Amazon RDS funciona com o IAM

Antes de usar o IAM para gerenciar o acesso ao Amazon RDS, você precisa entender quais recursos do IAM estão disponíveis para uso com o Amazon RDS.

A tabela a seguir lista os recursos do IAM que você pode usar com o Amazon RDS:

Atributo do IAM	Compatibilidade do Amazon RDS
Políticas baseadas em identidade	Sim
Políticas baseadas em recurso	Não
Ações de políticas	Sim
Recursos de políticas	Sim
Chaves de condição de política (específicas do serviço)	Sim
ACLs	Não

Atributo do IAM	Compatibilidade do Amazon RDS
Controle de acesso baseado em atributos (ABAC) (tags em políticas)	Sim
Credenciais temporárias	Sim
Sessões de acesso direto	Sim
Perfis de serviço	Sim
Perfis vinculados a serviço	Sim

Para obter uma visão de alto nível de como o Amazon RDS e outros serviços da AWS funcionam com o IAM, consulte Serviços da AWS compatíveis com o IAM no Guia do usuário do IAM.

Tópicos

- Políticas baseadas em identidade do Amazon RDS
- Políticas baseadas em recursos do Amazon RDS
- Ações de políticas para o Amazon RDS
- Recursos de políticas do Amazon RDS
- Chaves de condição de políticas do Amazon RDS
- Listas de controle de acesso (ACLs) no Amazon RDS
- Controle de acesso baseado em atributos (ABAC) em políticas com tags do Amazon RDS
- Usar credenciais temporárias com o Amazon RDS
- Sessões de acesso direto para o Amazon RDS
- Perfis de serviço do Amazon RDS
- Funções vinculadas a serviço do Amazon RDS

Políticas baseadas em identidade do Amazon RDS

Compatível com políticas baseadas em identidade: sim.

As políticas baseadas em identidade são documentos de políticas de permissões JSON que você pode anexar a uma identidade, como usuário, grupo de usuários ou perfil do IAM. Essas

políticas controlam quais ações os usuários e perfis podem realizar, em quais recursos e em que condições. Para saber como criar uma política baseada em identidade, consulte Definir permissões personalizadas do IAM com políticas gerenciadas pelo cliente no Guia do usuário do IAM.

Com as políticas baseadas em identidade do IAM, é possível especificar ações e recursos permitidos ou negados, assim como as condições sob as quais as ações são permitidas ou negadas. Você não pode especificar a entidade principal em uma política baseada em identidade porque ela se aplica ao usuário ou perfil ao qual ela está anexada. Para saber mais sobre todos os elementos que podem ser usados em uma política JSON, consulte Referência de elemento de política JSON do IAM no Guia do usuário do IAM.

Exemplos de políticas baseadas em identidade do Amazon RDS

Para visualizar exemplos de políticas baseadas em identidade do Amazon RDS, consulte Exemplos de políticas baseadas em identidade do Amazon RDS.

Políticas baseadas em recursos do Amazon RDS

Compatível com políticas baseadas em recursos: não.

Políticas baseadas em recursos são documentos de políticas JSON que você anexa a um recurso. São exemplos de políticas baseadas em recursos as políticas de confiança de perfil do IAM e as políticas de bucket do Amazon S3. Em serviços compatíveis com políticas baseadas em recursos, os administradores de serviço podem usá-las para controlar o acesso a um recurso específico. Para o atributo ao qual a política está anexada, a política define quais ações uma entidade principal especificado pode executar nesse atributo e em que condições. Você deve especificar uma entidade principal em uma política baseada em recursos. As entidades principais podem incluir contas, usuários, perfis, usuários federados ou Serviços da AWS.

Para permitir o acesso entre contas, você pode especificar uma conta inteira ou as entidades do IAM em outra conta como a entidade principal em uma política baseada em recursos. Adicionar uma entidade principal entre contas à política baseada em recurso é apenas metade da tarefa de estabelecimento da relação de confiança. Quando a entidade principal e o recurso estão em diferentes Contas da AWS, um administrador do IAM da conta confiável também deve conceder à entidade principal (usuário ou perfil) permissão para acessar o recurso. Eles concedem permissão ao anexar uma política baseada em identidade para a entidade. No entanto, se uma política baseada em recurso conceder acesso a uma entidade principal na mesma conta, nenhuma política baseada em identidade adicional será necessária. Consulte mais informações em Acesso a recursos entre contas no IAM no Guia do usuário do IAM.

Ações de políticas para o Amazon RDS

Compatível com ações de políticas: sim.

Os administradores podem usar as políticas JSON da AWS para especificar quem tem acesso a quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

O elemento `Action` de uma política JSON descreve as ações que podem ser usadas para permitir ou negar acesso em uma política. As ações de políticas geralmente têm o mesmo nome que a operação de API da AWS associada. Existem algumas exceções, como ações somente de permissão, que não têm uma operação de API correspondente. Algumas operações também exigem várias ações em uma política. Essas ações adicionais são chamadas de ações dependentes.

Incluem ações em uma política para conceder permissões para executar a operação associada.

As ações de políticas no Amazon RDS usam o seguinte prefixo antes da ação: `rds:`. Por exemplo, para conceder a alguém permissão para descrever instâncias de banco de dados com a operação da API `DescribeDBInstances` do Amazon RDS, inclua a ação `rds:DescribeDBInstances` na política. As instruções de política devem incluir um elemento `Action` ou `NotAction`. O Amazon RDS define seu próprio conjunto de ações que descrevem as tarefas que você pode executar com esse serviço.

Para especificar várias ações em uma única declaração, separe-as com vírgulas, conforme a seguir.

```
"Action": [
      "rds:action1",
      "rds:action2"
```

Você também pode especificar várias ações utilizando caracteres curinga (*). Por exemplo, para especificar todas as ações que começam com a palavra `Describe`, inclua a ação a seguir:

```
"Action": "rds:Describe*"
```

Para obter uma lista de ações do Amazon RDS, consulte Ações definidas pelo Amazon RDS na Referência de autorização do serviço.

Recursos de políticas do Amazon RDS

Compatível com recursos de políticas: sim.

Os administradores podem usar as políticas JSON da AWS para especificar quem tem acesso a quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

O elemento de política JSON `Resource` especifica o objeto ou os objetos aos quais a ação se aplica. As instruções devem incluir um elemento `Resource` ou `NotResource`. Como prática recomendada, especifique um recurso usando seu nome do recurso da Amazon (ARN). Isso pode ser feito para ações que oferecem compatibilidade com um tipo de recurso específico, conhecido como permissões em nível de recurso.

Para ações que não oferecem compatibilidade com permissões em nível de recurso, como operações de listagem, use um curinga (*) para indicar que a instrução se aplica a todos os recursos.

```
"Resource": "*"
```

O recurso de instância de banco de dados tem o nome do recurso da Amazon (ARN) a seguir.

```
arn:${Partition}:rds:${Region}:${Account}:{ResourceType}/${Resource}
```

Para obter mais informações sobre o formato de ARNs, consulte Nomes de recursos da Amazon (ARNs) e namespaces de serviços da AWS.

Por exemplo, para especificar a instância de banco de dados `dbtest` em sua instrução, use o ARN a seguir.

```
"Resource": "arn:aws:rds:us-west-2:123456789012:db:dbtest"
```

Para especificar todas as instâncias de banco de dados que pertencem a uma conta específica, use o caractere curinga (*).

```
"Resource": "arn:aws:rds:us-east-1:123456789012:db:*"
```

Algumas operações da API do RDS, como as operações para a criação de recursos, não podem ser executadas em um recurso específico. Nesses casos, use o caractere curinga (*).

```
"Resource": "*"
```

Muitas operações da API do Amazon RDS envolvem vários recursos. Por exemplo, o `CreateDBInstance` cria uma instância de banco de dados. Você pode especificar que um usuário do deve usar um grupo de segurança e um grupo de parâmetros específicos ao criar uma instância

de banco de dados. Para especificar vários recursos em uma única declaração, separe os ARNs com vírgulas.

```
"Resource": [
      "resource1",
      "resource2"
```

Para ver uma lista dos tipos de recursos do Amazon RDS e seus ARNs, consulte Tipos de recursos definidos pelo Amazon RDS na Referência de autorização do serviço. Para saber com quais ações você pode especificar o ARN de cada recurso, consulte Ações definidas pelo Amazon RDS.

Chaves de condição de políticas do Amazon RDS

Compatível com chaves de condição de política específicas de serviço: sim.

Os administradores podem usar as políticas JSON da AWS para especificar quem tem acesso a quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

O elemento `Condition` (ou bloco `Condition`) permite que você especifique condições nas quais uma instrução estiver em vigor. O elemento `Condition` é opcional. É possível criar expressões condicionais que usem agentes de condição, como “igual a” ou “menor que”, para fazer a condição da política corresponder aos valores na solicitação.

Se você especificar vários elementos de `Condition` em uma declaração ou várias chaves em um único elemento de `Condition`, a AWS os avaliará usando uma operação lógica `AND`. Se você especificar vários valores para uma única chave de condição, a AWS avaliará a condição usando uma operação lógica `OR`. Todas as condições devem ser atendidas antes que as permissões da instrução sejam concedidas.

Você também pode usar variáveis de espaço reservado ao especificar condições. Por exemplo, é possível conceder a um usuário do IAM permissão para acessar um recurso somente se ele estiver marcado com seu nome de usuário do IAM. Para obter mais informações, consulte Elementos da política do IAM: variáveis e tags no Guia do usuário do IAM.

A AWS oferece compatibilidade com chaves de condição globais e chaves de condição específicas do serviço. Para ver todas as chaves de condição globais da AWS, consulte Chaves de contexto de condição globais da AWS no Guia do usuário do IAM.

O Amazon RDS define seu próprio conjunto de chaves de condição e também é compatível com o uso de algumas chaves de condição globais. Para ver todas as chaves de condição globais da AWS, consulte Chaves de contexto de condição globais da AWS no Guia do usuário do IAM.

Todas as operações da API do RDS oferecem suporte à chave de condição `aws:RequestedRegion`.

Para ver uma lista das chaves de condição do Amazon RDS, consulte Chaves de condição do Amazon RDS na Referência de autorização do serviço. Para saber com quais ações e recursos você pode usar a chave de condição, consulte Ações definidas pelo Amazon RDS.

Listas de controle de acesso (ACLs) no Amazon RDS

Compatível com listas de controle de acesso (ACLs): não.

As listas de controle de acesso (ACLs) controlam quais entidades principais (membros, usuários ou perfis da conta) têm permissões para acessar um recurso. As ACLs são semelhantes às políticas baseadas em recursos, embora não usem o formato de documento de política JSON.

Controle de acesso baseado em atributos (ABAC) em políticas com tags do Amazon RDS

Compatível com tags de controle de acesso baseado em atributos (ABAC) em políticas: sim.

O controle de acesso por atributo (ABAC) é uma estratégia de autorização que define as permissões com base em atributos. Na AWS, esses atributos são chamados de tags. É possível anexar tags a entidades do IAM (usuários ou perfis) e a muitos recursos da AWS. Marcar de entidades e atributos é a primeira etapa do ABAC. Em seguida, você cria políticas de ABAC para permitir operações quando a tag da entidade principal corresponder à tag do recurso que ela estiver tentando acessar.

O ABAC é útil em ambientes que estão crescendo rapidamente e ajuda em situações em que o gerenciamento de políticas se torna um problema.

Para controlar o acesso baseado em tags, forneça informações sobre as tags no elemento de condição de uma política usando as `aws:ResourceTag/`*`key-name`*, `aws:RequestTag/`*`key-name`* ou chaves de condição `aws:TagKeys`.

Se um serviço for compatível com as três chaves de condição para cada tipo de recurso, o valor será Sim para o serviço. Se um serviço for compatível com as três chaves de condição somente para alguns tipos de recursos, o valor será Parcial

Para obter mais informações sobre o ABAC, consulte Definir permissões com autorização do ABAC no Guia do usuário do IAM. Para visualizar um tutorial com etapas para configurar o ABAC, consulte Usar controle de acesso baseado em atributos (ABAC) no Guia do usuário do IAM.

Para obter mais informações sobre recursos de marcação do Amazon RDS, consulte Especificar condições: usar tags personalizadas. Para visualizar um exemplo de política baseada em identidade para limitar o acesso a um recurso baseado em tags desse recurso, consulte Conceder permissão para ações em um recurso com uma tag específica com dois valores diferentes.

Usar credenciais temporárias com o Amazon RDS

Compatível com credenciais temporárias: sim.

Alguns Serviços da AWS não funcionam quando você faz login usando credenciais temporárias. Para obter informações adicionais, como quais Serviços da AWS funcionam com credenciais temporárias, consulte Serviços da AWS que funcionem com o IAM no Guia do usuário do IAM.

Você está usando credenciais temporárias se fizer login no AWS Management Console por qualquer método, exceto nome de usuário e uma senha. Por exemplo, quando você acessa a AWS usando o link de autenticação única (SSO) da sua empresa, esse processo cria credenciais temporárias automaticamente. Você também cria automaticamente credenciais temporárias quando faz login no console como usuário e, em seguida, alterna perfis. Para obter mais informações sobre como alternar perfis, consulte Mudar de um usuário para um perfil do IAM (console) no Guia do usuário do IAM.

É possível criar credenciais temporárias manualmente usando a API AWS CLI ou AWS. Em seguida, você pode usar essas credenciais temporárias para acessar a AWS. A AWS recomenda que você gere credenciais temporárias dinamicamente em vez de usar chaves de acesso de longo prazo. Para obter mais informações, consulte Credenciais de segurança temporárias no IAM.

Sessões de acesso direto para o Amazon RDS

Compatível com sessões de acesso direto: sim.

O usuário ou perfil do IAM usado para executar ações na AWS é considerado uma entidade principal. Ao usar alguns serviços, você pode executar uma ação que inicia outra ação em um serviço diferente. O recurso FAS usa as permissões da entidade principal chamando um AWS service (Serviço da AWS), bem como o AWS service (Serviço da AWS) solicitante, para fazer solicitações para serviços subsequentes. As solicitações de FAS são feitas somente quando um serviço recebe uma solicitação que exige interações com outros Serviços da AWS ou com recursos para serem concluídas. Nesse caso, você precisa ter permissões para executar ambas as ações. Para obter detalhes da política ao fazer solicitações de FAS, consulte Sessões de acesso direto.

Perfis de serviço do Amazon RDS

Compatível com perfis de serviço: sim.

O perfil de serviço é um perfil do IAM que um serviço assume para executar ações em seu nome. Um administrador do IAM pode criar, modificar e excluir um perfil de serviço do IAM. Para obter mais informações, consulte Criar um perfil para delegar permissões a um AWS service (Serviço da AWS) no Guia do Usuário do IAM.

> **Warning**
>
> A alteração das permissões de uma função de serviço pode interromper a funcionalidade do Amazon RDS. Edite perfis de serviço somente quando o Amazon RDS fornecer orientação para isso.

Funções vinculadas a serviço do Amazon RDS

Compatível com perfis vinculados a serviços: sim.

Um perfil vinculado a serviço é um tipo de perfil de serviço vinculado a um AWS service (Serviço da AWS). O serviço pode presumir o perfil de executar uma ação em seu nome. Perfis vinculadas ao serviço aparecem em sua Conta da AWS e são de propriedade do serviço. Um administrador do IAM pode visualizar, mas não editar as permissões para perfis vinculados ao serviço.

Para obter detalhes sobre como usar funções vinculadas o serviço do Amazon RDS, consulte Usar funções vinculadas ao serviço do Amazon RDS.

Exemplos de políticas baseadas em identidade do Amazon RDS

Por padrão, os conjuntos de permissões e perfis não têm permissão para criar nem modificar recursos do Amazon RDS. Eles também não podem executar tarefas usando o AWS Management Console, a AWS CLI ou uma API da AWS. Um administrador deve criar políticas do IAM que concedam aos conjuntos de permissões e perfis permissão para executar operações de API específicas nos recursos especificados de que precisam. Depois, o administrador deve anexar essas políticas aos conjuntos de permissões e perfis que exigem essas permissões.

Para saber como criar uma política baseada em identidade do IAM utilizando esses exemplos de documentos de política JSON, consulte Criar políticas na guia JSON no Guia do usuário do IAM.

Tópicos

- Práticas recomendadas de política
- Usar o console do Amazon RDS
- Permissões necessárias para usar o console
- Permitir que os usuários visualizem suas próprias permissões
- Políticas de permissão para criar, modificar e excluir recursos no Amazon RDS
- Políticas de exemplo: usar chaves de condição
- Especificar condições: usar tags personalizadas
- Conceder permissão para marcar recursos do Amazon RDS durante a criação

Práticas recomendadas de política

As políticas baseadas em identidade determinam se alguém pode criar, acessar ou excluir recursos do Amazon RDS em sua conta. Essas ações podem incorrer em custos para sua Conta da AWS. Ao criar ou editar políticas baseadas em identidade, siga estas diretrizes e recomendações:

- Comece com as políticas gerenciadas pela AWS e avance para as permissões de privilégio mínimo: para começar a conceder permissões a seus usuários e workloads, use as políticas gerenciadas pela AWS, que concedem permissões para muitos casos de uso comuns. Elas estão disponíveis em seus Conta da AWS. Recomendamos que você reduza ainda mais as permissões definindo políticas gerenciadas pelo cliente da AWS que são específicas para seus casos de uso. Para obter mais informações, consulte Políticas gerenciadas pela AWS ou Políticas gerenciadas pela AWS para funções de trabalho no Guia do usuário do IAM.
- Aplique permissões de privilégio mínimo: ao definir permissões com as políticas do IAM, conceda apenas as permissões necessárias para executar uma tarefa. Você faz isso definindo as ações que podem ser executadas em recursos específicos sob condições específicas, também conhecidas como permissões de privilégio mínimo. Para obter mais informações sobre como usar o IAM para aplicar permissões, consulte Políticas e permissões no IAM no Guia do usuário do IAM.
- Use condições nas políticas do IAM para restringir ainda mais o acesso: você pode adicionar uma condição às políticas para limitar o acesso a ações e recursos. Por exemplo, você pode escrever uma condição de política para especificar que todas as solicitações devem ser enviadas usando SSL. Também pode usar condições para conceder acesso a ações de serviço, se elas forem usadas por meio de um AWS service (Serviço da AWS) específico, como o AWS CloudFormation. Para obter mais informações, consulte Elementos da política JSON do IAM: condição no Guia do usuário do IAM.

- Use o IAM Access Analyzer para validar suas políticas do IAM a fim de garantir permissões seguras e funcionais: o IAM Access Analyzer valida as políticas novas e existentes para que elas sigam a linguagem de política do IAM (JSON) e as práticas recomendadas do IAM. O IAM Access Analyzer oferece mais de cem verificações de política e recomendações práticas para ajudar a criar políticas seguras e funcionais. Para obter mais informações, consulte Validação de políticas do IAM Access Analyzer no Guia do Usuário do IAM.
- Exigir autenticação multifator (MFA): se houver um cenário que exija usuários do IAM ou um usuário raiz em sua Conta da AWS, ative a MFA para obter segurança adicional. Para exigir MFA quando as operações de API forem chamadas, adicione condições de MFA às suas políticas. Para obter mais informações, consulte Configuração de acesso à API protegido por MFA no Guia do Usuário do IAM.

Para obter mais informações sobre as práticas recomendadas do IAM, consulte Práticas recomendadas de segurança no IAM no Guia do usuário do IAM.

Usar o console do Amazon RDS

Para acessar o console do Amazon RDS, você deve ter um conjunto mínimo de permissões. Essas permissões devem permitir que você liste e visualize detalhes sobre os recursos do Amazon RDS em sua Conta da AWS. Caso crie uma política baseada em identidade mais restritiva que as permissões mínimas necessárias, o console não funcionará como pretendido para entidades (usuários ou perfis) com essa política.

Não é necessário conceder permissões mínimas do console para usuários que fazem chamadas somente à AWS CLI ou à API do AWS. Em vez disso, permita o acesso somente às ações que corresponderem a operação da API que você estiver tentando executar.

Para garantir que essas entidades ainda possam usar o console do Amazon RDS, anexe também a seguinte política gerenciada pela AWS às entidades.

```
AmazonRDSReadOnlyAccess
```

Para obter mais informações, consulte Adicionar permissões a um usuário no Guia do usuário do IAM.

Permissões necessárias para usar o console

Para um usuário trabalhar com o console, esse usuário deve ter um conjunto de permissões mínimo. Essas permissões permitem que o usuário descreva os recursos do Amazon RDS para a conta da

AWS e forneça outras informações relacionadas, inclusive informações de segurança e rede do Amazon EC2.

Se você criar uma política do IAM que seja mais restritiva que as permissões mínimas necessárias, o console não funcionará como pretendido para os usuários com essa política do IAM. Para garantir que esses usuários ainda consigam usar o console, associe também a política gerenciada `AmazonRDSReadOnlyAccess` ao usuário, conforme descrito em Gerenciamento do acesso usando políticas.

Não é necessário conceder permissões mínimas do console para usuários que fazem chamadas somente à AWS CLI ou à API do Amazon RDS.

A seguinte política concede acesso completo a todos os recursos do Amazon RDS para a conta raiz da AWS:

```
AmazonRDSFullAccess
```

Permitir que os usuários visualizem suas próprias permissões

Este exemplo mostra como criar uma política que permita que os usuários do IAM visualizem as políticas gerenciadas e em linha anexadas a sua identidade de usuário. Essa política inclui permissões para concluir essa ação no console ou de forma programática usando a AWS CLI ou a API da AWS.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
```

```
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

Políticas de permissão para criar, modificar e excluir recursos no Amazon RDS

As seções a seguir apresentam exemplos de políticas de permissão que concedem e restringem o acesso aos recursos:

Permitir que um usuário crie instâncias de Bancos de Dados em uma conta da AWS

Veja a seguir um exemplo de política que permite à conta com o ID `123456789012` criar instâncias de banco de dados para a sua conta da AWS. A política exige que o nome da nova instância de banco de dados comece com `test`. A nova instância de banco de dados também deve usar o mecanismo de banco de dados MySQL e a classe de instância de banco de dados `db.t2.micro`. Além disso, a nova instância de banco de dados deve usar um grupo de opções e um grupo de parâmetros de banco de dados que começa com `default` e deve usar o grupo de sub-redes `default`.

JSON

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "AllowCreateDBInstanceOnly",
         "Effect": "Allow",
         "Action": [
```

```
            "rds:CreateDBInstance"
         ],
         "Resource": [
            "arn:aws:rds:*:123456789012:db:test*",
            "arn:aws:rds:*:123456789012:og:default*",
            "arn:aws:rds:*:123456789012:pg:default*",
            "arn:aws:rds:*:123456789012:subgrp:default"
         ],
         "Condition": {
            "StringEquals": {
               "rds:DatabaseEngine": "mysql",
               "rds:DatabaseClass": "db.t2.micro"
            }
         }
      }
   ]
}
```

A política inclui uma única instrução que especifica as seguintes permissões para o usuário do :

- A política permite que a conta crie uma instância de banco de dados usando a operação de API CreateDBInstance (isso também se aplica ao comando create-db-instance da AWS CLI e ao AWS Management Console).
- O elemento `Resource` especifica que o usuário pode realizar ações em ou com recursos. Você especifica recursos usando um nome de recurso da Amazon (ARN). O ARN inclui o nome do serviço ao qual o recurso pertence (`rds`), a região da AWS (`*` indica qualquer região neste exemplo), o número de conta da AWS (`123456789012` é o número de conta neste exemplo) e o tipo de recurso. Para obter mais informações sobre como criar ARNs, consulte Nomes de recurso da Amazon (ARNs) no Amazon RDS.

  O elemento `Resource` neste exemplo especifica as restrições da política a seguir em recursos para o usuário:

  - O identificador de instância de banco de dados para a nova instância de banco de dados deve começar com `test` (por exemplo, `testCustomerData1`, `test-region2-data`).
  - O grupo de opções para a nova instância de banco de dados deve começar com `default`.
  - O grupo de parâmetros de banco de dados para a nova instância de banco de dados deve começar com `default`.

  - O grupo de sub-redes para a nova instância de banco de dados deve ser o grupo de sub-redes `default`.
- O elemento `Condition` especifica que o mecanismo de banco de dados deve ser MySQL e a classe da instância de banco de dados deve ser `db.t2.micro`. O elemento `Condition` especifica as condições quando uma política deve entrar em vigor. Você pode adicionar permissões ou restrições usando o elemento `Condition`. Para obter mais informações sobre como especificar condições, consulte Chaves de condição de políticas do Amazon RDS. Este exemplo especifica condições `rds:DatabaseEngine` e `rds:DatabaseClass`. Para obter informações sobre valores de condição válidos para o `rds:DatabaseEngine`, consulte a lista no parâmetro `Engine` em CreateDBInstance. Para obter informações sobre os valores de condição válidos para `rds:DatabaseClass`, consulte Mecanismos de banco de dados compatíveis para classes de instância de banco de dados .

A política não especifica o elemento `Principal` porque, em uma política baseada em identidade, não se especifica o principal que obtém as permissões. Quando você anexar uma política um usuário, o usuário será a entidade principal implícita. Quando você anexa uma política de permissão a um perfil do IAM, o principal identificado na política de confiança do perfil obtém as permissões.

Para obter uma lista de ações do Amazon RDS, consulte Ações definidas pelo Amazon RDS na Referência de autorização do serviço.

Permitir que um usuário execute qualquer ação de descrição em qualquer recurso do RDS

A seguinte política de permissões concede permissões a um usuário para executar todas as ações que começam com `Describe`. Essas ações mostram informações sobre um recurso do RDS, como uma instância de banco de dados. O caractere curinga (*) no elemento `Resource` indica que as ações são permitidas para todos os recursos do Amazon RDS que pertencem à conta.

JSON

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "AllowRDSDescribe",
         "Effect": "Allow",
         "Action": "rds:Describe*",
         "Resource": "*"
```

```
      }
   ]
}
```

Permitir que um usuário crie uma instância de banco de dados que usa o grupo de parâmetros de banco de dados e o grupo de sub-redes especificados.

A seguinte política de permissões concede permissões para permitir que um usuário crie apenas uma instância de banco de dados que deve usar o grupo de parâmetros de banco de dados `mydbpg` e o grupo de sub-rede de banco de dados `mydbsubnetgroup`.

JSON

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "VisualEditor0",
         "Effect": "Allow",
         "Action": "rds:CreateDBInstance",
         "Resource": [
            "arn:aws:rds:*:*:pg:mydbpg",
            "arn:aws:rds:*:*:subgrp:mydbsubnetgroup"
         ]
      }
   ]
}
```

Conceder permissão para ações em um recurso com uma tag específica com dois valores diferentes

Você pode usar condições em sua política baseada em identidade para controlar o acesso aos recursos do Amazon RDS com base em tags. A política a seguir concede permissão para realizar a operação de API `CreateDBSnapshot` em instâncias de banco de dados com a etiqueta `stage` definida como `development` ou `test`.

JSON

```
{
```

```
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"AllowAnySnapshotName",
         "Effect":"Allow",
         "Action":[
            "rds:CreateDBSnapshot"
         ],
         "Resource":"arn:aws:rds:*:123456789012:snapshot:*"
      },
      {
         "Sid":"AllowDevTestToCreateSnapshot",
         "Effect":"Allow",
         "Action":[
            "rds:CreateDBSnapshot"
         ],
         "Resource":"arn:aws:rds:*:123456789012:db:*",
         "Condition":{
            "StringEquals":{
                "rds:db-tag/stage":[
                  "development",
                  "test"
               ]
            }
         }
      }
   ]
}
```

A política a seguir concede permissão para realizar a operação de API `ModifyDBInstance` em instâncias de banco de dados com a etiqueta `stage` definida como `development` ou `test`.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"AllowChangingParameterOptionSecurityGroups",
         "Effect":"Allow",
         "Action":[
```

```
            "rds:ModifyDBInstance"
         ],
         "Resource": [
            "arn:aws:rds:*:123456789012:pg:*",
            "arn:aws:rds:*:123456789012:secgrp:*",
            "arn:aws:rds:*:123456789012:og:*"
         ]
      },
      {
         "Sid":"AllowDevTestToModifyInstance",
         "Effect":"Allow",
         "Action":[
            "rds:ModifyDBInstance"
         ],
         "Resource":"arn:aws:rds:*:123456789012:db:*",
         "Condition":{
            "StringEquals":{
                "rds:db-tag/stage":[
                  "development",
                  "test"
               ]
            }
         }
      }
   ]
}
```

Impedir que um usuário exclua uma instância de banco de dados

A seguinte política de permissões concede permissões para impedir que um usuário exclua uma instância de banco de dados específica. Por exemplo, você pode querer negar a capacidade de excluir suas instâncias de banco de dados de produção para qualquer usuário que não seja um administrador.

JSON

```
{
   "Version": "2012-10-17",
```

```
   "Statement": [
      {
         "Sid": "DenyDelete1",
         "Effect": "Deny",
         "Action": "rds:DeleteDBInstance",
         "Resource": "arn:aws:rds:us-west-2:123456789012:db:my-mysql-instance"
      }
   ]
}
```

Negar todo o acesso a um recurso

É possível negar acesso explicitamente a um recurso. As políticas de negação têm precedência sobre as políticas de permissão. A política a seguir nega explicitamente a um usuário a capacidade de gerenciar um recurso:

JSON

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect": "Deny",
         "Action": "rds:*",
         "Resource": "arn:aws:rds:us-east-1:123456789012:db:mydb"
      }
   ]
}
```

Políticas de exemplo: usar chaves de condição

Os seguintes exemplos mostram como você pode usar chaves de condição em políticas de permissões do IAM do Amazon RDS.

Exemplo 1: conceder permissão para criar uma instância de banco de dados que usa um mecanismo de banco de dados específico e não é MultiAZ

A seguinte política usa uma chave de condição do RDS e permite que um usuário crie apenas instâncias de banco de dados que usam o mecanismo de banco de dados MySQL e não use o

MultiAZ. O elemento `Condition` indica a exigência de que o mecanismo de banco de dados seja MySQL.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "AllowMySQLCreate",
         "Effect": "Allow",
         "Action": "rds:CreateDBInstance",
         "Resource": "*",
         "Condition": {
            "StringEquals": {
               "rds:DatabaseEngine": "mysql"
            },
            "Bool": {
               "rds:MultiAz": false
            }
         }
      }
   ]
}
```

Exemplo 2: negar explicitamente a permissão para criar instâncias de bancos de dados para determinadas classes de instância de banco de dados e criar instâncias de bancos de dados que usam IOPS provisionadas

A seguinte política nega explicitamente a permissão para criar instâncias de bancos de dados que usam as classes de instância de banco de dados `r3.8xlarge` e `m4.10xlarge`, que são as classes de instância de banco de dados maiores e mais caras. Essa política também impede que os usuários criem instâncias de banco de dados que usam IOPS provisionadas, que resultam em custos adicionais.

A negação explícita da permissão substitui quaisquer outras permissões concedidas. Isso garante que as identidades não obtenham acidentalmente permissão que você nunca deseja conceder.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "DenyLargeCreate",
```

```
         "Effect": "Deny",
         "Action": "rds:CreateDBInstance",
         "Resource": "*",
         "Condition": {
            "StringEquals": {
               "rds:DatabaseClass": [
                  "db.r3.8xlarge",
                  "db.m4.10xlarge"
               ]
            }
         }
      },
      {
         "Sid": "DenyPIOPSCreate",
         "Effect": "Deny",
         "Action": "rds:CreateDBInstance",
         "Resource": "*",
         "Condition": {
            "NumericNotEquals": {
               "rds:Piops": "0"
            }
         }
      }
   ]
}
```

Exemplo 3: limitar o conjunto de chaves de tag e valores que podem ser usados para identificar um recurso

A política a seguir usa uma chave de condição do RDS e permite a adição de uma tag com a chave `stage` a ser adicionada a um recurso com os valores `test`, `qa` e `production`.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Action": [
            "rds:AddTagsToResource",
            "rds:RemoveTagsFromResource"
         ],
         "Resource": "*",
         "Condition": {
```

```
            "streq": {
               "rds:req-tag/stage": [
                  "test",
                  "qa",
                  "production"
               ]
            }
         }
      }
   ]
}
```

Especificar condições: usar tags personalizadas

O Amazon RDS oferece suporte para especificar condições em uma política do IAM usando tags personalizadas.

Por exemplo, suponha que você adicione uma tag chamada `environment` às suas instâncias de banco de dados com valores como `beta`, `staging`, `production` e assim por diante. Se fizer isso, você poderá criar uma política que restrinja determinados usuários a instâncias de banco de dados com base no valor da tag `environment`.

> **Note**
>
> Os identificadores de tags personalizados diferenciam maiúsculas de minúsculas.

A tabela a seguir lista os identificadores de tags do RDS que você pode usar em um elemento `Condition`.

Identificador de tag do RDS	Aplica-se a
`db-tag`	Instâncias de bancos de dados, incluindo réplicas de leitura
`snapshot-tag`	DB snapshots
`ri-tag`	Instâncias de bancos de dados reservadas
`og-tag`	Grupos de opções de banco de dados

Identificador de tag do RDS	Aplica-se a
`pg-tag`	Grupos de parâmetros do banco de dados
`subgrp-tag`	Grupos de sub-redes de banco de dados
`es-tag`	Assinaturas de eventos
`cluster-tag`	clusters de banco de dados
`cluster-pg-tag`	Grupos de parâmetros de cluster de banco de dados
`cluster-snapshot-tag`	Snapshots de cluster de banco de dados

A sintaxe de uma condição de tag personalizada é a seguinte:

```
"Condition":{"StringEquals":{"rds:rds-tag-identifier/tag-name":
["value"]} }
```

Por exemplo, o seguinte elemento `Condition` se aplica a instâncias de banco de dados com uma tag `environment` e um valor de tag de `production`.

```
"Condition":{"StringEquals":{"rds:db-tag/environment": ["production"]} }
```

Para obter informações sobre como criar tags, consulte Marcar recursos do Amazon RDS.

Important

Se você gerenciar o acesso aos recursos do RDS usando tags, recomendamos proteger o acesso às tags para os seus recursos do RDS. Você pode gerenciar o acesso a tags, criando políticas para as ações `AddTagsToResource` e `RemoveTagsFromResource`. Por exemplo, a seguinte política nega aos usuários a capacidade de adicionar ou remover tags para todos os recursos. Você pode então criar políticas para permitir que usuários específicos adicionem ou removam tags.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
```

```
            "Sid":"DenyTagUpdates",
            "Effect":"Deny",
            "Action":[
               "rds:AddTagsToResource",
               "rds:RemoveTagsFromResource"
            ],
            "Resource":"*"
         }
      ]
}
```

Para obter uma lista de ações do Amazon RDS, consulte Ações definidas pelo Amazon RDS na Referência de autorização do serviço.

Políticas de exemplo: usar tags personalizadas

Os seguintes exemplos mostram como você pode usar tags personalizadas em políticas de permissões do IAM do Amazon RDS. Para obter mais informações sobre como adicionar tags a um recurso do Amazon RDS, consulte Nomes de recurso da Amazon (ARNs) no Amazon RDS.

> **Note**
>
> Todos os exemplos usam a região us-west-2 e contêm IDs de conta fictícios.

Exemplo 1: conceder permissão para ações em um recurso com uma tag específica com dois valores diferentes

A política a seguir concede permissão para realizar a operação de API `CreateDBSnapshot` em instâncias de banco de dados com a etiqueta `stage` definida como `development` ou `test`.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"AllowAnySnapshotName",
         "Effect":"Allow",
         "Action":[
            "rds:CreateDBSnapshot"
         ],
```

```
         "Resource":"arn:aws:rds:*:123456789012:snapshot:*"
      },
      {
         "Sid":"AllowDevTestToCreateSnapshot",
         "Effect":"Allow",
         "Action":[
            "rds:CreateDBSnapshot"
         ],
         "Resource":"arn:aws:rds:*:123456789012:db:*",
         "Condition":{
            "StringEquals":{
                "rds:db-tag/stage":[
                  "development",
                  "test"
               ]
            }
         }
      }
   ]
}
```

A política a seguir concede permissão para realizar a operação de API `ModifyDBInstance` em instâncias de banco de dados com a etiqueta `stage` definida como `development` ou `test`.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"AllowChangingParameterOptionSecurityGroups",
         "Effect":"Allow",
         "Action":[
            "rds:ModifyDBInstance"
         ],
         "Resource":" [
            "arn:aws:rds:*:123456789012:pg:*",
            "arn:aws:rds:*:123456789012:secgrp:*",
            "arn:aws:rds:*:123456789012:og:*"
         ]
      },
      {
         "Sid":"AllowDevTestToModifyInstance",
         "Effect":"Allow",
         "Action":[
```

```
            "rds:ModifyDBInstance"
         ],
         "Resource":"arn:aws:rds:*:123456789012:db:*",
         "Condition":{
            "StringEquals":{
                "rds:db-tag/stage":[
                  "development",
                  "test"
               ]
            }
         }
      }
   ]
}
```

Exemplo 2: negar explicitamente a permissão para criar uma instância de banco de dados que usa grupos de parâmetros de banco de dados especificados

A seguinte política nega explicitamente a permissão para criar uma instância de banco de dados que usa grupos de parâmetros de banco de dados com valores de tag específicos. Você poderá aplicar essa política se precisar que um grupo de parâmetros de banco de dados específico criado pelo cliente sempre seja usado ao criar instâncias de bancos de dados. As políticas que utilizam `Deny` são mais frequentemente usadas para restringir o acesso que foi concedido por uma política mais ampla.

A negação explícita da permissão substitui quaisquer outras permissões concedidas. Isso garante que as identidades não obtenham acidentalmente permissão que você nunca deseja conceder.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"DenyProductionCreate",
         "Effect":"Deny",
         "Action":"rds:CreateDBInstance",
         "Resource":"arn:aws:rds:*:123456789012:pg:*",
         "Condition":{
            "StringEquals":{
               "rds:pg-tag/usage":"prod"
            }
```

```
         }
      }
   ]
}
```

Exemplo 3: conceder permissão para executar ações em uma instância de banco de dados com um nome de instância prefixado com um nome de usuário

A seguinte política permite chamar qualquer API (exceto para `AddTagsToResource` ou `RemoveTagsFromResource`) em uma instância de banco de dados que tem um nome prefixado com o nome do usuário e que tem uma tag `stage` igual a `devo` ou sem tag `stage`.

A linha `Resource` na política identifica um recurso pelo seu Nome de Recurso Amazon (ARN). Para obter mais informações sobre como usar ARNs com recursos do Amazon RDS, consulte Nomes de recurso da Amazon (ARNs) no Amazon RDS.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"AllowFullDevAccessNoTags",
         "Effect":"Allow",
         "NotAction":[
            "rds:AddTagsToResource",
            "rds:RemoveTagsFromResource"
         ],
         "Resource":"arn:aws:rds:*:123456789012:db:${aws:username}*",
         "Condition":{
            "StringEqualsIfExists":{
               "rds:db-tag/stage":"devo"
            }
         }
      }
   ]
}
```

Conceder permissão para marcar recursos do Amazon RDS durante a criação

Algumas operações de API do RDS permitem atribuir tags ao criar recursos. É possível usar tags de recursos para implementar o controle baseado em atributo (ABAC). Para ter mais informações, consulte What is ABAC for AWS? e Controlar o acesso a recursos da AWS usando tags.

Para permitir que os usuários marquem recursos na criação, eles devem ter permissões para usar a ação que cria o recurso, como `rds:CreateDBInstance`. Se as tags forem especificadas no ato da criação, o RDS concederá uma autorização adicional na ação `rds:AddTagsToResource` para verificar se os usuários têm permissões para criar tags. Portanto, os usuários também precisam ter permissões para usar a ação `rds:AddTagsToResource`.

Na definição de política do IAM para a ação `rds:AddTagsToResource`, você pode usar a chave de condição `aws:RequestTag` para exigir tags em uma solicitação para marcar um recurso.

Por exemplo, a seguinte política permite que os usuários criem instâncias de banco de dados e apliquem tags durante a criação, mas somente com chaves de tag específicas (`environment` ou `project`):

```
{
   "Version": "2012-10-17",
   "Statement": [
       {
           "Effect": "Allow",
           "Action": [
               "rds:CreateDBInstance"
           ],
           "Resource": "*"
       },
       {
           "Effect": "Allow",
           "Action": [
               "rds:AddTagsToResource"
           ],
           "Resource": "*",
           "Condition": {
               "StringEquals": {
                   "aws:RequestTag/environment": ["production", "development"],
                   "aws:RequestTag/project": ["dataanalytics", "webapp"]
               },
               "ForAllValues:StringEquals": {
                   "aws:TagKeys": ["environment", "project"]
               }
           }
       }
   ]
}
```

Essa política nega qualquer solicitação de criação de instância de banco de dados que inclua tags diferentes das tags `environment` ou `project` ou que não especifique nenhuma dessas tags. Além disso, os usuários devem especificar valores para as tags que correspondam aos valores permitidos na política.

A seguinte política permite que os usuários criem clusters de banco de dados e apliquem qualquer tag durante a criação, exceto a tag `environment=prod`:

```
{
   "Version": "2012-10-17",
   "Statement": [
       {
           "Effect": "Allow",
           "Action": [
               "rds:CreateDBCluster"
           ],
           "Resource": "*"
       },
       {
           "Effect": "Allow",
           "Action": [
               "rds:AddTagsToResource"
           ],
           "Resource": "*",
           "Condition": {
               "StringNotEquals": {
                   "aws:RequestTag/environment": "prod"
               }
           }
       }
   ]
}
```

Ações de API do RDS que permitem atribuir tags durante a criação

As ações de API do RDS a seguir permitem atribuir tags durante a criação de um recurso. Para estas ações, você pode atribuir tags ao criar o recurso:

- `CreateBlueGreenDeployment`
- `CreateCustomDBEngineVersion`
- `CreateDBCluster`

- CreateDBClusterEndpoint
- CreateDBClusterParameterGroup
- CreateDBClusterSnapshot
- CreateDBInstance
- CreateDBInstanceReadReplica
- CreateDBParameterGroup
- CreateDBProxy
- CreateDBProxyEndpoint
- CreateDBSecurityGroup
- CreateDBShardGroup
- CreateDBSnapshot
- CreateDBSubnetGroup
- CreateEventSubscription
- CreateGlobalCluster
- CreateIntegration
- CreateOptionGroup
- CreateTenantDatabase
- CopyDBClusterParameterGroup
- CopyDBClusterSnapshot
- CopyDBParameterGroup
- CopyDBSnapshot
- CopyOptionGroup
- RestoreDBClusterFromS3
- RestoreDBClusterFromSnapshot
- RestoreDBClusterToPointInTime
- RestoreDBInstanceFromDBSnapshot
- RestoreDBInstanceFromS3
- RestoreDBInstanceToPointInTime
- PurchaseReservedDBInstancesOffering

Se você usar a AWS CLI ou a API para criar um recurso com tags, o parâmetro `Tags` será usado para aplicar tags aos recursos durante a criação.

Para essas ações de API, se a atribuição de tags falhar, o recurso não será criado e a solicitação falhará e apresentará um erro. Isso garante que os recursos sejam criados com tags ou não sejam criados, o que evita a criação de recursos sem as tags pretendidas.

Políticas gerenciadas pela AWS para o Amazon RDS

Para adicionar permissões a conjuntos de permissões e perfis, é mais fácil usar políticas gerenciadas pela AWS do que elaborar políticas por conta própria. É necessário tempo e experiência para criar políticas gerenciadas pelo cliente do IAM que fornecem à sua equipe apenas as permissões de que precisam. Para começar rapidamente, você pode usar nossas políticas gerenciadas pela AWS. Essas políticas abrangem casos de uso comuns e estão disponíveis na sua Conta da AWS. Para ter mais informações sobre as políticas gerenciadas da AWS, consulte Políticas gerenciadas da AWS no Guia do usuário do IAM.

Os Serviços da AWS mantêm e atualizam políticas gerenciadas pela AWS. Não é possível alterar as permissões em políticas gerenciadas pela AWS. Os serviços ocasionalmente acrescentam permissões adicionais a uma política gerenciada pela AWS para oferecer suporte a novos recursos. Esse tipo de atualização afeta todas as identidades (conjuntos de permissões e perfis) às quais a política está anexada. É mais provável que os serviços atualizem uma política gerenciada pela AWS quando um novo recurso for iniciado ou novas operações se tornarem disponíveis. Os serviços não removem permissões de uma política gerenciada pela AWS. Portanto, as atualizações de políticas não suspendem suas permissões atuais.

Além disso, a AWS oferece suporte a políticas gerenciadas para funções de trabalho que abrangem vários serviços. Por exemplo, a política `ReadOnlyAccess` gerenciada pela AWS concede acesso somente leitura a todos os recursos e Serviços da AWS. Quando um serviço executa um novo recurso, a AWS adiciona permissões somente leitura para novas operações e recursos. Para obter uma lista e descrições das políticas de funções de trabalho, consulte Políticas gerenciadas pela AWS para funções de trabalho no Guia do usuário do IAM.

Tópicos

- Política gerenciada pela AWS: AmazonRDSReadOnlyAccess
- Política gerenciada pela AWS: AmazonRDSFullAccess
- Política gerenciada pela AWS: AmazonRDSDataFullAccess
- Política gerenciada pela AWS: AmazonRDSEnhancedMonitoringRole
- Política gerenciada pela AWS: AmazonRDSPerformanceInsightsReadOnly
- Política gerenciada pela AWS: AmazonRDSPerformanceInsightsFullAccess
- Política gerenciada pela AWS: AmazonRDSDirectoryServiceAccess
- Política gerenciada pela AWS: AmazonRDSServiceRolePolicy
- Política gerenciada pela AWS: AmazonRDSCustomServiceRolePolicy

- Política gerenciada da AWS: AmazonRDSCustomInstanceProfileRolePolicy
- Política gerenciada da AWS: AmazonRDSPreviewServiceRolePolicy
- Política gerenciada da AWS: AmazonRDSBetaServiceRolePolicy

Política gerenciada pela AWS: AmazonRDSReadOnlyAccess

Essa política permite o acesso somente leitura ao Amazon RDS por meio do AWS Management Console.

Detalhes de permissão

Esta política inclui as seguintes permissões:

- `rds`: permite que as entidades principais descrevam os recursos do Amazon RDS e listem as etiquetas dos respectivos recursos.
- `cloudwatch`: permite que as entidades principais obtenham estatísticas de métricas do Amazon CloudWatch.
- `ec2`: permite que as entidades principais descrevam zonas de disponibilidade e recursos de rede.
- `logs`: permite que as entidades principais descrevam fluxos de log do CloudWatch Logs de grupos de logs e obtenham eventos de log do CloudWatch Logs.
- `devops-guru`: permite que as entidades principais descrevam os recursos que têm cobertura do Amazon DevOps Guru, que é especificada pelos nomes das pilhas ou pelas tags de recursos do CloudFormation.

Para ter mais informações sobre essa política, incluindo o documento de política JSON, consulte AmazonRDSReadOnlyAccess no Guia de referência de políticas gerenciadas pela AWS.

Política gerenciada pela AWS: AmazonRDSFullAccess

Essa política concede acesso total ao Amazon RDS por meio do AWS Management Console.

Detalhes de permissão

Esta política inclui as seguintes permissões:

- `rds`: concede às entidades principais acesso total ao Amazon RDS.
- `application-autoscaling`: permite que as entidades principais descrevam e gerenciem destinos e políticas de escalabilidade do Application Auto Scaling.

- `cloudwatch`: permite que as entidades principais obtenham estáticas de métricas do CloudWatch e gerenciem os respectivos alarmes.
- `ec2`: permite que as entidades principais descrevam zonas de disponibilidade e recursos de rede.
- `logs`: permite que as entidades principais descrevam fluxos de log do CloudWatch Logs de grupos de logs e obtenham eventos de log do CloudWatch Logs.
- `outposts`: permite que as entidades principais obtenham tipos de instância AWS Outposts.
- `pi`: permite que as entidades principais obtenham métricas do Performance Insights.
- `sns`: permite que as entidades principais acessem assinaturas e tópicos do Amazon Simple Notification Service (Amazon SNS) e publiquem mensagens do Amazon SNS.
- `devops-guru`: permite que as entidades principais descrevam os recursos que têm cobertura do Amazon DevOps Guru, que é especificada pelos nomes das pilhas ou pelas tags de recursos do CloudFormation.

Para ter mais informações sobre essa política, incluindo o documento de política JSON, consulte AmazonRDSFullAccess no Guia de referência de políticas gerenciadas pela AWS.

Política gerenciada pela AWS: AmazonRDSDataFullAccess

Essa política permite acesso total ao uso da API DATA e ao editor de consultas em clusters do Aurora Serverless em uma Conta da AWS específica. Essa política permite que a Conta da AWS obtenha o valor de um segredo do AWS Secrets Manager.

É possível anexar a política `AmazonRDSDataFullAccess` a suas identidades do IAM.

Detalhes de permissão

Esta política inclui as seguintes permissões:

- `dbqms`: permite que as entidades principais acessem, criem, excluam, descrevam e atualizem consultas. O Database Query Metadata Service (`dbqms`) é um serviço somente interno. Ele fornece suas consultas recentes e salvas para o editor de consultas no AWS Management Console para vários Serviços da AWS, inclusive o Amazon RDS.
- `rds-data`: permite que as entidades principais executem instruções SQL em bancos de dados do Aurora Serverless.
- `secretsmanager`: permite que as entidades principais obtenham o valor de um segredo do AWS Secrets Manager.

Para ter mais informações sobre essa política, incluindo o documento de política JSON, consulte AmazonRDSDataFullAccess no Guia de referência de políticas gerenciadas pela AWS.

Política gerenciada pela AWS: AmazonRDSEnhancedMonitoringRole

Essa política fornece acesso ao Amazon CloudWatch Logs for Amazon RDS Enhanced Monitoring.

Detalhes de permissão

Esta política inclui as seguintes permissões:

- `logs`: permite que as entidades principais criem grupos de logs e políticas de retenção do CloudWatch Logs, bem como criem e descrevam fluxos de log do CloudWatch Logs de grupos de logs. Ela também permite que as entidades principais insiram e obtenham eventos de log do CloudWatch Logs.

Para ter mais informações sobre essa política, incluindo o documento de política JSON, consulte AmazonRDSEnhancedMonitoringRole no Guia de referência de políticas gerenciadas pela AWS.

Política gerenciada pela AWS: AmazonRDSPerformanceInsightsReadOnly

Essa política fornece acesso somente leitura ao Insights de Performance do Amazon RDS para instâncias de banco de dados do Amazon RDS e de clusters de banco de dados do Amazon Aurora.

Essa política agora inclui `Sid` (ID da instrução) como identificador para a declaração de política.

Detalhes de permissão

Esta política inclui as seguintes permissões:

- `rds`: permite que as entidades principais descrevam instâncias de banco de dados do Amazon RDS e de clusters de banco de dados Amazon Aurora
- `pi`: permite que as entidades principais façam chamadas para a API do Insights de Performance do Amazon RDS e acessem as métricas do Insights de Performance.

Para ter mais informações sobre essa política, incluindo o documento de política JSON, consulte AmazonRDSPerformanceInsightsReadOnly no Guia de referência de políticas gerenciadas pela AWS.

Política gerenciada pela AWS: AmazonRDSPerformanceInsightsFullAccess

Essa política fornece acesso total ao Insights de Performance do Amazon RDS para instâncias de banco de dados do Amazon RDS e clusters de banco de dados do Amazon Aurora.

Essa política agora inclui `Sid` (ID da instrução) como identificador para a declaração de política.

Detalhes de permissão

Esta política inclui as seguintes permissões:

- `rds`: permite que as entidades principais descrevam instâncias de banco de dados do Amazon RDS e de clusters de banco de dados Amazon Aurora
- `pi`: permite que as entidades principais façam chamadas para a API do Insights de Performance do Amazon RDS e criem, visualizem e excluam relatórios de análise de performance.
- `cloudwatch`: permite que as entidades principais listem métricas do Amazon CloudWatch e obtenham dados de métricas e estatística.

Para receber mais informações sobre essa política, incluindo o documento de política JSON, consulte AmazonRDSPerformanceInsightsFullAccess no Guia de referência de políticas gerenciadas pela AWS.

Política gerenciada pela AWS: AmazonRDSDirectoryServiceAccess

Essa política permite que o Amazon RDS faça chamadas ao AWS Directory Service.

Detalhes de permissão

Esta política inclui a seguinte permissão:

- `ds`: permite que as entidades principais descrevam diretórios do AWS Directory Service e controlem a autorização para diretórios do AWS Directory Service.

Para ter mais informações sobre essa política, incluindo o documento de política JSON, consulte AmazonRDSDirectoryServiceAccess no Guia de referência de políticas gerenciadas pela AWS.

Política gerenciada pela AWS: AmazonRDSServiceRolePolicy

Não é possível anexar a política `AmazonRDSServiceRolePolicy` às suas entidades do IAM. Essa política é anexada a uma função vinculada ao serviço que permite ao Amazon RDS realizar ações

em seu nome. Para ter mais informações, consulte Permissões de função vinculada ao serviço do Amazon RDS.

Política gerenciada pela AWS: AmazonRDSCustomServiceRolePolicy

Não é possível anexar a política `AmazonRDSCustomServiceRolePolicy` às suas entidades do IAM. Essa política é anexada a um perfil vinculado ao serviço que permite ao Amazon RDS chamar serviços da AWS em nome de seus recursos de banco de dados do RDS.

Esta política inclui as seguintes permissões:

- `ec2`: permite que o RDS Custom execute operações de backup na instância de banco de dados que fornece recursos de restauração para um ponto no tempo.
- `secretsmanager`: permite que o RDS Custom gerencie segredos específicos da instância de banco de dados criados pelo RDS Custom.
- `cloudwatch`: permite que o RDS Custom faça upload de métricas e de logs da instância de banco de dados no CloudWatch por meio do agente do CloudWatch.
- `events`, `sqs`: permite que o RDS Custom envie e receba informações de status sobre a instância de banco de dados.

Para ter mais informações sobre essa política, incluindo o documento de política JSON, consulte AmazonRDSCustomServiceRolePolicy no Guia de referência de políticas gerenciadas pela AWS.

Política gerenciada da AWS: AmazonRDSCustomInstanceProfileRolePolicy

Não é possível anexar `AmazonRDSCustomInstanceProfileRolePolicy` às suas entidades do IAM. Ele só deve ser anexado a uma função de perfil de instância usada para conceder permissões à instância de banco de dados do Amazon RDS Custom para realizar várias ações de automação e tarefas de gerenciamento de banco de dados. Passe o perfil de instância como o parâmetro `custom-iam-instance-profile` durante a criação da instância personalizada do RDS e o RDS Custom associará esse perfil à instância de banco de dados.

Detalhes de permissão

Esta política inclui as seguintes permissões:

- `ssm`, `ssmmessages`, `ec2messages`: permite que o RDS Custom se comunique, execute automação e mantenha agentes na instância de banco de dados por meio do Systems Manager.

- `ec2`, `s3`: permite que o RDS Custom execute operações de backup na instância de banco de dados que fornece recursos de restauração pontuais.
- `secretsmanager`: permite que o RDS Custom gerencie segredos específicos da instância de banco de dados criados pelo RDS Custom.
- `cloudwatch`, `logs`: permite que o RDS Custom faça upload de métricas e logs da instância de banco de dados no CloudWatch por meio do agente do CloudWatch.
- `events`, `sqs`: permite que o RDS Custom envie e receba informações de status sobre a instância de banco de dados.
- `kms`: permite que o RDS Custom use uma chave do KMS específica da instância para realizar a criptografia de segredos e objetos do S3 gerenciados pelo RDS Custom.

Para receber mais informações sobre essa política, incluindo o documento de política JSON, consulte AmazonRDSCustomInstanceProfileRolePolicy no Guia de referência de políticas gerenciadas da AWS.

Política gerenciada da AWS: AmazonRDSPreviewServiceRolePolicy

Não é possível anexar `AmazonRDSPreviewServiceRolePolicy` às suas entidades do IAM. Essa política é anexada a um perfil vinculado ao serviço que permite ao Amazon RDS chamar serviços da AWS em nome de seus recursos de banco de dados do RDS. Para ter mais informações, consulte Perfil vinculado ao serviço da Pré-visualização do Amazon RDS.

Detalhes de permissão

Esta política inclui as seguintes permissões:

- `ec2`: permite que as entidades principais descrevam Zonas de Disponibilidade e recursos de rede.
- `secretsmanager`: permite que as entidades principais obtenham o valor de um segredo do AWS Secrets Manager.
- `cloudwatch`, `logs`: permite que o Amazon RDS faça upload de métricas e logs da instância de banco de dados no CloudWatch por meio do agente do CloudWatch.

Para ter mais informações sobre essa política, incluindo o documento de política JSON, consulte AmazonRDSDataFullAccess no Guia de referência de políticas gerenciadas pela AWS.

Política gerenciada da AWS: AmazonRDSBetaServiceRolePolicy

Não é possível anexar `AmazonRDSBetaServiceRolePolicy` às suas entidades do IAM. Essa política é anexada a um perfil vinculado ao serviço que permite ao Amazon RDS chamar serviços da AWS em nome de seus recursos de banco de dados do RDS. Para ter mais informações, consulte Permissões de perfis vinculados ao serviço para o Amazon RDS Beta.

Detalhes de permissão

Esta política inclui as seguintes permissões:

- `ec2`: permite que o Amazon RDS execute operações de backup na instância de banco de dados que fornece recursos de restauração para um ponto no tempo.
- `secretsmanager`: permite ao Amazon RDS gerenciar segredos específicos da instância de banco de dados criados pelo Amazon RDS.
- `cloudwatch`, `logs`: permite que o Amazon RDS faça upload de métricas e logs da instância de banco de dados no CloudWatch por meio do agente do CloudWatch.

Para ter mais informações sobre essa política, incluindo o documento de política JSON, consulte AmazonRDSBetaServiceRolePolicy no Guia de referência de políticas gerenciadas da AWS.

Atualizações do Amazon RDS para políticas gerenciadas pela AWS

Visualize detalhes sobre atualizações em políticas gerenciadas pela AWS para o Amazon RDS desde que esse serviço começou a monitorar essas alterações. Para receber alertas automáticos sobre mudanças nesta página, assine o feed RSS na página Histórico de documentos do Amazon RDS.

Alteração	Descrição	Data
Política gerenciada da AWS: AmazonRDSPreviewServiceRolePolicy: atualizar para uma política existente.	O Amazon RDS removeu a permissão `sns:Publish` da `AmazonRDSPreviewServiceRolePolicy` do perfil vinculado ao serviço `AWSServiceRoleForRDSPreview`. Para ter mais informações, consulte Política gerenciada da AWS: AmazonRDSPreviewServiceRolePolicy.	7 de agosto de 2024
Política gerenciada da AWS: AmazonRDSBetaServiceRolePolicy: atualizar para uma política existente.	O Amazon RDS removeu a permissão `sns:Publish` da `AmazonRDSBetaServiceRolePolicy` do perfil vinculado ao serviço `AWSServiceRoleForRDSBeta`. Para ter mais informações, consulte Política gerenciada da AWS: AmazonRDSBetaServiceRolePolicy.	7 de agosto de 2024
Permissões de funções vinculadas a serviços para o Amazon RDS Custom:	O Amazon RDS adicionou novas permissões à `AmazonRDSCustomServiceRolePolicy` da	18 de julho de 2024

Alteração	Descrição	Data
atualizar para uma política existente.	função vinculada ao serviço `AWSServiceRoleForRDSCustom` . As permissões possibilitam que o RDS Custom se comunique com os serviços do Amazon RDS em outra Região da AWS e copie imagens do EC2. Para ter mais informações, consulte Permissões de funções vinculadas a serviços para o Amazon RDS Custom.	
Política gerenciada pela AWS: AmazonRDSServiceRolePolicy: atualizar para uma política existente.	O Amazon RDS removeu a permissão `sns:Publish` da `AmazonRDSServiceRolePolicy` do perfil vinculado ao serviço `AWSServiceRoleForRDS` . Para ter mais informações, consulte Política gerenciada pela AWS: AmazonRDSServiceRolePolicy.	2 de julho de 2024

Alteração	Descrição	Data
Permissões de funções vinculadas a serviços para o Amazon RDS Custom: atualizar para uma política existente.	O Amazon RDS adicionou novas permissões à `AmazonRDSCustomServiceRolePolicy` da função vinculada ao serviço `AWSServiceRoleForRDSCustom` . Essa nova permissão permite que o RDS Custom associe um perfil de serviço como um perfil de instância a uma instância do RDS Custom. Para ter mais informações, consulte Permissões de funções vinculadas a serviços para o Amazon RDS Custom.	19 de abril de 2024

Alteração	Descrição	Data
Políticas gerenciadas pela AWS para o Amazon RDS: atualizar para uma política existente.	O Amazon RDS adicionou uma nova permissão ao `AmazonRDSCustomServiceRolePolicy` do perfil vinculado ao serviço `AWSServiceRoleForRDSCustom` para permitir que o RDS Custom para SQL Server modifique o tipo de instância de host do banco de dados subjacente. O RDS também adicionou a permissão `ec2:DescribeInstanceTypes` para receber informações sobre o tipo de instância para o host do banco de dados. Para ter mais informações, consulte Políticas gerenciadas pela AWS para o Amazon RDS.	8 de abril de 2024
Políticas gerenciadas pela AWS para o Amazon RDS – Nova política	O Amazon RDS adicionou uma nova política gerenciada chamada `AmazonRDSCustomInstanceProfileRolePolicy` para permitir que o RDS Custom execute ações de automação e tarefas de gerenciamento de banco de dados por meio de um perfil de instância do EC2. Para ter mais informações, consulte Políticas gerenciadas pela AWS para o Amazon RDS.	27 de fevereiro de 2024

Alteração	Descrição	Data
Permissões de função vinculada ao serviço do Amazon RDS: atualizar para uma política existente	O Amazon RDS adicionou novos IDs de declarações à `AmazonRDSServiceRolePolicy` do perfil vinculado ao serviço `AWSServiceRoleForRDS` . Para ter mais informações, consulte Permissões de função vinculada ao serviço do Amazon RDS.	19 de janeiro de 2024
Políticas gerenciadas pela AWS para o Amazon RDS: atualizações em políticas existentes.	As políticas `AmazonRDSPerformanceInsightsFullAccess` gerenciadas `AmazonRDSPerformanceInsightsReadOnly` e as políticas agora incluem `Sid` (ID da instrução) como identificador na declaração de política. Para obter mais informações, consulte Política gerenciada pela AWS: AmazonRDSPerformanceInsightsReadOnly e Política gerenciada pela AWS: AmazonRDSPerformanceInsightsFullAccess.	23 de outubro de 2023

Alteração	Descrição	Data
Permissões de função vinculada ao serviço do Amazon RDS: atualizar para uma política existente	O Amazon RDS adicionou novas permissões à `AmazonRDSCustomServiceRolePolicy` da função vinculada ao serviço `AWSServiceRoleForRDSCustom` . Essas novas permissões possibilitam que o RDS Custom para Oracle crie, modifique e exclua regras gerenciadas do EventBridge. Para ter mais informações, consulte Permissões de funções vinculadas a serviços para o Amazon RDS Custom.	20 de setembro de 2023
Políticas gerenciadas pela AWS para o Amazon RDS: atualizar para uma política existente.	O Amazon RDS adicionou novas permissões à política gerenciada `AmazonRDSFullAccess` . As permissões autorizam que você gere, visualize e exclua o relatório de análise de performance por um período. Para receber mais informações sobre como configurar políticas de acesso para o Insights de Performance, consulte Configurar políticas de acesso para o Performance Insights.	17 de agosto de 2023

Alteração	Descrição	Data
Políticas gerenciadas pela AWS para o Amazon RDS: nova política e atualização da política existente.	O Amazon RDS adicionou novas permissões à política gerenciada `AmazonRDSPerformanceInsightsReadOnly` e uma nova política gerenciada chamada `AmazonRDSPerformanceInsightsFullAccess`. Essas permissões autorizam que você analise o Insights de Performance por um período, visualize os resultados da análise com as recomendações e exclua os relatórios. Para receber mais informações sobre como configurar políticas de acesso para o Insights de Performance, consulte Configurar políticas de acesso para o Performance Insights.	16 de agosto de 2023

Alteração	Descrição	Data
Permissões de função vinculada ao serviço do Amazon RDS: atualizar para uma política existente	O Amazon RDS adicionou novas permissões à `AmazonRDSCustomServiceRolePolicy` da função vinculada ao serviço `AWSServiceRoleForRDSCustom`. Essas novas permissões possibilitam que o RDS Custom para Oracle utilize snapshots de banco de dados. Para ter mais informações, consulte Permissões de funções vinculadas a serviços para o Amazon RDS Custom.	23 de junho de 2023
Permissões de função vinculada ao serviço do Amazon RDS: atualizar para uma política existente	O Amazon RDS adicionou novas permissões à `AmazonRDSCustomServiceRolePolicy` da função vinculada ao serviço `AWSServiceRoleForRDSCustom`. Essas novas permissões possibilitam que o RDS Custom para Oracle utilize snapshots de banco de dados. Para ter mais informações, consulte Permissões de funções vinculadas a serviços para o Amazon RDS Custom.	23 de junho de 2023

Alteração	Descrição	Data
Permissões de função vinculada ao serviço do Amazon RDS: atualizar para uma política existente	O Amazon RDS adicionou novas permissões à `AmazonRDSCustomServiceRolePolicy` da função vinculada ao serviço `AWSServiceRoleForRDSCustom` . Essas novas permissões possibilitam que o RDS Custom crie interfaces de rede. Para ter mais informações, consulte Permissões de funções vinculadas a serviços para o Amazon RDS Custom.	30 de maio de 2023
Permissões de função vinculada ao serviço do Amazon RDS: atualizar para uma política existente	O Amazon RDS adicionou novas permissões à `AmazonRDSCustomServiceRolePolicy` da função vinculada ao serviço `AWSServiceRoleForRDSCustom` . Essas novas permissões permitem que o RDS Custom chame o Amazon EBS para conferir a cota de armazenamento. Para ter mais informações, consulte Permissões de funções vinculadas a serviços para o Amazon RDS Custom.	18 de abril de 2023

Alteração	Descrição	Data
Permissões de função vinculada ao serviço do Amazon RDS: atualizar para uma política existente	O Amazon RDS Custom adicionou novas permissões a `AmazonRDSCustomServiceRolePolicy` da função vinculada ao serviço `AWSServiceRoleForRDSCustom` para integração com o Amazon SQS. O RDS Custom requer integração com o Amazon SQS para criar e gerenciar filas do SQS na conta do cliente. Os nomes das filas do SQS seguem o formato `do-not-delete-rds-custom-[identifier]` e são marcados com `Amazon RDS Custom`. A permissão para `ec2:CreateSnapshot` foi adicionada para permitir que o RDS Custom crie backups para volumes anexados à instância. Para ter mais informações, consulte Permissões de funções vinculadas a serviços para o Amazon RDS Custom.	6 de abril de 2023

Alteração	Descrição	Data
Políticas gerenciadas pela AWS para o Amazon RDS: atualizar para uma política existente	O Amazon RDS adicionou um novo namespace do Amazon CloudWatch `ListMetrics` a `AmazonRDSFullAccess` e a `AmazonRDSReadOnlyAccess` . Esse namespace é necessário para que o Amazon RDS liste métricas específicas de uso de recursos. Para ter mais informações, consulte Visão geral do gerenciamento de permissões de acesso aos recursos do CloudWatch no Guia do usuário do Amazon CloudWatch.	4 de abril de 2023

Alteração	Descrição	Data
Políticas gerenciadas pela AWS para o Amazon RDS: atualizar para uma política existente	O Amazon RDS adicionou uma nova permissão a `AmazonRDSFullAccess` e às políticas gerenciadas `AmazonRDSReadOnlyAccess` para possibilitar a exibição das descobertas do Amazon DevOps Guru no console do RDS. Essa permissão é necessária para possibilitar a exibição das descobertas do DevOps Guru. Para ter mais informações, consulte Atualizações do Amazon RDS para políticas gerenciadas pela AWS.	30 de março de 2023

Alteração	Descrição	Data
Permissões de função vinculada ao serviço do Amazon RDS: atualizar para uma política existente	O Amazon RDS adicionou novas permissões à `AmazonRDSServiceRolePolicy` da função vinculada ao serviço `AWSServiceRoleForRDS` para integração com o AWS Secrets Manager. O RDS requer integração com o Secrets Manager para gerenciar senhas do usuário principal no Secrets Manager. O segredo usa uma convenção de nomenclatura reservada e restringe as atualizações do cliente. Para ter mais informações, consulte Gerenciamento de senhas com Amazon RDS e AWS Secrets Manager.	22 de dezembro de 2022

Alteração	Descrição	Data
Permissões de função vinculada ao serviço do Amazon RDS: atualizar para uma política existente	O Amazon RDS adicionou novas permissões à `AmazonRDSCustomServiceRolePolicy` da função vinculada ao serviço `AWSServiceRoleForRDSCustom` . O RDS Custom é compatível com clusters de banco de dados. Essas novas permissões na política permitem que o RDS Custom chame Serviços da AWS em nome de seus clusters de banco de dados. Para ter mais informações, consulte Permissões de funções vinculadas a serviços para o Amazon RDS Custom.	9 de novembro de 2022

Alteração	Descrição	Data
Permissões de função vinculada ao serviço do Amazon RDS: atualizar para uma política existente	O Amazon RDS adicionou novas permissões ao perfil vinculado ao serviço `AWSServiceRoleForRDS` para integração com o AWS Secrets Manager. A integração com o Secrets Manager é necessária para que o SQL Server Reporting Services (SSRS) Email funcione no RDS. O SSRS Email cria um segredo em nome do cliente. O segredo usa uma convenção de nomenclatura reservada e restringe as atualizações do cliente. Para ter mais informações, consulte Como usar o SSRS Email para enviar relatórios.	26 de agosto de 2022

Alteração	Descrição	Data
Permissões de função vinculada ao serviço do Amazon RDS: atualizar para uma política existente	O Amazon RDS adicionou um novo namespace do Amazon CloudWatch ao `AmazonRDSPreviewServiceRolePolicy` para `PutMetricData`. Esse namespace é necessário para que o Amazon RDS publique métricas de uso de recursos. Para ter mais informações, consulte Usar chaves de condição para limitar o acesso a namespaces do CloudWatch no Guia do usuário do Amazon CloudWatch.	7 de junho de 2022
Permissões de função vinculada ao serviço do Amazon RDS: atualizar para uma política existente	O Amazon RDS adicionou um novo namespace do Amazon CloudWatch ao `AmazonRDSBetaServiceRolePolicy` para `PutMetricData`. Esse namespace é necessário para que o Amazon RDS publique métricas de uso de recursos. Para ter mais informações, consulte Usar chaves de condição para limitar o acesso a namespaces do CloudWatch no Guia do usuário do Amazon CloudWatch.	7 de junho de 2022

Alteração	Descrição	Data
Permissões de função vinculada ao serviço do Amazon RDS: atualizar para uma política existente	O Amazon RDS adicionou um novo namespace do Amazon CloudWatch ao `AWSServiceRoleForRDS` para `PutMetricData` . Esse namespace é necessário para que o Amazon RDS publique métricas de uso de recursos. Para ter mais informações, consulte Usar chaves de condição para limitar o acesso a namespaces do CloudWatch no Guia do usuário do Amazon CloudWatch.	22 de abril de 2022

Alteração	Descrição	Data
Permissões de função vinculada ao serviço do Amazon RDS: atualizar para uma política existente	O Amazon RDS adicionou novas permissões à função `AWSServiceRoleForRDS` vinculada a serviços para gerenciar permissões para grupos de IP de propriedade do cliente e tabelas de rotas de gateway local (LGW-RTBs). Essas permissões são necessárias para que o RDS on Outposts execute a replicação multi-AZ na rede local do Outposts. Para ter mais informaçõ es, consulte Trabalhar com implantações multi-AZ para o Amazon RDS no AWS Outposts.	19 de abril de 2022

Alteração	Descrição	Data
Políticas baseadas em identidade: atualizar para uma política existente	O Amazon RDS adicionou uma nova permissão à política gerenciada `AmazonRDSFullAccess` para descrever permissões em LGW-RTBs. Essa permissão é necessária para que o RDS on Outposts execute a replicação multi-AZ na rede local do Outposts. Para ter mais informações, consulte Trabalhar com implantações multi-AZ para o Amazon RDS no AWS Outposts.	19 de abril de 2022
Políticas gerenciadas pela AWS para o Amazon RDS – Nova política	O Amazon RDS adicionou uma nova política gerenciada denominada `AmazonRDSPerformanceInsightsReadOnly`, para permitir que o Amazon RDS chame serviços da AWS em nome de suas instâncias de banco de dados. Para receber mais informações sobre como configurar políticas de acesso para o Insights de Performance, consulte Configurar políticas de acesso para o Performance Insights.	10 de março de 2022

Alteração	Descrição	Data
Permissões de função vinculada ao serviço do Amazon RDS: atualizar para uma política existente	O Amazon RDS adicionou novos namespaces do Amazon CloudWatch ao `AWSServiceRoleForRDS` para `PutMetricData` . Esses namespaces são necessários para que o Amazon DocumentDB (compatível com MongoDB) e o Amazon Neptune publiquem métricas do CloudWatch. Para ter mais informações, consulte Usar chaves de condição para limitar o acesso a namespaces do CloudWatch no Guia do usuário do Amazon CloudWatch.	4 de março de 2022
Permissões de funções vinculadas a serviços para o Amazon RDS Custom – Nova política	O Amazon RDS adicionou uma nova função vinculada ao serviço chamada `AWSServiceRoleForRDSCustom` para permitir que o RDS Custom chame Serviços da AWS em nome das instâncias de banco de dados.	26 de outubro de 2021
O Amazon RDS passou a monitorar alterações	O Amazon RDS passou a monitorar alterações nas políticas gerenciadas pela AWS.	26 de outubro de 2021

Prevenção do problema do substituto confuso entre serviços

O problema “confused deputy” é um problema de segurança em que uma entidade que não tem permissão para executar uma ação pode coagir uma entidade mais privilegiada a executá-la. Na AWS, a personificação entre serviços pode resultar no problema do ‘confused deputy’.

A personificação entre serviços pode ocorrer quando um serviço (o serviço de chamada) chama outro serviço (o serviço chamado). O serviço de chamada pode ser manipulado de modo a usar suas permissões para atuar nos recursos de outro cliente de uma forma na qual ele não deveria ter permissão para acessar. Para evitar isso, a AWS fornece ferramentas que ajudam você a proteger seus dados para todos os serviços com entidades principais de serviço que receberam acesso aos recursos em sua conta. Para obter mais informações, consulte O problema confused deputy no Guia do usuário IAM.

Recomendamos o uso das chaves de contexto de condição global `aws:SourceArn` e `aws:SourceAccount` em políticas de recursos para limitar as permissões que o Amazon RDS concede a outro serviço para um recurso específico.

Em alguns casos, o valor `aws:SourceArn` não contém o ID da conta, por exemplo, quando você usa o nome do recurso da Amazon (ARN) para um bucket do Simple Storage Service (Amazon S3). Nesses casos, certifique-se de usar as duas chaves de contexto de condição global para limitar as permissões. Em alguns casos, você usa chaves de contexto de condição global e o valor `aws:SourceArn` contém o ID da conta. Nesses casos, verifique se o valor `aws:SourceAccount` e a conta no `aws:SourceArn` usa o mesmo ID de conta quando eles são usados na mesma instrução de política. Use `aws:SourceArn` se quiser que apenas um recurso seja associado ao acesso entre serviços. Use `aws:SourceAccount` se você quiser permitir que qualquer recurso nessa conta da AWS específica seja associado ao uso entre serviços.

Verifique se o valor de `aws:SourceArn` é um ARN para um tipo de recurso do Amazon RDS. Para obter mais informações, consulte Nomes de recurso da Amazon (ARNs) no Amazon RDS.

A maneira mais eficaz de se proteger contra o problema do substituto confuso é usar a chave de contexto de condição global `aws:SourceArn` com o ARN completo do recurso. Em alguns casos, talvez você não saiba o ARN completo do recurso ou pode estar especificando vários recursos. Nesses casos, use a chave de condição de contexto global com curingas `aws:SourceArn` (`*`) para as partes desconhecidas do ARN. Um exemplo é `arn:aws:rds:*:123456789012:*`.

O exemplo a seguir mostra como é possível usar as chaves de contexto de condição global `aws:SourceArn` e `aws:SourceAccount` no Amazon RDS, a fim de evitar o problema do substituto confuso.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Sid": "ConfusedDeputyPreventionExamplePolicy",
    "Effect": "Allow",
    "Principal": {
      "Service": "rds.amazonaws.com"
    },
    "Action": "sts:AssumeRole",
    "Condition": {
      "ArnLike": {
        "aws:SourceArn": "arn:aws:rds:us-east-1:123456789012:db:mydbinstance"
      },
      "StringEquals": {
        "aws:SourceAccount": "123456789012"
      }
    }
  }
}
```

Para obter mais exemplos de políticas que usam as chaves de contexto de condição global `aws:SourceArn` e `aws:SourceAccount`, consulte as seguintes seções:

- Conceder permissões para publicar notificações em um tópico do Amazon SNS
- Criar manualmente uma função do IAM para backup e restauração nativos
- Configuração da autenticação do Windows para instâncias de banco de dados do SQL Server
- Pré-requisitos para integrar o RDS for SQL Server com o S3
- Criar manualmente uma função do IAM para a Auditoria do SQL Server
- Configurar permissões do IAM para a integração do RDS para Oracle com o Amazon S3
- Configurar o acesso a um bucket do Amazon S3 (Importação do PostgreSQL)
- Configurar o acesso a um bucket do Amazon S3 (Exportação do PostgreSQL)

Autenticação do banco de dados do IAM para MariaDB, MySQL e PostgreSQL

Você pode se autenticar a instância de banco de dados usando a autenticação de banco de dados do AWS Identity and Access Management (IAM). A autenticação do banco de dados do IAM funciona com o MariaDB, MySQL e o PostgreSQL. Com esse método de autenticação, você não precisa usar uma senha ao conectar-se a um cluster de de banco de dados. Em vez disso, você usa um token de autenticação.

Um token de autenticação é uma string exclusiva de caracteres que o Amazon RDS gera mediante solicitação. Os tokens de autenticação são gerados usando o Signature da AWS versão 4. Cada token tem uma vida útil de 15 minutos. Você não precisa armazenar as credenciais de usuário no banco de dados, porque a autenticação é gerenciada externamente usando o IAM. Você também pode usar a autenticação de banco de dados padrão. O token é usado apenas para autenticação e não afetará a sessão depois que ela for estabelecida.

A autenticação do banco de dados do IAM oferece os seguintes benefícios:

- O tráfego de rede de e para o banco de dados é criptografado usando Secure Socket Layer (SSL) ou Transport Layer Security (TLS). Para obter mais informações sobre como usar SSL/TLS com o Amazon RDS, consulte Usar SSL/TLS para criptografar uma conexão com uma instância ou um cluster de banco de dados.
- Você pode usar o IAM para gerenciar centralmente o acesso aos recursos de banco de dados, em vez de gerenciar o acesso individualmente em cada instância de banco de dados.
- Para aplicações em execução no Amazon EC2, você pode usar as credenciais específicas da instância do EC2 para acessar o banco de dados em vez de uma senha para maior segurança.

Em geral, considere usar a autenticação de banco de dados do IAM quando suas aplicações criam menos de 200 conexões por segundo e você não deseja gerenciar nomes de usuário e senhas diretamente no código da aplicação.

O driver JDBC da Amazon Web Services (AWS) comporta a autenticação do banco de dados do IAM. Para ter mais informações, consulte AWS IAM Authentication Plugin no Amazon Web Services (AWS) JDBC Driver GitHub repository.

O driver Python da Amazon Web Services (AWS) comporta a autenticação do banco de dados do IAM. Para ter mais informações, consulte AWS IAM Authentication Plugin no Amazon Web Services (AWS) Python Driver GitHub repository.

Navegue pelos tópicos a seguir para aprender o processo de configuração do IAM para autenticação de banco de dados:

- Habilitar e desabilitar a autenticação de banco de dados do IAM
- Criar e usar uma política do IAM para acesso do banco de dados do IAM
- Criar uma conta de banco de dados usando autenticação do IAM
- Conectar-se à instância de banco de dados usando a autenticação do IAM

Disponibilidade de regiões e versões

A disponibilidade e a compatibilidade de recursos variam entre versões específicas de cada mecanismo de banco de dados. Para ter mais informações sobre a disponibilidade de versões e regiões com o Amazon RDS e a autenticação de banco de dados do IAM, consulte Regiões e mecanismos de banco de dados compatíveis com a autenticação de banco de dados do IAM no Amazon RDS.

Suporte para CLI e SDK

A autenticação de banco de dados do IAM está disponível para a AWS CLI e para os seguintes AWS SDKs específicos à linguagem:

- AWS SDK para .NET
- AWS SDK para C++
- AWS SDK para Go
- AWS SDK para Java
- AWS SDK para JavaScript
- AWS SDK para PHP
- AWS SDK para Python (Boto3)
- AWS SDK para Ruby

Limitações para a autenticação de banco de dados do IAM

Ao usar a autenticação do banco de dados do IAM, as seguintes limitações se aplicam:

- A autenticação do banco de dados do IAM limita as conexões a 200 por segundo.

As conexões que usam o mesmo token de autenticação não têm controle de utilização. Recomendamos que você reutilize os tokens de autenticação sempre que possível.

- Atualmente, a autenticação do banco de dados do IAM não oferece suporte a todas as chaves de contexto de condição global.

  Para obter mais informações sobre chaves de contexto de condição global, consulte Chaves de contexto de condição global AWS no Guia do usuário do IAM.
- No PostgreSQL, se o perfil do IAM (`rds_iam`) for adicionado a um usuário (por exemplo, o usuário principal do RDS), a autenticação do IAM terá precedência sobre a autenticação por senha, então o usuário precisará fazer login como um usuário do IAM.
- Para o PostgreSQL, o Amazon RDS não permite a habilitação dos métodos de autenticação do IAM e do Kerberos ao mesmo tempo.
- Para o PostgreSQL, você não pode usar a autenticação do IAM para estabelecer uma conexão de replicação.
- Não é possível usar um registro DNS personalizado do Route 53 ou um endpoint personalizado do Aurora em vez do endpoint do cluster da instância de banco de dados para gerar o token de autenticação.
- O CloudWatch e o CloudTrail não registram em log a autenticação do IAM. Esses serviços não rastreiam chamadas de API `generate-db-auth-token` que autorizam o perfil do IAM a habilitar a conexão com o banco de dados.
- A autenticação de banco de dados do IAM requer recursos de computação na instância de banco de dados. É necessário ter entre 300 MiB e 1.000 MiB extras de memória no banco de dados para ter uma conectividade confiável. Para ver a memória necessária para a workload, compare a coluna RES para processos do RDS na lista de processos do monitoramento avançado antes e depois de habilitar a autenticação de banco de dados do IAM. Consulte Como visualizar métricas do SO no console do RDS.

  Se você estiver usando uma instância de classe com capacidade de expansão, evite ficar sem memória reduzindo a memória usada por outros parâmetros, como buffers e cache, na mesma quantidade.
- Para o RDS para MySQL, não é possível usar autenticação baseada em senha para um usuário do banco de dados que você configura com autenticação do IAM.
- Não é possível usar a autenticação de banco de dados do IAM com o RDS no Outposts em nenhum mecanismo.

Recomendações para autenticação de banco de dados do IAM

Recomendamos o seguinte durante o uso da autenticação do banco de dados do IAM:

- Use a autenticação de banco de dados do IAM quando sua aplicação exigir menos de 200 novas conexões de autenticação de banco de dados do IAM por segundo.

  Os mecanismos de banco de dados que funcionam com o Amazon RDS não impõem quaisquer limites para as tentativas de autenticação por segundo. No entanto, quando você usa a autenticação de banco de dados do IAM, sua aplicação deve gerar um token de autenticação. Sua aplicação então usa esse token para se conectar à instância de banco de dados. Se você exceder o limite de novas conexões máximas por segundo, a sobrecarga extra da autenticação de banco de dados IAM poderá causar a limitação da conexão.

  Considere usar o agrupamento de conexões em suas aplicações para mitigar a criação constante de conexões. Isso pode reduzir a sobrecarga da autenticação de banco de dados do IAM e permitir que as aplicações reutilizem as conexões existentes. Como alternativa, considere usar o RDS Proxy para esses casos. O RDS Proxy tem custos adicionais. Consulte Preços do RDS Proxy.
- O tamanho de um token de autenticação de banco de dados do IAM depende de muitas coisas, incluindo o número de etiquetas do IAM, políticas de serviço do IAM, comprimentos de ARN, bem como outras propriedades do IAM e do banco de dados. O tamanho mínimo desse token geralmente é de cerca de 1 KB, mas pode ser maior. Como esse token é usado como senha na string de conexão com o banco de dados por meio da autenticação do IAM, você deve garantir que o driver de banco de dados (por exemplo, ODBC) e/ou quaisquer ferramentas não limitem nem trunquem esse token devido ao respectivo tamanho. Um token truncado fará com que a validação da autenticação feita pelo banco de dados e pelo IAM falhe.
- Se você estiver usando credenciais temporárias ao criar um token de autenticação do banco de dados do IAM, as credenciais temporárias ainda deverão ser válidas ao usar o token de autenticação do banco de dados do IAM para fazer uma solicitação de conexão.

Chaves de contexto de condição globais da AWS incompatíveis

A autenticação do banco de dados do IAM não é compatível com o seguinte subconjunto de chaves de contexto de condição global da AWS.

- `aws:Referer`
- `aws:SourceIp`

- `aws:SourceVpc`
- `aws:SourceVpce`
- `aws:UserAgent`
- `aws:VpcSourceIp`

Para obter mais informações, consulte Chaves de contexto de condição global da AWS no Guia do usuário do IAM.

Habilitar e desabilitar a autenticação de banco de dados do IAM

Por padrão, a autenticação de banco de dados do IAM está desabilitada nas instâncias de banco de dados. É possível habilitar ou desabilitar a autenticação de banco de dados do IAM usando o AWS Management Console, a AWS CLI ou a API.

É possível habilitar a autenticação de banco de dados do IAM ao executar uma das seguintes ações:

- Para criar uma nova instância de banco de dados com autenticação de banco de dados do IAM habilitada, consulte Criar uma instância de banco de dados do Amazon RDS.
- Para modificar uma instância de banco de dados para habilitar a autenticação de banco de dados do IAM, consulte Modificar uma instância de banco de dados do Amazon RDS.
- Para restaurar uma instância de banco de dados de um snapshot com a autenticação de banco de dados do IAM habilitada, consulte Restaurar uma instância de banco de dados.
- Para restaurar uma instância de banco de dados em um momento específico com a autenticação de banco de dados do IAM habilitada, consulte Restaurar uma instância de banco de dados para um momento especificado no Amazon RDS.

A autenticação do IAM para de instâncias de banco de dados PostgreSQL exige que o valor de SSL seja 1. Você não pode habilitar a autenticação IAM para uma instância de banco de dados PostgreSQL caso o valor SSL seja 0. Você não pode alterar o valor SSL para 0 caso a autenticação IAM seja habilitada para uma instância de banco de dados PostgreSQL.

Console

Cada fluxo de trabalho de criação ou modificação tem uma seção Database authentication (Autenticação de banco de dados), onde é possível habilitar ou desabilitar a autenticação de banco de dados do IAM. Nessa seção, escolha Password and IAM database authentication (Senha e autenticação do banco de dados do IAM) para habilitar a autenticação do banco de dados do IAM.

Para habilitar ou desabilitar a autenticação do banco de dados do IAM para uma instância de banco de dados existente

1. Abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Databases (Bancos de dados).
3. Escolha a instância de banco de dados que você deseja modificar.

> **Note**
>
> Certifique-se de que a instância de banco de dados seja compatível com a autenticação do IAM. Verifique os requisitos de compatibilidade em Disponibilidade de regiões e versões.

4. Selecione Modify.
5. Na seção Database authentication (Autenticação do banco de dados), escolha Password and IAM database authentication (Senha e autenticação do banco de dados do IAM) para habilitar a autenticação do banco de dados do IAM. Escolha Autenticação de senha ou Senha e autenticação Kerberos para desabilitar a autenticação do IAM.
6. Você também pode optar por habilitar a publicação de logs de autenticação de banco de dados do IAM no CloudWatch Logs. Em Exportações de log, escolha a opção Log iam-db-auth-error. A publicação de logs no CloudWatch Logs consome armazenamento e gera cobranças por esse armazenamento. Se não precisar mais de algum log do CloudWatch, lembre-se de excluí-lo.
7. Escolha Continue.
8. Para aplicar as alterações imediatamente, escolha Immediately (Imediatamente) na seção Scheduling of modifications (Programação de modificações).
9. Escolha Modify DB Instance (Modificar instância de banco de dados) para salvar suas alterações.

AWS CLI

Para criar uma nova instância de banco de dados com a autenticação do IAM usando a AWS CLI, use o comando `create-db-instance`. Especifique a opção `--enable-iam-database-authentication`, conforme exibido no exemplo a seguir.

```
aws rds create-db-instance \
    --db-instance-identifier mydbinstance \
```

```
    --db-instance-class db.m3.medium \
    --engine MySQL \
    --allocated-storage 20 \
    --master-username masterawsuser \
    --manage-master-user-password \
    --enable-iam-database-authentication
```

Para atualizar uma instância de banco de dados existente a fim de ter ou não autenticação do IAM, use o comando AWS CLI da `modify-db-instance`. Especifique a opção `--enable-iam-database-authentication` ou `--no-enable-iam-database-authentication`, conforme apropriado.

> **Note**
>
> Certifique-se de que a instância de banco de dados seja compatível com a autenticação do IAM. Verifique os requisitos de compatibilidade em Disponibilidade de regiões e versões.

Por padrão, o Amazon RDS modifica a instância de banco de dados durante a próxima janela de manutenção. Se você quiser habilitar a autenticação de banco de dados do IAM o mais rápido possível, use o parâmetro `--apply-immediately`.

Os exemplos a seguir mostram como habilitar imediatamente a autenticação do IAM para uma instância de banco de dados existente.

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --apply-immediately \
    --enable-iam-database-authentication
```

Se você está restaurando uma instância de banco de dados, use um dos comandos da AWS CLI a seguir:

- `restore-db-instance-to-point-in-time`
- `restore-db-instance-from-db-snapshot`

A autenticação de banco de dados do IAM assumirá como padrão aquela do snapshot de origem. Para alterar essa configuração, defina a opção `--enable-iam-database-authentication` ou `--no-enable-iam-database-authentication`, conforme apropriado.

API do RDS

Para criar uma nova instância de banco de dados com a autenticação do IAM usando a API, use a operação da API `CreateDBInstance`. Defina o parâmetro `EnableIAMDatabaseAuthentication` como `true`.

Para atualizar uma instância de banco de dados existente a fim de ter ou não autenticação do IAM, use a operação da API `ModifyDBInstance`. Defina o parâmetro `EnableIAMDatabaseAuthentication` como `true` para habilitar a autenticação do IAM, ou `false` para desabilitá-la.

> **Note**
>
> Certifique-se de que a instância de banco de dados seja compatível com a autenticação do IAM. Verifique os requisitos de compatibilidade em Disponibilidade de regiões e versões.

Se você está restaurando um cluster de de banco de dados, use uma das operações da API a seguir:

- `RestoreDBInstanceFromDBSnapshot`
- `RestoreDBInstanceToPointInTime`

A autenticação de banco de dados do IAM assumirá como padrão aquela do snapshot de origem. Para alterar essa configuração, defina o parâmetro `EnableIAMDatabaseAuthentication` como `true` para habilitar a autenticação do IAM, ou `false` para desabilitá-la.

Criar e usar uma política do IAM para acesso do banco de dados do IAM

Para permitir que um usuário ou um perfil se conecte à instância de banco de dados, você deve criar uma política do IAM. Depois disso, associe a política a um conjunto de permissões ou a um perfil.

> **Note**
>
> Para saber mais sobre as políticas do IAM, consulte Gerenciamento de identidade e acesso no Amazon RDS.

O exemplo de política a seguir permite que um usuário se conecte a uma instância de banco de dados usando a autenticação de banco de dados do IAM.

JSON

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Action": [
             "rds-db:connect"
         ],
         "Resource": [
             "arn:aws:rds-db:us-east-2:1234567890:dbuser:db-ABCDEFGHIJKL01234/
db_user"
         ]
      }
   ]
}
```

JSON

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Action": [
             "rds-db:connect"
         ],
         "Resource": [
             "arn:aws:rds-db:us-east-2:1234567890:dbuser:cluster-
ABCDEFGHIJKL01234/db_user"
         ]
      }
   ]
}
```

> **Important**
>
> Um usuário com permissões de administrador pode acessar instâncias de banco de dados sem permissões explícitas em uma política do IAM. Se você quiser restringir o acesso do administrador a instâncias de banco de dados, é possível criar um perfil do IAM com as permissões adequadas e menos privilegiadas e atribuí-lo ao administrador.

> **Note**
>
> Não confunda o prefixo `rds-db:` com outros prefixos de operações da API do RDS que começam com `rds:`. Você usa o prefixo `rds-db:` e a ação `rds-db:connect` somente para a autenticação de banco de dados do IAM. Eles não são válidos em nenhum outro contexto.

Os exemplos de política incluem uma única instrução com os seguintes elementos:

- `Effect`: especifica `Allow` para conceder acesso à instância de banco de dados. Se você não permitir explicitamente o acesso, o acesso será negado por padrão.
- `Action`: especifica `rds-db:connect` para permitir conexões com a instância de banco de dados.
- `Resource`: especifica um nome do recurso da Amazon (ARN) que descreva uma conta de banco de dados em um instância. de banco de dados. O formato do ARN é o seguinte.

```
arn:aws:rds-db:region:account-id:dbuser:DbiResourceId/db-user-name
```

  Neste formato, substitua o seguinte:

  - *`region`* é a região da AWS para a instância de Bancos de Dados . No exemplo de política, a região da AWS é `us-east-2`.
  - *`account-id`* é o número da conta da AWS para a instância de Bancos de Dados. No exemplo de política, o número da conta é `1234567890`. O usuário deve estar na mesma conta que a conta da instância de banco de dados.

    Para realizar o acesso entre contas, crie um perfil do IAM com a política mostrada acima na conta da instância de banco de dados e permita que sua outra conta assuma o perfil.

- *DbiResourceId* é o identificador da instância de banco de dados. Esse identificador é exclusivo para uma região da AWS e nunca muda. Na exemplo de política, o identificador é db-ABCDEFGHIJKL01234.

  Para encontrar um ID de recurso de instância de banco de dados no AWS Management Console do Amazon RDS, escolha a instância de banco de dados para ver os respectivos detalhes. Em seguida, escolha a guia Configuration (Configuração). O Resource ID (ID de recurso) é exibido na seção Configuration (Configuração).

  Como alternativa, use o comando da AWS CLI para listar os identificadores e os IDs de recurso de todas as suas instâncias de Bancos de Dados na região atual da AWS, conforme mostrado a seguir.

  ```
  aws rds describe-db-instances --query "DBInstances[*].
  [DBInstanceIdentifier,DbiResourceId]"
  ```

  Se você estiver usando o Amazon Aurora, especifique um `DbClusterResourceId` em vez de um `DbiResourceId`. Para obter mais informações, consulte Criar e usar uma política do IAM para acesso do banco de dados do IAM no Guia do usuário do Amazon Aurora.

  > **Note**
  >
  > Se você estiver se conectando a um banco de dados por meio do RDS Proxy, especifique o ID do recurso de proxy, como `prx-ABCDEFGHIJKL01234`. Para obter informações sobre como usar a autenticação de banco de dados do IAM com RDS Proxy, consulte Conectar-se a um proxy usando autenticação do IAM.

- *db-user-name* é o nome da conta de banco de dados para associar à autenticação do IAM. No exemplo de política, a conta de banco de dados é `db_user`.

Você pode criar outros ARNs que sejam compatíveis com vários padrões de acesso. A política a seguir permite o acesso a duas contas de banco de dados diferentes em uma instância de banco de dados.

JSON

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Action": [
             "rds-db:connect"
         ],
         "Resource": [
             "arn:aws:rds-db:us-east-2:123456789012:dbuser:db-ABCDEFGHIJKL01234/
jane_doe",
             "arn:aws:rds-db:us-east-2:123456789012:dbuser:db-ABCDEFGHIJKL01234/
mary_roe"
         ]
      }
   ]
}
```

JSON

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Action": [
             "rds-db:connect"
         ],
         "Resource": [
             "arn:aws:rds-db:us-east-2:123456789012:dbuser:cluster-
ABCDEFGHIJKL01234/jane_doe",
             "arn:aws:rds-db:us-east-2:123456789012:dbuser:cluster-
ABCDEFGHIJKL01234/mary_roe"
         ]
      }
   ]
}
```

A política a seguir usa o caractere "*" para comparar todas as instâncias de Bancos de Dados e todas as contas de banco de dados para uma conta daAWS e uma região específicas da AWS.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "rds-db:connect"
            ],
            "Resource": [
                "arn:aws:rds-db:us-east-2:1234567890:dbuser:*/*"
            ]
        }
    ]
}
```

A política a seguir compara todas as instâncias de Bancos de Dados de uma conta da AWS e uma região da AWS específicas. Contudo, a política concede acesso somente às instâncias de banco de dados que têm uma conta de banco de dados `jane_doe`.

JSON

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Action": [
             "rds-db:connect"
         ],
         "Resource": [
             "arn:aws:rds-db:us-east-2:123456789012:dbuser:*/jane_doe"
```

```
            ]
        }
    ]
}
```

O usuário ou perfil tem acesso apenas aos bancos de dados que o usuário do banco de dados tem. Por exemplo, suponha que sua instância de banco de dados tenha um banco de dados chamado dev e outro chamado test. Se a usuária do banco de dados `jane_doe` tiver acesso apenas a dev, os usuários ou perfis que acessarem essa instância de banco de dados com a usuária `jane_doe` também terão acesso apenas a dev. Essa restrição de acesso também é válida para outros objetos de banco de dados, como tabelas, visualizações e assim por diante.

Um administrador deve criar políticas do IAM que concedam às entidades permissões para executar operações de API específicas nos recursos especificados de que precisam. Depois, o administrador deve anexar essas políticas aos conjuntos de permissões e perfis que exigem essas permissões. Para obter exemplos de políticas, consulte Exemplos de políticas baseadas em identidade do Amazon RDS.

Anexar uma política do IAM a um conjunto de permissões ou perfil

Depois de criar uma política do IAM para permitir a autenticação de banco de dados, você precisa anexar a política a um conjunto de permissões ou a um perfil. Para obter um tutorial sobre esse tópico, consulte Criar e anexar sua primeira política gerenciada pelo cliente no Guia do usuário do IAM.

À medida que avança pelo tutorial, você pode usar um dos exemplos de política mostrados nessa seção como um ponto de partida e adequá-lo às suas necessidades. No fim do tutorial, você terá um conjunto de permissões com uma política anexada que pode usar a ação `rds-db:connect`.

> **Note**
>
> Você pode mapear vários conjuntos de permissões ou perfis para a mesma conta de usuário do banco de dados. Por exemplo, suponha que a sua política do IAM especificou o seguinte recurso do ARN.
>
> ```
> arn:aws:rds-db:us-east-2:123456789012:dbuser:db-12ABC34DEFG5HIJ6KLMNOP78QR/
> jane_doe
> ```

Se você anexar a política a Jane, Bob e Diego, todos eles poderão se conectar à instância de banco de dados em questão usando a conta de banco de dados `jane_doe`.

Criar uma conta de banco de dados usando autenticação do IAM

Com a autenticação de banco de dados do IAM, você não precisa atribuir senhas de banco de dados à conta de usuário que você criar. Se você remover um usuário que está mapeado a uma conta de banco de dados, também deverá remover a conta de banco de dados com a instrução `DROP USER`.

> **Note**
>
> O nome de usuário usado para autenticação do IAM deve corresponder ao caso do nome de usuário no banco de dados.

Tópicos

- Usar autenticação do IAM com o MariaDB e o MySQL
- Usar autenticação do IAM com o PostgreSQL

Usar autenticação do IAM com o MariaDB e o MySQL

Com o MariaDB e o MySQL, a autenticação é processada por `AWSAuthenticationPlugin`: um plug-in fornecido pela AWS que funciona perfeitamente com o IAM para autenticar seus usuários. Conecte-se à instância de banco de dados como usuário principal ou um usuário diferente que possa criar usuários e conceder privilégios. Depois de se conectar, emita a instrução `CREATE USER` conforme mostrado no exemplo a seguir.

```
CREATE USER 'jane_doe' IDENTIFIED WITH AWSAuthenticationPlugin AS 'RDS';
```

A cláusula `IDENTIFIED WITH` permite que o MariaDB e o MySQL usem o `AWSAuthenticationPlugin` para autenticar a conta de banco de dados (`jane_doe`). A `AS 'RDS'` cláusula refere-se ao método de autenticação. Verifique se o nome do usuário do banco de dados especificado é o mesmo que um recurso na política do IAM para acesso ao banco de dados

do IAM. Para obter mais informações, consulte Criar e usar uma política do IAM para acesso do banco de dados do IAM.

> **Note**
>
> Se você ver a mensagem a seguir, significa que o plugin fornecido pela AWS não está disponível para a instância de Bancos de Dados atual .
> `ERROR 1524 (HY000): Plugin 'AWSAuthenticationPlugin' is not loaded`
> Para solucionar esse erro, verifique se você está usando uma configuração compatível e se habilitou a autenticação de banco de dados do IAM em sua instância de banco de dados. Para obter mais informações, consulte Disponibilidade de regiões e versões e Habilitar e desabilitar a autenticação de banco de dados do IAM.

Apos criar uma conta usando `AWSAuthenticationPlugin`, você a gerencia do mesmo modo que as outras contas de banco de dados. Por exemplo, você pode modificar os privilégios da conta com os atributos `GRANT` e `REVOKE`, ou modificar os vários atributos da conta com a instrução `ALTER USER`.

O tráfego de rede do banco de dados é criptografado utilizando SSL/TLS ao usar o IAM. Para permitir conexões SSL, modifique a conta do usuário com o comando a seguir.

```
ALTER USER 'jane_doe'@'%' REQUIRE SSL;
```

Usar autenticação do IAM com o PostgreSQL

Para usar a autenticação do IAM com o PostgreSQL, conecte-se à instância de banco de dados como usuário principal ou um usuário diferente que possa criar usuários e conceder privilégios. Depois de conectar-se, crie usuários de banco de dados e, depois, conceda a eles a função `rds_iam` conforme mostrado no exemplo a seguir.

```
CREATE USER db_userx;
GRANT rds_iam TO db_userx;
```

Verifique se o nome do usuário do banco de dados especificado é o mesmo que um recurso na política do IAM para acesso ao banco de dados do IAM. Para obter mais informações, consulte Criar e usar uma política do IAM para acesso do banco de dados do IAM. Você deve conceder o perfil

`rds_iam` para usar a autenticação do IAM. Também é possível usar associações aninhadas ou concessões indiretas do perfil.

Conectar-se à instância de banco de dados usando a autenticação do IAM

Com a autenticação de banco de dados do IAM, você usa um token de autenticação ao se conectar à instância de banco de dados. Um token de autenticação é uma string de caracteres que você usa em vez de uma senha. Depois que você gerar um token de autenticação, ele será válido por 15 minutos antes de expirar. Se você tentar se conectar usando um token expirado, a solicitação de conexão será negada.

Todo token de autenticação deve ser acompanhado por uma assinatura válida, usando o Signature da AWS versão 4. (Para ter mais informações, consulte Processo de assinatura do Signature versão 4 na Referência geral da AWS.) A AWS CLI e um SDK da AWS, como AWS SDK para Java ou AWS SDK para Python (Boto3), podem assinar automaticamente cada token que você criar.

Você pode usar um token de autenticação quando se conectar ao Amazon RDS de outro serviço da AWS, como o AWS Lambda. Ao usar um token, você não precisa inserir uma senha no seu código. Como alternativa, você pode usar o SDK da AWS para criar e assinar programaticamente um token de autenticação.

Depois que tiver um token de autenticação do IAM assinado, você poderá se conectar a uma instância de banco de dados do Amazon RDS. Veja a seguir como fazer isso usando uma ferramenta de linha de comando ou um SDK da AWS, como o AWS SDK para Java ou AWS SDK para Python (Boto3).

Para ter mais informações, consulte as seguintes postagens no blog:

- Use IAM authentication to connect with SQL Workbench/J to Aurora MySQL or Amazon RDS para MySQL
- Using IAM authentication to connect with pgAdmin Amazon Aurora PostgreSQL or Amazon RDS para PostgreSQL

Pré-requisitos

Veja a seguir os pré-requisitos para se conectar à instância de banco de dados usando a autenticação do IAM:

- Habilitar e desabilitar a autenticação de banco de dados do IAM

- Criar e usar uma política do IAM para acesso do banco de dados do IAM
- Criar uma conta de banco de dados usando autenticação do IAM

Tópicos

- Conectar-se à instância de banco de dados usando a autenticação do IAM com os drivers da AWS.
- Conectando-se à sua instância de banco de dados usando a autenticação do IAM na linha de comando: AWS CLI e cliente mysql
- Conectar a instância de banco de dados usando a autenticação do IAM na linha de comando: AWS CLI e cliente psql
- Conectar-se à instância de banco de dados usando a autenticação do IAM e o AWS SDK para .NET
- Conectar-se à instância de banco de dados usando a autenticação do IAM e o AWS SDK para Go
- Conectar-se à instância de banco de dados usando a autenticação do IAM e o AWS SDK para Java
- Conectar-se à instância de banco de dados usando a autenticação do IAM e o AWS SDK para Python (Boto3)

Conectar-se à instância de banco de dados usando a autenticação do IAM com os drivers da AWS.

O pacote de drivers da AWS foram projetados para comportar tempos mais rápidos de transição e de failover, além de autenticação com o AWS Secrets Manager, o AWS Identity and Access Management (IAM) e identidades federadas. Os drivers da AWS dependem do monitoramento do status da instância de banco de dados e do conhecimento da topologia da instância para determinar o novo gravador. Essa abordagem reduz os tempos de transição e de failover para segundos de um dígito, em comparação com dezenas de segundos para drivers de código aberto.

Para ter mais informações sobre os drivers da AWS, consulte o driver de linguagem correspondente para a instância de banco de dados do RDS para MariaDB, RDS para MySQL ou RDS para PostgreSQL.

> **Note**
>
> Os únicos recursos compatíveis com o RDS para MariaDB são a autenticação com o AWS Secrets Manager, o AWS Identity and Access Management (IAM) e a identidade federada.

Conectando-se à sua instância de banco de dados usando a autenticação do IAM na linha de comando: AWS CLI e cliente mysql

Você pode se conectar de uma linha de comando a uma instância de banco de dados do Amazon RDS com a AWS CLI e a ferramenta da linha de comando `mysql`, conforme descrito a seguir.

Pré-requisitos

Veja a seguir os pré-requisitos para se conectar à instância de banco de dados usando a autenticação do IAM:

- Habilitar e desabilitar a autenticação de banco de dados do IAM
- Criar e usar uma política do IAM para acesso do banco de dados do IAM
- Criar uma conta de banco de dados usando autenticação do IAM

> **Note**
>
> Para obter informações sobre como se conectar ao banco de dados usando o SQL Workbench/J com autenticação do IAM, consulte a publicação do blog Use IAM authentication to connect with SQL Workbench/J to Aurora MySQL or Amazon RDS para MySQL.

Tópicos

- Gerar um token de autenticação do IAM
- Conexão à instância de banco de dados

Gerar um token de autenticação do IAM

O exemplo a seguir mostra como obter um token de autenticação assinado usando a AWS CLI.

```
aws rds generate-db-auth-token \
   --hostname rdsmysql.123456789012.us-west-2.rds.amazonaws.com \
   --port 3306 \
   --region us-west-2 \
   --username jane_doe
```

No exemplo, os parâmetros são os seguintes:

- `--hostname`: o nome do host da instância de banco de dados que você deseja acessar
- `--port`: o número da porta usada para se conectar à instância de banco de dados
- `--region`: a região da AWS na qual a instância do banco de dados está em execução.
- `--username`: a conta de banco de dados que você deseja acessar

Os primeiros caracteres do token são parecidos com os seguintes.

```
rdsmysql.123456789012.us-west-2.rds.amazonaws.com:3306/?
Action=connect&DBUser=jane_doe&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Expires=900...
```

> **Note**
>
> Não é possível usar um registro DNS personalizado do Route 53 em vez do endpoint do da instância de banco de dados para gerar o token de autenticação.

Conexão à instância de banco de dados

O formato geral para se conectar é mostrado a seguir.

```
mysql --host=hostName --port=portNumber --ssl-ca=full_path_to_ssl_certificate --enable-
cleartext-plugin --user=userName --password=authToken
```

Os parâmetros são os seguintes:

- `--host`: o nome do host da instância de banco de dados que você deseja acessar
- `--port`: o número da porta usada para se conectar à instância de banco de dados
- `--ssl-ca`: o caminho completo para o arquivo de certificado SSL que contém a chave pública

  Para ter mais informações sobre o suporte SSL/TLS para o MariaDB, consulte Suporte de SSL/TLS para instâncias de banco de dados do MariaDB no Amazon RDS.

  Para ter mais informações sobre o suporte SSL/TLS para o MySQL, consulte Suporte do SSL/TLS para instâncias de banco de dados do MySQL no Amazon RDS.

  Para baixar um certificado SSL, consulte Usar SSL/TLS para criptografar uma conexão com uma instância ou um cluster de banco de dados.

- `--enable-cleartext-plugin`: um valor que especifica que o `AWSAuthenticationPlugin` deve ser usado para essa conexão

  Se você estiver usando um cliente MariaDB, a opção`--enable-cleartext-plugin` não será necessária.
- `--user`: a conta de banco de dados que você deseja acessar
- `--password`: um token de autenticação do IAM assinado

Um token de autenticação é composto de várias centenas de caracteres. Ele pode ser incômodo para a linha de comando. Um modo de contornar isso é salvar o token em uma variável de ambiente, e usar essa variável quando você se conectar. O exemplo a seguir mostra um modo de executar essa solução alternativa. No exemplo, */sample_dir/* corresponde ao caminho completo do arquivo de certificado SSL contendo a chave pública.

```
RDSHOST="mysqldb.123456789012.us-east-1.rds.amazonaws.com"
TOKEN="$(aws rds generate-db-auth-token --hostname $RDSHOST --port 3306 --region us-
west-2 --username jane_doe )"

mysql --host=$RDSHOST --port=3306 --ssl-ca=/sample_dir/global-bundle.pem --enable-
cleartext-plugin --user=jane_doe --password=$TOKEN
```

Quando você se conecta usando o `AWSAuthenticationPlugin`, a conexão é protegida usando SSL. Para verificar isso, digite o seguinte no prompt de comando `mysql>`.

```
show status like 'Ssl%';
```

As seguintes linhas na saída mostram mais detalhes.

```
+---------------+-------------+
| Variable_name | Value

                                                                    |
+---------------+-------------+
| ...           | ...
| Ssl_cipher    | AES256-SHA

                                                                    |
| ...           | ...
```

```
| Ssl_version   | TLSv1.1

                                                                    |
| ...           | ...
+-----------------------------+
```

Se você quiser se conectar a uma instância de banco de dados por meio de um proxy, consulte Conectar-se a um proxy usando autenticação do IAM.

Conectar a instância de banco de dados usando a autenticação do IAM na linha de comando: AWS CLI e cliente psql

Conecte-se pela linha de comando a uma instância de banco de dados do Amazon RDS para PostgreSQL com a AWS CLI e a ferramenta da linha de comando psql conforme descrito a seguir.

Pré-requisitos

Veja a seguir os pré-requisitos para se conectar à instância de banco de dados usando a autenticação do IAM:

- Habilitar e desabilitar a autenticação de banco de dados do IAM
- Criar e usar uma política do IAM para acesso do banco de dados do IAM
- Criar uma conta de banco de dados usando autenticação do IAM

> **Note**
>
> Para obter informações sobre como se conectar ao seu banco de dados usando pgAdmin com autenticação do IAM, consulte a publicação do blog Using IAM authentication to connect with pgAdmin Amazon Aurora PostgreSQL or Amazon RDS para PostgreSQL.

Tópicos

- Gerar um token de autenticação do IAM
- Conectar-se a uma instância PostgreSQL do Amazon RDS

Gerar um token de autenticação do IAM

Um token de autenticação é composto de várias centenas de caracteres. Dessa maneira, ele pode ficar estranho na linha de comando. Um modo de contornar isso é salvar o token em uma variável

de ambiente, e usar essa variável quando você se conectar. O exemplo a seguir mostra como usar a AWS CLI para obter um token de autenticação assinado usando o comando `generate-db-auth-token` e armazená-lo em uma variável de ambiente `PGPASSWORD`.

```
export RDSHOST="rdspostgres.123456789012.us-west-2.rds.amazonaws.com"
export PGPASSWORD="$(aws rds generate-db-auth-token --hostname $RDSHOST --port 5432 --
region us-west-2 --username jane_doe )"
```

No exemplo, os parâmetros para o comando `generate-db-auth-token` são os seguintes:

- `--hostname`: o nome do host da instância de banco de dados que você deseja acessar
- `--port`: o número da porta usada para se conectar à instância de banco de dados
- `--region`: a região da AWS na qual a instância do banco de dados está em execução.
- `--username`: a conta de banco de dados que você deseja acessar

Os primeiros caracteres do token gerado são parecidos com os seguintes.

```
rdspostgres.123456789012.us-west-2.rds.amazonaws.com:5432/?
Action=connect&DBUser=jane_doe&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Expires=900...
```

> **Note**
>
> Não é possível usar um registro DNS personalizado do Route 53 em vez do endpoint da instância de banco de dados para gerar o token de autenticação.

Conectar-se a uma instância PostgreSQL do Amazon RDS

O formato geral para usar psql na conexão é mostrado a seguir.

```
psql "host=hostName port=portNumber sslmode=verify-full
 sslrootcert=full_path_to_ssl_certificate dbname=DBName user=userName
 password=authToken"
```

Os parâmetros são os seguintes:

- `host`: o nome do host da instância de banco de dados que você deseja acessar

- `port`: o número da porta usada para se conectar à instância de banco de dados
- `sslmode`: o modo SSL a ser usado

  Quando você usa `sslmode=verify-full`, a conexão SSL verifica o endpoint da instância de banco de dados em relação ao endpoint no certificado SSL.
- `sslrootcert`: o caminho completo para o arquivo de certificado SSL que contém a chave pública

  Para ter mais informações, consulte Usar o SSL com uma instância de banco de dados PostgreSQL.

  Para baixar um certificado SSL, consulte Usar SSL/TLS para criptografar uma conexão com uma instância ou um cluster de banco de dados.
- `dbname`: o banco de dados que você deseja acessar
- `user`: a conta de banco de dados que você deseja acessar
- `password`: um token de autenticação do IAM assinado

> **Note**
>
> Não é possível usar um registro DNS personalizado do Route 53 em vez do endpoint da instância de banco de dados para gerar o token de autenticação.

O exemplo a seguir mostra o uso do psql para conexão. No exemplo, psql usa a variável de ambiente `RDSHOST` para o host e a variável de ambiente `PGPASSWORD` para o token gerado. Além disso, *`/sample_dir/`* corresponde ao caminho completo do arquivo de certificado SSL contendo a chave pública.

```
export RDSHOST="rdspostgres.123456789012.us-west-2.rds.amazonaws.com"
export PGPASSWORD="$(aws rds generate-db-auth-token --hostname $RDSHOST --port 5432 --
region us-west-2 --username jane_doe )"

psql "host=$RDSHOST port=5432 sslmode=verify-full sslrootcert=/sample_dir/global-
bundle.pem dbname=DBName user=jane_doe password=$PGPASSWORD"
```

Se você quiser se conectar a uma instância de banco de dados por meio de um proxy, consulte Conectar-se a um proxy usando autenticação do IAM.

Conectar-se à instância de banco de dados usando a autenticação do IAM e o AWS SDK para .NET

Você pode se conectar a uma instância de banco de dados do RDS para MariaDB, RDS para MySQL ou RDS para PostgreSQL com a AWS SDK para .NET, conforme descrito a seguir.

Pré-requisitos

Veja a seguir os pré-requisitos para se conectar à instância de banco de dados usando a autenticação do IAM:

- Habilitar e desabilitar a autenticação de banco de dados do IAM
- Criar e usar uma política do IAM para acesso do banco de dados do IAM
- Criar uma conta de banco de dados usando autenticação do IAM

Exemplos

O exemplo de código a seguir mostra como gerar um token de autenticação e usá-lo para se conectar a uma instância de banco de dados.

Para executar esse exemplo de código, você precisa do AWS SDK para .NET, encontrado no site AWS. Os pacotes `AWSSDK.CORE` e `AWSSDK.RDS` são necessários. Para se conectar a uma instância de banco de dados, use o conector de banco de dados .NET para o mecanismo de banco de dados, como MySqlConnector para MariaDB ou MySQL ou Npgsql para PostgreSQL.

Esse código se conecta a uma instância de banco de dados MariaDB ou MySQL. Modifique os valores das seguintes variáveis, conforme necessário:

- `server`: o endpoint da instância de banco de dados que você deseja acessar
- `user`: a conta de banco de dados que você deseja acessar
- `database`: o banco de dados que você deseja acessar
- `port`: o número da porta usada para se conectar à instância de banco de dados
- `SslMode`: o modo SSL a ser usado

  Quando você usa `SslMode=Required`, a conexão SSL verifica o endpoint da instância de banco de dados em relação ao endpoint no certificado SSL.
- `SslCa`: o caminho completo para o certificado SSL do Amazon RDS

  Para baixar um certificado, consulte Usar SSL/TLS para criptografar uma conexão com uma instância ou um cluster de banco de dados.

> **Note**
>
> Não é possível usar um registro DNS personalizado do Route 53 em vez do endpoint da instância de banco de dados para gerar o token de autenticação.

```
using System;
using System.Data;
using MySql.Data;
using MySql.Data.MySqlClient;
using Amazon;

namespace ubuntu
{
  class Program
  {
    static void Main(string[] args)
    {
      var pwd =
 Amazon.RDS.Util.RDSAuthTokenGenerator.GenerateAuthToken(RegionEndpoint.USEast1,
 "mysqldb.123456789012.us-east-1.rds.amazonaws.com", 3306, "jane_doe");
      // for debug only Console.Write("{0}\n", pwd);  //this verifies the token is
 generated

      MySqlConnection conn = new MySqlConnection($"server=mysqldb.123456789012.us-
east-1.rds.amazonaws.com;user=jane_doe;database=mydB;port=3306;password={pwd};SslMode=Required;
      conn.Open();

      // Define a query
      MySqlCommand sampleCommand = new MySqlCommand("SHOW DATABASES;", conn);

      // Execute a query
      MySqlDataReader mysqlDataRdr = sampleCommand.ExecuteReader();

      // Read all rows and output the first column in each row
      while (mysqlDataRdr.Read())
        Console.WriteLine(mysqlDataRdr[0]);

      mysqlDataRdr.Close();
      // Close connection
      conn.Close();
    }
```

```
  }
}
```

Esse código se conecta a uma instância de banco de dados PostgreSQL.

Modifique os valores das seguintes variáveis, conforme necessário:

- `Server`: o endpoint da instância de banco de dados que você deseja acessar
- `User ID`: a conta de banco de dados que você deseja acessar
- `Database`: o banco de dados que você deseja acessar
- `Port`: o número da porta usada para se conectar à instância de banco de dados
- `SSL Mode`: o modo SSL a ser usado

  Quando você usa `SSL Mode=Required`, a conexão SSL verifica o endpoint da instância de banco de dados em relação ao endpoint no certificado SSL.
- `Root Certificate`: o caminho completo para o certificado SSL do Amazon RDS

  Para baixar um certificado, consulte Usar SSL/TLS para criptografar uma conexão com uma instância ou um cluster de banco de dados.

> **Note**
>
> Não é possível usar um registro DNS personalizado do Route 53 em vez do endpoint da instância de banco de dados para gerar o token de autenticação.

```
using System;
using Npgsql;
using Amazon.RDS.Util;

namespace ConsoleApp1
{
    class Program
    {
        static void Main(string[] args)
        {
            var pwd =
 RDSAuthTokenGenerator.GenerateAuthToken("postgresmydb.123456789012.us-
east-1.rds.amazonaws.com", 5432, "jane_doe");
```

```
// for debug only Console.Write("{0}\n", pwd);  //this verifies the token is generated

            NpgsqlConnection conn = new
 NpgsqlConnection($"Server=postgresmydb.123456789012.us-east-1.rds.amazonaws.com;User
 Id=jane_doe;Password={pwd};Database=mydb;SSL Mode=Require;Root
 Certificate=full_path_to_ssl_certificate");
            conn.Open();

            // Define a query
                   NpgsqlCommand cmd = new NpgsqlCommand("select count(*) FROM
 pg_user", conn);

            // Execute a query
            NpgsqlDataReader dr = cmd.ExecuteReader();

            // Read all rows and output the first column in each row
            while (dr.Read())
                Console.Write("{0}\n", dr[0]);

            // Close connection
            conn.Close();
        }
    }
}
```

Se você quiser se conectar a uma instância de banco de dados por meio de um proxy, consulte Conectar-se a um proxy usando autenticação do IAM.

Conectar-se à instância de banco de dados usando a autenticação do IAM e o AWS SDK para Go

Você pode se conectar a uma instância de banco de dados do RDS para MariaDB, RDS para MySQL ou RDS para PostgreSQL com a AWS SDK para Go, conforme descrito a seguir.

Pré-requisitos

Veja a seguir os pré-requisitos para se conectar à instância de banco de dados usando a autenticação do IAM:

- Habilitar e desabilitar a autenticação de banco de dados do IAM
- Criar e usar uma política do IAM para acesso do banco de dados do IAM
- Criar uma conta de banco de dados usando autenticação do IAM

Exemplos

Para executar esses exemplos de código, você precisa do AWS SDK para Go, encontrado no site AWS.

Modifique os valores das seguintes variáveis, conforme necessário:

- `dbName`: o banco de dados que você deseja acessar
- `dbUser`: a conta de banco de dados que você deseja acessar
- `dbHost`: o endpoint da instância de banco de dados que você deseja acessar

> **Note**
>
> Não é possível usar um registro DNS personalizado do Route 53 em vez do endpoint do da instância de banco de dados para gerar o token de autenticação.

- `dbPort`: o número da porta usada para se conectar à instância de banco de dados
- `region`: a região da AWS na qual a instância do banco de dados está em execução.

Além disso, verifique se as bibliotecas importadas no código de exemplo existem no sistema.

> **Important**
>
> Os exemplos nesta seção usam o seguinte código para fornecer credenciais que acessam um banco de dados a partir de um ambiente local:
> `creds := credentials.NewEnvCredentials()`
> Se estiver acessando um banco de dados de um serviço da AWS, como o Amazon EC2 ou Amazon ECS, você poderá substituir o código pelo seguinte código:
> `sess := session.Must(session.NewSession())`
> `creds := sess.Config.Credentials`
> Se você fizer essa alteração, certifique-se de adicionar a seguinte importação:
> `"github.com/aws/aws-sdk-go/aws/session"`

Tópicos

- Conectar-se usando a autenticação do IAM e o AWS SDK para Go V2
- Conectar-se usando a autenticação do IAM e o AWS SDK para Go V1.

Conectar-se usando a autenticação do IAM e o AWS SDK para Go V2

Conecte-se a uma instância de banco de dados usando a autenticação do IAM e o AWS SDK para Go V2.

O exemplo de código a seguir mostra como gerar um token de autenticação e usá-lo para se conectar a uma instância de banco de dados.

Esse código se conecta a uma instância de banco de dados MariaDB ou MySQL.

```
package main

import (
     "context"
     "database/sql"
     "fmt"

     "github.com/aws/aws-sdk-go-v2/config"
     "github.com/aws/aws-sdk-go-v2/feature/rds/auth"
     _ "github.com/go-sql-driver/mysql"
)

func main() {

     var dbName string = "DatabaseName"
     var dbUser string = "DatabaseUser"
     var dbHost string = "mysqldb.123456789012.us-east-1.rds.amazonaws.com"
     var dbPort int = 3306
     var dbEndpoint string = fmt.Sprintf("%s:%d", dbHost, dbPort)
     var region string = "us-east-1"

    cfg, err := config.LoadDefaultConfig(context.TODO())
    if err != nil {
     panic("configuration error: " + err.Error())
    }

    authenticationToken, err := auth.BuildAuthToken(
     context.TODO(), dbEndpoint, region, dbUser, cfg.Credentials)
    if err != nil {
     panic("failed to create authentication token: " + err.Error())
    }

    dsn := fmt.Sprintf("%s:%s@tcp(%s)/%s?tls=true&allowCleartextPasswords=true",
        dbUser, authenticationToken, dbEndpoint, dbName,
```

```
    )

    db, err := sql.Open("mysql", dsn)
    if err != nil {
        panic(err)
    }

    err = db.Ping()
    if err != nil {
        panic(err)
    }
}
```

Esse código se conecta a uma instância de banco de dados PostgreSQL.

```
package main

import (
     "context"
     "database/sql"
     "fmt"

     "github.com/aws/aws-sdk-go-v2/config"
     "github.com/aws/aws-sdk-go-v2/feature/rds/auth"
     _ "github.com/lib/pq"
)

func main() {

     var dbName string = "DatabaseName"
     var dbUser string = "DatabaseUser"
     var dbHost string = "postgresmydb.123456789012.us-east-1.rds.amazonaws.com"
     var dbPort int = 5432
     var dbEndpoint string = fmt.Sprintf("%s:%d", dbHost, dbPort)
     var region string = "us-east-1"

    cfg, err := config.LoadDefaultConfig(context.TODO())
    if err != nil {
     panic("configuration error: " + err.Error())
    }

    authenticationToken, err := auth.BuildAuthToken(
     context.TODO(), dbEndpoint, region, dbUser, cfg.Credentials)
```

```
    if err != nil {
     panic("failed to create authentication token: " + err.Error())
    }

    dsn := fmt.Sprintf("host=%s port=%d user=%s password=%s dbname=%s",
        dbHost, dbPort, dbUser, authenticationToken, dbName,
    )

    db, err := sql.Open("postgres", dsn)
    if err != nil {
        panic(err)
    }

    err = db.Ping()
    if err != nil {
        panic(err)
    }
}
```

Se você quiser se conectar a uma instância de banco de dados por meio de um proxy, consulte Conectar-se a um proxy usando autenticação do IAM.

Conectar-se usando a autenticação do IAM e o AWS SDK para Go V1.

Conecte-se a um cluster de banco de dados usando a autenticação do IAM e o AWS SDK para Go V1

O exemplo de código a seguir mostra como gerar um token de autenticação e usá-lo para se conectar a uma instância de banco de dados.

Esse código se conecta a uma instância de banco de dados MariaDB ou MySQL.

```
package main

import (
    "database/sql"
    "fmt"
    "log"

    "github.com/aws/aws-sdk-go/aws/credentials"
    "github.com/aws/aws-sdk-go/service/rds/rdsutils"
    _ "github.com/go-sql-driver/mysql"
)
```

```
func main() {
    dbName := "app"
    dbUser := "jane_doe"
    dbHost := "mysqldb.123456789012.us-east-1.rds.amazonaws.com"
    dbPort := 3306
    dbEndpoint := fmt.Sprintf("%s:%d", dbHost, dbPort)
    region := "us-east-1"

    creds := credentials.NewEnvCredentials()
    authToken, err := rdsutils.BuildAuthToken(dbEndpoint, region, dbUser, creds)
    if err != nil {
        panic(err)
    }

    dsn := fmt.Sprintf("%s:%s@tcp(%s)/%s?tls=true&allowCleartextPasswords=true",
        dbUser, authToken, dbEndpoint, dbName,
    )

    db, err := sql.Open("mysql", dsn)
    if err != nil {
        panic(err)
    }

    err = db.Ping()
    if err != nil {
        panic(err)
    }
}
```

Esse código se conecta a uma instância de banco de dados PostgreSQL.

```
package main

import (
 "database/sql"
 "fmt"

 "github.com/aws/aws-sdk-go/aws/credentials"
 "github.com/aws/aws-sdk-go/service/rds/rdsutils"
 _ "github.com/lib/pq"
)
```

```
func main() {
    dbName := "app"
    dbUser := "jane_doe"
    dbHost := "postgresmydb.123456789012.us-east-1.rds.amazonaws.com"
    dbPort := 5432
    dbEndpoint := fmt.Sprintf("%s:%d", dbHost, dbPort)
    region := "us-east-1"

    creds := credentials.NewEnvCredentials()
    authToken, err := rdsutils.BuildAuthToken(dbEndpoint, region, dbUser, creds)
    if err != nil {
        panic(err)
    }

    dsn := fmt.Sprintf("host=%s port=%d user=%s password=%s dbname=%s",
        dbHost, dbPort, dbUser, authToken, dbName,
    )

    db, err := sql.Open("postgres", dsn)
    if err != nil {
        panic(err)
    }

    err = db.Ping()
    if err != nil {
        panic(err)
    }
}
```

Se você quiser se conectar a uma instância de banco de dados por meio de um proxy, consulte Conectar-se a um proxy usando autenticação do IAM.

Conectar-se à instância de banco de dados usando a autenticação do IAM e o AWS SDK para Java

Você pode se conectar a uma instância de banco de dados do RDS para MariaDB, RDS para MySQL ou RDS para PostgreSQL com a AWS SDK para Java, conforme descrito a seguir.

Pré-requisitos

Veja a seguir os pré-requisitos para se conectar à instância de banco de dados usando a autenticação do IAM:

- Habilitar e desabilitar a autenticação de banco de dados do IAM

- Criar e usar uma política do IAM para acesso do banco de dados do IAM
- Criar uma conta de banco de dados usando autenticação do IAM
- Set up the AWS SDK for Java (Configurar o AWS SDK for Java)

Consulte exemplos de como usar o SDK para Java 2.x em Amazon RDS examples using SDK for Java 2.x. Você também pode usar o AWS Advanced JDBC Wrapper. Para isso, consulte a documentação do AWS Advanced JDBC Wrapper.

Tópicos

- Gerar um token de autenticação do IAM
- Criar manualmente um token de autenticação do IAM
- Conexão à instância de banco de dados

Gerar um token de autenticação do IAM

Se estiver escrevendo programas usando o AWS SDK para Java, você pode obter um token de autenticação assinado usando a classe `RdsIamAuthTokenGenerator`. O uso dessa classe exige que você forneça as credenciais da AWS. Para fazer isso, crie uma instância da classe `DefaultAWSCredentialsProviderChain`. `DefaultAWSCredentialsProviderChain` usa a primeira chave de acesso da AWS e a chave secreta encontradas na cadeia de fornecedores de credencial padrão. Para ter mais informações sobre chaves de acesso da AWS, consulte Gerenciar chaves de acesso para usuários.

> **Note**
>
> Não é possível usar um registro DNS personalizado do Route 53 em vez do endpoint do da instância de banco de dados para gerar o token de autenticação.

Após criar uma instância de `RdsIamAuthTokenGenerator`, você pode chamar o método `getAuthToken` para obter um token assinado. Forneça a região da AWS, o nome do host, o número da porta e o nome do usuário. O exemplo de código a seguir ilustra como fazer isso.

```
package com.amazonaws.codesamples;

import com.amazonaws.auth.DefaultAWSCredentialsProviderChain;
import com.amazonaws.services.rds.auth.GetIamAuthTokenRequest;
```

```
import com.amazonaws.services.rds.auth.RdsIamAuthTokenGenerator;

public class GenerateRDSAuthToken {

    public static void main(String[] args) {

     String region = "us-west-2";
     String hostname = "rdsmysql.123456789012.us-west-2.rds.amazonaws.com";
     String port = "3306";
     String username = "jane_doe";

     System.out.println(generateAuthToken(region, hostname, port, username));
    }

    static String generateAuthToken(String region, String hostName, String port, String
 username) {

     RdsIamAuthTokenGenerator generator = RdsIamAuthTokenGenerator.builder()
      .credentials(new DefaultAWSCredentialsProviderChain())
      .region(region)
      .build();

     String authToken = generator.getAuthToken(
      GetIamAuthTokenRequest.builder()
      .hostname(hostName)
      .port(Integer.parseInt(port))
      .userName(username)
      .build());

     return authToken;
    }

}
```

Criar manualmente um token de autenticação do IAM

No Java, o modo mais fácil de gerar um token de autenticação é usar o `RdsIamAuthTokenGenerator`. Essa classe cria um token de autenticação para você e assina-o usando o Signature da AWS versão 4. Para ter mais informações, consulte Processo de assinatura do Signature versão 4 na Referência geral da AWS.

No entanto, você também pode construir e assinar um token de autenticação manualmente, conforme mostrado no exemplo de código a seguir.

```
package com.amazonaws.codesamples;

import com.amazonaws.SdkClientException;
import com.amazonaws.auth.DefaultAWSCredentialsProviderChain;
import com.amazonaws.auth.SigningAlgorithm;
import com.amazonaws.util.BinaryUtils;
import org.apache.commons.lang3.StringUtils;

import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;
import java.nio.charset.Charset;
import java.security.MessageDigest;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.SortedMap;
import java.util.TreeMap;

import static com.amazonaws.auth.internal.SignerConstants.AWS4_TERMINATOR;
import static com.amazonaws.util.StringUtils.UTF8;

public class CreateRDSAuthTokenManually {
    public static String httpMethod = "GET";
    public static String action = "connect";
    public static String canonicalURIParameter = "/";
    public static SortedMap<String, String> canonicalQueryParameters = new TreeMap();
    public static String payload = StringUtils.EMPTY;
    public static String signedHeader = "host";
    public static String algorithm = "AWS4-HMAC-SHA256";
    public static String serviceName = "rds-db";
    public static String requestWithoutSignature;

    public static void main(String[] args) throws Exception {

        String region = "us-west-2";
        String instanceName = "rdsmysql.123456789012.us-west-2.rds.amazonaws.com";
        String port = "3306";
        String username = "jane_doe";

        Date now = new Date();
        String date = new SimpleDateFormat("yyyyMMdd").format(now);
        String dateTimeStamp = new
 SimpleDateFormat("yyyyMMdd'T'HHmmss'Z'").format(now);
```

```
       DefaultAWSCredentialsProviderChain creds = new
DefaultAWSCredentialsProviderChain();
    String awsAccessKey = creds.getCredentials().getAWSAccessKeyId();
    String awsSecretKey = creds.getCredentials().getAWSSecretKey();
       String expiryMinutes = "900";

       System.out.println("Step 1:  Create a canonical request:");
       String canonicalString = createCanonicalString(username, awsAccessKey, date,
dateTimeStamp, region, expiryMinutes, instanceName, port);
       System.out.println(canonicalString);
       System.out.println();

       System.out.println("Step 2:  Create a string to sign:");
       String stringToSign = createStringToSign(dateTimeStamp, canonicalString,
awsAccessKey, date, region);
       System.out.println(stringToSign);
       System.out.println();

       System.out.println("Step 3:  Calculate the signature:");
       String signature = BinaryUtils.toHex(calculateSignature(stringToSign,
newSigningKey(awsSecretKey, date, region, serviceName)));
       System.out.println(signature);
       System.out.println();

       System.out.println("Step 4:  Add the signing info to the request");

       System.out.println(appendSignature(signature));
       System.out.println();

   }

   //Step 1: Create a canonical request date should be in format YYYYMMDD and dateTime
should be in format YYYYMMDDTHHMMSSZ
   public static String createCanonicalString(String user, String accessKey, String
date, String dateTime, String region, String expiryPeriod, String hostName, String
port) throws Exception {
       canonicalQueryParameters.put("Action", action);
       canonicalQueryParameters.put("DBUser", user);
       canonicalQueryParameters.put("X-Amz-Algorithm", "AWS4-HMAC-SHA256");
       canonicalQueryParameters.put("X-Amz-Credential", accessKey + "%2F" + date +
"%2F" + region + "%2F" + serviceName + "%2Faws4_request");
       canonicalQueryParameters.put("X-Amz-Date", dateTime);
       canonicalQueryParameters.put("X-Amz-Expires", expiryPeriod);
       canonicalQueryParameters.put("X-Amz-SignedHeaders", signedHeader);
```

```
        String canonicalQueryString = "";
        while(!canonicalQueryParameters.isEmpty()) {
            String currentQueryParameter = canonicalQueryParameters.firstKey();
            String currentQueryParameterValue =
 canonicalQueryParameters.remove(currentQueryParameter);
            canonicalQueryString = canonicalQueryString + currentQueryParameter + "=" +
 currentQueryParameterValue;
            if (!currentQueryParameter.equals("X-Amz-SignedHeaders")) {
                canonicalQueryString += "&";
            }
        }
        String canonicalHeaders = "host:" + hostName + ":" + port + '\n';
        requestWithoutSignature = hostName + ":" + port + "/?" + canonicalQueryString;

        String hashedPayload = BinaryUtils.toHex(hash(payload));
        return httpMethod + '\n' + canonicalURIParameter + '\n' + canonicalQueryString
 + '\n' + canonicalHeaders + '\n' + signedHeader + '\n' + hashedPayload;

    }

    //Step 2: Create a string to sign using sig v4
    public static String createStringToSign(String dateTime, String canonicalRequest,
 String accessKey, String date, String region) throws Exception {
        String credentialScope = date + "/" + region + "/" + serviceName + "/
aws4_request";
        return algorithm + '\n' + dateTime + '\n' + credentialScope + '\n' +
 BinaryUtils.toHex(hash(canonicalRequest));

    }

    //Step 3: Calculate signature
    /**
     * Step 3 of the &AWS; Signature version 4 calculation. It involves deriving
     * the signing key and computing the signature. Refer to
     * http://docs.aws.amazon
     * .com/general/latest/gr/sigv4-calculate-signature.html
     */
    public static byte[] calculateSignature(String stringToSign,
                                            byte[] signingKey) {
        return sign(stringToSign.getBytes(Charset.forName("UTF-8")), signingKey,
                SigningAlgorithm.HmacSHA256);
    }

    public static byte[] sign(byte[] data, byte[] key,
```

```
                        SigningAlgorithm algorithm) throws SdkClientException {
        try {
            Mac mac = algorithm.getMac();
            mac.init(new SecretKeySpec(key, algorithm.toString()));
            return mac.doFinal(data);
        } catch (Exception e) {
            throw new SdkClientException(
                    "Unable to calculate a request signature: "
                            + e.getMessage(), e);
        }
    }

    public static byte[] newSigningKey(String secretKey,
                                   String dateStamp, String regionName, String
serviceName) {
        byte[] kSecret = ("AWS4" + secretKey).getBytes(Charset.forName("UTF-8"));
        byte[] kDate = sign(dateStamp, kSecret, SigningAlgorithm.HmacSHA256);
        byte[] kRegion = sign(regionName, kDate, SigningAlgorithm.HmacSHA256);
        byte[] kService = sign(serviceName, kRegion,
                SigningAlgorithm.HmacSHA256);
        return sign(AWS4_TERMINATOR, kService, SigningAlgorithm.HmacSHA256);
    }

    public static byte[] sign(String stringData, byte[] key,
                      SigningAlgorithm algorithm) throws SdkClientException {
        try {
            byte[] data = stringData.getBytes(UTF8);
            return sign(data, key, algorithm);
        } catch (Exception e) {
            throw new SdkClientException(
                    "Unable to calculate a request signature: "
                            + e.getMessage(), e);
        }
    }

    //Step 4: append the signature
    public static String appendSignature(String signature) {
        return requestWithoutSignature + "&X-Amz-Signature=" + signature;
    }

    public static byte[] hash(String s) throws Exception {
        try {
            MessageDigest md = MessageDigest.getInstance("SHA-256");
            md.update(s.getBytes(UTF8));
```

```
            return md.digest();
        } catch (Exception e) {
            throw new SdkClientException(
                    "Unable to compute hash while signing request: "
                            + e.getMessage(), e);
        }
    }
}
```

Conexão à instância de banco de dados

O exemplo de código a seguir mostra como gerar um token de autenticação e usá-lo para se conectar a uma instância que esteja executando o MariaDB ou o MySQL.

Para executar esse exemplo de código, você precisa do AWS SDK para Java, encontrado no site AWS. Além disso, você precisa do seguinte:

- MySQL Connector/J. Este exemplo de código foi testado com `mysql-connector-java-5.1.33-bin.jar`.
- Um certificado intermediário do Amazon RDS específico de uma região da AWS. (Para ter mais informações, consulte Usar SSL/TLS para criptografar uma conexão com uma instância ou um cluster de banco de dados.) No tempo de execução, o carregador de classe procura o certificado no mesmo diretório do exemplo de código Java para que possa encontrá-lo.
- Modifique os valores das seguintes variáveis, conforme necessário:
  - `RDS_INSTANCE_HOSTNAME`: o nome do host da instância de banco de dados que você deseja acessar.
  - `RDS_INSTANCE_PORT`: o número da porta usado na conexão com a instância de banco de dados PostgreSQL.
  - `REGION_NAME`: a região da AWS na qual a instância do banco de dados está em execução.
  - `DB_USER`: a conta de banco de dados que você deseja acessar.
  - `SSL_CERTIFICATE`: um certificado SSL para o Amazon RDS que é específico de uma região da AWS.

    Para baixar o certificado para a sua região da AWS, consulte Usar SSL/TLS para criptografar uma conexão com uma instância ou um cluster de banco de dados. Coloque o certificado do SSL no mesmo diretório que esse arquivo do programa Java para que o carregador de classe possa encontrar o certificado no tempo de execução.

Esse exemplo de código obtém as credenciais da AWS a partir dacadeia de fornecedores de credencial padrão.

> **Note**
>
> Especifique uma senha para `DEFAULT_KEY_STORE_PASSWORD` diferente do prompt mostrado aqui como prática recomendada de segurança.

```
package com.amazonaws.samples;

import com.amazonaws.services.rds.auth.RdsIamAuthTokenGenerator;
import com.amazonaws.services.rds.auth.GetIamAuthTokenRequest;
import com.amazonaws.auth.BasicAWSCredentials;
import com.amazonaws.auth.DefaultAWSCredentialsProviderChain;
import com.amazonaws.auth.AWSStaticCredentialsProvider;

import java.io.File;
import java.io.FileOutputStream;
import java.io.InputStream;
import java.security.KeyStore;
import java.security.cert.CertificateFactory;
import java.security.cert.X509Certificate;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import java.util.Properties;

import java.net.URL;

public class IAMDatabaseAuthenticationTester {
    //&AWS; Credentials of the IAM user with policy enabling IAM Database Authenticated
 access to the db by the db user.
    private static final DefaultAWSCredentialsProviderChain creds = new
 DefaultAWSCredentialsProviderChain();
    private static final String AWS_ACCESS_KEY =
 creds.getCredentials().getAWSAccessKeyId();
    private static final String AWS_SECRET_KEY =
 creds.getCredentials().getAWSSecretKey();
```

```
    //Configuration parameters for the generation of the IAM Database Authentication
 token
    private static final String RDS_INSTANCE_HOSTNAME = "rdsmysql.123456789012.us-
west-2.rds.amazonaws.com";
    private static final int RDS_INSTANCE_PORT = 3306;
    private static final String REGION_NAME = "us-west-2";
    private static final String DB_USER = "jane_doe";
    private static final String JDBC_URL = "jdbc:mysql://" + RDS_INSTANCE_HOSTNAME +
 ":" + RDS_INSTANCE_PORT;

    private static final String SSL_CERTIFICATE = "rds-ca-2019-us-west-2.pem";

    private static final String KEY_STORE_TYPE = "JKS";
    private static final String KEY_STORE_PROVIDER = "SUN";
    private static final String KEY_STORE_FILE_PREFIX = "sys-connect-via-ssl-test-
cacerts";
    private static final String KEY_STORE_FILE_SUFFIX = ".jks";
    private static final String DEFAULT_KEY_STORE_PASSWORD = "changeit";

    public static void main(String[] args) throws Exception {
        //get the connection
        Connection connection = getDBConnectionUsingIam();

        //verify the connection is successful
        Statement stmt= connection.createStatement();
        ResultSet rs=stmt.executeQuery("SELECT 'Success!' FROM DUAL;");
        while (rs.next()) {
             String id = rs.getString(1);
            System.out.println(id); //Should print "Success!"
        }

        //close the connection
        stmt.close();
        connection.close();

        clearSslProperties();

    }

    /**
     * This method returns a connection to the db instance authenticated using IAM
 Database Authentication
     * @return
     * @throws Exception
```

```
     */
    private static Connection getDBConnectionUsingIam() throws Exception {
        setSslProperties();
        return DriverManager.getConnection(JDBC_URL, setMySqlConnectionProperties());
    }

    /**
     * This method sets the mysql connection properties which includes the IAM Database
 Authentication token
     * as the password. It also specifies that SSL verification is required.
     * @return
     */
    private static Properties setMySqlConnectionProperties() {
        Properties mysqlConnectionProperties = new Properties();
        mysqlConnectionProperties.setProperty("verifyServerCertificate","true");
        mysqlConnectionProperties.setProperty("useSSL", "true");
        mysqlConnectionProperties.setProperty("user",DB_USER);
        mysqlConnectionProperties.setProperty("password",generateAuthToken());
        return mysqlConnectionProperties;
    }

    /**
     * This method generates the IAM Auth Token.
     * An example IAM Auth Token would look like follows:
     * btusi123.cmz7kenwo2ye.rds.cn-north-1.amazonaws.com.cn:3306/?
Action=connect&DBUser=iamtestuser&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-
Date=20171003T010726Z&X-Amz-SignedHeaders=host&X-Amz-Expires=899&X-Amz-
Credential=AKIAPFXHGVDI5RNFO4AQ%2F20171003%2Fcn-north-1%2Frds-db%2Faws4_request&X-Amz-
Signature=f9f45ef96c1f770cdad11a53e33ffa4c3730bc03fdee820cfdf1322eed15483b
     * @return
     */
    private static String generateAuthToken() {
        BasicAWSCredentials awsCredentials = new BasicAWSCredentials(AWS_ACCESS_KEY,
 AWS_SECRET_KEY);

        RdsIamAuthTokenGenerator generator = RdsIamAuthTokenGenerator.builder()
                .credentials(new
 AWSStaticCredentialsProvider(awsCredentials)).region(REGION_NAME).build();
        return generator.getAuthToken(GetIamAuthTokenRequest.builder()

 .hostname(RDS_INSTANCE_HOSTNAME).port(RDS_INSTANCE_PORT).userName(DB_USER).build());
    }

    /**
```

```
    * This method sets the SSL properties which specify the key store file, its type
and password:
    * @throws Exception
    */
   private static void setSslProperties() throws Exception {
       System.setProperty("javax.net.ssl.trustStore", createKeyStoreFile());
       System.setProperty("javax.net.ssl.trustStoreType", KEY_STORE_TYPE);
       System.setProperty("javax.net.ssl.trustStorePassword",
DEFAULT_KEY_STORE_PASSWORD);
   }

   /**
    * This method returns the path of the Key Store File needed for the SSL
verification during the IAM Database Authentication to
    * the db instance.
    * @return
    * @throws Exception
    */
   private static String createKeyStoreFile() throws Exception {
       return createKeyStoreFile(createCertificate()).getPath();
   }

   /**
    *  This method generates the SSL certificate
    * @return
    * @throws Exception
    */
   private static X509Certificate createCertificate() throws Exception {
       CertificateFactory certFactory = CertificateFactory.getInstance("X.509");
       URL url = new File(SSL_CERTIFICATE).toURI().toURL();
       if (url == null) {
           throw new Exception();
       }
       try (InputStream certInputStream = url.openStream()) {
           return (X509Certificate) certFactory.generateCertificate(certInputStream);
       }
   }

   /**
    * This method creates the Key Store File
    * @param rootX509Certificate - the SSL certificate to be stored in the KeyStore
    * @return
    * @throws Exception
    */
```

```
    private static File createKeyStoreFile(X509Certificate rootX509Certificate) throws
 Exception {
        File keyStoreFile = File.createTempFile(KEY_STORE_FILE_PREFIX,
 KEY_STORE_FILE_SUFFIX);
        try (FileOutputStream fos = new FileOutputStream(keyStoreFile.getPath())) {
            KeyStore ks = KeyStore.getInstance(KEY_STORE_TYPE, KEY_STORE_PROVIDER);
            ks.load(null);
            ks.setCertificateEntry("rootCaCertificate", rootX509Certificate);
            ks.store(fos, DEFAULT_KEY_STORE_PASSWORD.toCharArray());
        }
        return keyStoreFile;
    }

    /**
     * This method clears the SSL properties.
     * @throws Exception
     */
    private static void clearSslProperties() throws Exception {
           System.clearProperty("javax.net.ssl.trustStore");
           System.clearProperty("javax.net.ssl.trustStoreType");
           System.clearProperty("javax.net.ssl.trustStorePassword");
    }

}
```

Se você quiser se conectar a uma instância de banco de dados por meio de um proxy, consulte Conectar-se a um proxy usando autenticação do IAM.

Conectar-se à instância de banco de dados usando a autenticação do IAM e o AWS SDK para Python (Boto3)

Você pode se conectar a uma instância de banco de dados do RDS para MariaDB, RDS para MySQL ou RDS para PostgreSQL com a AWS SDK para Python (Boto3), conforme descrito a seguir.

Pré-requisitos

Veja a seguir os pré-requisitos para se conectar à instância de banco de dados usando a autenticação do IAM:

- Habilitar e desabilitar a autenticação de banco de dados do IAM
- Criar e usar uma política do IAM para acesso do banco de dados do IAM
- Criar uma conta de banco de dados usando autenticação do IAM

Além disso, verifique se as bibliotecas importadas no código de exemplo existem no sistema.

Exemplos

Os exemplos de código usam perfis para credenciais compartilhadas. Para obter informações sobre a especificação de credenciais, consulte Credenciais na documentação AWS SDK para Python (Boto3).

O exemplo de código a seguir mostra como gerar um token de autenticação e usá-lo para se conectar a uma instância de banco de dados.

Para executar esse exemplo de código, você precisa do AWS SDK para Python (Boto3), encontrado no site AWS.

Modifique os valores das seguintes variáveis, conforme necessário:

- `ENDPOINT`: o endpoint da instância de banco de dados que você deseja acessar
- `PORT`: o número da porta usada para se conectar à instância de banco de dados
- `USER`: a conta de banco de dados que você deseja acessar
- `REGION`: a região da AWS na qual a instância do banco de dados está em execução.
- `DBNAME`: o banco de dados que você deseja acessar
- `SSLCERTIFICATE`: o caminho completo para o certificado SSL do Amazon RDS

  Para `ssl_ca`, defina um certificado SSL. Para baixar um certificado SSL, consulte Usar SSL/TLS para criptografar uma conexão com uma instância ou um cluster de banco de dados.

> **Note**
>
> Não é possível usar um registro DNS personalizado do Route 53 em vez do endpoint do da instância de banco de dados para gerar o token de autenticação.

Esse código se conecta a uma instância de banco de dados MariaDB ou MySQL.

Antes de executar esse código, instale o driver PyMySQL seguindo as instruções no Python Package Index.

```
import pymysql
import sys
```

```
import boto3
import os

ENDPOINT="mysqldb.123456789012.us-east-1.rds.amazonaws.com"
PORT="3306"
USER="jane_doe"
REGION="us-east-1"
DBNAME="mydb"
os.environ['LIBMYSQL_ENABLE_CLEARTEXT_PLUGIN'] = '1'

#gets the credentials from .aws/credentials
session = boto3.Session(profile_name='default')
client = session.client('rds')

token = client.generate_db_auth_token(DBHostname=ENDPOINT, Port=PORT, DBUsername=USER,
 Region=REGION)

try:
    conn =
 pymysql.connect(auth_plugin_map={'mysql_clear_password':None},host=ENDPOINT,
 user=USER, password=token, port=PORT, database=DBNAME, ssl_ca='SSLCERTIFICATE',
 ssl_verify_identity=True, ssl_verify_cert=True)
    cur = conn.cursor()
    cur.execute("""SELECT now()""")
    query_results = cur.fetchall()
    print(query_results)
except Exception as e:
    print("Database connection failed due to {}".format(e))
```

Esse código se conecta a uma instância de banco de dados PostgreSQL.

Antes de executar esse código, instale `psycopg2`, seguindo as instruções na documentação de Psycopg.

```
import psycopg2
import sys
import boto3
import os

ENDPOINT="postgresmydb.123456789012.us-east-1.rds.amazonaws.com"
PORT="5432"
```

```
USER="jane_doe"
REGION="us-east-1"
DBNAME="mydb"

#gets the credentials from .aws/credentials
session = boto3.Session(profile_name='RDSCreds')
client = session.client('rds')

token = client.generate_db_auth_token(DBHostname=ENDPOINT, Port=PORT, DBUsername=USER,
 Region=REGION)

try:
    conn = psycopg2.connect(host=ENDPOINT, port=PORT, database=DBNAME, user=USER,
 password=token, sslrootcert="SSLCERTIFICATE")
    cur = conn.cursor()
    cur.execute("""SELECT now()""")
    query_results = cur.fetchall()
    print(query_results)
except Exception as e:
    print("Database connection failed due to {}".format(e))
```

Se você quiser se conectar a uma instância de banco de dados por meio de um proxy, consulte Conectar-se a um proxy usando autenticação do IAM.

Solucionar problemas referentes à autenticação de banco de dados do IAM

A seguir, é possível encontrar ideias de solução de problemas para alguns problemas comuns de autenticação de banco de dados do IAM e informações sobre logs e métricas do CloudWatch para autenticação de banco de dados do IAM.

Exportar logs de erros de autenticação de banco de dados do IAM para o CloudWatch Logs

Os logs de erros de autenticação de banco de dados do IAM são armazenados no host do banco de dados e podem ser exportados para sua conta do CloudWatch Logs. Use os logs e os métodos de correção nesta página para solucionar problemas de autenticação de banco de dados do IAM.

Você pode habilitar as exportações de log para o CloudWatch Logs por meio do console, da AWS CLI e da API do RDS. Para obter instruções sobre o console, consulte Publicação de logs de banco de dados no Amazon CloudWatch Logs.

Para exportar logs de erros de autenticação de banco de dados do IAM para o CloudWatch Logs ao criar uma instância de banco de dados por meio da AWS CLI, use o seguinte comando:

```
aws rds create-db-instance --db-instance-identifier mydbinstance \
--region us-east-1 \
--db-instance-class db.t3.large \
--allocated-storage 50 \
--engine postgres \
--engine-version 16 \
--port 5432 \
--master-username master \
--master-user-password password \
--publicly-accessible \
--enable-iam-database-authentication \
--enable-cloudwatch-logs-exports=iam-db-auth-error
```

Para exportar logs de erros de autenticação de banco de dados do IAM para o CloudWatch Logs ao modificar uma instância de banco de dados por meio da AWS CLI, use o seguinte comando:

```
aws rds modify-db-instance --db-instance-identifier mydbinstance \
--region us-east-1 \
--cloudwatch-logs-export-configuration '{"EnableLogTypes":["iam-db-auth-error"]}'
```

Para verificar se a instância de banco de dados está exportando logs de autenticação de banco de dados do IAM para o CloudWatch Logs, confira se o parâmetro `EnabledCloudwatchLogsExports` está definido como `iam-db-auth-error` na saída do comando `describe-db-instances`.

```
aws rds describe-db-instances --region us-east-1 --db-instance-identifier mydbinstance
            ...

             "EnabledCloudwatchLogsExports": [
                "iam-db-auth-error"
            ],
            ...
```

Métricas do CloudWatch de autenticação de banco de dados do IAM

O Amazon RDS fornece métricas quase em tempo real sobre a autenticação de banco de dados do IAM para sua conta do Amazon CloudWatch. A seguinte tabela lista as métricas de autenticação de banco de dados do IAM disponíveis usando o CloudWatch:

Métrica	Descrição
`IamDbAuthConnectionRequests`	Número total de solicitações de conexão feitas com a autenticação de banco de dados do IAM.
`IamDbAuthConnectionSuccess`	Número total de solicitações de autenticação de banco de dados do IAM bem-sucedidas.
`IamDbAuthConnectionFailure`	Número total de solicitações de autenticação de banco de dados do IAM malsucedidas.
`IamDbAuthConnectionFailureInvalidToken`	Número total de solicitações de autenticação de banco de dados do IAM malsucedidas devido a token inválido.
`IamDbAuthConnectionFailureInsufficientPermissions`	Número total de solicitações de autenticação de banco de dados do IAM malsucedidas devido a políticas ou permissões incorretas.
`IamDbAuthConnectionFailureThrottling`	Número total de solicitações de autenticação de banco de dados do IAM malsucedidas devido ao controle de utilização da autenticação de banco de dados do IAM.
`IamDbAuthConnectionFailureServerError`	Número total de solicitações de autenticação de banco de dados do IAM malsucedidas devido a um erro interno do servidor no recurso de autenticação de banco de dados do IAM.

Problemas e soluções comuns de

Você pode encontrar os problemas a seguir ao usar a autenticação de banco de dados do IAM. Use as etapas de correção na tabela para resolver os problemas:

Erro	Métrica(s)	Causa	Solução

`[ERROR] Failed to authenticate the connection request for user db_user because the provided token is malformed or otherwise invalid. (Status Code: 400, Error Code: InvalidToken)`	`IamDbAuthConnectionFailure`
	
	`IamDbAuthConnectionFailureInvalidToken`
		O token de autenticação de banco de dados do IAM na solicitação de conexão não é um token SigV4a válido ou não está formatado corretamente.	Verifique a estratégia de geração de tokens em sua aplicação. Em alguns casos, observe se você está passando o token com uma formatação válida. Truncar o token (ou formatar incorretamente a string) tornará o token inválido.

`[ERROR] Failed to authenticate the connection request for user db_user because the token age is longer than 15 minutes. (Status Code: 400, Error Code:ExpiredToken)`	`IamDbAuthConnectionFailure`
	
	`IamDbAuthConnectionFailureInvalidToken`
		O token de autenticação de banco de dados do IAM expirou. Os tokens são válidos somente por 15 minutos.	Verifique a lógica de armazenamento em cache e/ou a reutilização de tokens em sua aplicação. Você não deve reutilizar tokens com mais de 15 minutos.

`[ERROR] Failed to authorize the connection request for user db_user because the IAM policy assumed by the caller 'arn:aws:`	`IamDbAuthConnectionFailure`
	
	`IamDbAuthConnectionFailureInsufficientPermissions`
		Esse erro pode ocorrer devido aos seguintes motivos:
		
		• A política do IAM assumida pela aplicação não autoriza a
			Verifique o perfil e/ou a política do IAM que você está assumindo em sua aplicação. Você deve assumir a mesma política para gerar o token e para se conectar ao banco de dados.

Erro	Métrica(s)	Causa	Solução
`sts::1234 56789012: assumed-r ole/ <RoleName >/ <RoleSess ion>' is not authorized to perform `rds-db:connect` on the DB instance. (Status Code: 403, Error Code:NotA uthorized)`		ação `rds-db:co nnect` . • Você está assumindo o perfil/ política incorreto s para *db_user* para se conectar ao banco de dados. • Você está assumindo a política correta para *db_user*, mas não está se conectando ao banco de dados correto.	

`[ERROR] Failed to authorize the connection request for user *db_user* due to IAM DB authentic ation throttlin g. (Status Code: 429, Error Code: Throttlin gException)`	`IamDbAuth Connectio nFailure`
	
	`IamDbAuth Connectio nFailureT hrottling`
		Você está fazendo muitas solicitações de conexão ao banco de dados em um curto espaço de tempo. O limite de controle de utilização da autentica ção de banco de dados do IAM é duzentas conexões por segundo.	Reduza a taxa de estabelecimento de novas conexões por meio da autenticação do IAM. Considere implementar o agrupamento de conexões usando o RDS Proxy para reutilizar as conexões estabelecidas em sua aplicação.

Erro	Métrica(s)	Causa	Solução

`[ERROR] Failed to authorize the connection request for user db_user due to an internal IAM DB authentication error. (Status Code: 500, Error Code: InternalError)`	`IamDbAuthConnectionFailure`
	
	`IamDbAuthConnectionFailureThrottling`
		Houve um erro interno ao autorizar a conexão do banco de dados com a autenticação de banco de dados do IAM.	Entre em contato com https://aws.amazon.com/premiumsupport/ para investigar o problema.

Solução de problemas de identidade e acesso do Amazon RDS

Use as seguintes informações para ajudar a diagnosticar e corrigir problemas comuns que podem ser encontrados ao trabalhar com o Amazon RDS e o IAM.

Tópicos

- Não estou autorizado a realizar uma ação no Amazon RDS
- Não estou autorizado a executar iam:PassRole
- Quero permitir que pessoas fora de minha conta da AWS acessem meus recursos do Amazon RDS

Não estou autorizado a realizar uma ação no Amazon RDS

Se o AWS Management Console informar que você não está autorizado a executar uma ação, você deverá entrar em contato com o administrador para obter assistência. Seu administrador é a pessoa que forneceu a você suas credenciais de início de sessão.

O exemplo de erro a seguir ocorre quando o usuário `mateojackson` tenta usar o console para visualizar detalhes sobre um *`widget`*, mas não tem as permissões `rds:`*`GetWidget`*.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 rds:GetWidget on resource: my-example-widget
```

Neste caso, Mateo pede ao administrador para atualizar suas políticas para permitir a ele o acesso ao recurso *my-example-widget* usando a ação `rds:GetWidget`.

Não estou autorizado a executar iam:PassRole

Se você receber uma mensagem de erro informando que você não está autorizado a executar a ação `iam:PassRole`, entre em contato com o administrador para obter assistência. Seu administrador é a pessoa que forneceu a você suas credenciais de início de sessão. Peça a essa pessoa para atualizar suas políticas para permitir que você passe uma função para o Amazon RDS.

Alguns serviços da AWS permitem que você passe uma função existente para o serviço, em vez de criar um novo perfil de serviço ou perfil vinculado ao serviço. Para fazer isso, um usuário deve ter permissões para passar o perfil para o serviço.

O erro de exemplo a seguir ocorre quando uma usuária chamada `marymajor` tenta usar o console para executar uma ação no Amazon RDS. No entanto, a ação exige que o serviço tenha permissões concedidas por um perfil de serviço. Mary não tem permissões para passar a função para o serviço.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

Neste caso, Mary pede ao administrador para atualizar suas políticas para permitir que ela execute a ação `iam:PassRole`.

Quero permitir que pessoas fora de minha conta da AWS acessem meus recursos do Amazon RDS

Você pode criar uma função que os usuários de outras contas ou pessoas fora da sua organização podem usar para acessar seus recursos. Você pode especificar quem é confiável para assumir a função. Para serviços que oferecem suporte a políticas baseadas em recursos ou listas de controle de acesso (ACLs), você pode usar essas políticas para conceder às pessoas acesso aos seus recursos.

Para saber mais, consulte o seguinte:

- Para saber se o Amazon RDS oferece suporte a esses recursos, consulte Como o Amazon RDS funciona com o IAM.
- Para saber como conceder acesso a seus recursos em todas as contas da AWS pertencentes a você, consulte Fornecimento de acesso a um usuário do IAM em outra conta da AWS pertencente a você no Guia de usuário do IAM.

- Para saber como conceder acesso aos recursos para contas da AWS de terceiros, consulte Fornecer acesso a contas da AWS pertencentes a terceiros no Guia do usuário do IAM.
- Para saber como conceder acesso por meio da federação de identidades, consulte Conceder acesso a usuários autenticados externamente (federação de identidades) no Guia do usuário do IAM.
- Para saber a diferença entre usar perfis e políticas baseadas em recursos para acesso entre contas, consulte Como os perfis do IAM diferem de políticas baseadas em recursos no Guia do usuário do IAM.

Registrar em log e monitorar no Amazon RDS

O monitoramento é uma parte importante da manutenção da confiabilidade, da disponibilidade e da performance do Amazon RDS e de suas soluções da AWS. É necessário coletar dados de monitoramento de todas as partes de sua solução da AWS para depurar uma falha de vários pontos com maior facilidade, caso ocorra. A AWS oferece várias ferramentas para monitorar seus recursos do Amazon RDS e responder a possíveis incidentes:

Alarmes do Amazon CloudWatch

Com o uso de alarmes do Amazon CloudWatch, você observa uma única métrica durante um período especificado. Se a métrica exceder determinado limite, uma notificação será enviada para um tópico do Amazon SNS ou para uma política do AWS Auto Scaling. Os alarmes do CloudWatch não invocam ações só porque estão em um determinado estado. O estado deve ter sido alterado e mantido por uma quantidade especificada de períodos.

AWS CloudTrailLogs do

O CloudTrail fornece um registro de ações executadas por um usuário, perfil ou serviço da AWS no Amazon RDS. O CloudTrail captura todas as chamadas de API para o Amazon RDS como eventos, inclusive as chamadas do console e de chamadas do código para operações de API do Amazon RDS. Usando as informações coletadas pelo CloudTrail, é possível determinar a solicitação que foi feita ao Amazon RDS, o endereço IP da solicitação, quem fez a solicitação, quando ela foi feita e outros detalhes. Para obter mais informações, consulte Monitorar chamadas de API do Amazon RDSno AWS CloudTrail.

Monitoramento avançado

O Amazon RDS fornece métricas em tempo real para o sistema operacional (SO) no qual a instância de banco de dados é executado. Você pode visualizar as métricas dainstância de banco

de dados usando o console ou consumir o resultado do JSON de monitoramento avançado do Amazon CloudWatch Logs em um sistema de monitoramento de sua escolha. Para obter mais informações, consulte Monitorar métricas do SO com o monitoramento avançado.

Insights de Performance do Amazon RDS

O Insights de Performance expande os recursos de monitoramento existentes do Amazon RDS para ilustrar a performance do banco de dados e ajudar a analisar todos os problemas que o afetam. Com o painel do Performance Insights, você pode visualizar a carga do banco de dados e filtrá-la por esperas, instruções SQL, hosts ou usuários. Para obter mais informações, consulte Monitorar a carga de banco de dados com o Performance Insights no Amazon RDS.

Logs de banco de dados

Você pode visualizar, baixar e observar os logs de banco de dados usando o AWS Management Console, a AWS CLI ou a API do RDS. Para obter mais informações, consulte Monitorar arquivos de log do Amazon RDS.

Recomendações do Amazon RDS

O Amazon RDS oferece recomendações automatizadas para recursos de banco de dados. Essas recomendações fornecem orientações de práticas recomendadas analisando a configuração, o uso e os dados de performance da instância de banco de dados. Para obter mais informações, consulte Recomendações do Amazon RDS.

Notificação de eventos do Amazon RDS

OAmazon RDS usa o Amazon Simple Notification Service (Amazon SNS) para fornecer uma notificação quando um evento do Amazon RDS ocorre. Essas notificações podem estar em qualquer formato de notificação compatível com o Amazon SNS para uma região da AWS, como um e-mail, uma mensagem de texto ou uma chamada para um endpoint HTTP. Para obter mais informações, consulte Trabalhar com a notificação de eventos do Amazon RDS.

AWS Trusted Advisor

O Trusted Advisor conta com as práticas recomendadas aprendidas com o atendimento a centenas de milhões de clientes da AWS. O Trusted Advisor inspeciona seu ambiente da AWS e faz recomendações quando há oportunidades para economizar dinheiro, melhorar a performance e a disponibilidade do sistema e ajuda a corrigir falhas de segurança. Todos os clientes da AWS têm acesso a cinco verificações do Trusted Advisor. Os clientes com um plano de suporte Business ou Enterprise podem ver todas as verificações do Trusted Advisor.

O Trusted Advisor tem as seguintes verificações relacionadas ao Amazon RDS:

- Instâncias ociosas de banco de dados do Amazon RDS
- Risco de acesso de grupo de segurança do Amazon RDS
- Backups do Amazon RDS
- Multi-az do Amazon RDS

Para obter mais informações sobre essas verificações, consulte Práticas recomendadas do Trusted Advisor (verificações).

Para ter mais informações sobre monitoramento do Amazon RDS, consulte Métricas de monitoramento em uma instância do Amazon RDS.

Validação de conformidade do Amazon RDS

Auditores de terceiros avaliam a segurança e a compatibilidade do Amazon RDS como parte de vários programas de conformidade da AWS. Isso inclui SOC, PCI, FedRAMP, HIPAA e outros.

Para obter uma lista dos serviços da AWS no escopo de programas de conformidade específicos, consulte Serviços da AWS no escopo por programa de conformidade. Para obter informações gerais, consulte Programas de conformidade da AWS.

É possível baixar relatórios de auditoria de terceiros usando o AWS Artifact. Para obter mais informações, consulte Baixar os relatórios no AWS Artifact.

Sua responsabilidade com relação à conformidade ao usar o Amazon RDS é determinada pela confidencialidade dos dados, pelos objetivos de compatibilidade da organização e pelos regulamentos e leis aplicáveis. A AWS fornece os seguintes recursos para ajudar com a compatibilidade:

- Guias de início rápido de segurança e conformidade: esses guias de implantação abordam as considerações de arquitetura e fornecem etapas para a implantação de ambientes de linha de base concentrados em conformidade e segurança na AWS.
- Architecting for HIPAA Security and Compliance on Amazon Web Services (Arquitetura para segurança e conformidade com HIPAA na Amazon Web Services): esse artigo técnico descreve como as empresas podem usar a AWS para criar aplicações em conformidade com os padrões HIPAA.
- Recursos de conformidade da AWS: esta coleção de manuais e guias pode ser aplicada ao seu setor e local.
- AWS Config: este produto da AWS avalia até que ponto suas configurações de recursos atendem adequadamente às práticas internas e às diretrizes e regulamentações do setor.
- AWS Security Hub: este AWS service (Serviço da AWS) fornece uma visão abrangente do seu estado de segurança na AWS. O Security Hub usa controles de segurança para avaliar os recursos da AWS e verificar a conformidade com os padrões e as práticas recomendadas do setor de segurança. Para obter uma lista dos serviços e controles aceitos, consulte a Referência de controles do Security Hub.

Resiliência no Amazon RDS

A infraestrutura global da AWS é criada com base em regiões da AWSe zonas de disponibilidade da AWS. As regiões fornecem várias zonas de disponibilidade separadas e isoladas fisicamente, as quais são conectadas com baixa latência, alto throughput e redes altamente redundantes. Com as zonas de disponibilidade, você pode projetar e operar aplicações e bancos de dados que executam o failover automaticamente entre as zonas de disponibilidade sem interrupção. As zonas de disponibilidade são mais altamente disponíveis, tolerantes a falhas e escaláveis que uma ou várias infraestruturas de data center tradicionais.

Para obter mais informações sobre regiões e zonas de disponibilidade da AWS, consulte Infraestrutura global da AWS.

Além da infraestrutura global da AWS, o Amazon RDSoferece recursos para ajudar a oferecer suporte às suas necessidades de resiliência e backup de dados.

Backup e restauração

O Amazon RDS cria e salva os backups automáticos da instância de banco de dados. O Amazon RDS cria um snapshot do volume de armazenamento de sua instância de banco de dados, fazendo o backup de toda a instância de banco de dados, não apenas dos bancos de dados individuais.

O Amazon RDS cria backups automatizados de sua instância de banco de dados durante a janela de backup de sua instância de banco de dados. O Amazon RDS salva os backups automatizados da instância de banco de dados de acordo com o período de retenção de backup especificado. Se necessário, você poderá recuperar seu banco de dados em qualquer ponto no tempo durante o período de retenção do backup. Também faça backup da instância de banco de dados manualmente, criando manualmente um snapshot de banco de dados.

Você poderá criar uma instância de banco de dados restaurando a partir desse snapshot de banco de dados como uma solução de recuperação de desastre se a instância de banco de dados de origem falhar.

Para obter mais informações, consulte Backup, restauração e exportação de dados.

Replicação

O Amazon RDS usa a funcionalidade de replicação integrada nos mecanismos de banco de dados MariaDB, MySQL, Oracle e PostgreSQL para criar um tipo especial de instância de banco de dados

chamada de réplica de leitura de uma instância de banco de dados de origem. As atualizações feitas à instância de banco de dados de origem são copiadas de forma assíncrona na réplica de leitura. É possível reduzir a carga na instância de banco de dados de origem roteando as consultas de leitura das aplicações para a réplica de leitura. Com o uso de réplicas de leitura, é possível expandir de maneira elástica para além das limitações de capacidade de uma única instância de banco de dados para workloads de banco de dados com muita leitura. É possível promover uma réplica de leitura para uma instância autônoma como solução de recuperação de desastres, caso haja uma falha na instância de banco de dados de origem. Para alguns mecanismos de banco de dados, o Amazon RDS também oferece suporte a outras opções de replicação.

Para obter mais informações, consulte Trabalhar com réplicas de leitura de instância de banco de dados.

Failover

O Amazon RDS oferece alta disponibilidade e suporte a failover para instâncias de banco de dados usando implantações multi-AZ. O Amazon RDS usa várias tecnologias diferentes para fornecer suporte a failover. As implantações multi-AZ para instâncias de banco de dados Oracle, PostgreSQL, MySQL e MariaDB usam a tecnologia de failover da Amazon. As instâncias de banco de dados do SQL Server usam o SQL Server Database Mirroring (DBM).

Para obter mais informações, consulte Configurar e gerenciar uma implantação multi-AZ para o Amazon RDS.

Segurança da infraestrutura no Amazon RDS

Como um serviço gerenciado, o Amazon Relational Database Service é protegido pela segurança de rede global da AWS. Para obter informações sobre serviços de segurança da AWS e como a AWS protege a infraestrutura, consulte Segurança na Nuvem AWS. Para projetar seu ambiente da AWS usando as práticas recomendadas de segurança da infraestrutura, consulte Proteção de Infraestrutura em Pilar de Segurança: AWS Well-Architected Framework.

Você usa chamadas de API publicadas da AWS para acessar o Amazon RDS por meio da rede. Os clientes devem oferecer suporte para:

- Transport Layer Security (TLS). Exigimos TLS 1.2 e recomendamos TLS 1.3.
- Conjuntos de criptografia com perfect forward secrecy (PFS) como DHE (Ephemeral Diffie-Hellman) ou ECDHE (Ephemeral Elliptic Curve Diffie-Hellman). A maioria dos sistemas modernos, como Java 7 e versões posteriores, comporta esses modos.

Além disso, as solicitações devem ser assinadas usando um ID da chave de acesso e uma chave de acesso secreta associada a uma entidade principal do IAM. Ou você pode usar o AWS Security Token Service (AWS STS) para gerar credenciais de segurança temporárias para assinar solicitações.

Além disso, o Amazon RDS oferece recursos para ajudar a oferecer suporte à segurança da infraestrutura.

Grupos de segurança

Os grupos de segurança controlam o acesso que o tráfego tem dentro e fora de uma instância de banco de dados. Por padrão, o acesso à rede é desativado para uma instância de banco de dados. É possível especificar regras em um grupo de segurança que permitem o acesso de um intervalo de endereço IP, de uma porta ou de grupo de segurança. Depois que as regras de entrada são configuradas, as mesmas regras se aplicam a todas as instâncias de banco de dados associadas a esse grupo de segurança.

Para ter mais informações, consulte Controlar acesso com grupos de segurança.

Public accessibility

Quando você inicia uma instância de banco de dados dentro de uma nuvem privada virtual (VPC) com base no serviço da Amazon VPC, pode ativar ou desativar a acessibilidade pública para essa

instância de banco de dados. Para designar se a instância de banco de dados que você cria tem um nome DNS que é determinado como um endereço IP público, use o parâmetro Public accessibility. Usando esse parâmetro, você pode designar se há acesso público à instância do banco de dados. Você pode modificar uma instância de banco de dados para ativar ou desativar a acessibilidade pública modificando o parâmetro Public accessibility (Acessibilidade pública).

Para obter mais informações, consulte Ocultar uma instância de banco de dados em uma VPC da Internet.

> **Note**
>
> Se sua instância de banco de dados estiver em uma VPC, mas não estiver acessível publicamente, também será possível usar uma conexão AWS Site-to-Site VPN ou uma conexão do AWS Direct Connect para acessá-la de uma rede privada. Para ter mais informações, consulte Privacidade do tráfego entre redes.

API do Amazon RDS e endpoints da VPC de interface (AWS PrivateLink)

É possível estabelecer uma conexão privada entre a VPC e os endpoints da API do Amazon RDS criando um VPC endpoint de interface. Os endpoints de interface são desenvolvidos pelo AWS PrivateLink.

O AWS PrivateLink permite que você acesse as operações de API do Amazon RDS sem um gateway da Internet, um dispositivo NAT, uma conexão VPN ou uma conexão do AWS Direct Connect. As instâncias de banco de dados na VPC não precisam de endereços IP públicos para se comunicarem com endpoints de API do Amazon RDS para executar, modificar ou encerrar instâncias de banco de dados. As instâncias de banco de dados também não precisam de endereços IP públicos para usar qualquer uma das operações de API do RDS disponíveis. O tráfego entre seu VPC e Amazon RDS não deixa a rede da Amazon.

Cada endpoint de interface é representado por uma ou mais interfaces de rede elástica nas sub-redes. Para obter mais informações sobre interfaces de rede elástica, consulte Interfaces de rede elástica no Guia do usuário do Amazon EC2.

Para obter mais informações sobre limites de VPC , consulte VPC endpoints de interface (AWS PrivateLink) no Guia do usuário da Amazon VPC. Para obter informações sobre as operações da API do RDS, consulte Referência da API do Amazon RDS.

Você não precisa de um endpoint da VPC de interface para se conectar a uma instância de banco de dados. Para ter mais informações, consulte Cenários para acessar uma instância de banco de dados em uma VPC.

Considerações sobre VPC endpoints

Antes de configurar um VPC endpoint de interface para endpoints da API do Amazon RDS, revise Propriedades e limitações do endpoint de interface no Guia do usuário da Amazon VPC.

Todas as operações de API do RDS relevantes para o gerenciamento de recursos do Amazon RDS estão disponíveis na VPC usando o AWS PrivateLink.

As políticas de endpoint da VPC têm suporte para endpoints da API do RDS. Por padrão, o acesso total às operações de API do RDS é permitido através do endpoint. Para obter mais informações, consulte Controlar o acesso a serviços com VPC endpoints no Guia do usuário da Amazon VPC.

Disponibilidade

No momento, a API do Amazon RDS é compatível com os limites da VPC nas seguintes regiões da AWS:

- Leste dos EUA (Ohio)
- Leste dos EUA (N. da Virgínia)
- Oeste dos EUA (N. da Califórnia)
- Oeste dos EUA (Oregon)
- África (Cidade do Cabo)
- Ásia-Pacífico (Hong Kong)
- Ásia-Pacífico (Mumbai)
- Ásia-Pacífico (Osaka)
- Ásia-Pacífico (Seul)
- Ásia-Pacífico (Singapura)
- Ásia-Pacífico (Sydney)
- Ásia-Pacífico (Tóquio)
- Canadá (Central)
- Oeste do Canadá (Calgary)
- China (Pequim)
- China (Ningxia)
- Europa (Frankfurt)
- Europa (Zurique)
- Europa (Irlanda)
- Europa (Londres)
- Europa (Paris)
- Europa (Estocolmo)
- Europa (Milão)
- Israel (Tel Aviv)
- Middle East (Bahrain)
- South America (São Paulo)
- AWS GovCloud (Leste dos EUA)

- AWS GovCloud (Oeste dos EUA)

Criar um VPC endpoint de interface para a API Amazon RDS

Você pode criar um endpoint da VPC para o serviço de APIs do Amazon RDS usando o console da Amazon VPC ou a AWS Command Line Interface (AWS CLI). Para obter mais informações, consulte Criar um endpoint de interface no Guia do usuário da Amazon VPC.

Crie um VPC endpoint para a API Amazon RDS usando o nome de serviço `com.amazonaws.region.rds`.

Ao excluir regiões da AWS na China, se você habilitar o DNS privado para o endpoint, poderá fazer solicitações de API ao Amazon RDS com o endpoint da VPC usando seu nome DNS padrão para a região da AWS, por exemplo, `rds.us-east-1.amazonaws.com`. Para as regiões China (Pequim) e China (Ningxia) da AWS, é possível fazer solicitações de API com o endpoint da VPC usando `rds-api.cn-north-1.amazonaws.com.cn` e `rds-api.cn-northwest-1.amazonaws.com.cn`, respectivamente.

Para obter mais informações, consulte Acessar um serviço por um endpoint de interface no Guia do usuário da Amazon VPC.

Criar uma política de VPC endpoint para a API Amazon RDS

É possível anexar uma política de endpoint ao VPC endpoint que controla o acesso à API Amazon RDS Essa política especifica as seguintes informações:

- A entidade principal que pode executar ações.
- As ações que podem ser executadas.
- Os recursos sobre os quais as ações podem ser realizadas.

Para obter mais informações, consulte Controlar o acesso a serviços com VPC endpoints no Guia do usuário da Amazon VPC.

Exemplo: política de VPC endpoint para ações da API Amazon RDS

Veja a seguir um exemplo de uma política de endpoint da API Amazon RDS. Quando anexada a um endpoint, essa política concede acesso às ações indicadas da API Amazon RDS para todos os principais em todos os recursos.

```
{
   "Statement":[
      {
         "Principal":"*",
         "Effect":"Allow",
         "Action":[
            "rds:CreateDBInstance",
            "rds:ModifyDBInstance",
            "rds:CreateDBSnapshot"
         ],
         "Resource":"*"
      }
   ]
}
```

Exemplo: política de endpoint da VPC que nega todo o acesso de uma conta da AWS especificada

A política de endpoint da VPC a seguir nega à conta da AWS `123456789012` todo o acesso aos recursos que usam o limite. A política permite todas as ações de outras contas.

```
{
  "Statement": [
    {
      "Action": "*",
      "Effect": "Allow",
      "Resource": "*",
      "Principal": "*"
    },
    {
      "Action": "*",
      "Effect": "Deny",
      "Resource": "*",
      "Principal": { "AWS": [ "123456789012" ] }
     }
   ]
}
```

Práticas recomendadas de segurança do Amazon RDS

Use contas do AWS Identity and Access Management (IAM) para controlar o acesso a operações da API do Amazon RDS, especialmente operações que criam, modificam ou excluem recursos do

Amazon RDS. Esses recursos incluem instâncias de banco de dados, grupos de segurança e grupos de parâmetros. Além disso, use o IAM para controlar ações que executam ações administrativas comuns, como fazer backup e restaurar instâncias de banco de dados.

- Crie um usuário individual para cada pessoa que gerencia recursos do Amazon RDS, incluindo você. Não use as credenciais-raiz da AWS para gerenciar recursos do Amazon RDS.
- Conceda a cada usuário o conjunto mínimo de permissões necessárias para realizar suas funções.
- Use grupos do IAM para gerenciar efetivamente permissões para vários usuários.
- Mude suas credenciais do IAM regularmente.
- Configure o AWS Secrets Manager para alternar automaticamente os segredos para o Amazon RDS. Para ter mais informações, consulte Alternar os segredos do AWS Secrets Manager no Guia do usuário do AWS Secrets Manager. Também é possível recuperar a credencial do AWS Secrets Manager forma programática. Para ter mais informações, consulte Recuperar o valor do segredo no Guia do usuário do AWS Secrets Manager.

Para ter mais informações sobre a segurança do Amazon RDS, consulte Segurança no Amazon RDS. Para ter mais informações sobre o IAM, consulte AWS Identity and Access Management. Para obter informações sobre as práticas recomendadas do IAM, acesse Melhores práticas do IAM.

O AWS Security Hub utiliza controles de segurança para avaliar configurações de recursos e padrões de segurança que ajudam você a cumprir várias frameworks de conformidade. Para ter mais informações sobre como usar o Security Hub para avaliar os recursos do RDS, consulte Controles do Amazon Relational Database Service no Guia do usuário do AWS Security Hub.

É possível monitorar o uso do IAM em relação às práticas recomendadas de segurança com o Security Hub. Para ter mais informações, consulte O que é o AWS Security Hub?.

Use o AWS Management Console, a AWS CLI ou a API do RDS para alterar a senha do usuário mestre. Se você usar outra ferramenta, como um cliente SQL, para alterar a senha do usuário mestre, isso poderá resultar na revogação de privilégios ao usuário involuntariamente.

Controlar acesso com grupos de segurança

Os grupos de segurança de VPC controlam o acesso que o tráfego tem dentro e fora de uma instância de banco de dados. Por padrão, o acesso à rede é desativado para uma instância de banco de dados. É possível especificar regras em um grupo de segurança que permitem o acesso de um intervalo de endereço IP, de uma porta ou de grupo de segurança. Depois que as regras de

entrada são configuradas, as mesmas regras se aplicam a todas as instâncias de banco de dados associadas a esse grupo de segurança. Você pode especificar até 20 regras no grupo de segurança.

Visão geral dos grupos de segurança de VPC

Cada regra de grupo de segurança de VPC possibilita que uma origem específica acesse uma instância de banco de dados em uma VPC que esteja associada a esse grupo de segurança de VPC. A origem pode ser uma gama de endereços (por exemplo, 203.0.113.0/24) ou outro grupo de segurança da VPC. Ao especificar um grupo de segurança de VPC como origem, você permite o tráfego recebido de todas as instâncias (geralmente servidores de aplicações) que usam o grupo de segurança de VPC de origem. Os grupos de segurança de VPC podem ter regras que controlam o tráfego de entrada e saída. No entanto, as regras de tráfego de saída normalmente não se aplicam a instâncias de banco de dados. As regras de tráfego de saída se aplicarão apenas se a instância de banco de dados atuar como um cliente. Por exemplo, as regras de tráfego de saída se aplicam a uma instância de banco de dados do Oracle com links de banco de dados de saída. É necessário usar a API do Amazon EC2 ou a opção Security Group (Grupo de segurança) no console da VPC para criar grupos de segurança de VPC.

Quando você cria regras para o grupo de segurança de VPC que permitem acessar as instâncias na sua VPC, você deve especificar uma porta para cada intervalo de endereços ao qual a regra permite o acesso. Por exemplo, se quiser ativar o acesso via Secure Shell (SSH) para instâncias na VPC, crie uma regra que permitirá o acesso à porta TCP 22 para o intervalo de endereços especificado.

É possível configurar vários grupos de segurança de VPC que permitem o acesso a diferentes portas para diferentes instâncias na sua VPC. Por exemplo, você pode criar um grupo de segurança da VPC que permite acessar a porta TCP 80 para servidores web na VPC. Você pode criar outro grupo de segurança de VPC que permita acessar a porta TCP 3306 para instâncias de bancos de dados do RDS para MySQL na VPC.

Para ter mais informações sobre grupos de segurança da VPC, consulte Grupos de segurança no Guia do usuário da Amazon Virtual Private Cloud.

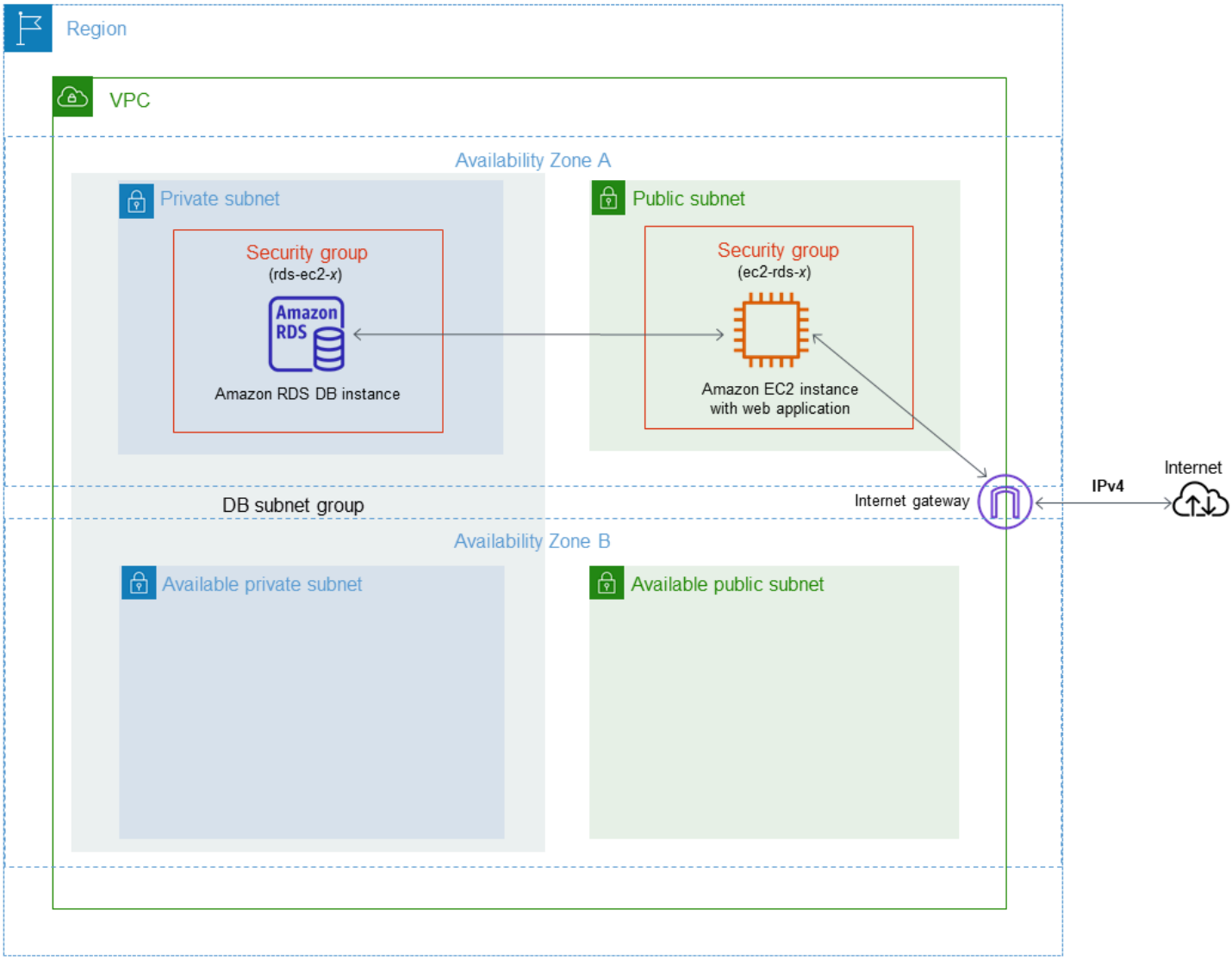
> **Note**
>
> Se a instância de banco de dados estiver em uma VPC, mas não acessível publicamente, também será possível usar um Site-to-Site VPN ou uma conexão do AWS Direct Connect para acessá-la de uma rede privada. Para obter mais informações, consulte Privacidade do tráfego entre redes.

Cenário de grupos de segurança

Um uso comum de uma instância de banco de dados em uma VPC é compartilhar dados com um servidor de aplicações executado em uma instância do Amazon EC2 na mesma VPC, acessado por uma aplicação cliente fora da VPC. Para este cenário, use as páginas do RDS e da VPC no AWS Management Console ou nas operações de API do RDS e do EC2 para criar as instâncias e os grupos de segurança necessários:

1. Crie um grupo de segurança de VPC (por exemplo, `sg-0123ec2example`) e defina as regras de entrada que utilizam os endereços IP da aplicação cliente como a origem. Esse grupo de segurança permite que sua aplicação cliente se conecte a instâncias do EC2 em uma VPC que usa esse grupo de segurança.
2. Crie uma instância do EC2 para a aplicação e adicione a instância do EC2 ao grupo de segurança de VPC (`sg-0123ec2example`) que você criou na etapa anterior.
3. Crie um segundo grupo de segurança de VPC (por exemplo, `sg-6789rdsexample`) e crie uma nova regra especificando o grupo de segurança de VPC que você criou na etapa 1 (`sg-0123ec2example`) como origem.
4. Crie uma instânciaum cluster de banco de dados e adicione a instância de banco de dados ao grupo de segurança de VPC (`sg-6789rdsexample`) que você criou na etapa anterior. Quando você criar a instância de banco de dados, use o mesmo número de porta especificado para a regra do grupo de segurança de VPC (`sg-6789rdsexample`) que você criou na etapa 3.

O diagrama a seguir mostra esse cenário.



Para obter instruções detalhadas sobre como configurar uma VPC para esse cenário, consulte Tutorial: Criar uma VPC para usar com uma instância de banco de dados (somente IPv4). Para ter mais informações sobre como usar uma VPC, consulte Amazon VPC e Amazon RDS.

Criar um grupo de segurança de VPC

Você pode criar um grupo de segurança de VPC para uma instância de banco de dados usando o console da VPC. Para obter informações sobre como criar um grupo de segurança, consulte Fornecer acesso à instância de banco de dados na VPC criando um grupo de segurança e Grupos de segurança no Guia do usuário da Amazon Virtual Private Cloud.

Associar um grupo de segurança a uma instância de banco de dados

Você pode associar um grupo de segurança a uma instância de banco de dados usando Modify (Modificar) no console do RDS, a API `ModifyDBInstance` do Amazon RDS ou o comando `modify-db-instance` da AWS CLI.

O exemplo da CLI a seguir associa um grupo de segurança da VPC específico e remove grupos de segurança de banco de dados da instância de banco de dados.

```
aws rds modify-db-instance --db-instance-identifier dbName --vpc-security-group-ids sg-ID
```

Para ter mais informações sobre como modificar uma instância de banco de dados, consulte Modificar uma instância de banco de dados do Amazon RDS. Com relação a considerações de grupo de segurança quando você restaura uma instância de banco de dados de um snapshot de banco de dados, consulte Considerações sobre os grupos de segurança.

> **Note**
>
> O console do RDS exibirá diferentes nomes de regras de grupos de segurança para seu banco de dados se o valor da porta estiver configurado como um valor não padrão.

Para instâncias de banco de dados do RDS para Oracle, grupos de segurança adicionais podem ser associados preenchendo a configuração de opções do grupo de segurança para as opções Oracle Enterprise Manager Database Express (OEM), Oracle Management Agent for Enterprise Manager Cloud Control (OEM Agent) e Oracle Secure Sockets Layer. Nesse caso, os grupos de segurança associados à instância de banco de dados e as configurações de opções aplicam-se à instância de banco de dados. Para ter mais informações sobre esses grupos de opções, consulte Oracle Enterprise Manager, Oracle Management Agent para Enterprise Manager Cloud Control e Oracle Secure Sockets Layer.

Privilégios da conta de usuário mestre

Ao criar uma instância de banco de dados, o usuário principal padrão usado obtém determinados privilégios para essa instância de banco de dados. Não é possível alterar o nome de usuário principal depois que a instância de banco de dados é criado.

> **Important**
>
> É altamente recomendável não usar o usuário mestre diretamente nas aplicações. Em vez disso, siga as práticas recomendadas de usar um usuário do banco de dados criado com os privilégios mínimos obrigatórios para a aplicação.

> **Note**
>
> Se você excluir acidentalmente as permissões do usuário principal, poderá restaurá-las modificando a instância de banco de dados e definindo uma nova senha de usuário principal. Para ter mais informações sobre como modificar uma instância de banco de dados, consulte Modificar uma instância de banco de dados do Amazon RDS.

A tabela a seguir mostra os privilégios e as funções de banco de dados que o usuário mestre obtém para cada um dos mecanismos do banco de dados.

Mecanism do banco de dados	Privilégio do sistema	Função do banco de dados
RDS para Db2	O usuário principal é atribuído ao grupo `masterdba` e recebe o `master_user_role` . `SYSMON,DBADM` com `DATAACCESS` E `ACCCESSCTRL` , `BINDADD,CONNECT, CREATETAB` ,`CREATE_SECURE_OBJECT` , `EXPLAIN,IMPLICIT_SCHEMA` , `LOAD,SQLADM,WLMADM`	`DBA,DBA_RESTRICTED` , `DEVELOPER` ,`ROLE_NULLID_PACKAGES` , `ROLE_PROCEDURES` ,`ROLE_TABLESPACES` Para obter mais informaçõ es, consulte Perfis padrão do Amazon RDS para Db2.

Mecanism do banco de dados	Privilégio do sistema	Função do banco de dados
RDS para MariaDB	`SELECT,INSERT,UPDATE,DELETE, CREATE,DROP,RELOAD, PROCESS,REFERENCES ,INDEX, ALTER,SHOW DATABASES ,CREATE TEMPORARY TABLES ,LOCK TABLES, EXECUTE,REPLICATION CLIENT,CREATE VIEW ,SHOW VIEW,CREATE ROUTINE, ALTER ROUTINE,CREATE USER, EVENT,TRIGGER,REPLICATION SLAVE` A partir do RDS para MariaDB versão 11.4, o usuário principal também recebe o privilégio `SHOW CREATE ROUTINE`.	—
RDS para MySQL 8.0.36 e posterior	`SELECT,INSERT,UPDATE, DELETE,CREATE,DROP, RELOAD,PROCESS, REFERENCES ,INDEX,ALTER, SHOW DATABASES ,CREATE TEMPORARY TABLES,LOCK TABLES,EXECUTE, REPLICATION SLAVE ,REPLICATION CLIENT , CREATE VIEW,SHOW VIEW,CREATE ROUTINE,ALTER ROUTINE,CREATE USER,EVENT,TRIGGER, CREATE ROLE,DROP ROLE, APPLICATION_PASSWORD_ADMIN , ROLE_ADMIN ,SET_USER_ID , XA_RECOVER_ADMIN`	`rds_superuser_role` Para obter mais informações sobre o `rds_superuser_role` , consulte Modelo de privilégios baseados em perfis para o RDS para MySQL.
Versões do RDS para MySQL anteriores à 8.0.36	`SELECT,INSERT,UPDATE, DELETE,CREATE,DROP, RELOAD,PROCESS, REFERENCES ,INDEX,ALTER, SHOW DATABASES ,CREATE TEMPORARY TABLES,LOCK TABLES,EXECUTE, REPLICATION CLIENT ,CREATE VIEW, SHOW VIEW,CREATE ROUTINE,ALTER ROUTINE,CREATE USER,EVENT, TRIGGER,REPLICATION SLAVE`	—

Mecanism do banco de dados	Privilégio do sistema	Função do banco de dados
RDS para PostgreSQL	`CREATE ROLE,CREATE DB, PASSWORD VALID UNTIL INFINITY,CREATE EXTENSION ,ALTER EXTENSION ,DROP EXTENSION ,CREATE TABLESPACE ,ALTER <OBJECT> OWNER,CHECKPOINT , PG_CANCEL_BACKEND( ) , PG_TERMINATE_BACKEND() ,SELECT PG_STAT_REPLICATION ,EXECUTE PG_STAT_STATEMENTS_RESET() ,OWN POSTGRES_ FDW_HANDLER() ,OWN POSTGRES_FDW_VALIDATOR() ,OWN POSTGRES_FDW , EXECUTE PG_BUFFERCACHE_PAGES() ,SELECT PG_BUFFERCACHE`	`RDS_SUPERUSER` Para obter mais informações sobre RDS_SUPERUSER, consulte Noções básicas de perfis e permissões do PostgreSQL.

Mecanism do banco de dados	Privilégio do sistema	Função do banco de dados
RDS para Oracle	ADMINISTER DATABASE TRIGGER ,ALTER DATABASE LINK,ALTER PUBLIC DATABASE LINK, AUDIT SYSTEM,CHANGE NOTIFICATION , DROP ANY DIRECTORY ,EXEMPT ACCESS POLICY,EXEMPT IDENTITY POLICY,EXEMPT REDACTION POLICY,FLASHBACK ANY TABLE, GRANT ANY OBJECT PRIVILEGE ,RESTRICTED SESSION ,SELECT ANY TABLE,UNLIMITED TABLESPACE	DBA **Note** O perfil DBA está isento dos seguintes privilégios: ALTER DATABASE,ALTER SYSTEM, CREATE ANY DIRECTORY ,CREATE EXTERNAL JOB,CREATE PLUGGABLE DATABASE, GRANT ANY PRIVILEGE ,GRANT ANY ROLE,READ ANY FILE GROUP
Amazon RDS para Microsoft SQL Server	ADMINISTER BULK OPERATIONS ,ALTER ANY CONNECTION ,ALTER ANY CREDENTIAL , ALTER ANY EVENT SESSION,ALTER ANY LINKED SERVER,ALTER ANY LOGIN,ALTER ANY SERVER AUDIT,ALTER ANY SERVER ROLE, ALTER SERVER STATE,ALTER TRACE, CONNECT SQL,CREATE ANY DATABASE, VIEW ANY DATABASE,VIEW ANY DEFINITION ,VIEW SERVER STATE,ALTER ON ROLE SQLAgentOperatorRole	DB_OWNER (função no nível do banco de dados), PROCESSADMIN (função no nível do servidor), SETUPADMIN (função no nível do servidor) , SQLAgentUserRole (função no nível do banco de dados)

Usar funções vinculadas ao serviço do Amazon RDS

O Amazon RDS usa funções vinculadas ao serviço do AWS Identity and Access Management (IAM). O perfil vinculado ao serviço é um tipo exclusivo de perfil do IAM vinculado diretamente ao Amazon RDS. Os perfis vinculados a serviços são predefinidos pelo Amazon RDS e incluem todas as permissões que o serviço requer para chamar outros serviços da AWS em seu nome.

Uma função vinculada ao serviço facilita o uso do Amazon RDS porque você não precisa adicionar as permissões necessárias manualmente. O Amazon RDS define as permissões das funções vinculadas ao serviço e, exceto se definido de outra forma, somente o Amazon RDS pode assumir suas funções. As permissões definidas incluem a política de confiança e a política de permissões, e essa política não pode ser anexada a nenhuma outra entidade do IAM.

Você pode excluir as funções somente depois de primeiro excluir seus recursos relacionados. Isso protege seus recursos do Amazon RDS, pois você não pode remover por engano as permissões para acessar os recursos.

Para obter informações sobre outros serviços compatíveis com funções vinculadas a serviços, consulte Serviços da AWS compatíveis com o IAM e procure serviços que tenham Sim na coluna de função vinculada ao serviço. Escolha um Sim com um link para visualizar a documentação da função vinculada a serviço desse serviço.

Permissões de função vinculada ao serviço do Amazon RDS

O Amazon RDS utiliza a função vinculada a serviço chamada AWSServiceRoleForRDS para permitir que o Amazon RDS chame serviços da AWS em nome das suas instâncias de banco de dados.

A função vinculada ao serviço AWSServiceRoleForRDS confia nos seguintes serviços para assumir a função:

- `rds.amazonaws.com`

Essa função vinculada a serviços tem uma política de permissões anexada a ela, chamada `AmazonRDSServiceRolePolicy`, que concede permissões para operar na conta.

Para ter mais informações sobre essa política, incluindo o documento de política JSON, consulte AmazonRDSServiceRolePolicy no Guia de referência de políticas gerenciadas pela AWS.

> **Note**
>
> É necessário configurar permissões para que uma entidade do IAM (por exemplo, um usuário, grupo ou função) crie, edite ou exclua uma função vinculada ao serviço. Se encontrar a seguinte mensagem de erro:
> Impossível criar o recurso. Você se você tem permissão para criar o perfil vinculado ao serviço. Caso contrário, aguarde e tente novamente mais tarde.
> Certifique-se de que você tem as seguintes permissões ativadas:
>
> ```
> {
>     "Action": "iam:CreateServiceLinkedRole",
>     "Effect": "Allow",
>     "Resource": "arn:aws:iam::*:role/aws-service-role/rds.amazonaws.com/
> AWSServiceRoleForRDS",
>     "Condition": {
>         "StringLike": {
>             "iam:AWSServiceName":"rds.amazonaws.com"
>         }
>     }
> }
> ```
>
> Para ter mais informações, consulte Service-linked role permissions (Permissões de nível vinculado a serviços) no Guia do usuário do IAM.

Criar uma função vinculada ao serviço para o Amazon RDS

Você não precisa criar manualmente uma função vinculada a serviço. Ao criar uma instância de banco de dados, o Amazon RDS cria a função vinculada ao serviço para você.

> **Important**
>
> Se você já usava o serviço Amazon RDS antes de 1.º de dezembro de 2017, quando ele começou a comportar funções vinculadas a serviços, o Amazon RDS já criou a função AWSServiceRoleForRDS em sua conta. Para saber mais, consulte Uma nova função apareceu na minha conta da AWS.

Se você excluir essa função vinculada ao serviço e precisar criá-la novamente, poderá usar esse mesmo processo para recriar a função em sua conta. Ao criar uma instância de banco de dados, o Amazon RDS cria a função vinculada ao serviço para você novamente.

Editar uma função vinculada ao serviço para o Amazon RDS

O Amazon RDS não permite que você edite a função vinculada ao serviço AWSServiceRoleForRDS. Depois que criar uma função vinculada ao serviço, você não poderá alterar o nome da função, pois várias entidades podem fazer referência a ela. No entanto, será possível editar a descrição do perfil usando o IAM. Para ter mais informações, consulte Editar uma função vinculada a serviço no Guia do usuário do IAM.

Excluir uma função vinculada ao serviço para o Amazon RDS

Se você não precisar mais usar um recurso ou serviço que requer uma função vinculada a serviço, é recomendável excluí-la. Dessa forma, você não tem uma entidade não utilizada que não seja monitorada ativamente ou mantida. Contudo, você deve excluir todas as suas instâncias de banco de dados antes de poder excluir a função vinculada ao serviço.

Limpar uma função vinculada ao serviço

Antes de você poder usar o IAM para excluir uma função vinculada ao serviço, você deve primeiro confirmar que a função não tem sessões ativas e remover quaisquer recursos usados pela função.

Para verificar se a função vinculada ao serviço tem uma sessão ativa no console do IAM

1. Faça login em AWS Management Console e abra o console do IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação do console do IAM, escolha Roles (Perfis). A seguir, escolha o nome (não a caixa de seleção) da função AWSServiceRoleForRDS.
3. Na página Resumo do perfil escolhido, selecione a guia Último acesso.
4. Na guia Último acesso, analise a atividade recente para o perfil vinculado ao serviço.

> **Note**
>
> Se não tiver certeza se o Amazon RDS está usando a função AWSServiceRoleForRDS, você pode tentar excluir a função. Se o serviço estiver usando a função, a exclusão falhará e você poderá visualizar as regiões da AWS em que a função está sendo usada.

Se a função está sendo usada, você deve aguardar a sessão final antes de excluir a função. Você não pode revogar a sessão para uma função vinculada a serviço.

Para remover a função AWSServiceRoleForRDS, primeiro é necessário excluir todas as instâncias de banco de dados.

Exclusão de todas as instâncias

Use um destes procedimentos para excluir cada uma de suas instâncias.

Para excluir uma instância (console)

1. Abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Databases (Bancos de dados).
3. Escolha a instância que você deseja excluir.
4. Em Actions, escolha Delete.
5. Se for exibido Create final Snapshot? (Criar snapshot final?), escolha Yes (Sim) ou No (Não).
6. Se você escolher Yes (Sim) na etapa anterior, em Final snapshot name (Nome do snapshot final), digite o nome do snapshot final.
7. Escolha Excluir.

Para excluir uma instância (CLI)

Consulte `delete-db-instance` no AWS CLI Command Reference.

Para excluir uma instância (API)

Consulte `DeleteDBInstance` no Amazon RDS API Reference.

Também é possível usar o console do IAM, a CLI do IAM ou a API do IAM para excluir a função AWSServiceRoleForRDS vinculada a serviço. Para ter mais informações, consulte Excluir uma função vinculada ao serviço no Guia do usuário do IAM.

Permissões de funções vinculadas a serviços para o Amazon RDS Custom

O Amazon RDS Custom utiliza o perfil vinculado ao serviço chamado `AWSServiceRoleForRDSCustom` para permitir que o RDS Custom chame serviços da AWS em nome dos recursos de banco de dados do RDS.

A função vinculada ao serviço AWSServiceRoleForRDSCustom confia nos seguintes serviços para assumir a função:

- `custom.rds.amazonaws.com`

Essa função vinculada a serviços tem uma política de permissões anexada a ela, chamada `AmazonRDSCustomServiceRolePolicy`, que concede permissões para operar na conta.

O processo de criar, editar ou excluir a função vinculada a serviço para o RDS Custom funciona da mesma maneira do que para o Amazon RDS. Para obter mais informações, consulte Política gerenciada pela AWS: AmazonRDSCustomServiceRolePolicy.

> **Note**
>
> É necessário configurar permissões para que uma entidade do IAM (por exemplo, um usuário, grupo ou função) crie, edite ou exclua uma função vinculada ao serviço. Se encontrar a seguinte mensagem de erro:
> Impossível criar o recurso. Você se você tem permissão para criar o perfil vinculado ao serviço. Caso contrário, aguarde e tente novamente mais tarde.
> Certifique-se de que você tem as seguintes permissões ativadas:
>
> ```
> {
>     "Action": "iam:CreateServiceLinkedRole",
>     "Effect": "Allow",
>     "Resource": "arn:aws:iam::*:role/aws-service-role/custom.rds.amazonaws.com/
> AmazonRDSCustomServiceRolePolicy",
>     "Condition": {
>         "StringLike": {
>             "iam:AWSServiceName":"custom.rds.amazonaws.com"
>         }
>     }
> }
> ```
>
> Para ter mais informações, consulte Service-linked role permissions (Permissões de nível vinculado a serviços) no Guia do usuário do IAM.

Permissões de perfis vinculados ao serviço para o Amazon RDS Beta

O Amazon RDS usa o perfil vinculado ao serviço chamado `AWSServiceRoleForRDSBeta` para permitir que o Amazon RDS chame serviços da AWS em nome dos recursos de banco de dados do RDS.

O perfil vinculado ao serviço AWSServiceRoleForRDSBeta confia nos seguintes serviços para assumir o perfil:

- `rds.amazonaws.com`

Essa função vinculada a serviços tem uma política de permissões anexada a ela, chamada `AmazonRDSBetaServiceRolePolicy`, que concede permissões para operar na conta. Para obter mais informações, consulte Política gerenciada da AWS: AmazonRDSBetaServiceRolePolicy.

> **Note**
>
> É necessário configurar permissões para que uma entidade do IAM (por exemplo, um usuário, grupo ou função) crie, edite ou exclua uma função vinculada ao serviço. Se encontrar a seguinte mensagem de erro:
> Impossível criar o recurso. Você se você tem permissão para criar o perfil vinculado ao serviço. Caso contrário, aguarde e tente novamente mais tarde.
> Certifique-se de que você tem as seguintes permissões ativadas:
>
> ```
> {
>     "Action": "iam:CreateServiceLinkedRole",
>     "Effect": "Allow",
>     "Resource": "arn:aws:iam::*:role/aws-service-role/custom.rds.amazonaws.com/
> AmazonRDSBetaServiceRolePolicy",
>     "Condition": {
>         "StringLike": {
>             "iam:AWSServiceName":"custom.rds.amazonaws.com"
>         }
>     }
> }
> ```
>
> Para ter mais informações, consulte Service-linked role permissions (Permissões de nível vinculado a serviços) no Guia do usuário do IAM.

Perfil vinculado ao serviço da Pré-visualização do Amazon RDS

O Amazon RDS usa o perfil vinculado ao serviço chamado `AWSServiceRoleForRDSPreview` para permitir que o Amazon RDS chame serviços da AWS em nome dos recursos de banco de dados do RDS.

O perfil vinculado ao serviço AWSServiceRoleForRDSPreview confia nos seguintes serviços para assumir o perfil:

- `rds.amazonaws.com`

Essa função vinculada a serviços tem uma política de permissões anexada a ela, chamada `AmazonRDSPreviewServiceRolePolicy`, que concede permissões para operar na conta. Para obter mais informações, consulte Política gerenciada da AWS: AmazonRDSPreviewServiceRolePolicy.

> **Note**
>
> É necessário configurar permissões para que uma entidade do IAM (por exemplo, um usuário, grupo ou função) crie, edite ou exclua uma função vinculada ao serviço. Se encontrar a seguinte mensagem de erro:
> Impossível criar o recurso. Você se você tem permissão para criar o perfil vinculado ao serviço. Caso contrário, aguarde e tente novamente mais tarde.
> Certifique-se de que você tem as seguintes permissões ativadas:
>
> ```
> {
>     "Action": "iam:CreateServiceLinkedRole",
>     "Effect": "Allow",
>     "Resource": "arn:aws:iam::*:role/aws-service-role/custom.rds.amazonaws.com/
> AmazonRDSPreviewServiceRolePolicy",
>     "Condition": {
>         "StringLike": {
>             "iam:AWSServiceName":"custom.rds.amazonaws.com"
>         }
>     }
> }
> ```

Para ter mais informações, consulte Service-linked role permissions (Permissões de nível vinculado a serviços) no Guia do usuário do IAM.

Amazon VPC e Amazon RDS

A Amazon Virtual Private Cloud (Amazon VPC) possibilita a execução de recursos da AWS, como instâncias de bancos de dados do Amazon RDS, em uma nuvem privada virtual (VPC).

Ao usar uma VPC, você tem controle sobre o ambiente de rede virtual. É possível escolher seu próprio intervalo de endereços IP, criar sub-redes e configurar o roteamento e listas de controle de acesso. Não há custos adicionais para executar a instância de banco de dados em uma VPC.

As contas têm uma VPC padrão. Todas as novas instâncias de banco de dados são criadas na VPC padrão, a menos que você especifique o contrário.

Tópicos

- Trabalhar com uma instância de banco de dados em uma VPC
- Atualizar a VPC para uma instância de banco de dados
- Cenários para acessar uma instância de banco de dados em uma VPC
- Tutorial: Criar uma VPC para usar com uma instância de banco de dados (somente IPv4)
- Tutorial: Criar uma VPC para uso com uma instância de banco de dados (modo de pilha dupla)
- Mover uma instância de banco de dados fora de uma VPC para uma VPC

Veja a seguir uma discussão sobre a funcionalidade da VPC relevante para instâncias de banco de dados do Amazon RDS. Para obter mais informações sobre uma Amazon VPC, consulte o Guia de conceitos básicos da Amazon VPC e Guia do usuário da Amazon VPC.

Trabalhar com uma instância de banco de dados em uma VPC

Sua instância deve estar em uma nuvem privada virtual (VPC). Uma VPC é uma rede virtual logicamente isolada de outras redes virtuais na Nuvem AWS. A Amazon VPC permite que você execute recursos da AWS, como uma instância de banco de dados do Amazon RDS ou instância do Amazon EC2, em uma VPC. A VPC pode ser uma VPC padrão que vem com sua conta ou aquele que você criou. Todas as VPCs estão associadas à sua conta da AWS.

Sua VPC padrão possui três sub-redes que você pode usar para isolar recursos dentro da VPC. A VPC padrão também possui um gateway da Internet que pode ser usado para fornecer acesso a recursos na VPC de fora da VPC.

Para obter uma lista de cenários envolvendo instâncias de banco de dados do Amazon RDS em uma VPC e fora de uma VPC, consulte Cenários para acessar uma instância de banco de dados em uma VPC.

Tópicos

- Trabalhar com uma instância de banco de dados em uma VPC
- Trabalhar com grupos de sub-redes de banco de dados
- Sub-redes compartilhadas
- Endereçamento IP do Amazon RDS
- Ocultar uma instância de banco de dados em uma VPC da Internet
- Criar uma instância de banco de dados em uma VPC

Nos tutoriais a seguir, você pode aprender a criar uma VPC a ser usar para um cenário comum do Amazon RDS:

- Tutorial: Criar uma VPC para usar com uma instância de banco de dados (somente IPv4)
- Tutorial: Criar uma VPC para uso com uma instância de banco de dados (modo de pilha dupla)

Trabalhar com uma instância de banco de dados em uma VPC

Veja a seguir algumas dicas sobre como trabalhar com uma instância de banco de dados em uma VPC:

- A VPC deve ter pelo menos duas sub-redes. Essas sub-redes devem estar em duas zonas de disponibilidade diferentes na Região da AWS onde você deseja implantar a instância de banco de dados. Uma sub-rede é um segmento do intervalo de endereços IP de uma VPC que você pode especificar e que permite agrupar instâncias com base nas suas necessidades operacionais e de segurança.

  Para implantações multi-AZ, a definição de uma sub-rede para duas ou mais zonas de disponibilidade em uma região da Região da AWS permite que o Amazon RDS crie uma nova espera em outra zona de disponibilidade, conforme necessário. Faça isso mesmo para implantações single-AZ, caso queira convertê-las em implantações multi-AZ em algum momento.

> **Note**
>
> O grupo de sub-redes de banco de dados para uma região local pode ter somente uma sub-rede.

- Se quiser que sua instância de banco de dados na VPC seja publicamente acessível, você deverá habilitar os atributos DNS hostnames (Nomes de host de DNS) e DNS resolution (Resolução de DNS).
- Sua VPC deve ter um grupo de sub-redes de banco de dados que você criou. Crie um grupo de sub-redes de banco de dados especificando as sub-redes criadas. O Amazon RDS escolhe uma sub-rede e um endereço IP dentro desse grupo de sub-redes para associar à sua instância de banco de dados. A instância de banco de dados usa a zona de disponibilidade que contém a sub-rede.
- Sua VPC deve ter um grupo de segurança de VPC que permita o acesso à instância de banco de dados.

  Para ter mais informações, consulte Cenários para acessar uma instância de banco de dados em uma VPC.
- Os blocos CIDR em cada uma das suas sub-redes devem ser suficientemente grandes para acomodar endereços IP sobressalentes para o Amazon RDS usar durante atividades de manutenção, incluindo failover e escalabilidade de computação. Por exemplo, um intervalo como 10.0.0.0/24 e 10.0.1.0/24 normalmente é grande o suficiente.
- Uma VPC pode ter um atributo instance tenancy (locação de instâncias) com o valor default (padrão) ou dedicated (dedicado). Todas as VPC padrão têm o atributo de locação de instâncias definido como padrão, e uma VPC padrão pode oferecer suporte a qualquer classe de instância de banco de dados.

  Se você optar por ter sua instância de banco de dados em uma VPC dedicada em que o atributo de locação de instâncias esteja definido como dedicado, a classe de sua instância de banco de dados deverá ser um dos tipos aprovados de instância dedicada do Amazon EC2. Por exemplo, a instância dedicada r5.large do EC2 corresponde à classe de instância de banco de dados db.r5.large. Para obter informações sobre a locação de instâncias em uma VPC, consulte Instâncias dedicadas no Guia do usuário do Amazon Elastic Compute Cloud.

  Consulte mais informações sobre os tipos de instância que podem estar em uma instância dedicada em Instâncias dedicadas do Amazon EC2, na página de preços do Amazon EC2.

> **Note**
>
> Quando você define o atributo de locação de instâncias como dedicado para uma instância de banco de dados, ele não garante que essa instância terá execução em um host dedicado.

- Quando um grupo de opções é atribuído a uma instância de banco de dados, ele é associado à VPC da instância de banco de dados. Esse vínculo significa que você não pode usar o grupo de opções atribuído a uma instância de banco de dados se tentar restaurar essa instância de banco de dados em uma VPC diferente.
- Se você restaurar uma instância de banco de dados em uma VPC diferente, atribua o grupo de opções padrão à instância de banco de dados, atribua um grupo de opções que esteja vinculado a essa VPC ou crie um grupo de opções e atribua-o à instância de banco de dados. Com opções persistentes ou permanentes, como o Oracle TDE, você deve criar um novo grupo de opções que inclua a opção persistente ou permanente ao restaurar uma instância de banco de dados em uma VPC diferente.

Trabalhar com grupos de sub-redes de banco de dados

Sub-redes são segmentos do intervalo de endereços IP de uma VPC que você designa para agrupar seus recursos com base em necessidades operacionais e de segurança. Um grupo de sub-redes de banco de dados é uma coleção de sub-redes (geralmente privadas) que você cria em uma VPC e designa para suas instâncias de banco de dados. Ao usar um grupo de sub-redes de banco de dados, você pode especificar uma VPC específica ao criar instâncias de banco de dados usando a AWS CLI ou a API. Se você usar o console, poderá escolher somente a VPC e os grupos de sub-redes que deseja usar.

Cada grupo de sub-redes de banco de dados deve ter sub-redes em pelo menos duas zonas de disponibilidade em determinada Região da AWS. Ao criar uma instância de banco de dados em uma VPC, escolha um grupo de sub-redes de banco de dados. Do grupo de sub-rede de banco de dados, o Amazon RDS escolhe uma sub-rede e um endereço IP dentro dessa sub-rede para associar . O banco de dados usa a zona de disponibilidade que contém a sub-rede. O Amazon RDS sempre atribui um endereço IP de uma sub-rede que tem espaço de endereço IP livre.

Se a instância do banco de dados primário de uma implantação multi-AZ falhar, o Amazon RDS poderá promover a espera correspondente e, posteriormente, criar uma espera usando um endereço IP da sub-rede em uma das outras zonas de disponibilidade.

As sub-redes em um grupo de sub-redes de banco de dados são públicas ou privadas. As sub-redes são públicas ou privadas, dependendo da configuração definida para as listas de controle de acesso à rede (ACLs de rede) e tabelas de roteamento. Para que uma instância de banco de dados seja acessível ao público, todas as sub-redes em seu grupo de sub-redes de banco de dados devem ser públicas. Se uma sub-rede associada a uma instância de banco de dados acessível ao público mudar de pública para privada, ela poderá afetar a disponibilidade da instância de banco de dados.

Para criar um grupo de sub-redes de banco de dados que seja compatível com o modo de pilha dupla, verifique se cada sub-rede adicionada ao grupo de sub-redes de banco de dados tem um bloco CIDR do Internet Protocol versão 6 (IPv6) associado a ele. Para ter mais informações, consulte Endereçamento IP do Amazon RDS e Migrar para IPv6 no Guia do usuário da Amazon VPC.

> **Note**
>
> O grupo de sub-redes de banco de dados para uma região local pode ter somente uma sub-rede.

Quando o Amazon RDS cria uma instância de banco de dados em uma VPC, ele atribui uma interface de rede à sua instância de banco de dados usando um endereço IP do seu grupo de sub-redes de banco de dados. No entanto, é altamente recomendável que você use o Sistema de Nomes de Domínio (DNS) para se conectar à sua instância de banco de dados. Recomendamos isso porque o endereço IP subjacente muda durante o failover.

> **Note**
>
> Para cada instância de banco de dados executada em uma VPC, é necessário reservar pelo menos um endereço em cada sub-rede no grupo de sub-redes de banco de dados para uso pelo Amazon RDS para ações de recuperação.

Sub-redes compartilhadas

Você pode criar uma instância de banco de dados em uma VPC compartilhada.

Algumas considerações que você deve ter em mente ao usar VPCs compartilhadas:

- Você pode mover uma instância de banco de dados de uma sub-rede de VPC compartilhada para uma sub-rede de VPC não compartilhada e vice-versa.

- Os participantes de uma VPC compartilhada devem criar um grupo de segurança na VPC para permitir que criem uma instância de banco de dados.
- Proprietários e participantes de uma VPC compartilhada podem acessar o banco de dados usando consultas SQL. No entanto, somente o criador de um recurso pode fazer chamadas de API no recurso.

Endereçamento IP do Amazon RDS

Os endereços IP habilitam recursos na sua VPC para se comunicar com outros e com recursos na Internet. O Amazon RDS comporta protocolos de endereçamento IPv4 e IPv6. Por padrão, o Amazon RDS e a Amazon VPC usam o protocolo de endereçamento IPv4. Você não pode desativar esse comportamento. Ao criar uma VPC, especifique um bloco CIDR IPv4 (um intervalo de endereços IPv4 privados). Você também pode atribuir um bloco CIDR IPv6 à VPC e às sub-redes, bem como endereços IPv6 desse bloco a instâncias de banco de dados em sua sub-rede.

A compatibilidade com o protocolo IPv6 expande o número de endereços IP compatíveis. Ao usar o protocolo IPv6, você tem a garantia de que terá endereços disponíveis suficientes para o crescimento futuro da Internet. Os recursos do RDS novos e existentes podem usar endereços IPv4 e IPv6 na VPC. Configurar, proteger e converter o tráfego de rede entre os dois protocolos usados em diferentes partes de uma aplicação pode causar sobrecarga operacional. Você pode padronizar o protocolo IPv6 para recursos do Amazon RDS a fim de simplificar sua configuração de rede.

Tópicos

- Endereços IPv4
- Endereços IPv6
- Modo de pilha dupla

Endereços IPv4

Quando você cria uma VPC, é necessário especificar um intervalo de endereços IPv4 para a VPC em forma de um bloco CIDR, como `10.0.0.0/16`. m grupo de sub-redes de banco de dados define o intervalo de endereços IP nesse bloco CIDR que uma instância de banco de dados pode usar. Esses endereços IP podem ser públicos ou privados.

Um endereço IPv4 privado é um endereço IP que não é acessível pela Internet. Você pode usar endereços IPv4 privados para comunicação entre a instância de banco de dados e outros recursos,

como instâncias do Amazon EC2, na mesma VPC. Cada instância de banco de dados tem um endereço IP privado para comunicação na VPC.

Um endereço IP público é um endereço IPv4 acessível pela Internet. Você pode usar endereços públicos para comunicação entre a instância de banco de dados e os recursos na Internet, como um cliente SQL. Você controla se a instância de banco de dados recebe um endereço IP público.

Para ver um tutorial que mostra como criar uma VPC somente com endereços IPv4 privados que você pode usar para um cenário comum do Amazon RDS, consulte Tutorial: Criar uma VPC para usar com uma instância de banco de dados (somente IPv4).

Endereços IPv6

Como opção, você pode associar um bloco CIDR IPv6 a sua VPC e sub-redes e atribuir endereços IPv6 desse bloco a recursos em sua VPC. Todo endereço IPv6 é globalmente exclusivo.

O bloco CIDR IPv6 da VPC é automaticamente atribuído do grupo de endereços IPv6 da Amazon. Você não pode escolher o intervalo.

Ao se conectar a um endereço IPv6, verifique se as seguintes condições são atendidas:

- O cliente está configurado para que o tráfego do cliente para o banco de dados via IPv6 seja permitido.
- Os grupos de segurança do RDS usados pela instância de banco de dados estão configurados corretamente para que o tráfego do cliente para o banco de dados via IPv6 seja permitido.
- A pilha do sistema operacional do cliente permite tráfego no endereço IPv6, e os drivers e bibliotecas do sistema operacional estão configurados para escolher o endpoint de instância de banco de dados padrão correto (IPv4 ou IPv6).

Para ter mais informações sobre IPv6, consulte Endereçamento IP no Guia do usuário da Amazon VPC.

Modo de pilha dupla

Quando uma instância de banco de dados consegue se comunicar pelos protocolos de endereçamento IPv4 e IPv6, a execução ocorre em modo de pilha dupla. Portanto, os recursos podem se comunicar com a instância de banco de dados por IPv4, IPv6 ou ambos. O RDS desabilita o acesso ao gateway da Internet para endpoints IPv6 de instâncias de banco de dados privadas

do modo de pilha dupla. O RDS faz isso para garantir que seus endpoints IPv6 sejam privados e possam ser acessados somente de dentro de sua VPC.

Tópicos

- Modo de pilha dupla e grupos de sub-redes de banco de dados
- Trabalhar com instâncias de banco de dados de modo de pilha de dupla
- Modificar instâncias de banco de dados somente IPv4 para usar o modo de pilha dupla
- Disponibilidade de região e versão
- Limitações para instâncias de banco de dados de rede de pilha dupla

Para ver um tutorial que mostra como criar uma VPC com endereços IPv4 e IPv6 que você pode usar para um cenário comum do Amazon RDS, consulte Tutorial: Criar uma VPC para uso com uma instância de banco de dados (modo de pilha dupla).

Modo de pilha dupla e grupos de sub-redes de banco de dados

Para usar o modo de pilha dupla, verifique se cada sub-rede no grupo de sub-redes de banco de dados que você associa à instância de banco de dados tem um bloco CIDR IPv6 associado a ela. Você pode criar um grupo de sub-redes de banco de dados ou modificar um existente para atender a esse requisito. Depois que uma instância de banco de dados entra no modo de pilha dupla, os clientes podem se conectar normalmente. Os firewalls de segurança do cliente e os grupos de segurança de instâncias de banco de dados do RDS devem ser configurados com precisão para permitir tráfego por IPv6. Para se conectar, os clientes usam o endpoint da instância de banco de dados.endpoint da instância principal do cluster de banco de dados. As aplicações cliente podem especificar qual protocolo é o preferencial ao se conectar a um banco de dados. No modo de pilha dupla, a instância de banco de dados detecta o protocolo de rede de preferência do cliente, IPv4 ou IPv6, e o usa para a conexão.

Se um grupo de sub-redes de banco de dados deixar de ser compatível com o modo de pilha dupla devido à exclusão de sub-rede ou desassociação do CIDR, existe o risco de um estado de rede incompatível para instâncias de banco de dados associadas ao grupo de sub-redes de banco de dados. Além disso, não é possível usar o grupo de sub-redes de banco de dados ao criar uma nova instância de banco de dados do modo de pilha dupla.

Para determinar se um grupo de sub-redes de banco de dados é compatível com o modo de pilha dupla usando o AWS Management Console, visualize o Network type (Tipo de rede) na página de detalhes do grupo de sub-redes de banco de dados. Para determinar se um grupo de sub-redes de

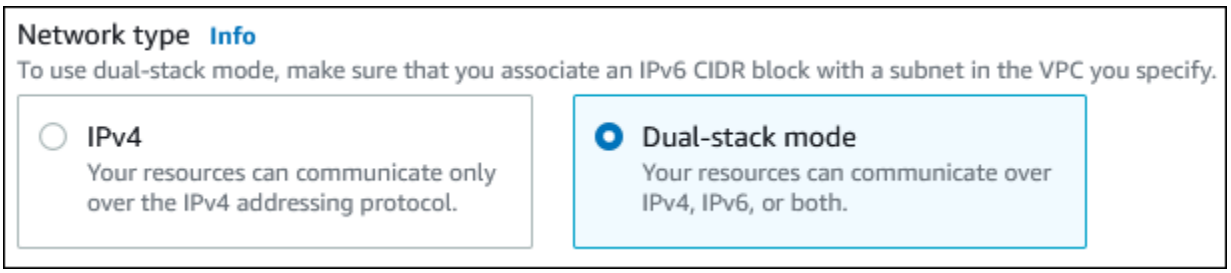
banco de dados comporta o modo de pilha dupla usando a AWS CLI, execute o comando describe-db-subnet-groups e visualize `SupportedNetworkTypes` na saída.

As réplicas de leitura são tratadas como instâncias de banco de dados independentes e podem ter um tipo de rede diferente da instância de banco de dados primária. Se você alterar o tipo de rede da instância de banco de dados primária de uma réplica de leitura, esta não será afetada. Você pode restaurar uma instância de banco de dados para qualquer tipo de rede compatível.

Trabalhar com instâncias de banco de dados de modo de pilha de dupla

Ao criar ou modificar uma instância de banco de dados, você pode especificar o modo de pilha dupla para permitir que os recursos se comuniquem com a instânciao cluster de banco de dados por IPv4, IPv6 ou ambos.

Ao usar o AWS Management Console para criar ou modificar uma instância de banco de dados, você pode especificar o modo de pilha dupla na seção Network type (Tipo de rede). A imagem a seguir mostra a seção Network type (Tipo de rede) no console.



Ao usar a AWS CLI para criar ou modificar uma instância de banco de dados, defina a opção `--network-type` como `DUAL` para usar o modo de pilha dupla. Ao usar a API do RDS para criar ou modificar uma instância de banco de dados, defina o parâmetro `NetworkType` como `DUAL` para usar o modo de pilha dupla. Quando você estiver modificando o tipo de rede de uma instância de banco de dados, é possível que ocorra tempo de inatividade. Se o modo de pilha dupla não for compatível com a versão do mecanismo de banco de dados especificado nem com o grupo de sub-redes de banco de dados, o erro `NetworkTypeNotSupported` será retornado.

Para ter mais informações sobre como criar uma instância de banco de dados, consulte Criar uma instância de banco de dados do Amazon RDS. Para ter mais informações sobre a modificação de uma instância de banco de dados, consulte Modificar uma instância de banco de dados do Amazon RDS.

Para determinar se uma instância de banco de dados está no modo de pilha dupla usando o console, visualize o Network type (Tipo de rede) na guia Connectivity & security (Conectividade e segurança) para a instância de banco de dados.

Modificar instâncias de banco de dados somente IPv4 para usar o modo de pilha dupla

Você pode modificar instâncias de banco de dados somente IPv4 para usar o modo de pilha dupla. Para isso, altere o tipo de rede da instância de banco de dados. A modificação pode ocasionar tempo de inatividade.

É recomendável que você altere o tipo de rede das instâncias de banco de dados do Amazon RDS durante uma janela de manutenção. No momento, não é possível definir o tipo de rede de novas instâncias para o modo de pilha dupla. É possível definir o tipo de rede manualmente usando o comando `modify-db-instance` .

Antes de modificar uma instância de banco de dados para usar o modo de pilha dupla, verifique se seu grupo de sub-redes de banco de dados é compatível com o modo de pilha dupla. Se o grupo de sub-redes de banco de dados associado à instância de banco de dados não for compatível com o modo de pilha dupla, ao modificar a instância de banco de dados especifique um grupo de sub-redes de banco de dados diferente que seja compatível. Modificar o grupo de sub-redes de banco de dados de uma instância de banco de dados pode ocasionar tempo de inatividade.

Se você modificar o grupo de sub-redes de banco de dados de uma instância de banco de dados antes de alterar a instância de banco de dados para usar o modo de pilha dupla, verifique se o grupo de sub-redes de banco de dados é válido para a instância de banco de dados antes e depois da alteração.

Para instâncias single-AZ do RDS para PostgreSQL, RDS para MySQL, RDS para Oracle e RDS para MariaDB, recomendamos que você execute o comando modify-db-instance apenas com o parâmetro `--network-type` com valor `DUAL` para alterar a rede para o modo de pilha dupla. Adicionar outros parâmetros com o parâmetro `--network-type` na mesma chamada de API pode ocasionar tempo de inatividade. Para modificar vários parâmetros, primeiro é preciso concluir com êxito a modificação do tipo de rede para só então enviar outra solicitação `modify-db-instance` com outros parâmetros.

As modificações do tipo de rede para instâncias de banco de dados multi-AZ do RDS para PostgreSQL, RDS para MySQL, RDS para Oracle e RDS para MariaDB causarão um breve tempo de inatividade e acionarão um failover se você usar apenas o parâmetro `--network-type` ou combinar parâmetros em um comando modify-db-instance.

As modificações do tipo de rede em instâncias de banco de dados multi-AZ ou single-AZ do RDS para SQL Server causarão tempo de inatividade se você usar apenas o parâmetro `--network-type` ou combinar parâmetros em um comando `modify-db-instance`. As modificações do tipo de rede causam failover em uma instância multi-AZ do SQL Server.

Se você não conseguir se conectar à instância de banco de dados após a alteração, verifique se os firewalls de segurança do cliente e do banco de dados e as tabelas de rotas estão configurados com precisão para permitir tráfego para o banco de dados na rede selecionada (IPv4 ou IPv6). Você também pode precisar modificar parâmetros, bibliotecas ou drivers do sistema operacional para se conectar por meio de um endereço IPv6.

Quando se modifica uma instância de banco de dados para usar o modo de pilha dupla, não pode haver uma alteração pendente de uma implantação single-AZ para uma implantação multi-AZ ou de uma implantação multi-AZ para uma implantação single-AZ.

Como modificar instâncias de banco de dados somente IPv4 para usar o modo de pilha dupla

1. Modifique um grupo de sub-redes de banco de dados para ser compatível com o modo de pilha dupla ou crie um grupo de sub-redes de banco de dados que seja compatível com esse modo:

   a. Associe um bloco CIDR IPv6 à VPC.

      Para obter instruções, consulte Adicionar um bloco CIDR IPv6 a sua VPC no Manual do usuário do Amazon VPC.

   b. Anexe o bloco CIDR IPv6 a todas as sub-redes do grupo de sub-redes do banco de dados.

      Para obter instruções, consulte Adicionar um bloco CIDR IPv6 a sua sub-rede no Manual do usuário do Amazon VPC.

   c. Verifique se o grupo de sub-redes de banco de dados é compatível com o modo de pilha dupla.

      Se você estiver usando o AWS Management Console, selecione o grupo de sub-redes de banco de dados e verifique se o valor Supported network types (Tipos de rede compatíveis) é Dual, IPv4 (Duplo, IPv4).

      Se estiver usando a AWS CLI, execute o comando describe-db-subnet-groups e verifique se o valor `SupportedNetworkType` para a instância de banco de dados é `Dual, IPv4`.

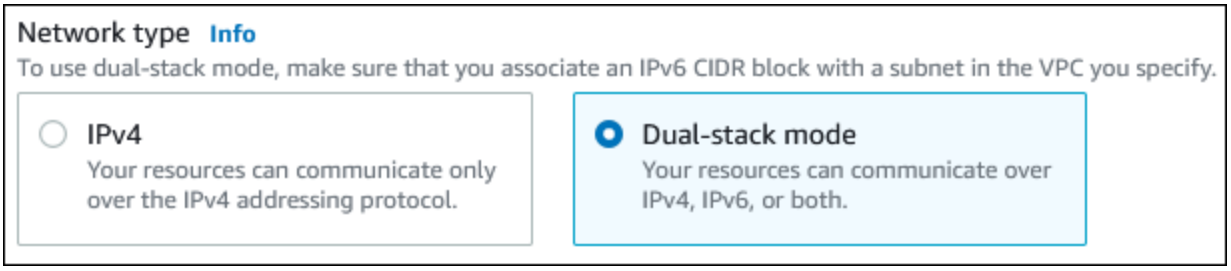
2. Modifique o grupo de segurança associado à instância de banco de dados para permitir conexões IPv6 com o banco de dados ou crie um grupo de segurança que permita conexões IPv6.

   Para obter instruções, consulte Regras do grupo de segurança no Guia do usuário da Amazon VPC.

3. Modifique a instância de banco de dados para oferecer suporte ao modo de pilha dupla. Para fazer isso, defina a opção Network type (Tipo de rede) como Dual-stack mode (Modo de pilha dupla).

   Se você estiver usando o console, verifique se as seguintes configurações estão corretas:

   - Network type (Tipo de rede): Dual-stack mode (Modo de pilha dupla).



   - DB subnet group (Grupo de sub-redes do banco de dados): o grupo de sub-redes de banco de dados que você configurou em uma etapa anterior
   - Security group (Grupo de segurança): o grupo de segurança que você configurou em uma etapa anterior.

   Se você estiver usando a AWS CLI, verifique se as seguintes configurações estão corretas:

   - `--network-type` – `dual`
   - `--db-subnet-group-name`: o grupo de sub-redes de banco de dados que você configurou em uma etapa anterior
   - `--vpc-security-group-ids`: o grupo de segurança da VPC que você configurou em uma etapa anterior.

   Por exemplo:

```
aws rds modify-db-instance --db-instance-identifier my-instance --network-type
 "DUAL"
```

4. Verifique se a instância de banco de dados é compatível com o modo de pilha dupla.

   Se estiver usando o console, selecione a guia Connectivity & security (Conectividade e segurança) para a instância de banco de dados. Nessa guia, verifique se o valor de Network type (Tipo de rede) é Dual-stack mode (Modo de pilha dupla).

Se estiver usando a AWS CLI, execute o comando describe-db-instances e verifique se o valor `NetworkType` para a instância de banco de dados é `dual`.

Execute o comando `dig` no endpoint da instância de banco de dados para identificar o endereço IPv6 associado a ele.

```
dig db-instance-endpoint AAAA
```

Use o endpoint da instância de banco de dados, não o endereço IPv6, para se conectar à instância de banco de dados.

Disponibilidade de região e versão

A disponibilidade e a compatibilidade de recursos variam entre versões específicas de cada mecanismo de banco de dados e entre Regiões da AWS. Para obter mais informações sobre a disponibilidade de versões e regiões com o modo de pilha dupla, consulte Regiões e mecanismos de banco de dados compatíveis com o modo de pilha dupla no Amazon RDS.

Limitações para instâncias de banco de dados de rede de pilha dupla

As seguintes limitações se aplicam às instâncias de banco de dados de rede de pilha dupla:

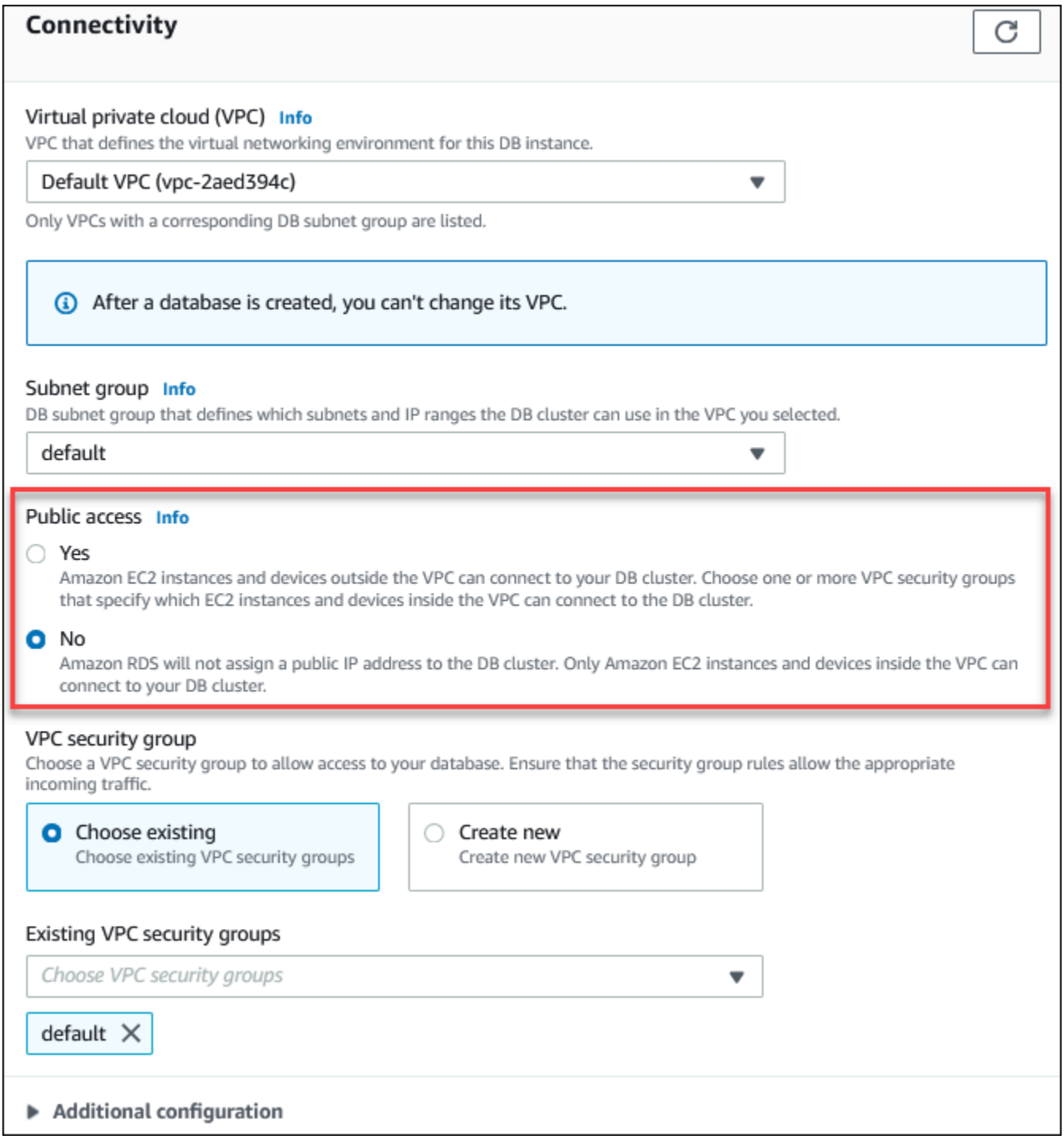
- Instâncias de banco de dados não podem usar o protocolo IPv6 exclusivamente. É possível usar IPv4 exclusivamente ou o protocolo IPv4 e IPv6 (modo de pilha dupla).
- O Amazon RDS não é compatível com sub-redes IPv6 nativas.
- Instâncias de banco de dados que usam o modo de pilha dupla devem ser privados. Eles não podem ser acessíveis ao público.
- Para o RDS para SQL Server, as instâncias de banco de dados de modo de pilha dupla que usam endpoints de ouvinte do grupo de disponibilidade Always On AGs apresentam apenas endereços IPv4.
- Você não pode usar o RDS Proxy com instâncias de banco de dados do modo de pilha dupla.
- Você não pode usar o modo de pilha dupla com o RDS em instâncias de banco de dados do AWS Outposts.
- Você não pode usar o modo de pilha dupla com instâncias de banco de dados em uma zona local.

Ocultar uma instância de banco de dados em uma VPC da Internet

Um cenário comum do Amazon RDS é ter uma VPC na qual existe uma instância do Amazon EC2 com uma aplicação Web voltada para o público e uma instância de banco de dados com um banco de dados não acessível ao público geral. Por exemplo, você pode criar uma VPC que tenha uma sub-rede pública e uma sub-rede privada. As instâncias do EC2 que funcionam como servidores Web podem ser implantadas na sub-rede pública. As instâncias de banco de dados são implantadas na sub-rede privada. Nessa implantação, apenas os servidores Web têm acesso às instâncias de banco de dados. Para obter uma ilustração desse cenário, consulte Uma instância de banco de dados em uma VPC acessada por uma instância do Amazon EC2 na mesma VPC..

Quando você executa uma instância de banco de dados dentro de uma VPC, a instância de banco de dados tem um endereço IP privado para tráfego dentro da VPC. Esse endereço IP privado não é acessível ao público geral. Você pode usar a opção Public access (Acesso público) para designar se a instância de banco de dados também deve ter um endereço IP público além do endereço IP privado. Se a instância de banco de dados for acessível ao público, seu endpoint DNS resolverá para o endereço IP privado de dentro da VPC. Ele é resolvido para o endereço IP público de fora da VPC. O acesso à instância de banco de dados é controlado em última análise pelo grupo de segurança usado. Esse acesso público não será permitido se o grupo de segurança atribuído à instância de banco de dados não incluir regras de entrada que permitam isso. Além disso, para que uma instância de banco de dados seja acessível publicamente, as sub-redes no grupo de sub-redes de banco de dados devem ter um gateway da Internet. Para ter mais informações, consulte Não é possível conectar-se à instância de banco de dados do Amazon RDS.

Você pode modificar uma instância de banco de dados para ativar ou desativar a acessibilidade ao público geral modificando a opção Public access (Acesso público). A ilustração a seguir mostra a opção Public access (Acesso público) na seção Additional connectivity configuration (Configuração de conectividade adicional). Para definir a opção, abra a seção Additional connectivity configuration (Configuração de conectividade adicional) na seção Connectivity (Conectividade).



Para obter informações sobre como modificar uma instância de banco de dados para definir a opção de Public access (Acesso público), consulte Modificar uma instância de banco de dados do Amazon RDS.

Criar uma instância de banco de dados em uma VPC

Os procedimentos a seguir ajudam você a criar uma instância de banco de dados em uma VPC. Para usar a VPC padrão, você pode começar na etapa 2 e usar a VPC e o grupo de sub-redes de banco de dados que já foram criados para você. Se quiser criar uma VPC adicional, você poderá criar uma nova VPC.

> **Note**
>
> Se você deseja que sua instância de banco de dados na VPC seja acessível ao público geral, atualize as informações de DNS da VPC, habilitando os atributos DNS hostnames (Nomes de host de DNS) e DNS resolution (Resolução de DNS). Para obter informações sobre como atualizar as informações de DNS para uma instância de VPC, consulte Atualização do suporte a DNS para sua VPC.

Siga estas etapas para criar uma instância de banco de dados em uma VPC:

- Etapa 1: Criar uma VPC
- Etapa 2: Criar um grupo de sub-redes de banco de dados
- Etapa 3: Criar um grupo de segurança da VPC
- Etapa 4: Criar uma instância de banco de dados na VPC

Etapa 1: Criar uma VPC

Crie uma VPC com duas sub-redes em pelo menos duas zonas de disponibilidade. Você usará essas sub-redes ao criar um grupo de sub-redes de banco de dados. Se você tiver uma VPC padrão, uma sub-rede será criada automaticamente para você em cada zona de disponibilidade na Região da AWS.

Para ter mais informações, consulte Criar uma VPC com sub-redes públicas e privadas ou Criar uma VPC no Guia do usuário da Amazon VPC.

Etapa 2: Criar um grupo de sub-redes de banco de dados

Um grupo de sub-redes de banco de dados é uma coleção de sub-redes (geralmente privadas) que você cria para uma VPC e designa às suas instâncias de banco de dados. Um grupo de sub-redes

de banco de dados permite que você especifique uma VPC particular ao criar instâncias de banco de dados usando a AWS CLI ou a API do RDS. Se você usar o console, poderá escolher somente a VPC e as sub-redes que deseja usar. Cada grupo de sub-redes de banco de dados deve ter pelo menos uma sub-rede em pelo menos duas zonas de disponibilidade na Região da AWS. Como prática recomendada, cada grupo de sub-redes de banco de dados deve ter no mínimo uma sub-rede para cada zona de disponibilidade na Região da AWS.

Para implantações multi-AZ, definir uma sub-rede para todas as zonas de disponibilidade em uma Região da AWS permite que o Amazon RDS crie uma réplica em espera em outra zona de disponibilidade, se necessário. Você pode seguir essa prática recomendada mesmo para implantações single-AZ, pois poderá convertê-las em implantações multi-AZ no futuro.

Para que uma instância de banco de dados seja acessível publicamente, as sub-redes no grupo de sub-redes de banco de dados devem ter um gateway da Internet. Para ter mais informações sobre gateways da Internet para sub-redes, consulte "Estabelecer conexão com a Internet usando um gateway da Internet no Manual do usuário da Amazon VPC.

> **Note**
>
> O grupo de sub-redes de banco de dados para uma região local pode ter somente uma sub-rede.

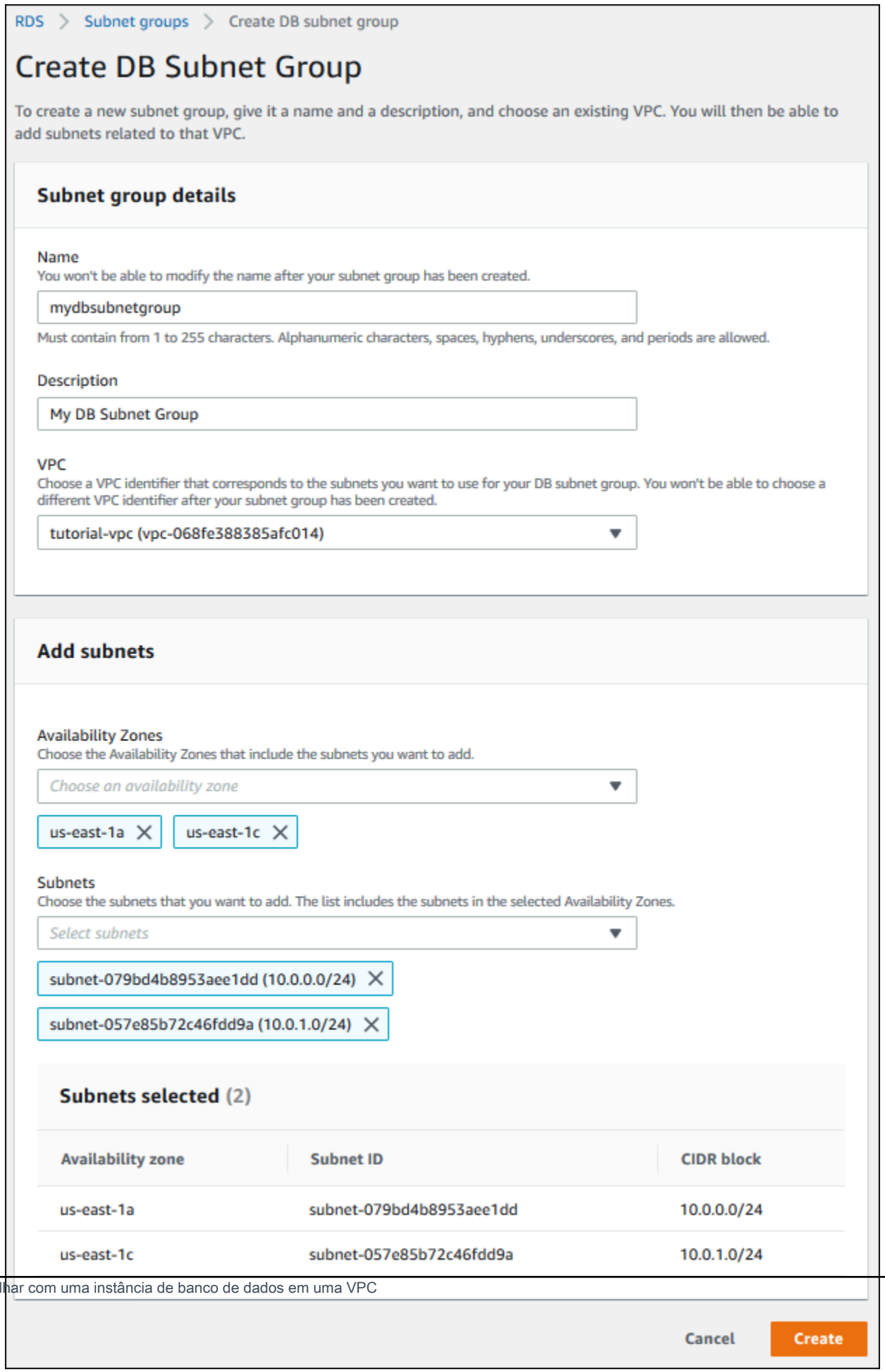
Ao criar uma instância de banco de dados em uma VPC, você pode escolher um grupo de sub-redes de banco de dados. O Amazon RDS escolhe uma sub-rede e um endereço IP dentro dessa sub-rede para associar à instância de banco de dados. Se não houver nenhum grupo de sub-redes de banco de dados, o Amazon RDS criará um grupo de sub-redes padrão quando você criar uma instância de banco de dados. O Amazon RDS cria e associa uma interface de rede elástica à sua instância de banco de dados com esse endereço IP. A instância de banco de dados usa a zona de disponibilidade que contém a sub-rede.

Para implantações multi-AZ, a definição de uma sub-rede para duas ou mais zonas de disponibilidade em uma região da Região da AWS permite que o Amazon RDS crie uma nova espera em outra zona de disponibilidade, caso haja necessidade. Você precisa fazer isso mesmo para implantações single-AZ, caso queira convertê-las em implantações multi-AZ em algum momento.

Nesta etapa, você criará um grupo de sub-redes de banco de dados e adicionará as sub-redes criadas à sua VPC.

Como criar um grupo de sub-redes de banco de dados

1. Abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Subnet groups (Grupos de sub-redes).
3. Escolha Create DB Subnet Group (Criar grupo de sub-redes de banco de dados).
4. Em Name (Nome), digite o nome do grupo de sub-redes de banco de dados.
5. Em Description (Descrição), digite uma descrição para o grupo de sub-redes de banco de dados.
6. Para VPC, escolha a VPC padrão ou a VPC criada por você.
7. Na seção Adicionar sub-redes, escolha as zonas de disponibilidade que incluem as sub-redes de Zonas de disponibilidade e escolha as sub-redes de Sub-redes.


RDS > Subnet groups > Create DB subnet group
Create DB Subnet Group
To create a new subnet group, give it a name and a description, and choose an existing VPC. You will then be able to add subnets related to that VPC.
Subnet group details
Name
You won't be able to modify the name after your subnet group has been created.
mydbsubnetgroup
Must contain from 1 to 255 characters. Alphanumeric characters, spaces, hyphens, underscores, and periods are allowed.
Description
My DB Subnet Group
VPC
Choose a VPC identifier that corresponds to the subnets you want to use for your DB subnet group. You won't be able to choose a different VPC identifier after your subnet group has been created.
tutorial-vpc (vpc-068fe388385afc014)
Add subnets
Availability Zones
Choose the Availability Zones that include the subnets you want to add.
Choose an availability zone
us-east-1a
us-east-1c
Subnets
Choose the subnets that you want to add. The list includes the subnets in the selected Availability Zones.
Select subnets
subnet-079bd4b8953aee1dd (10.0.0.0/24)
subnet-057e85b72c46fdd9a (10.0.1.0/24)
Subnets selected (2)
Availability zone
Subnet ID
CIDR block
us-east-1a
subnet-079bd4b8953aee1dd
10.0.0.0/24
us-east-1c
subnet-057e85b72c46fdd9a
10.0.1.0/24
Cancel
Create

> **Note**
>
> Se você habilitou uma zona local, poderá escolher um grupo de zonas de disponibilidade na página Create DB subnet group (Criar grupo de sub-redes de banco de dados). Nesse caso, escolha o Availability Zone group (Grupo de zonas de disponibilidade), Availability Zones (Zonas de disponibilidade) e Subnets (Sub-redes).

8. Escolha Create (Criar).

   Seu novo grupo aparece na lista de grupos de sub-redes de banco de dados no console do RDS. Você pode selecionar o grupo de sub-redes de banco de dados para obter detalhes, incluindo todas as sub-redes associadas a esse grupo, no painel de detalhes, na parte inferior da janela.

Etapa 3: Criar um grupo de segurança da VPC

Antes de criar uma instância de banco de dados, você pode criar um grupo de segurança da VPC para associar a essa instância. Se você não criar um grupo de segurança da VPC, poderá usar o grupo de segurança padrão ao criar uma instância de banco de dados. Para obter instruções sobre como criar um grupo de segurança para a instância de banco de dados, consulte Criar um grupo de segurança da VPC para uma instância de banco de dados privada ou Controle o tráfego para recursos usando grupos de segurança no Guia do usuário da Amazon VPC.

Etapa 4: Criar uma instância de banco de dados na VPC

Nessa etapa, você cria uma instância de banco de dados e usa o nome da VPC, o grupo de sub-redes de banco de dados e o grupo de segurança de VPC que você criou nas etapas anteriores.

> **Note**
>
> Se quiser que sua instância de banco de dados na VPC seja acessível ao público geral, você deverá habilitar os atributos DNS hostnames (Nomes de host de DNS) e DNS resolution (Resolução de DNS). Para ter mais informações, consulte Atributos de DNS para sua VPC no Guia do usuário da Amazon VPC.

Para obter detalhes sobre como criar uma instância de banco de dados, consulte Criar uma instância de banco de dados do Amazon RDS.

Quando solicitado na seção Connectivity (Conectividade), insira o nome da VPC, o grupo de sub-redes de banco de dados e o grupo de segurança da VPC.

Atualizar a VPC para uma instância de banco de dados

É possível usar o AWS Management Console para mover a instância de banco de dados para uma VPC diferente.

Para mais informações sobre a modificação de uma instância de banco de dados , consulte Modificar uma instância de banco de dados do Amazon RDS. Na seção Connectivity (Conectividade) da página de modificações mostrada a seguir, insira o novo grupo de sub-redes do banco de dados em DB subnet group (Grupo de sub-redes do banco de dados). O novo grupo de sub-redes deve ser um grupo em uma nova VPC.



Não é possível alterar a VPC de uma instância de banco de dados se as seguintes condições se aplicarem:

- A instância de banco de dados está em várias zonas de disponibilidade. Você pode converter a instância de banco de dados em uma única zona de disponibilidade, movê-la para uma nova VPC e convertê-la de volta para uma instância de banco de dados multi-AZ. Para obter mais informações, consulte Configurar e gerenciar uma implantação multi-AZ para o Amazon RDS.
- A instância de banco de dados possui uma ou mais réplicas de leitura. Você pode remover as réplicas de leitura, mover a instância de banco de dados para uma nova VPC e adicionar as réplicas de leitura novamente. Para obter mais informações, consulte Trabalhar com réplicas de leitura de instância de banco de dados.
- A instância de banco de dados é uma réplica de leitura Você pode promover a réplica de leitura e, em seguida, mover a instância de banco de dados autônoma para uma nova VPC. Para obter mais informações, consulte Promoção de uma réplica de leitura a uma instância de banco de dados autônoma.

- O grupo de sub-redes na VPC de destino não tem sub-redes na zona de disponibilidade da instância de banco de dados. Você pode adicionar sub-redes na zona de disponibilidade da instância de banco de dados ao grupo de sub-redes de banco de dados e, em seguida, mover a instância de banco de dados para a nova VPC. Para ter mais informações, consulte Trabalhar com grupos de sub-redes de banco de dados.

Cenários para acessar uma instância de banco de dados em uma VPC

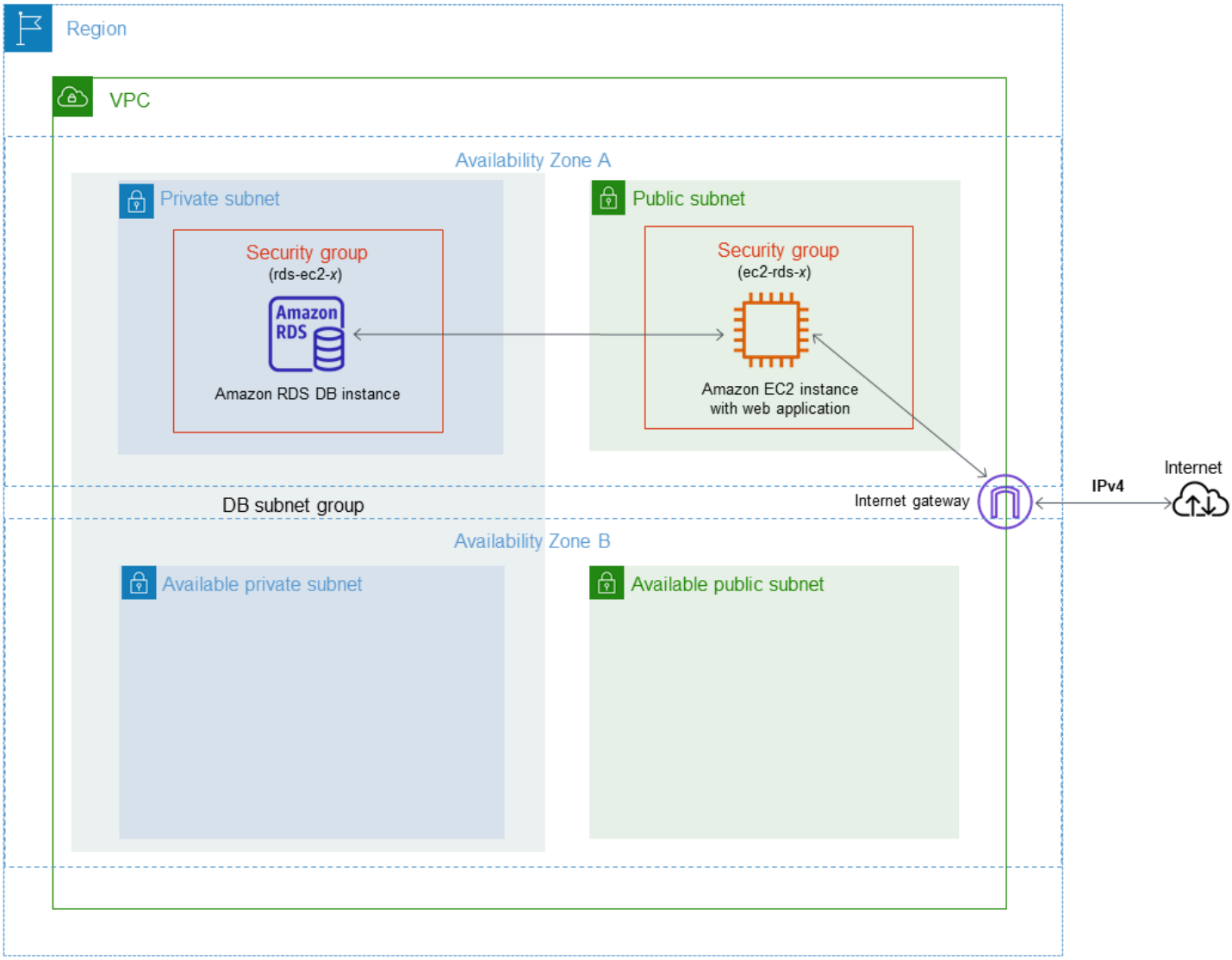
O Amazon RDS é compatível com os seguintes cenários de acesso a uma instância de banco de dados em uma VPC:

- Uma instância do Amazon EC2 na mesma VPC
- Uma instância do EC2 em uma VPC diferente
- Uma aplicação cliente pela Internet
- Uma rede privada

Uma instância de banco de dados em uma VPC acessada por uma instância do Amazon EC2 na mesma VPC.

Um uso comum de uma instância de banco de dados em uma VPC é compartilhar dados com um servidor de aplicações que está sendo executado em uma instância do Amazon EC2 na mesma VPC.

O diagrama a seguir mostra esse cenário.



A maneira mais simples de gerenciar o acesso entre instâncias do EC2 e instâncias de banco de dados na mesma VPC é fazer o seguinte:

- Crie um grupo de segurança de VPC no qual as suas instâncias de banco de dados estarão. Esse grupo de segurança pode ser usado para restringir o acesso às instâncias de banco de dados. Por exemplo, você pode criar uma regra personalizada para esse grupo de segurança. Isso pode permitir o acesso TCP usando a porta que você atribuiu à instância de banco de dados quando a criou e um endereço IP para acessar a instância de banco de dados para desenvolvimento ou outros fins.
- Crie um grupo de segurança da VPC em que as suas instâncias do EC2 (servidores Web e clientes) estarão. Esse grupo de segurança pode, se necessário, permitir o acesso à instância do

EC2 pela Internet usando a tabela de roteamento da VPC. Por exemplo, você pode definir regras nesse grupo de segurança para permitir o acesso TCP à instância do EC2 pela porta 22.

- Crie regras personalizadas no grupo de segurança para suas instâncias de banco de dados que permitam conexões do grupo de segurança que você criou para suas instâncias do EC2. Essas regras podem permitir que qualquer membro do grupo de segurança acesse as instâncias de banco de dados.

Há uma sub-rede pública e privada adicional em uma zona de disponibilidade separada. Um grupo de sub-redes do banco de dados RDS exige uma sub-rede em, pelo menos, duas zonas de disponibilidade. A sub-rede adicional facilita a migração para uma implantação de instância de banco de dados multi-AZ no futuro.

Para um tutorial que mostra como criar uma VPC com sub-redes públicas e privadas para esse cenário, consulte Tutorial: Criar uma VPC para usar com uma instância de banco de dados (somente IPv4).

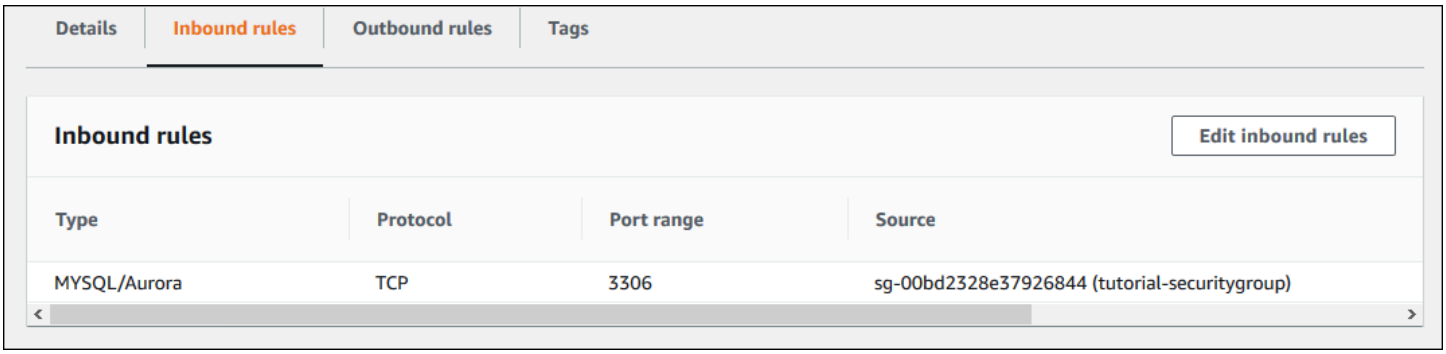
> **Tip**
>
> Você pode configurar a conectividade de rede entre uma instância do Amazon EC2 e uma instância de banco de dados automaticamente ao criar a instância de banco de dados. Para obter mais informações, consulte Configurar a conectividade automática de rede com uma instância do EC2.

Para criar uma regra em um grupo de segurança de VPC que permita conexões de outro grupo de segurança, faça o seguinte:

1. Faça login no AWS Management Console e abra o console da Amazon VPC em https://console.aws.amazon.com/vpc.
2. No painel de navegação, selecione Grupos de segurança.
3. Escolha ou crie um grupo de segurança ao qual você deseja conceder acesso para membros de outro grupo de segurança. No cenário anterior, esse é o grupo de segurança usado para suas instâncias de banco de dados. Vá para a guia Inbound Rules (Regras de entrada) e escolha Edit rules (Editar regras).
4. Na página Edit inbound rules (Editar regras de entrada), escolha Add Rule (Adicionar regra).
5. Em Type (Tipo), escolha a entrada que corresponde à porta usada ao criar sua instância de banco de dados, como MYSQL/Aurora.

6. Na caixa Origem, comece a digitar o ID do grupo de segurança para listar os grupos de segurança correspondentes. Escolha o grupo de segurança com membros que você deseja que tenham acesso aos recursos protegidos por esse grupo de segurança. Esse é o grupo de segurança usado para sua instância do EC2 no cenário anterior.
7. Se necessário, repita as etapas para o protocolo TCP, criando uma regra com Todos os TCP como Tipo e seu grupo de segurança na caixa Origem. Se você pretende usar o protocolo UDP, crie uma regra com All UDP (Todos os UDP) como Type (Tipo) e seu grupo de segurança em Source (Origem).
8. Escolha Salvar regras.

A tela a seguir mostra uma regra de entrada com um grupo de segurança como origem.

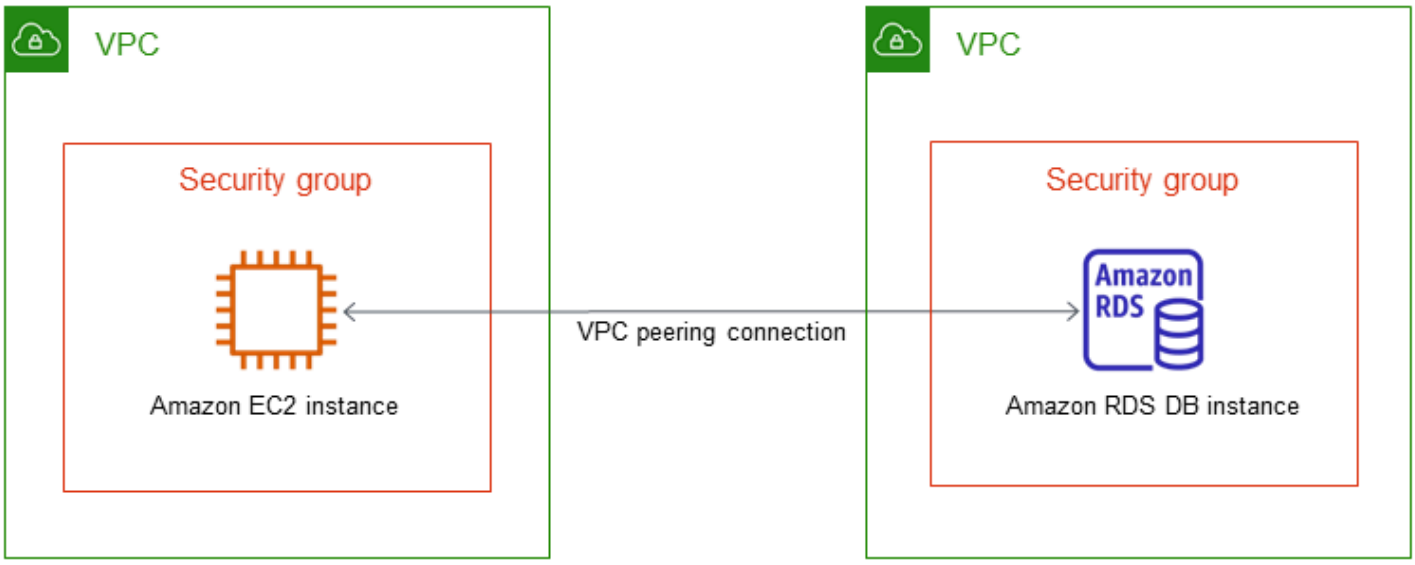


Para ter mais informações sobre como se conectar ao à instância de banco de dados por meio da instância do EC2, consulte Conectar a uma instância de banco de dados do Amazon RDS.

Uma instância de banco de dados em uma VPC acessada por uma instância do EC2 em uma VPC diferente

Quando sua instância de banco de dados está em uma VPC diferente da instância do EC2 que você está usando para acessá-la, você pode usar emparelhamento de VPC para acessar a instância de banco de dados.

O diagrama a seguir mostra esse cenário.

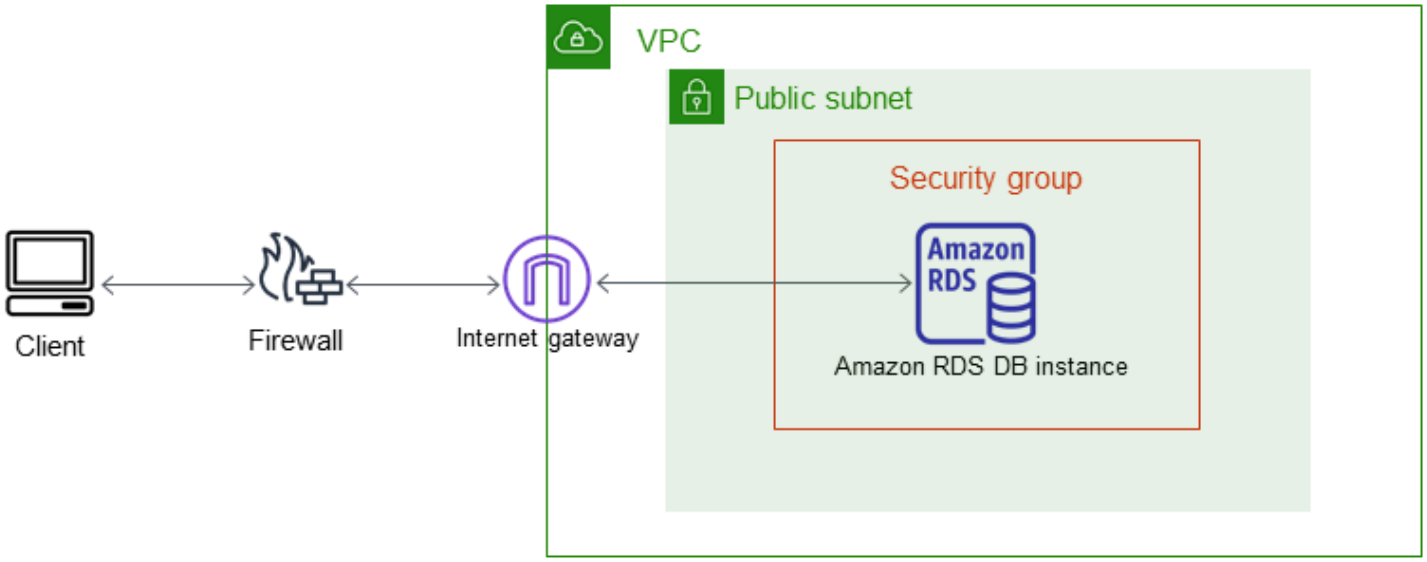


Uma conexão de emparelhamento da VPC é uma conexão de redes entre duas VPCs que permite direcionar o tráfego entre elas usando endereços IP privados. Recursos em qualquer VPC podem se comunicar uns com os outros como se estivessem na mesma rede. Você pode criar uma conexão de emparelhamento da VPC entre suas próprias VPCs, com uma VPC em outra conta da AWS ou com uma VPC em uma Região da AWS diferente. Para saber mais sobre o emparelhamento de VPCs, consulte Emparelhamento de VPCs, no Guia do usuário do Amazon Virtual Private Cloud.

Uma instância de banco de dados em uma VPC acessada por uma aplicação cliente pela internet

Para acessar uma instância de banco de dados em uma VPC de uma aplicação cliente via Internet, configure uma VPC com uma sub-rede pública única e um gateway da Internet para permitir a comunicação pela Internet.

O diagrama a seguir mostra esse cenário.



Recomendamos a seguinte configuração:

- Uma VPC de tamanho /16 (por exemplo, CIDR: 10.0.0.0/16). Esse tamanho fornece 65.536 endereços IP privados.
- Uma sub-rede de tamanho /24 (por exemplo CIDR: 10.0.0.0/24). Esse tamanho fornece 256 endereços IP privados.
- Um cluster de banco de dados do Amazon RDS associado à VPC e à sub-rede. O Amazon RDS atribui um endereço IP da sub-rede à sua instância de banco de dados.
- Um gateway da Internet que conecta a VPC à Internet e a outros produtos da AWS.
- Um grupo de segurança associado com a instância de banco de dados. As regras de entrada de grupo de segurança permitem que sua aplicação cliente acesse a sua instância de banco de dados.

Para ter mais informações sobre como criar uma instância de banco de dados em uma VPC, consulte Criar uma instância de banco de dados em uma VPC.

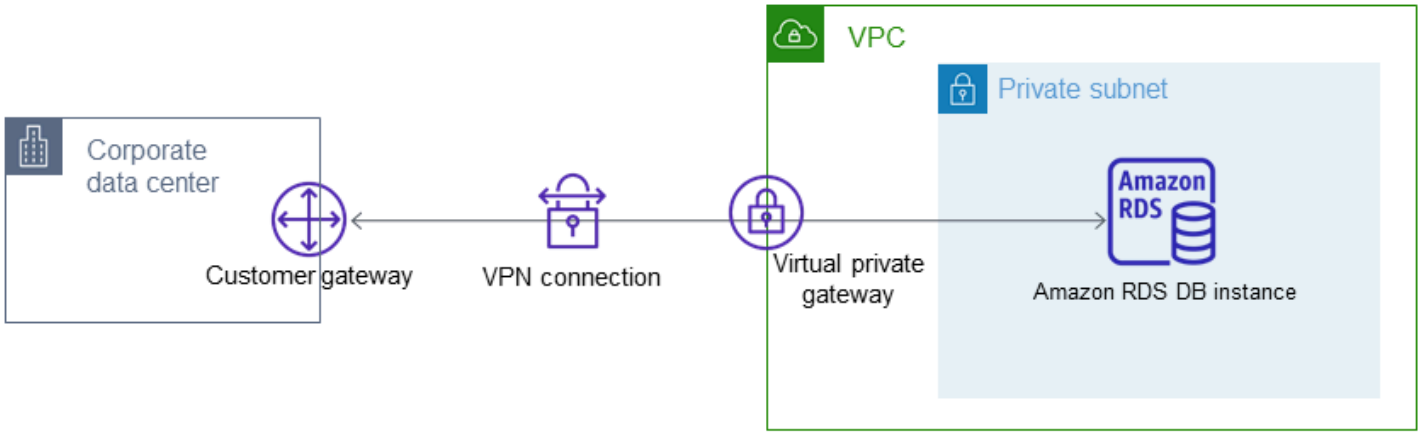
Uma instância de banco de dados em uma VPC acessada por uma rede privada

Se sua instância de banco de dados não estiver acessível publicamente, você terá as seguintes opções para acessá-la a partir de uma rede privada:

- Uma conexão AWS Site-to-Site VPN. Para ter mais informações, consulte O que é o AWS Site-to-Site VPN?

- Uma conexão AWS Direct Connect. Para obter mais informações, consulte O que é o AWS Direct Connect?
- Uma conexão AWS Client VPN. Para ter mais informações, consulte O que é o AWS Client VPN?

O diagrama a seguir mostra um cenário com uma conexão AWS Site-to-Site VPN.

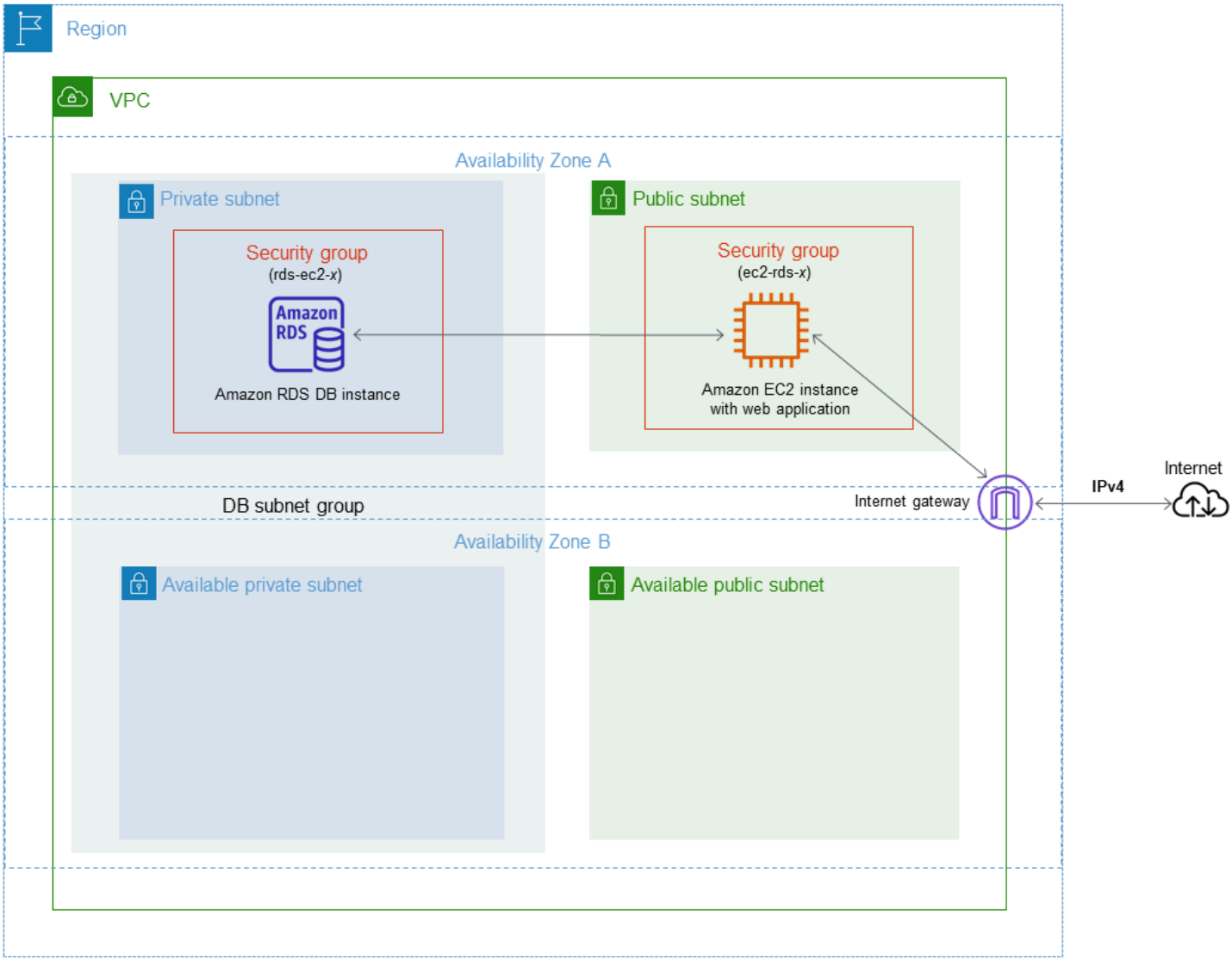


Para ter mais informações, consulte Privacidade do tráfego entre redes.

Tutorial: Criar uma VPC para usar com uma instância de banco de dados (somente IPv4)

Um cenário comum inclui uma instância de banco de dados em uma nuvem privada virtual (VPC) com base no serviço Amazon VPC. Essa VPC compartilha dados com um servidor Web em execução na mesma VPC. Neste tutorial, você cria a VPC para esse cenário.

O diagrama a seguir mostra esse cenário. Para obter informações sobre outros cenários, consulte Cenários para acessar uma instância de banco de dados em uma VPC.



Sua instância de banco de dados só precisa estar disponível para seu servidor Web e não para a Internet pública. Assim, você cria uma VPC com sub-redes públicas e privadas. O servidor Web está hospedado na sub-rede pública, para que ele possa chegar à Internet pública. A instância de banco

de dados está hospedada em uma sub-rede privada. O servidor Web pode se conectar à instância de banco de dados porque ela está hospedada na mesma VPC. Mas a instância de banco de dados não está disponível para a Internet pública, o que oferece maior segurança.

Esse tutorial configura uma sub-rede pública e privada adicional em uma zona de disponibilidade separada. Essas sub-redes não são usadas no tutorial. Um grupo de sub-redes do banco de dados RDS exige uma sub-rede em, pelo menos, duas zonas de disponibilidade. A sub-rede adicional facilita a migração para uma implantação de instância de banco de dados multi-AZ no futuro.

Este tutorial descreve como configurar uma VPC para instâncias de banco de dados do Amazon RDS. Para obter um tutorial que mostra como criar um servidor Web para esse cenário de VPC, consulte Tutorial: crie um servidor Web e uma instância de banco de dados do Amazon RDS. Para obter mais informações sobre uma Amazon VPC, consulte o Guia de conceitos básicos da Amazon VPC e Guia do usuário da Amazon VPC.

> **Tip**
>
> Você pode configurar a conectividade de rede entre uma instância do Amazon EC2 e uma instância de banco de dados automaticamente ao criar a instância de banco de dados. A configuração da rede é semelhante à descrita neste tutorial. Para obter mais informações, consulte Configurar a conectividade automática de rede com uma instância do EC2.

Criar uma VPC com sub-redes públicas e privadas

Use o seguinte procedimento para criar uma VPC com sub-redes públicas e privadas.

Para criar uma VPC e sub-redes

1. Abra o console da Amazon VPC, em https://console.aws.amazon.com/vpc/.
2. No canto superior direito do AWS Management Console, escolha a região na qual deseja criar sua VPC. Este exemplo usa a região Oeste dos EUA (Oregon).
3. No canto superior esquerdo, escolha VPC dashboard (Painel da VPC). Para começar a criar uma VPC, escolha Create VPC (Criar VPC).
4. Para Resources to create (Recursos a serem criados) em VPC settings (Configurações da VPC), escolha VPC and more (VPC e mais).
5. Para VPC settings (Configurações da VPC), defina os seguintes valores:

- Name tag auto-generation (Geração automática da etiqueta de nome): **`tutorial`**
- IPv4 CIDR block (Bloco CIDR IPv4): **`10.0.0.0/16`**
- IPv6 CIDR block (Bloco CIDR IPv6): No IPv6 CIDR Block (Nenhum bloco CIDR IPv6)
- Tenancy (Locação): Default (Padrão)
- Number of Availability Zones (AZs) [Número de zonas de disponibilidade (AZs)]: 2
- Customize AZs (Personalizar AZs): mantenha os valores padrão.
- Number of public subnet (Número de sub-redes públicas):2
- Number of private subnets (Número de sub-redes privadas): 2
- Customize subnets CIDR blocks (Personalizar blocos CIDR de sub-redes): mantenha os valores padrão.
- NAT gateways ($) (Gateways NAT ($)): None (Nenhum)
- VPC endpoints (Endpoints da VPC): None (Nenhum)
- DNS options (Opções de DNS): mantenha os valores padrão.

> **Note**
>
> O Amazon RDS exige pelo menos duas sub-redes em duas zonas de disponibilidade diferentes para oferecer suporte a implantações de instâncias de banco de dados multi-AZ. Este tutorial cria uma implantação single-AZ, mas o requisito facilita a conversão para uma implantação de instância de banco de dados multi-AZ no futuro.

6. Escolha Criar VPC.

Criar um grupo de segurança de VPC para um servidor Web público

Em seguida, você criará um grupo de segurança para acesso público. Para se conectar a instâncias públicas do EC2 em sua VPC, adicione regras de entrada ao seu grupo de segurança de VPC. Isso permite que o tráfego se conecte pela Internet.

Para criar um grupo de segurança de VPC

1. Abra o console da Amazon VPC, em https://console.aws.amazon.com/vpc/.
2. Escolha VPC Dashboard (Painel da VPC), Security Groups (Grupos de segurança) e depois Create grupo de segurança (Criar grupo de segurança).

3. Na página Create grupo de segurança (Criar grupo de segurança), defina esses valores:

   - Security group name (Nome do grupo de segurança: **`tutorial-securitygroup`**
   - Descrição: **`Tutorial Security Group`**
   - VPC: escolha a VPC criada na etapa anterior; por exemplo, vpc-***`identifier`*** (tutorial-vpc)

4. Adicione regras de entrada ao grupo de segurança.

   a. Determine o endereço IP a ser usado para se conectar a instâncias do EC2 em sua VPC usando Secure Shell (SSH). Para determinar seu endereço IP público, em uma janela ou guia diferente do navegador, é possível usar o serviço em https://checkip.amazonaws.com. Um exemplo de endereço IP: `203.0.113.25/32`.

      Em muitos casos, você pode se conectar por meio de um provedor de serviços de Internet (ISP) ou atrás de um firewall sem um endereço IP estático. Se sim, especifique o intervalo de endereços IP utilizado por computadores cliente.

      > **Warning**
      >
      > Se usar `0.0.0.0/0` para acesso do SSH, você possibilitará que todos os endereços IP acessem suas instâncias públicas usando SSH. Essa abordagem é aceitável por um período curto em um ambiente de teste, mas não é seguro em ambientes de produção. Em produção, autorize somente um endereço IP específico ou um intervalo de endereços para acessar suas instâncias usando SSH.

   b. Na seção Regras de entrada, escolha Adicionar regra.

   c. Defina os valores a seguir para a sua nova regra de entrada, para possibilitar o acesso do SSH à sua instância do Amazon EC2. Se você fizer isso, poderá se conectar à sua instância do Amazon EC2 para instalar o servidor Web e outros utilitários. Você também se conecta à sua instância do EC2 para fazer upload de conteúdo para seu servidor Web.

      - Digite: **`SSH`**
      - Origem: o endereço IP ou o intervalo da etapa A, por exemplo: **`203.0.113.25/32`**.

   d. Escolha Add rule (Adicionar regra).

   e. Defina os seguintes valores para a sua nova regra de entrada a fim de permitir o acesso HTTP ao servidor Web.

      - Digite: **`HTTP`**

- Origem: **`0.0.0.0/0`**

5. Escolha Create grupo de segurança (Criar grupo de segurança) para criar o grupo de segurança.

   Anote o ID do grupo de segurança porque você precisa dele posteriormente neste tutorial.

Criar um grupo de segurança da VPC para uma instância de banco de dados privada

Para manter sua instância de banco de dados particular, crie um segundo grupo de segurança para acesso privado. Para se conectar a instâncias privadas de banco de dados na VPC, você adiciona regras de entrada ao grupo de segurança da VPC que permitem o tráfego somente pelo seu servidor web.

Para criar um grupo de segurança de VPC

1. Abra o console da Amazon VPC, em https://console.aws.amazon.com/vpc/.
2. Escolha VPC Dashboard (Painel da VPC), Security Groups (Grupos de segurança) e depois Create grupo de segurança (Criar grupo de segurança).
3. Na página Create grupo de segurança (Criar grupo de segurança), defina esses valores:
   - Security group name (Nome do grupo de segurança: **`tutorial-db-securitygroup`**
   - Descrição: **`Tutorial DB Instance Security Group`**
   - VPC: escolha a VPC criada na etapa anterior; por exemplo, vpc-***`identifier`*** (tutorial-vpc)
4. Adicione regras de entrada ao grupo de segurança.
   a. Na seção Regras de entrada, escolha Adicionar regra.
   b. Defina os valores a seguir para a sua nova regra de entrada, para permitir o tráfego MySQL na porta 3306 de sua instância do Amazon EC2. Se fizer isso, você poderá se conectar de seu servidor Web à sua instância de banco de dados. Ao fazer isso, você poderá armazenar e recuperar dados de sua aplicação Web para seu banco de dados.
      - Digite: **`MySQL/Aurora`**
      - Source (Origem): o identificador do grupo de segurança tutorial-securitygroup criado anteriormente neste tutorial; por exemplo, sg-9edd5cfb.
5. Escolha Create grupo de segurança (Criar grupo de segurança) para criar o grupo de segurança.

Criar um grupo de sub-redes de banco de dados

Um grupo de sub-redes de banco de dados é uma coleção de sub-redes que você cria em uma VPC e depois designa para suas instâncias de bancos de dados. Um grupo de sub-redes de banco de dados permite que você especifique uma VPC ao criar instâncias de banco de dados.

Para criar um grupo de sub-redes de banco de dados

1. Identifique as sub-redes privadas do seu banco de dados na VPC.

   a. Abra o console da Amazon VPC, em https://console.aws.amazon.com/vpc/.

   b. Escolha VPC Dashboard (Painel da VPC), depois selecione Subnets (Sub-redes).

   c. Anote os IDs das sub-redes chamadas tutorial-subnet-private1-us-west-2a e tutorial-subnet-private2-us-west-2b.

      Você precisará dos IDs de sub-rede ao criar seu grupo de sub-redes de banco de dados.

2. Abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.

   Conecte-se ao console do Amazon RDS, não ao console da Amazon VPC.

3. No painel de navegação, escolha Subnet groups (Grupos de sub-redes).

4. Escolha Create DB Subnet Group (Criar grupo de sub-redes de banco de dados).

5. Na página Create DB subnet group (Criar grupo de sub-redes de banco de dados), defina esses valores em Subnet group details (Detalhes do grupo de sub-redes):

   - Nome: **`tutorial-db-subnet-group`**
   - Descrição: **`Tutorial DB Subnet Group`**
   - VPC: tutorial-vpc (vpc-***`identifier`***)

6. Na seção Adicionar sub-redes, escolha Zonas de disponibilidade e Sub-redes.

   Para este tutorial, escolha us-west-2a e us-west-2b para as Availability Zones (Zonas de disponibilidade). Para Subnets (Sub-redes), escolha as sub-redes privadas que você identificou na etapa anterior.

7. Escolha Criar.

   Seu novo grupo aparece na lista de grupos de sub-redes de banco de dados no console do RDS. Você pode selecionar o grupo de sub-redes de banco de dados para visualizar detalhes

no painel de detalhes na parte inferior da janela. Esses detalhes incluem todas as sub-redes associadas ao grupo.

> **Note**
>
> Se você criou essa VPC para concluir Tutorial: crie um servidor Web e uma instância de banco de dados do Amazon RDS, crie a instância de banco de dados seguindo as instruções em Criar uma instância de banco de dados do Amazon RDS .

Como excluir a VPC

Depois de criar a VPC e outros recursos para este tutorial, você poderá excluí-los se deixarem de ser necessários.

> **Note**
>
> Se você incluiu recursos na VPC que criou para este tutorial, talvez seja necessário excluí-los para poder excluir a VPC. Por exemplo, esses recursos podem incluir instâncias do Amazon EC2 ou instâncias de banco de dados do Amazon RDS. Para obter mais informações, consulte Como excluir a sua VPC no Guia do usuário da Amazon VPC.

Para excluir uma VPC e recursos relacionados

1. Exclua o grupo de sub-redes de banco de dados.
   a. Abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
   b. No painel de navegação, escolha Subnet groups (Grupos de sub-redes).
   c. Selecione o grupo de sub-rede de banco de dados que deseja excluir, como, por exemplo, tutorial-db-subnet-group.
   d. Escolha Delete (Excluir) e, em seguida, Delete (Excluir) na janela de confirmação.
2. Anote o ID da VPC.
   a. Abra o console da Amazon VPC, em https://console.aws.amazon.com/vpc/.
   b. Escolha VPC Dashboard (Painel da VPC) e, em seguida, VPCs.
   c. Na lista, identifique a VPC que criou, como tutorial-vpc.

d. Anote o VPC ID (ID da VPC) da VPC que você criou. Você precisará do ID da VPC nas etapas posteriores.

3. Exclua os grupos de segurança.

   a. Abra o console da Amazon VPC, em https://console.aws.amazon.com/vpc/.
   b. Escolha VPC Dashboard (Painel da VPC) e, em seguida, Security Groups (Grupos de segurança).
   c. Selecione o grupo de segurança para a instância de banco de dados do Amazon RDS, como, por exemplo, tutorial-db-securitygroup.
   d. Em Actions (Ações), escolha Delete grupo de seguranças (Excluir grupos de segurança) e, depois, Delete (Excluir) na página de confirmação.
   e. Na página Security Groups (Grupos de segurança), selecione o grupo de segurança para a instância do Amazon EC2, como, por exemplo, tutorial-securitygroup.
   f. Em Actions (Ações), escolha Delete grupo de seguranças (Excluir grupos de segurança) e, depois, Delete (Excluir) na página de confirmação.

4. Exclua a VPC.

   a. Abra o console da Amazon VPC, em https://console.aws.amazon.com/vpc/.
   b. Escolha VPC Dashboard (Painel da VPC) e, em seguida, VPCs.
   c. Selecione a VPC que deseja excluir, como, por exemplo, tutorial-vpc.
   d. Em Actions (Ações), escolha Delete VPC (Excluir a VPC).

      A página de confirmação mostra outros recursos associados à VPC que também serão excluídos, incluindo as sub-redes associadas a ela.

   e. Na página de confirmação, insira **`delete`** e, em seguida, escolha Delete (Excluir).

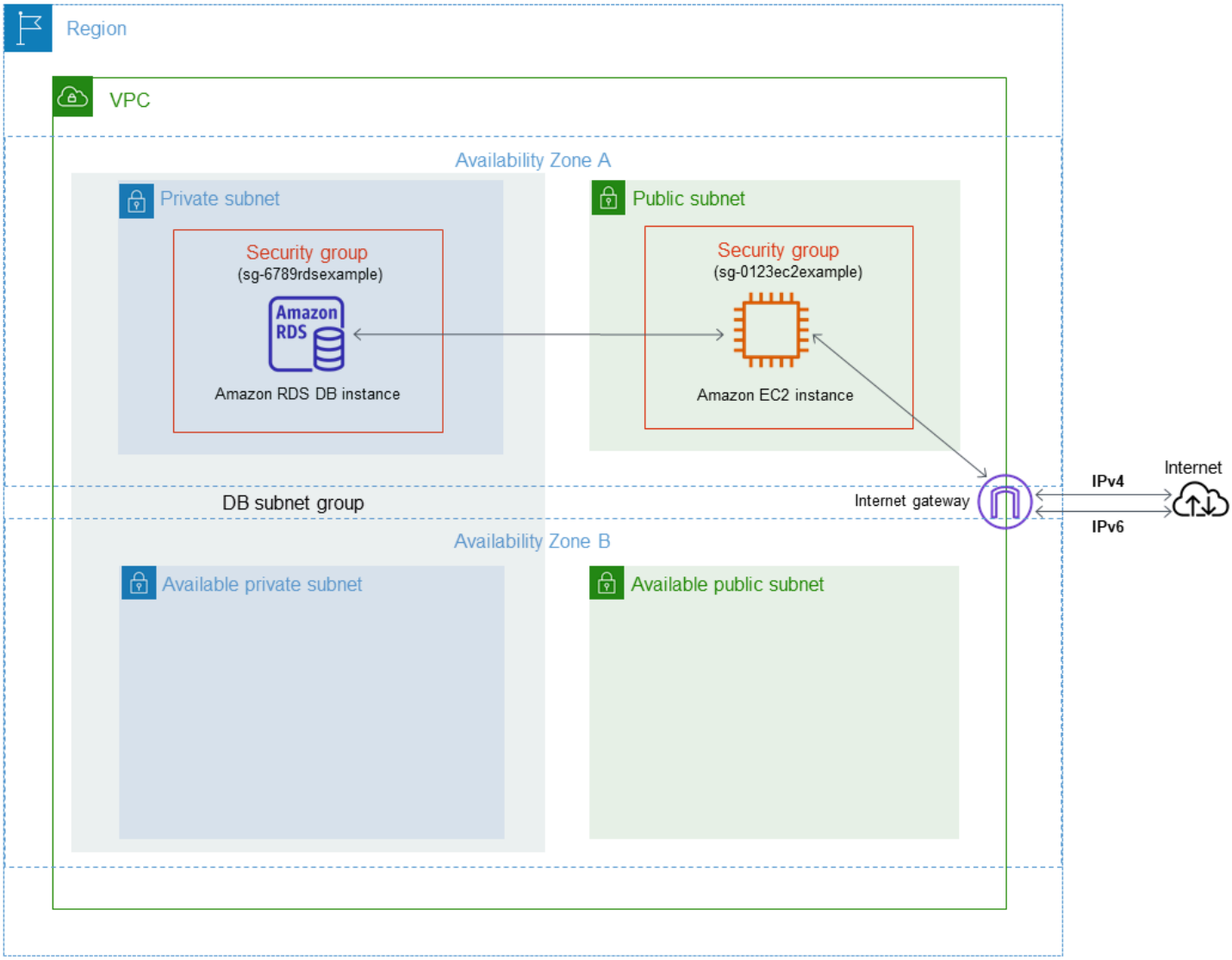
Tutorial: Criar uma VPC para uso com uma instância de banco de dados (modo de pilha dupla)

Um cenário comum inclui uma instância de banco de dados em uma nuvem privada virtual (VPC) com base no serviço Amazon VPC. Essa VPC compartilha dados com uma instância pública do Amazon EC2 que está sendo executada na mesma VPC.

Neste tutorial, você criará a VPC para esse cenário que funciona com um banco de dados em execução no modo de pilha dupla. Modo de pilha dupla para permitir a conexão pelo protocolo de endereçamento IPv6. Para obter mais informações sobre endereçamento IP, consulte Endereçamento IP do Amazon RDS.

As instâncias de rede de pilha dupla são compatíveis na maioria das regiões. Para obter mais informações, consulte Disponibilidade de região e versão. Para ver as limitações do modo de pilha dupla, consulte Limitações para instâncias de banco de dados de rede de pilha dupla.

O diagrama a seguir mostra esse cenário.



Para obter informações sobre outros cenários, consulte Cenários para acessar uma instância de banco de dados em uma VPC.

Sua instância de banco de dados só precisa estar disponível para sua instância do Amazon EC2 e não para a Internet pública. Assim, você cria uma VPC com sub-redes públicas e privadas. A instância do Amazon EC2 é hospedada na sub-rede pública, para que ela possa acessar a Internet pública. A instância de banco de dados está hospedada em uma sub-rede privada. A instância do Amazon EC2 pode se conectar à instância de banco de dados porque ela está hospedada na mesma VPC. No entanto, a instância de banco de dados não está disponível para a Internet pública, o que oferece maior segurança.

Esse tutorial configura uma sub-rede pública e privada adicional em uma zona de disponibilidade separada. Essas sub-redes não são usadas no tutorial. Um grupo de sub-redes do banco de dados

RDS exige uma sub-rede em, pelo menos, duas zonas de disponibilidade. A sub-rede adicional facilita a migração para uma implantação de instância de banco de dados multi-AZ no futuro.

Para criar uma instância de banco de dados que use o modo de pilha dupla, especifique Dual-stack mode (Modo de pilha dupla) para a configuração Network type (Tipo de rede). Você também pode modificar uma instância de banco de dados com a mesma configuração. Para ter mais informações, consulte Criar uma instância de banco de dados do Amazon RDS e Modificar uma instância de banco de dados do Amazon RDS.

Este tutorial descreve como configurar uma VPC para instâncias de banco de dados do Amazon RDS. Para obter mais informações sobre o Amazon VPC, consulte o Manual do usuário do Amazon VPC.

Criar uma VPC com sub-redes públicas e privadas

Use o seguinte procedimento para criar uma VPC com sub-redes públicas e privadas.

Para criar uma VPC e sub-redes

1. Abra o console da Amazon VPC em https://console.aws.amazon.com/vpc/.
2. No canto superior direito do AWS Management Console, escolha a região na qual deseja criar sua VPC. Este exemplo usa a região Leste dos EUA (Ohio).
3. No canto superior esquerdo, escolha VPC dashboard (Painel da VPC). Para começar a criar uma VPC, escolha Create VPC (Criar VPC).
4. Para Resources to create (Recursos a serem criados) em VPC settings (Configurações da VPC), escolha VPC and more (VPC e mais).
5. Para o restante de VPC settings (Configurações da VPC), defina os seguintes valores:

    - Name tag auto-generation (Geração automática da etiqueta de nome): **`tutorial-dual-stack`**
    - IPv4 CIDR block (Bloco CIDR IPv4): **`10.0.0.0/16`**
    - IPv6 CIDR block (Bloco CIDR IPv6): Amazon-provided IPv6 CIDR block (Bloco CIDR IPv6 fornecido pela Amazon)
    - Tenancy (Locação): Default (Padrão)
    - Number of Availability Zones (AZs) [Número de zonas de disponibilidade (AZs)]: 2
    - Customize AZs (Personalizar AZs): mantenha os valores padrão.
    - Number of public subnet (Número de sub-redes públicas):2

- Number of private subnets (Número de sub-redes privadas): 2
- Customize subnets CIDR blocks (Personalizar blocos CIDR de sub-redes): mantenha os valores padrão.
- NAT gateways ($) (Gateways NAT ($)): None (Nenhum)
- Egress only internet gateway (Gateway da Internet somente de saída): No (Não)
- VPC endpoints (Endpoints da VPC): None (Nenhum)
- DNS options (Opções de DNS): mantenha os valores padrão.

> **Note**
>
> O Amazon RDS exige pelo menos duas sub-redes em duas zonas de disponibilidade diferentes para oferecer suporte a implantações de instâncias de banco de dados multi-AZ. Este tutorial cria uma implantação single-AZ, mas o requisito facilita a conversão para uma implantação de instância de banco de dados multi-AZ no futuro.

6. Escolha Criar VPC.

Criar um grupo de segurança da VPC para uma instância pública do Amazon EC2

Em seguida, você criará um grupo de segurança para acesso público. Para se conectar a instâncias públicas do EC2 na sua VPC, adicione regras de entrada ao grupo de segurança de sua VPC que permitam que o tráfego se conecte da Internet.

Para criar um grupo de segurança de VPC

1. Abra o console da Amazon VPC em https://console.aws.amazon.com/vpc/.
2. Escolha VPC Dashboard (Painel da VPC), Security Groups (Grupos de segurança) e depois Create grupo de segurança (Criar grupo de segurança).
3. Na página Create grupo de segurança (Criar grupo de segurança), defina esses valores:
   - Security group name (Nome do grupo de segurança: **`tutorial-dual-stack-securitygroup`**
   - Descrição: **`Tutorial Dual-Stack Security Group`**
   - VPC: escolha a VPC criada na etapa anterior, por exemplo vpc-***`identifier`***(tutorial-dual-stack-vpc)

4. Adicione regras de entrada ao grupo de segurança.

   a. Determine o endereço IP a ser usado para se conectar a instâncias do EC2 em sua VPC usando Secure Shell (SSH).

      Um exemplo de endereço IPv4 (Internet Protocol versão 4) é `203.0.113.25/32`.
      Um exemplo de intervalo de endereços de Protocolo de Internet versão 6 (IPv6) é `2001:db8:1234:1a00::/64`.

      Em muitos casos, você pode se conectar por meio de um provedor de serviços de Internet (ISP) ou atrás de um firewall sem um endereço IP estático. Se sim, especifique o intervalo de endereços IP utilizado por computadores cliente.

      > **Warning**
      >
      > Se usar `0.0.0.0/0` para IPv4 ou `::0` para IPv6, você possibilitará que todos os endereços IP acessem suas instâncias públicas usando SSH. Essa abordagem é aceitável por um período curto em um ambiente de teste, mas não é seguro em ambientes de produção. No ambiente de produção, autorize apenas um endereço IP específico ou um intervalo de endereços a acessar as instâncias.

   b. Na seção Regras de entrada, escolha Adicionar regra.

   c. Defina os valores a seguir para a sua nova regra de entrada, para permitir o acesso Secure Shell (SSH) à sua instância do Amazon EC2. Se você fizer isso, poderá se conectar à instância do EC2 para instalar clientes SQL e outras aplicações. Especifique um endereço IP para que você possa acessar sua instância do EC2:

      - Digite: **`SSH`**
      - Source (Fonte): o endereço IP ou o intervalo da etapa a. Um exemplo de endereço IP IPv4 é **`203.0.113.25/32`**. Um exemplo de endereço IP IPv6 é **`2001:DB8::/32`**.

5. Escolha Create grupo de segurança (Criar grupo de segurança) para criar o grupo de segurança.

   Anote o ID do grupo de segurança porque você precisa dele posteriormente neste tutorial.

Criar um grupo de segurança da VPC para uma instância de banco de dados privada

Para manter sua instância de banco de dados particular, crie um segundo grupo de segurança para acesso privado. Para se conectar a instâncias de banco de dados privados em sua VPC, adicione regras de entrada ao seu grupo de segurança de VPC. Eles permitem o tráfego somente de sua instância do Amazon EC2.

Para criar um grupo de segurança de VPC

1. Abra o console da Amazon VPC em https://console.aws.amazon.com/vpc/.
2. Escolha VPC Dashboard (Painel da VPC), Security Groups (Grupos de segurança) e depois Create grupo de segurança (Criar grupo de segurança).
3. Na página Create grupo de segurança (Criar grupo de segurança), defina esses valores:
   - Security group name (Nome do grupo de segurança: **`tutorial-dual-stack-db-securitygroup`**
   - Descrição: **`Tutorial Dual-Stack DB Instance Security Group`**
   - VPC: escolha a VPC criada na etapa anterior, por exemplo vpc-***`identifier`***(tutorial-dual-stack-vpc)
4. Adicione regras de entrada ao grupo de segurança:
   a. Na seção Regras de entrada, escolha Adicionar regra.
   b. Defina os valores a seguir para a sua nova regra de entrada, para permitir o tráfego MySQL na porta 3306 de sua instância do Amazon EC2. Se fizer isso, você poderá se conectar de sua instância do EC2 à sua instância de banco de dados. Ao fazer isso, você poderá enviar dados da instância do EC2 para o banco de dados.
      - Digite: MySQL/Aurora
      - Source (Origem): o identificador do grupo de segurança tutorial-dual-stack-securitygroup criado anteriormente neste tutorial; por exemplo, sg-9edd5cfb.
5. Para criar o grupo de segurança, escolha Criar grupo de segurança.

Criar um grupo de sub-redes de banco de dados

Um grupo de sub-redes de banco de dados é uma coleção de sub-redes que você cria em uma VPC e depois designa para suas instâncias de bancos de dados. Ao usar um grupo de sub-redes de banco de dados, você pode especificar uma VPC específica ao criar instâncias de banco de dados.

Para criar um grupo de sub-redes de banco de dados que seja compatível com DUAL, todas as sub-redes devem ser compatíveis com DUAL. Para ser compatível com DUAL, uma sub-rede deve ter um CIDR IPv6 associado a ela.

Como criar um grupo de sub-redes de banco de dados

1. Identifique as sub-redes privadas do seu banco de dados na VPC.
   a. Abra o console da Amazon VPC em https://console.aws.amazon.com/vpc/.
   b. Escolha VPC Dashboard (Painel da VPC), depois selecione Subnets (Sub-redes).
   c. Anote os IDs das sub-redes chamadas tutorial-dual-stack-subnet-private1-us-west-2a e tutorial-dual-stack-subnet-private2-us-west-2b.

      Você precisará dos IDs de sub-rede ao criar seu grupo de sub-redes de banco de dados.
2. Abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.

   Conecte-se ao console do Amazon RDS, não ao console da Amazon VPC.
3. No painel de navegação, escolha Subnet groups (Grupos de sub-redes).
4. Escolha Create DB Subnet Group (Criar grupo de sub-redes de banco de dados).
5. Na página Create DB subnet group (Criar grupo de sub-redes de banco de dados), defina esses valores em Subnet group details (Detalhes do grupo de sub-redes):
   - Nome: **`tutorial-dual-stack-db-subnet-group`**
   - Descrição: **`Tutorial Dual-Stack DB Subnet Group`**
   - VPC: tutorial-dual-stack-vpc (vpc-***`identifier`***)
6. Na seção Add subnets (Adicionar sub-redes), escolha as opções Availability Zones (Zonas de disponibilidade) e Subnets (Sub-redes).

   Para este tutorial, escolha us-east-2a e us-east-2b para as Availability Zones (Zonas de disponibilidade). Para Subnets (Sub-redes), escolha as sub-redes privadas que você identificou na etapa anterior.
7. Escolha Create (Criar).

Seu novo grupo aparece na lista de grupos de sub-redes de banco de dados no console do RDS. Você pode escolher o grupo de sub-redes de banco de dados para ver os detalhes. Isso inclui os

protocolos de endereçamento compatíveis e todas as sub-redes associadas ao grupo e o tipo de rede compatível com o grupo de sub-redes de banco de dados.

Criar uma instância do Amazon EC2 no modo de pilha dupla

Para criar uma instância do Amazon EC2, siga as instruções em Iniciar uma instância usando o novo assistente de inicialização de instância, versão beta no Guia do usuário do Amazon EC2.

Na página Configure Instance Details (Configurar os detalhes da instância), defina esses valores e mantenha os outros valores como padrão:

- Rede: escolha uma VPC existente com sub-redes públicas e privadas, como tutorial-dual-stack-vpc (vpc-*identifier*), criada em Criar uma VPC com sub-redes públicas e privadas.
- Subnet (Sub-rede): escolha uma sub-rede pública existente, como subnet-***identifier*** | tutorial-dual-stack-subnet-public1-us-east-2a | us-east-2a criada em Criar um grupo de segurança da VPC para uma instância pública do Amazon EC2.
- Auto-assign Public IP (Atribuir automaticamente IP público): escolha Enable (Habilitar).
- Auto-assign IPv6 IP (Atribuir automaticamente IPv6): escolha Enable (Habilitar).
- Firewall (security groups) (Firewall (grupos de segurança)): escolha Select an existing security group (Selecionar um grupo de segurança existente).
- Common security groups (Grupos de segurança comuns): selecione um grupo de segurança existente, como o `tutorial-securitygroup` criado em Criar um grupo de segurança da VPC para uma instância pública do Amazon EC2. Verifique se o grupo de segurança escolhido inclui regras de entrada para Secure Shell (SSH) e acesso HTTP.

Criar uma instância de banco de dados no modo de pilha dupla

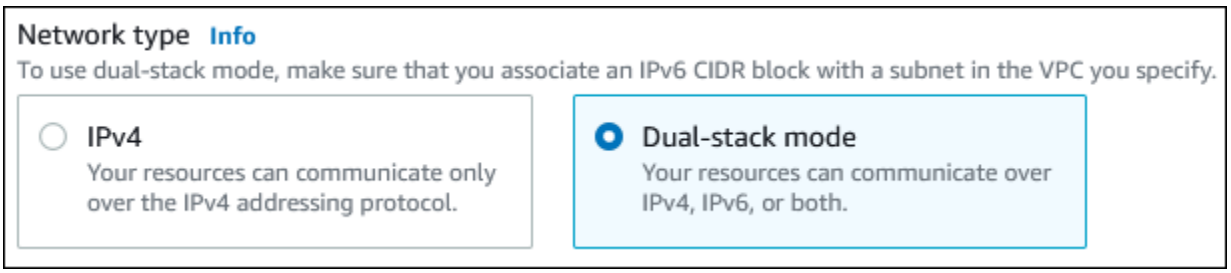
Nesta etapa, crie uma instância de banco de dados do Amazon RDS para execução no modo de pilha dupla.

Como criar uma instância de banco de dados

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No canto superior direito do console, escolha a Região da AWS na qual você quer criar a instância de banco de dados. Este exemplo usa a região Leste dos EUA (Ohio).
3. No painel de navegação, escolha Databases (Bancos de dados).

4. Selecione Criar banco de dados.
5. Na página Create database (Criar banco de dados), verifique se a opção Standard create (Criação padrão) está selecionada e escolha o tipo de mecanismo de banco de dados MySQL.
6. Na seção Conectividade, defina estes valores:

   - Network type (Tipo de rede): escolha Dual-stack mode (Modo de pilha dupla).



   - Virtual private cloud (VPC) (Nuvem privada virtual (VPC): escolha uma VPC existente com sub-redes públicas e privadas, como a tutorial-dual-stack-vpc (vpc-*identifier*) criada em Criar uma VPC com sub-redes públicas e privadas.

     A VPC deve ter sub-redes em zonas de disponibilidade diferentes.
   - DB subnet group (Grupo de sub-redes de banco de dados): escolha um grupo de sub-redes de banco de dados para a VPC, como tutorial-dual-stack-db-subnet-group criado em Criar um grupo de sub-redes de banco de dados.
   - Public access (Acesso público): escolha No (Não).
   - VPC security group (firewall) (Grupo de segurança da VPC (firewall)): selecione Choose existing (Escolher existente).
   - Existing VPC grupo de seguranças (Grupos de segurança da VPC existentes): escolha um grupo de segurança da VPC existente configurado para acesso privado, como tutorial-dual-stack-db-securitygroup criado em Criar um grupo de segurança da VPC para uma instância de banco de dados privada.

     Remova outros grupos de segurança, como o grupo de segurança padrão, escolhendo o X associado.
   - Availability Zone (Zona de disponibilidade): escolha us-west-2a.

     Para evitar o tráfego entre zonas, certifique-se de que a instância de banco de dados e a instância do EC2 estejam na mesma zona de disponibilidade.

7. Nas seções restantes, especifique suas configurações de instância de banco de dados. Para obter informações sobre cada configuração, consulte Configurações para instâncias de banco de dados.

Conectar-se à sua instância do Amazon EC2 e à instância de banco de dados

Depois de criar a instância do Amazon EC2 e a instância de banco de dados no modo de pilha dupla, você poderá se conectar a cada uma usando o protocolo IPv6. Para se conectar a uma instância do Amazon EC2 usando o protocolo IPv6, siga as instruções em Conecte-se à sua instância do Linux no Guia do usuário do Amazon EC2.

Para se conectar à instância de banco de dados do RDS para MySQL por meio da instância do Amazon EC2, siga as instruções em Conectar-se a uma instância de banco de dados MySQL.

Como excluir a VPC

Depois de criar a VPC e outros recursos para este tutorial, você poderá excluí-los se deixarem de ser necessários.

Se você incluiu recursos na VPC que criou para este tutorial, talvez seja necessário excluí-los para poder excluir a VPC. Exemplos de recursos são instâncias do Amazon EC2 ou instâncias de banco de dados. Para obter mais informações, consulte Como excluir a sua VPC no Guia do usuário da Amazon VPC.

Para excluir uma VPC e recursos relacionados

1. Exclua o grupo de sub-redes de banco de dados:
    a. Abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
    b. No painel de navegação, escolha Subnet groups (Grupos de sub-redes).
    c. Selecione o grupo de sub-redes de banco de dados a ser excluído, por exemplo, tutorial-db-subnet-group.
    d. Escolha Delete (Excluir) e, em seguida, Delete (Excluir) na janela de confirmação.
2. Anote o ID da VPC:
    a. Abra o console da Amazon VPC em https://console.aws.amazon.com/vpc/.
    b. Escolha VPC Dashboard (Painel da VPC) e, em seguida, VPCs.
    c. Na lista, identifique a VPC que você criou, como tutorial-dual-stack-vpc.

d. Anote o valor VPC ID (ID da VPC) da VPC que você criou. Você precisará do ID da VPC nas etapas subsequentes.

3. Exclua os grupos de segurança:

   a. Abra o console da Amazon VPC em https://console.aws.amazon.com/vpc/.

   b. Escolha VPC Dashboard (Painel da VPC) e, em seguida, Security Groups (Grupos de segurança).

   c. Selecione o grupo de segurança para a instância de banco de dados do Amazon RDS, como tutorial-dual-stack-db-securitygroup.

   d. Em Actions (Ações), escolha Delete grupo de seguranças (Excluir grupos de segurança) e, depois, Delete (Excluir) na página de confirmação.

   e. Na página Security Groups (Grupos de segurança), selecione o grupo de segurança para a instância do Amazon EC2, como tutorial-dual-stack-securitygroup.

   f. Em Actions (Ações), escolha Delete grupo de seguranças (Excluir grupos de segurança) e, depois, Delete (Excluir) na página de confirmação.

4. Exclua o gateway NAT:

   a. Abra o console da Amazon VPC em https://console.aws.amazon.com/vpc/.

   b. Escolha VPC Dashboard (Painel da VPC) e, em seguida, NAT Gateways (Gateways NAT).

   c. Selecione o gateway NAT da VPC que você criou. Use o ID da VPC para identificar o gateway NAT correto.

   d. Em Actions (Ações), escolha Delete NAT gateway (Excluir gateway NAT).

   e. Na página de confirmação, insira **`delete`** e, em seguida, escolha Delete (Excluir).

5. Exclua a VPC:

   a. Abra o console da Amazon VPC em https://console.aws.amazon.com/vpc/.

   b. Escolha VPC Dashboard (Painel da VPC) e, em seguida, VPCs.

   c. Selecione a VPC que deseja excluir, como a tutorial-dual-stack-vpc.

   d. Em Actions (Ações), escolha Delete VPC (Excluir a VPC).

      A página de confirmação mostra outros recursos associados à VPC que também serão excluídos, incluindo as sub-redes associadas a ela.

   e. Na página de confirmação, insira **`delete`** e, em seguida, escolha Delete (Excluir).

6. Libere os endereços de IP elásticos:

a. Abra o console do Amazon EC2 em https://console.aws.amazon.com/ec2/.
b. Escolha EC2 Dashboard (Painel do EC2) e, em seguida, Elastic IPs (IPs elásticos).
c. Selecione o endereço de IP elástico que deseja liberar.
d. Em Actions (Ações), escolha Release Elastic IP addresses (Liberar endereços de IP elásticos).
e. Na página de confirmação, escolha Release (Liberar).

Mover uma instância de banco de dados fora de uma VPC para uma VPC

Algumas instâncias de banco de dados herdadas na plataforma EC2-Classic não estão em uma VPC. Se sua instância de banco de dados não estiver em uma VPC, você poderá usar o AWS Management Console para movê-la facilmente para uma VPC. Antes de poder mover uma instância de banco de dados fora de uma VPC para uma VPC, você deve criar a VPC.

O EC2-Classic foi removido em 15 de agosto de 2022. Se ainda não migrou do EC2-Classic para uma VPC, recomendamos que você migre o mais rápido possível. Para obter mais informações, consulte Migre do EC2-Classic para uma VPC no Guia do usuário do Amazon EC2 e o blog EC2-Classic Networking is Retiring – Here's How to Prepare (O EC2-Classic Networking será descontinuado. Veja como se preparar.).

Important

Se você é um cliente novo do Amazon RDS e nunca criou uma instância de banco de dados, ou se estiver criando uma instância de banco de dados em uma região da AWS nunca usada antes, em quase todos os casos, você estará na plataforma EC2-VPC e terá uma VPC padrão. Para obter informações sobre como trabalhar com instâncias de banco de dados em uma VPC, consulte Trabalhar com uma instância de banco de dados em uma VPC.

Siga estas etapas para criar uma VPC para sua instância de banco de dados.

- Etapa 1: Criar uma VPC
- Etapa 2: Criar um grupo de sub-redes de banco de dados
- Etapa 3: Criar um grupo de segurança da VPC

Depois de criar a VPC, siga estas etapas para mover sua instância de banco de dados para essa VPC.

- Atualizar a VPC para uma instância de banco de dados

É altamente recomendável que você crie um backup de sua instância de banco de dados imediatamente antes da migração. Isso garante que você poderá restaurar os dados se a migração falhar. Para obter mais informações, consulte Backup, restauração e exportação de dados.

Veja a seguir algumas limitações ao mover sua instância de banco de dados para a VPC.

- Classes de instância de banco de dados de gerações anteriores: as classes de instância de banco de dados de gerações anteriores podem não ter suporte na plataforma VPC. Ao mover uma instância de banco de dados para uma VPC, escolha uma classe de instância de banco de dados db.m3 ou db.r3. Depois de mover a instância de banco de dados para uma VPC, é possível dimensionar a instância de banco de dados para usar uma classe de instância de banco de dados posterior. Para obter uma lista completa das classes de instância compatíveis com a VPC, consulte Tipos de instância do Amazon RDS.
- multi-AZ: atualmente, não há suporte para mover uma instância de banco de dados multi-AZ fora de uma VPC para uma VPC. Para mover sua instância de banco de dados para uma VPC, primeiro modifique a instância de banco de dados para que ela seja uma implantação single-AZ. Altere a configuração de implantação multi-AZ para No. Depois de mover a instância de banco de dados para uma VPC, modifique-a novamente para torná-la uma implantação multi-AZ. Para obter mais informações, consulte Modificar uma instância de banco de dados do Amazon RDS.
- Réplicas de leitura: no momento, não há suporte para mover uma instância de banco de dados com réplicas de leitura fora de uma VPC para uma VPC. Para mover sua instância de banco de dados para uma VPC, primeiro exclua todas as réplicas de leitura. Depois de mover a instância de banco de dados para uma VPC, recrie as réplicas de leitura. Para obter mais informações, consulte Trabalhar com réplicas de leitura de instância de banco de dados.
- Grupos de opções: se você mover sua instância de banco de dados para uma VPC e a instância de banco de dados estiver usando um grupo de opções personalizado, altere o grupo de opções associado à sua instância de banco de dados. Grupos de opções são específicos da plataforma, e mover para uma VPC é uma alteração na plataforma. Para usar um grupo de opções personalizada nesse caso, atribua o grupo de opções de VPC padrão à instância de banco de dados, atribua um grupo de opções que seja usado por outras instâncias de banco de dados na VPC para a qual você está movendo ou crie um novo grupo de opções e atribua-o à instâncias de banco de dados. Para obter mais informações, consulte Trabalhar com grupos de opções.

Alternativas para mover uma instância de banco de dados fora de uma VPC para uma VPC com tempo de inatividade mínimo

Usando as seguintes alternativas, é possível mover uma instância de banco de dados que não esteja em uma VPC para uma VPC com o mínimo de tempo de inatividade. Essas alternativas causam interrupção mínima na instância de banco de dados de origem e permitem que ela atenda ao tráfego

do usuário durante a migração. No entanto, o tempo necessário para migrar para uma VPC varia de acordo com o tamanho do banco de dados e as características da workload ativa.

- AWS Database Migration Service (AWS DMS): o AWS DMS permite a migração em tempo real de dados, mantendo a instância de banco de dados de origem totalmente operacional, mas replica apenas um conjunto limitado de instruções DDL. O AWS DMS não propaga itens como índices, usuários, privilégios, procedimentos armazenados e outras alterações de banco de dados não diretamente relacionadas aos dados da tabela. Além disso, o AWS DMS não usa automaticamente snapshots do RDS para a criação inicial da instância de banco de dados, o que pode aumentar o tempo de migração. Para mais informações, consulte AWS Database Migration Service.
- Restauração de snapshot de banco de dados ou recuperação em um ponto anterior no tempo: é possível mover uma instância de banco de dados para uma VPC restaurando um snapshot da instância de banco de dados ou restaurando uma instância de banco de dados para um point-in-time. Para obter mais informações, consulte Restaurar uma instância de banco de dados e Restaurar uma instância de banco de dados para um momento especificado no Amazon RDS.

Cotas e restrições do Amazon RDS

A seguir, você pode encontrar uma descrição das cotas de recursos e restrições de nomenclatura do Amazon RDS.

Tópicos

- Cotas no Amazon RDS
- Restrições de nomenclatura no Amazon RDS
- Número máximo de conexões de banco de dados
- Limites de tamanho de arquivo no Amazon RDS

Cotas no Amazon RDS

Cada conta da AWS tem cotas, para cada região da AWS, sobre o número de recursos do Amazon RDS que podem ser criados. Depois que a cota de um recurso é atingida, as chamadas adicionais para criá-lo falham, com uma exceção.

A tabela a seguir lista os recursos e suas cotas por região da AWS.

Nome	Padrão	Ajustá	Descrição
Autorizações por grupos de segurança de banco de dados	Cada região compatível: 20	Não	Número de autorizações de grupo de segurança por grupo de segurança do banco de dados.
Versões de mecanismos personalizados	Cada região compatível: 40	Sim	O número máximo de versões de mecanismo personalizadas permitida s nessa conta na região atual.
Grupos de parâmetros de cluster de banco de dados	Cada região compatível: 50	Não	O número máximo de grupos de parâmetros de cluster de banco de dados.

Nome	Padrão	Ajustá	Descrição
Clusters do banco de dados	Cada região compatível: 40	Sim	O número máximo de clusters do Aurora para essa conta na Região atual.
Instâncias de banco de dados	Cada região compatível: 40	Sim	O número máximo de instâncias de banco de dados permitidas nessa conta na Região atual.
Grupos de sub-redes de banco de dados	Cada região compatível: 50	Sim	O número máximo de grupos de sub-redes de banco de dados.
Tamanho do corpo da solicitação HTTP da API de dados	Cada região compatível: 4 megabytes	Não	O tamanho máximo permitido para o corpo de solicitação HTTP.
Máximo de pares simultâneos de segredo de cluster da API de dados	Cada região compatível: 30	Não	O número máximo de pares exclusivos de segredos e clusters de banco de dados do Aurora Sem Servidor v1 em solicitações simultâneas da API de dados para essa conta na região da AWS atual.

Nome	Padrão	Ajustá	Descrição
Número máximo de solicitações simultâneas da API de dados	Cada região compatível: 500	Não	O número máximo de solicitações da API de dados para um cluster de banco de dados do Aurora Sem Servidor v1 que usam o mesmo segredo e podem ser processadas simultane amente. Solicitações adicionais são colocadas em fila e processadas à medida que as solicitaç ões em andamento são concluídas.
Tamanho máximo do conjunto de resultados da API de dados	Cada região compatível: 1 megabyte	Não	O tamanho máximo do conjunto de resultados do banco de dados que pode ser retornado pela API de dados.
Tamanho máximo da API de dados da string de resposta JSON	Cada região compatível: 10 megabytes	Não	O tamanho máximo da string de resposta JSON simplificada retornada pela API de dados do RDS.

Nome	Padrão	Ajustá	Descrição
Solicitações de API de dados por segundo	Cada região compatível: 1.000 por segundo	Não	O número máximo de solicitações para a API de dados por segundo permitido nessa conta na região da AWS atual. Essa cota aplica-se somente aos clusters do Amazon Aurora Sem Servidor v1.
Assinaturas de eventos	Cada região compatível: 20	Sim	O número máximo de assinaturas de eventos.
Perfis do IAM por cluster de banco de dados	Cada região compatível: 5	Sim	O número máximo de perfis do IAM associados a um cluster de banco de dados.
Perfis do IAM por instância de banco de dados	Cada região compatível: 5	Sim	O número máximo de perfis do IAM associados a uma instância de banco de dados.
Integrações	Cada região compatível: 100	Não	O número máximo de integrações permitidas nessa conta na região da AWS atual.
Snapshots de cluster de banco de dados manual	Cada região compatível: 100	Sim	O número máximo de snapshots manuais de cluster de banco de dados.

Nome	Padrão	Ajustá	Descrição
Snapshots manuais da instância de banco de dados	Cada região compatível: 100	Sim	O número máximo de snapshots manuais de instância de banco de dados.
Grupos de opções	Cada região compatível: 20	Sim	O número máximo de grupos de opções.
Grupos de parâmetros	Cada região compatível: 50	Sim	O número máximo de grupos de parâmetros.
Proxies	Cada região compatível: 20	Sim	O número máximo de proxies permitidos nessa conta na Região da AWS atual.
Réplicas de leitura por primário	Cada região compatível: 15	Sim	O número máximo de réplicas de leitura por instância de banco de dados primária. Essa cota não pode ser ajustada para o Amazon Aurora.
Instâncias de bancos de dados reservadas	Cada região compatível: 40	Sim	O número máximo de instâncias de banco de dados reservadas permitidas nessa conta na Região da AWS atual.
Regras por grupo de segurança	Cada região compatível: 20	Não	O número máximo de regras por grupo de segurança do banco de dados.

Nome	Padrão	Ajustá	Descrição
Grupos de segurança	Cada região compatível: 25	Sim	O número máximo de grupos de segurança do banco de dados.
Grupos de segurança (VPC)	Cada região compatível: 5	Não	O número máximo de grupos de segurança do banco de dados por Amazon VPC.
Sub-redes por grupo de sub-redes do banco de dados	Cada região compatível: 20	Não	O número máximo de sub-redes para por grupo de sub-redes de banco de dados.
Tags por recurso	Cada região compatível: 50	Não	O número máximo de etiquetas por recurso do Amazon RDS.
Armazenamento total para todas as instâncias de banco de dados	Todas as regiões compatíveis: 100 mil gigabytes	Sim	O armazenamento total máximo (em GB) em volumes do EBS para todas as instâncias de banco de dados do Amazon RDS somadas. Essa cota não se aplica ao Amazon Aurora, que tem um volume máximo de cluster de 128 TiB para cada cluster de banco de dados.

> **Note**
>
> Por padrão, você pode ter um total de 40 instâncias de banco de dados. As instâncias de banco de dados do RDS, as instâncias de bancos de dados Aurora, as instâncias do Amazon Neptune e as instâncias do Amazon DocumentDB se aplicam a essa cota.
> As seguintes limitações se aplicam às instâncias de banco de dados do Amazon RDS:
>
> - 10 de cada edição do SQL Server (Enterprise, Standard, Web e Express) no modelo "licença incluída"
> - 10 para Oracle no modelo "licença incluída"
> - 40 para Db2 no modelo de licenciamento “traga a sua própria licença” (BYOL)
> - 40 MySQL, MariaDB ou PostgreSQL
> - 40 Oracle no modelo "bring-your-own-license" (BYOL)
>
> Se a sua aplicação exigir mais instâncias de banco de dados, você poderá solicitar instâncias de banco de dados adicionais abrindo o Console de Service Quotas. No painel de navegação, escolha AWSServiços da . Escolha Amazon Relational Database Service (Amazon RDS), escolha uma cota e siga as instruções para solicitar um aumento de cota. Para obter mais informações, consulte Como solicitar um aumento de cota no Guia do usuário do Service Quotas.
> Para RDS para Oracle e RDS para SQL Server, o limite de réplicas de leitura é de cinco por banco de dados de origem para cada região.
> Os backups gerenciados pelo AWS Backup são considerados snapshots manuais de de banco de dados, mas não são contabilizados para a cota de snapshot manual do . Para obter mais informações sobre o AWS Backup, consulte o Guia do desenvolvedor do AWS Backup.

Se você usar qualquer operação de API do RDS e exceder a cota padrão do número de chamadas por segundo, a API do Amazon RDS emitirá um erro como o seguinte.

ClientError: ocorreu um erro (ThrottlingException) ao chamar a operação *API_name*: taxa excedida.

Aqui, reduza o número de chamadas por segundo. A cota destina-se a cobrir a maioria dos casos de uso. Se cotas maiores forem necessárias, solicite um aumento de cota usando uma das seguintes opções:

- No console, abra o console do Service Quotas.

- Na AWS CLI, use o comando request-service-quota-increase da AWS CLI.

Para obter mais informações, consulte o Manual do usuário do Service Quotas.

Restrições de nomenclatura no Amazon RDS

As restrições de nomenclatura no Amazon RDS são as seguintes:

- Identificador da instância de banco de dados:
  - Devem conter 1 a 63 caracteres alfanuméricos ou hifens.
  - O primeiro caractere deve ser uma letra.
  - Não pode terminar com um hífen ou conter dois hifens consecutivos.
  - Deve ser exclusivo para todas as instâncias de Banco de Dados por conta da AWS, por região da AWS.
- Nome do banco de dados inicial:
  - Restrições de nomes de banco de dados diferem para cada mecanismo de banco de dados. Para ter mais informações, consulte as configurações disponíveis ao criar cada instância de banco de dados.
  - SQL Server: crie os bancos de dados depois de criar a instância de banco de dados.
- Nome de usuário principal: restrições de nomes de usuário principal são diferentes para cada mecanismo de banco de dados. Para ter mais informações, consulte as configurações disponíveis ao criar cada instância de banco de dados.
- Senha principal:
  - A senha do usuário principal do banco de dados pode incluir qualquer caractere ASCII imprimível, exceto `/`, `'`, `"`, `@` ou um espaço.

    Para Oracle, `&` é uma limitação adicional de caracteres.
  - Dependendo do mecanismo de banco de dados, a senha pode conter o seguinte número de caracteres ASCII imprimíveis:
    - Db2: 8–255
    - MariaDB e MySQL: 8–41
    - Oracle: 8–30
    - SQL Server e PostgreSQL: 8–128
- Grupo de parâmetros de banco de dados:

  - Devem conter de 1 a 255 caracteres alfanuméricos.
  - O primeiro caractere deve ser uma letra.
  - Os hifens são permitidos, mas o nome não pode terminar com um hífen nem conter dois hifens consecutivos.
- Grupo de sub-redes do banco de dados:
  - Devem conter de 1 a 255 caracteres.
  - Caracteres alfanuméricos, espaços, hifens, sublinhados e pontos são permitidos.

Número máximo de conexões de banco de dados

O número máximo de conexões simultâneas de banco de dados varia de acordo com o tipo de mecanismo de banco de dados e com a alocação de memória para a classe da instância de banco de dados. O número máximo de conexões geralmente é definido no grupo de parâmetros associado à instância de banco de dados. A exceção é o Microsoft SQL Server, em que ele é definido nas propriedades do servidor para a instância de banco de dados no SQL Server Management Studio (SSMS).

Conexões de banco de dados consomem memória. Definir um desses parâmetros muito alto poderá fazer com que uma condição de memória baixa possa fazer com que uma instância de banco de dados receba o status incompatible-parameters. Para ter mais informações, consulte Diagnosticar e resolver o status de parâmetros incompatíveis para um limite de memória.

Se suas aplicações abrem e fecham conexões com frequência ou mantêm um grande número de conexões de longa duração abertas, recomendamos usar o Amazon RDS Proxy. O RDS Proxy é um proxy de banco de dados totalmente gerenciado e altamente disponível que usa grupos de conexões para compartilhar conexões de banco de dados de forma segura e eficiente. Para saber mais sobre o RDS Proxy, consulte Amazon RDS Proxy.

> **Note**
>
> Para o Oracle, defina o número máximo de processos do usuário e sessões do usuário e do sistema.
> Para o Db2, não é possível definir o máximo de conexões. O limite é 64 mil.

A tabela a seguir mostra informações sobre o máximo de conexões de banco de dados para diferentes mecanismos de banco de dados.

Mecanismo de banco de dados	Parâmetro	Valores permitidos	Valor padrão	Descrição
MariaDB e MySQL	`max_connections`	1–100000	Padrão para todas as versões do MariaDB e MySQL, exceto MariaDB versões 10.5 e 10.6: {DBInstanceClassMemory/12582880} A fórmula é efetivamente equivalente a MB/12. Padrão para o MariaDB versões 10.5 e 10.6: LEAST({DBInstanceClassMemory/25165760},12000) A fórmula é efetivamente equivalente a MB/25. Em ambos os casos, se o cálculo de valor padrão resultar em um valor superior a 16 mil, o Amazon RDS definirá o limite para 16 mil para instâncias de banco de dados do MariaDB e do MySQL.	Número de conexões simultâneas de clientes permitidas

Mecanismo de banco de dados	Parâmetro	Valores permitidos	Valor padrão	Descrição
Oracle	`processes`	80–20000	LEAST({DBInstanceClassMemory/9868951}, 20000)	Processos do usuário
Oracle	`sessions`	100–65535	Não aplicável	Sessões do usuário e do sistema
PostgreSQL	`max_connections`	6–8388607	LEAST({DBInstanceClassMemory/9531392}, 5000)	Número máximo de conexões simultâneas
SQL Server	Número máximo de conexões simultâneas	0–32767	0 (ilimitado)	Número máximo de conexões simultâneas

`DBInstanceClassMemory` está em bytes. Para obter detalhes sobre como esse valor é calculado, consulte Especificação de parâmetros de banco de dados. Especificamente, devido à memória reservada para o sistema operacional e os processos de gerenciamento do RDS, esse tamanho de memória é menor do que o valor em gibibytes (GiB) mostrado em Especificações de hardware para classes de instância de banco de dados .

Por exemplo, algumas classes de instância de banco de dados têm 8 GiB de memória, o que equivale a 8.589.934.592 bytes. Para uma instância de banco de dados do MySQL executada em uma classe de instância de banco de dados com 8 GiB de memória, como db.m7g.large, a equação que usa a memória total seria `8589934592/12582880=683`. Entretanto, a variável `DBInstanceClassMemory` subtrai automaticamente os valores reservados para o sistema operacional e os processos do RDS que gerenciam a instância de banco de dados. O restante da subtração é então dividido por 12.582.880. Esse cálculo resulta em aproximadamente 630 para o valor de `max_connections`, em vez de 683. Esse valor depende da classe de instância de banco de dados e do mecanismo de banco de dados.

Quando uma instância de banco de dados do MariaDB ou MySQL está sendo executada em uma classe de instância de banco de dados pequena, como db.t3.micro ou db.t3.small, a memória total disponível é baixa. Para essas classes de instância de banco de dados, o RDS reserva uma parte

significativa da memória disponível, o que afeta o valor `max_connections`. Por exemplo, o número máximo padrão de conexões para uma instância de banco de dados do MySQL executada em uma classe de instância de banco de dados db.t3.micro é aproximadamente 60. Você pode determinar o valor `max_connections` da instância de banco de dados do MariaDB ou MySQL conectando-se a ela e executando o seguinte comando SQL:

```
SHOW GLOBAL VARIABLES LIKE 'max_connections';
```

Limites de tamanho de arquivo no Amazon RDS

Limites de tamanho de arquivo se aplicam a determinadas instâncias de banco de dados do Amazon RDS. Para obter mais informações, consulte os seguintes limites específicos de mecanismos:

- Limites de tamanho de arquivo do MariaDB no Amazon RDS
- Limites de tamanho de arquivo do MySQL no Amazon RDS
- Limites de tamanho de arquivo do Oracle no Amazon RDS

Solução de problemas para o Amazon RDS

Use as seções a seguir para solucionar problemas que possam surgir com instâncias de bancos de dados do Amazon RDS e do Amazon Aurora.

Tópicos

- Não é possível conectar-se à instância de banco de dados do Amazon RDS
- Problemas de segurança do Amazon RDS
- Solução de problemas de estado de rede incompatível
- Redefinir a senha de proprietário da instância de banco de dados
- Interrupção ou reinicialização da instância de banco de dados do Amazon RDS
- Alterações de parâmetros de banco de dados do Amazon RDS que não entram em vigor
- Instância de banco de dados do Amazon RDS ficando sem espaço de armazenamento
- Instâncias de banco de dados disponíveis do Amazon RDS insuficientes
- Problemas de memória liberável no Amazon RDS
- Problemas no MySQL e MariaDB
- Não é possível definir o período de retenção de backup como 0

Para obter informações sobre como depurar problemas usando a API do Amazon RDS, consulte Solução de problemas de aplicações no Amazon RDS.

Não é possível conectar-se à instância de banco de dados do Amazon RDS

Quando você não consegue se conectar a uma instância de banco de dados, as causas a seguir são motivos comuns:

- Regras de entrada – as regras de acesso impostas pelo firewall local e os endereços IP autorizados a acessar a instância de banco de dados podem não corresponder. O problema está provavelmente nas regras de entrada do seu grupo de segurança.

  Por padrão, as instâncias de banco de dados não permitem acesso. O acesso é concedido por meio de um grupo de segurança associado à VPC que permite o tráfego de entrada e saída

da instância de banco de dados. Se necessário, adicione regras de entrada e saída para sua situação específica ao grupo de segurança. Você pode especificar um endereço IP, um intervalo de endereços IP ou outro grupo de segurança da VPC.

> **Note**
>
> Ao adicionar uma nova regra de entrada, escolha My IP (Meu IP) para a Source (Origem) a fim de permitir o acesso à instância de banco de dados do endereço IP detectado em seu navegador.

Para ter mais informações sobre como configurar um grupo de segurança, consulte Fornecer acesso à instância de banco de dados na VPC criando um grupo de segurança.

> **Note**
>
> Conexões de cliente de endereços IP dentro do intervalo 169.254.0.0/16 não são permitidas. Esse é o APIPA (Automatic Private IP Addressing Range, Intervalo de endereçamento IP privado automático), usado para o endereçamento de link local.

- Acessibilidade pública – para se conectar à sua instância de banco de dados de fora da VPC, como por exemplo, usando uma aplicação cliente, a instância deve ter um endereço IP público atribuído a ela.

  Para tornar a instância acessível publicamente, modifique-a e escolha Yes (Sim) em Public accessibility (Acessibilidade pública). Para ter mais informações, consulte Ocultar uma instância de banco de dados em uma VPC da Internet.
- Porta: a porta que você especificou quando criou a instância de banco de dados não pode ser usada para enviar ou receber comunicações devido às restrições de firewall locais. Verifique com seu administrador de rede para determinar se a rede permite que a porta especificada seja usada para a comunicação de entrada e saída.
- Disponibilidade – a instância de banco de dados recém-criada fica com o status `creating` até que esteja pronta para uso. Quando o estado for alterado para `available`, será possível se conectar à instância de banco de dados. Dependendo do tamanho da sua instância de banco de dados, pode demorar até 20 minutos para que uma instância esteja disponível.
- Gateway da Internet – para que uma instância de banco de dados seja acessível publicamente, as sub-redes no grupo de sub-redes de banco de dados devem ter um gateway da Internet.

Como configurar um gateway da Internet para uma sub-rede

1. Faça login no AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Databases (Bancos de dados) e escolha o nome da instância de banco de dados.
3. Na guia Connectivity & security (Conectividade e segurança) anote os valores do ID da VPC em VPC e o ID da sub-rede em Subnets (Sub-redes).
4. Abra o console da Amazon VPC em https://console.aws.amazon.com/vpc/.
5. No painel de navegação, escolha Gateways da Internet. Verifique se há um gateway de internet associado à sua VPC. Caso contrário, escolha Criar gateway da internet para criar um gateway da Internet. Selecione o gateway de internet e escolha Associar à VPC e siga as instruções para associá-la à sua VPC.
6. No painel de navegação, escolha Sub-redes e selecione sua sub-rede.
7. Na guia Tabela de rotas, verifique que há uma rota com `0.0.0.0/0` como o destino e o gateway de Internet para sua VPC como destino.

   Se você estiver se conectando à sua instância usando o endereço IPv6, verifique se há uma rota para todo o tráfego IPv6 (`::/0`) que aponta para o gateway de Internet. Caso contrário, faça o seguinte:

   a. Escolha o ID da tabela de rotas (rtb-xxxxxxxx) para navegar para a tabela de rotas.
   b. Na guia Routes (Rotas), escolha Edit routes (Editar rotas). Escolha Add route (Adicionar rota), use `0.0.0.0/0` como o destino, e o gateway da Internet como o destino.

      Para IPv6, escolha Add route (Adicionar rota), use `::/0` como o destino, e o gateway da Internet como o destino.
   c. Escolha Save routes (Salvar rotas).

   Além disso, se você estiver tentando se conectar ao endpoint IPv6, verifique se o intervalo de endereços IPv6 do cliente está autorizado a se conectar à instância de banco de dados.

Para ter mais informações, consulte Trabalhar com uma instância de banco de dados em uma VPC.

Para problemas de conexão específicos do mecanismo, consulte os seguintes tópicos:

- Solução de problemas de conexões com a instância de banco de dados do SQL Server
- Solução de problema de conexão com a instância de banco de dados Oracle
- Solucionar problemas de conexões com a sua instância RDS for PostgreSQL
- Máximo de conexões MySQL e MariaDB

Testar uma conexão a uma instância de banco de dados

É possível testar sua conexão a uma instância de banco de dados usando ferramentas comuns do Linux ou do Microsoft Windows.

Em um terminal Linux ou Unix, teste a conexão inserindo o seguinte. Substitua *DB-instance-endpoint* pelo endpoint e *port* pela porta de sua instância de banco de dados.

```
nc -zv DB-instance-endpoint port
```

Veja a seguir um exemplo de comando e o valor de retorno.

```
nc -zv postgresql1.c6c8mn7fake0.us-west-2.rds.amazonaws.com 8299

  Connection to postgresql1.c6c8mn7fake0.us-west-2.rds.amazonaws.com 8299 port [tcp/
vvr-data] succeeded!
```

Os usuários do Windows podem usar o Telnet para testar a conexão com uma instância de banco de dados. As ações do Telnet não têm suporte além do teste da conexão. Se uma conexão for bem-sucedida, a ação não retorna uma mensagem. Se uma conexão não for bem-sucedida, você receberá uma mensagem de erro, como a seguinte:

```
C:\>telnet sg-postgresql1.c6c8mntfake0.us-west-2.rds.amazonaws.com 819

  Connecting To sg-postgresql1.c6c8mntfake0.us-west-2.rds.amazonaws.com...Could not
 open
  connection to the host, on port 819: Connect failed
```

Se as ações do Telnet retornarem êxito, seu grupo de segurança está corretamente configurado.

> **Note**
>
> O Amazon RDS não aceita o tráfego pelo protocolo ICMP (protocolo de mensagens de controle da Internet), incluindo ping.

Solução de problemas de autenticação da conexão

Em alguns casos, você pode se conectar à sua instância de banco de dados, mas recebe erros de autenticação. Nesses casos, convém redefinir a senha do usuário principal para a instância de banco de dados. Isso pode ser feito ao modificar a instância do RDS.

Para ter mais informações sobre a modificação de uma instância de banco de dados , consulte Modificar uma instância de banco de dados do Amazon RDS.

Problemas de segurança do Amazon RDS

Para evitar problemas de segurança, nunca use o nome do usuário-raiz e a senha da Conta da AWS para uma conta de usuário. A prática recomendada é usar o usuário-raiz para criar usuários e atribuí-los a contas de usuário de banco de dados. Também é possível usar o usuário-raiz para criar outras contas de usuário, se necessário.

Para obter informações sobre a criação de usuários, consulte Criar um usuário do IAM na sua Conta da AWS. Para obter informações sobre como criar usuários no AWS IAM Identity Center, consulte Manage identities in IAM Identity Center (Gerenciar identidades no IAM Identity Center).

Mensagem de erro "Falha ao recuperar atributos da conta, certas funções do console podem ser prejudicadas."

Esse erro pode ser exibido por vários motivos. Pode ser porque sua conta não tem as permissões ou não tenha sido configurada corretamente. Se a sua conta for nova, talvez você não tenha esperado que ela ficasse pronta. Se for uma conta existente, talvez você não tenha permissões nas suas políticas de acesso para realizar determinadas ações, como criar uma instância de banco de dados. Para corrigir o problema, o administrador precisa fornecer os perfis necessários para a sua conta. Para ter mais informações, consulte a documentação do IAM.

Solução de problemas de estado de rede incompatível

Estado de rede incompatível significa que, embora o banco de dados ainda possa estar acessível em nível de banco de dados, ele não pode ser modificado nem reinicializado.

Causas

O estado de rede incompatível de sua instância de banco de dados pode resultar de uma das seguintes ações:

- Modificar a classe da instância de banco de dados.
- Modificar a instância de banco de dados para usar a implantação do cluster de banco de dados multi-AZ.
- Substituição de um host devido a um evento de manutenção.
- Iniciar uma instância de banco de dados substituta.
- Restaurar por meio de um backup de snapshot.
- Iniciar uma instância de banco de dados que foi interrompida.

Resolução

Usar o comando start-db-instance

Para corrigir um banco de dados em um estado de rede incompatível, siga estas instruções:

1. Abra ohttps://console.aws.amazon.com/rds/ e escolhaBancos de dados no painel de navegação.
2. Escolha a instância de banco de dados que está no estado de rede incompatível e anote o identificador da respectiva instância, o ID da VPC e os IDs de sub-rede da guia Segurança e conexão.
3. Use a AWS CLI para executar o comando `start-db-instance`. Especifique o valor `--db-instance-identifier`.

> **Note**
>
> A execução desse comando quando o banco de dados está no modo incompatível pode causar algum tempo de inatividade.

> O comando `start-db-instance` não resolve esse problema para instâncias de banco de dados do RDS para SQL Server.

O status do banco de dados mudará para Disponível se o comando for executado com êxito.

Se o banco de dados for reiniciado, a instância de banco de dados poderá executar a última operação executada na instância antes de ser movida para um estado de rede incompatível. Isso pode levar a instância de volta ao estado de rede incompatível.

Se o comando `start-db-instance` não for bem-sucedido ou a instância voltar ao estado de rede incompatível, abra a página Bancos de dados no console do RDS e selecione o banco de dados. Acesse a seção Logs e eventos. A seção Eventos recentes exibe outras etapas de resolução que devem ser seguidas. As mensagens são classificadas da seguinte forma:

- VERIFICAÇÃO INTERNA DE RECURSOS: pode haver problemas com os recursos internos.
- VERIFICAÇÃO DE DNS: verifique a resolução de DNS e os nomes de host da VPC no console da VPC.
- VERIFICAÇÃO DE ENI: a interface de rede elástica (ENI) para o banco de dados talvez não exista.
- VERIFICAÇÃO DE GATEWAY: o gateway da Internet para seu banco de dados disponível ao público não está conectado à VPC.
- VERIFICAÇÃO DE IP: não há endereços IP gratuitos nas sub-redes.
- VERIFICAÇÃO DE GRUPO DE SEGURANÇA: não há grupos de segurança associados ao banco de dados ou os grupos de segurança são inválidos.
- VERIFICAÇÃO DE SUB-REDE: não há sub-redes válidas no grupo de sub-redes de banco de dados ou há problemas na sub-rede.
- VERIFICAÇÃO DE VPC: a VPC associada ao banco de dados é inválida.

Realize a recuperação para um ponto no tempo.

É uma prática recomendada ter um backup (instantâneo ou lógico), caso o banco de dados entre em um estado de rede incompatível. Consulte Introdução aos backups. Ao ativar backups automatizados, interrompa temporariamente qualquer gravação no banco de dados e execute uma recuperação para um ponto no tempo.

> **Note**
>
> Depois que uma instância entra no estado de rede incompatível, a instância de banco de dados pode não estar acessível para realizar um backup lógico.

Se você não ativou backups automatizados, crie uma instância de banco de dados. Em seguida, migre os dados usando o AWS Database Migration Service(AWS DMS) ou usando uma ferramenta de backup e restauração.

Se isso não resolver o problema, entre em contato com o Suportepara obter maior assistência.

Redefinir a senha de proprietário da instância de banco de dados

Se não conseguir acessar sua instância de banco de dados, você poderá fazer login como o usuário mestre. Depois, você poderá redefinir as credenciais de outros usuários administrativos ou funções. Se não conseguir fazer login como usuário mestre, o proprietário da conta da AWS poderá redefinir a senha do usuário mestre. Para obter detalhes sobre quais contas administrativas ou funções deverão ser redefinidas, consulte Privilégios da conta de usuário mestre.

É possível alterar a senha da instância de banco de dados usando o console do Amazon RDS, o comando da AWS CLI modify-db-instance ou a operação de API ModifyDBInstance. Para ter mais informações sobre a modificação de uma instância de banco de dados , consulte Modificar uma instância de banco de dados do Amazon RDS.

Interrupção ou reinicialização da instância de banco de dados do Amazon RDS

Uma interrupção da instância de banco de dados pode ocorrer quando a instância de banco de dados é reinicializada. A interrupção também pode ocorrer quando a instância de banco de dados é colocada em um estado que impede o acesso a ela e quando o banco de dados é reiniciado. Uma reinicialização pode ocorrer ao reinicializar manualmente a instância de banco de dados. Uma reinicialização também pode ocorrer quando você altera uma configuração da instância de banco de dados que exija uma reinicialização para que tenha efeito.

A reinicialização de uma instância de banco de dados só ocorre quando você altera uma configuração que exija uma reinicialização ou quando você faz uma reinicialização manualmente.

Uma reinicialização poderá ocorrer imediatamente se você alterar uma configuração e solicitar que ela tenha efeito imediato. Ou isso pode ocorrer durante a janela de manutenção da instância de banco de dados.

Uma reinicialização de instância de banco de dados ocorre imediatamente quando ocorre um dos seguintes eventos:

- Você altera o período de retenção de backup para uma instância de banco de dados de 0 para um valor diferente de zero ou vice-versa. Depois, defina Apply Immediately (Aplicar imediatamente) como `true`.
- Você altera a classe de instância de banco de dados, e Apply Immediately (Aplicar imediatamente) é definido como `true`.
- Você altera o tipo de armazenamento de Magnetic (Standard) (Magnético (padrão)) para General Purpose (SSD) (Finalidade geral (SSD)) ou Provisioned IOPS (SSD) (IOPS provisionadas (SSD)) ou de Provisioned IOPS (SSD) (IOPS provisionadas (SSD)) ou General Purpose (SSD) (Finalidade geral (SSD)) para Magnetic (Standard) (Magnético (padrão)).

A reinicialização de uma instância de banco de dados ocorre durante a janela de manutenção quando ocorre um dos seguintes:

- Você altera o período de retenção de backup para uma instância de banco de dados de 0 para um valor diferente de zero ou vice-versa e define Apply Immediately (Aplicar imediatamente) como `false`.
- Você altera a classe de instância de banco de dados, e Apply Immediately (Aplicar imediatamente) é definido como `false`.

Quando você altera um parâmetro estático em um grupo de parâmetros de banco de dados, a alteração não terá efeito até que a instância de banco de dados associada ao grupo de parâmetros seja reinicializada. A alteração requer uma reinicialização manual. A instância de banco de dados não é reinicializada automaticamente durante a janela de manutenção.

Para ver uma tabela que mostra as ações das instâncias de bancos de dados e o efeito de configurar o valor Apply Immediately, consulte Modificar uma instância de banco de dados do Amazon RDS.

Alterações de parâmetros de banco de dados do Amazon RDS que não entram em vigor

Em alguns casos, talvez você altere um parâmetro em um grupo de parâmetros do banco de dados, mas não veja as alterações entrarem em vigor. Nesse caso, provavelmente será necessário reinicializar a instância de banco de dados associada ao grupo de parâmetros do banco de dados. Quando você altera um parâmetro dinâmico, a alteração entra em vigor imediatamente. Quando você altera um parâmetro estático, a alteração não entrará em vigor até que você reinicie a instância de banco de dados associada ao grupo de parâmetros.

Você pode reinicializar uma instância de banco de dados usando o console do RDS. Ou você pode chamar explicitamente a operação `RebootDBInstance` da API. Você poderá reinicializar sem failover se a instância de banco de dados estiver em uma implantação multi-AZ. A exigência de reinicializar a instância de banco de dados associada após uma alteração de parâmetro estático ajuda a atenuar o risco de que uma configuração incorreta de parâmetro afete uma chamada de API. Um exemplo disso é chamar `ModifyDBInstance` para alterar a classe de instância de banco de dados. Para ter mais informações, consulte Modificar parâmetros em um grupo de parâmetros de banco de dados no Amazon RDS.

Instância de banco de dados do Amazon RDS ficando sem espaço de armazenamento

Se a sua instância de banco de dados ficar sem espaço de armazenamento, talvez ela se torne indisponível. Recomendamos que você monitore constantemente a métrica `FreeStorageSpace` publicada no CloudWatch para garantir que a sua instância de banco de dados tenha espaço de armazenamento livre suficiente.

Se a instância de banco de dados ficar sem armazenamento, seu status será alterado para `storage-full`. Por exemplo, uma chamada para a operação de API `DescribeDBInstances` para uma instância de banco de dados que esgotou seu armazenamento emitirá o seguinte:

```
aws rds describe-db-instances --db-instance-identifier mydbinstance

DBINSTANCE  mydbinstance  2009-12-22T23:06:11.915Z  db.m5.large  mysql8.0  50  sa
storage-full  mydbinstance.clla4j4jgyph.us-east-1.rds.amazonaws.com  3306
us-east-1b  3
 SECGROUP  default  active
```

```
 PARAMGRP  default.mysql8.0  in-sync
```

Para solucionar esse cenário, adicione mais espaço de armazenamento à instância usando a operação de API `ModifyDBInstance` ou o comando da AWS CLI a seguir.

Para Linux, macOS ou Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --allocated-storage 60 \
    --apply-immediately
```

Para Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --allocated-storage 60 ^
    --apply-immediately
```

```
DBINSTANCE  mydbinstance  2009-12-22T23:06:11.915Z  db.m5.large  mysql8.0  50  sa
storage-full  mydbinstance.clla4j4jgyph.us-east-1.rds.amazonaws.com  3306
us-east-1b  3  60
 SECGROUP  default  active
 PARAMGRP  default.mysql8.0  in-sync
```

Agora, quando descrever sua instância de banco de dados, você verá que ela terá o status `modifying`, o que indica que o armazenamento está sendo dimensionado.

```
aws rds describe-db-instances --db-instance-identifier mydbinstance
```

```
DBINSTANCE  mydbinstance  2009-12-22T23:06:11.915Z  db.m5.large  mysql8.0  50  sa
modifying  mydbinstance.clla4j4jgyph.us-east-1.rds.amazonaws.com
3306  us-east-1b  3  60
 SECGROUP  default  active
 PARAMGRP  default.mysql8.0  in-sync
```

Após a conclusão da escalabilidade do armazenamento, o status da instância de banco de dados mudará para `available`.

```
aws rds describe-db-instances --db-instance-identifier mydbinstance
```

```
DBINSTANCE  mydbinstance  2009-12-22T23:06:11.915Z  db.m5.large  mysql8.0  60  sa
available  mydbinstance.clla4j4jgyph.us-east-1.rds.amazonaws.com  3306
us-east-1b  3
 SECGROUP  default  active
 PARAMGRP  default.mysql8.0  in-sync
```

Ao usar a operação `DescribeEvents`, é possível que você receba notificações quando seu espaço estiver esgotado. Por exemplo, nesse cenário, se fizer uma chamada `DescribeEvents` depois dessas operações, você verá a seguinte saída:

```
aws rds describe-events --source-type db-instance --source-identifier mydbinstance
```

```
2009-12-22T23:44:14.374Z  mydbinstance  Allocated storage has been exhausted db-
instance
2009-12-23T00:14:02.737Z  mydbinstance  Applying modification to allocated storage db-
instance
2009-12-23T00:31:54.764Z  mydbinstance  Finished applying modification to allocated
 storage
```

Instâncias de banco de dados disponíveis do Amazon RDS insuficientes

O erro `InsufficientDBInstanceCapacity` pode ser retornado ao tentar criar, iniciar ou modificar uma instância de banco de dados. Ele também pode ser retornado ao tentar restaurar uma instância de banco de dados de um snapshot de banco de dados. Quando esse erro é retornado, uma causa comum é que a classe de instância de banco de dados específica não está disponível na zona de disponibilidade solicitada. É possível tentar uma das seguinte opções para resolver o problema:

- Repita a solicitação com uma classe de instância de banco de dados diferente.
- Repita a solicitação com uma zona de disponibilidade diferente.
- Repita a solicitação sem especificar uma zona de disponibilidade explícita.

Para obter informações sobre como solucionar problemas de capacidade de instância para o Amazon EC2, consulte Capacidade da instância insuficiente no Guia do usuário do Amazon EC2.

Para ter mais informações sobre como modificar uma instância de banco de dados , consulte Modificar uma instância de banco de dados do Amazon RDS.

Problemas de memória liberável no Amazon RDS

Memória liberávelé a memória total de acesso aleatório (RAM) em uma instância de banco de dados que pode ser disponibilizada para o mecanismo de banco de dados. É a soma da memória livre do sistema operacional (SO) e o buffer e a memória cache de página disponíveis. O mecanismo de banco de dados usa a maior parte da memória no host, mas os processos do sistema operacional também usam RAM. A memória atualmente alocada ao mecanismo de banco de dados ou usada pelos processos do sistema operacional não está incluída na memória liberável. Quando o mecanismo de banco de dados está ficando sem memória, a instância de banco de dados pode usar o espaço temporário normalmente usado para buffer e armazenamento em cache. Como mencionado anteriormente, esse espaço temporário está incluído na memória liberável.

Você usa a métrica `FreeableMemory` no Amazon CloudWatch para monitorar a memória liberável. Para ter mais informações, consulte Ferramentas de monitoramento do Amazon RDS.

Se a instância de banco de dados for executada consistentemente com memória liberável ou usar espaço de troca, pense em aumentar a escala verticalmente para uma classe de instância de banco de dados maior. Para ter mais informações, consulte Classes de instâncias de banco de dados .

Também é possível alterar as configurações de memória. Por exemplo, no RDS para MySQL, você pode ajustar o tamanho do parâmetro `innodb_buffer_pool_size`. Esse parâmetro é definido por padrão como 75% da memória física. Para obter mais dicas sobre solução de problemas do MySQL, consulte How can I troubleshoot low freeable memory in an Amazon RDS para MySQL database? (Como posso solucionar problemas de pouca memória liberável em um banco de dados do Amazon RDS para MySQL?)

Problemas no MySQL e MariaDB

É possível diagnosticar e corrigir problemas em instâncias de banco de dados MySQL e MariaDB.

Tópicos

- Máximo de conexões MySQL e MariaDB
- Diagnosticar e resolver o status de parâmetros incompatíveis para um limite de memória
- Diagnosticar e resolver atrasos entre réplicas de leitura
- Diagnosticar e resolver uma falha de replicação de leitura do MySQL ou MariaDB

- Criar triggers com o registro de logs binários habilitado requer o privilégio SUPER
- Diagnosticar e resolver falhas de restauração pontual
- Erro de replicação interrompida
- Falha na criação da réplica de leitura ou interrupção da replicação com o erro fatal 1236
- A replicação da réplica de leitura falha ao inicializar a estrutura de metadados

Máximo de conexões MySQL e MariaDB

O número máximo de conexões permitidas para uma instância de banco de dados RDS para MySQL ou RDS para MariaDB baseia-se na quantidade de memória disponível para a classe de instância de banco de dados. Uma classe de instância de banco de dados com mais memória disponível resulta em um número maior de conexões disponíveis. Para ter mais informações sobre classes de instância de banco de dados, consulte Classes de instâncias de banco de dados .

O limite de conexão para uma instância de banco de dados é definido por padrão como o máximo para a classe de instância de banco de dados. É possível limitar o número de conexões simultâneas a qualquer valor até o número máximo de conexões permitidas. Use o parâmetro `max_connections` no grupo de parâmetros para a instância de banco de dados. Para ter mais informações, consulte Número máximo de conexões de banco de dados e Grupos de parâmetros para Amazon RDS.

É possível recuperar o número máximo de conexões permitidas para uma instância de banco de dados MySQL ou MariaDB executando a consulta a seguir.

```
SELECT @@max_connections;
```

É possível recuperar o número de conexões ativas para uma instância de banco de dados MySQL ou MariaDB executando a consulta a seguir.

```
SHOW STATUS WHERE `variable_name` = 'Threads_connected';
```

Diagnosticar e resolver o status de parâmetros incompatíveis para um limite de memória

Uma instância de banco de dados MariaDB ou MySQL pode ser colocada no status parâmetros incompatíveis para limitar a memória quando as seguintes condições forem atendidas:

- A instância de banco de dados foi reiniciada pelo menos três vezes em uma hora ou pelo menos cinco vezes em um dia quando o status da instância de banco de dados era Disponível.
- Uma tentativa de reiniciar a instância de banco de dados falha porque uma ação de manutenção ou processo de monitoramento não conseguiu reiniciar a instância de banco de dados.
- O uso de memória potencial da instância de banco de dados excedeu 1,2 vez a memória alocada para a respectiva classe de instância de banco de dados.

Quando uma instância de banco de dados é reiniciada pela terceira vez em uma hora ou pela quinta vez em um dia, será executada uma verificação de uso de memória. A verificação faz com que o cálculo do potencial uso de memória da instância de banco de dados. O valor retornado pelo cálculo é a soma dos seguintes valores:

- Valor 1 – a soma dos seguintes parâmetros:
  - `innodb_additional_mem_pool_size`
  - `innodb_buffer_pool_size`

    É possível modificar o valor de `innodb_buffer_pool_size`. No entanto, o valor nem sempre corresponderá ao que você inseriu. Essa incompatibilidade ocorre por vários motivos. Primeiro, se a instância de banco de dados for uma microinstância de banco de dados, substituiremos o valor padrão e o definiremos como 256 MB. Para ter mais informações, consulte Substituir innodb_buffer_pool_size.

    Em segundo lugar, garantimos que 500 MB de memória sejam reservados na instância de banco de dados para o gerenciador de host, o mecanismo, o sistema operacional e o kernel.

    Por fim, otimizamos `innodb_buffer_pool_size` dividindo-o em unidades. O gerenciador de host arredonda para baixo para o múltiplo mais próximo dessas unidades. As unidades são calculadas multiplicando-se `innodb_buffer_pool_chunk_size` por `innodb_buffer_pool_instances`. Para ter mais informações, consulte Configuring InnoDB Buffer Pool Size, na documentação do MySQL.

    O padrão para `innodb_buffer_pool_instances` é 8, a menos que `innodb_buffer_pool_size` seja menor que 1 GB. Se `innodb_buffer_pool_size` for menor que 1 GB, o padrão para `innodb_buffer_pool_instances` será 1. O padrão para `innodb_buffer_pool_chunk_size` é 128 MB.
  - `innodb_log_buffer_size`
  - `key_buffer_size`

  - `query_cache_size` (MySQL versão 5.7 somente)
  - `tmp_table_size`
- Valor 2 – o `max_connections` parâmetro multiplicado pela soma dos seguintes parâmetros:
  - `binlog_cache_size`
  - `join_buffer_size`
  - `read_buffer_size`
  - `read_rnd_buffer_size`
  - `sort_buffer_size`
  - `thread_stack`
- Valor 3 – se o parâmetro `performance_schema` estiver habilitado, multiplique o parâmetro `max_connections` por `429498`.

  Se o parâmetro `performance_schema` estiver desabilitado, esse valor será zero.

Então, o valor retornado pelo cálculo é o seguinte:

```
Value 1 + Value 2 + Value 3
```

Quando esse valor excede 1,2 vezes a memória alocada para a classe da instância de banco de dados, essa instância é colocada no status incompatible-parameters (parâmetros incompatíveis). Para ter mais informações sobre a memória alocada para as classes de instâncias de banco de dados , consulte Especificações de hardware para classes de instância de banco de dados .

O cálculo multiplica o valor do parâmetro `max_connections` pela soma de vários parâmetros. Se o parâmetro `max_connections` for definido para um valor grane, é possível que a verificação retorne um valoro elevado para o uso potencial de memória da instância de banco de dados. Nesse caso, avalie a possibilidade de reduzir o valor do parâmetro `max_connections`.

Para resolver o problema, conclua as seguintes etapas:

1. Ajuste os parâmetros de memória no grupo de parâmetros de banco de dados associado à instância de banco de dados. Faça isso para que o uso potencial de memória seja menor que 1,2 vezes a memória alocada para a respectiva classe de instância de banco de dados.

   Para obter informações sobre como configurar parâmetros, consulte Modificar parâmetros em um grupo de parâmetros de banco de dados no Amazon RDS.
2. Reinicie a instância de banco de dados.

Para obter informações sobre como configurar parâmetros, consulte Iniciar uma instância de banco de dados do Amazon RDS que foi anteriormente interrompida.

Diagnosticar e resolver atrasos entre réplicas de leitura

Depois de criar uma réplica de leitura MySQL ou MariaDB e quando ela estiver disponível, o Amazon RDS primeiro replicará as alterações feitas na instância de banco de dados de origem a partir do momento em que a operação de criação da réplica de leitura foi iniciada. Durante essa fase, o tempo de atraso da replicação para a réplica de leitura será maior que 0. Você pode monitorar esse tempo de atraso no Amazon CloudWatch visualizando a métrica `ReplicaLag` do Amazon RDS.

A métrica `ReplicaLag` informa o valor do campo `Seconds_Behind_Master` do comando `SHOW REPLICA STATUS` do MariaDB ou MySQL. Para ter mais informações, consulte Instrução SHOW REPLICA STATUS na documentação do MySQL.

Quando a métrica `ReplicaLag` chega a 0, isso mostra que a réplica alcançou a instância do banco de dados de origem. Se a métrica `ReplicaLag` retornar -1, a replicação pode não estar ativa. Para solucionar um erro de replicação, consulte Diagnosticar e resolver uma falha de replicação de leitura do MySQL ou MariaDB. Um `ReplicaLag` com um valor de -1 também pode significar que o valor de `Seconds_Behind_Master` não pode ser determinado ou é `NULL`.

> **Note**
>
> Versões anteriores do MariaDB usavam `SHOW SLAVE STATUS` em vez de `SHOW REPLICA STATUS`. Se você estiver usando uma versão do MariaDB anterior à 10.5, use `SHOW SLAVE STATUS`.

A métrica `ReplicaLag` retorna -1 durante uma interrupção da rede ou quando um patch é aplicado durante a janela de manutenção. Nesse caso, aguarde até que a conectividade de rede seja restaurada ou a janela de manutenção seja finalizada antes de verificar novamente a métrica `ReplicaLag`.

A tecnologia de replicação de leitura do MySQL e do MariaDB é assíncrona. Portanto, você pode esperar aumentos ocasionais da métrica `BinLogDiskUsage` na instância de banco de dados de origem e da métrica `ReplicaLag` na réplica de leitura. Por exemplo, considere uma situação em que um alto volume de operações de gravação para a instância de banco de dados de origem ocorra em

paralelo. Ao mesmo tempo, as operações de gravação na réplica de leitura são serializadas usando um único thread de E/S. Tal situação pode levar a um atraso entre a instância de origem e a réplica de leitura.

Para ter mais informações sobre réplicas de leitura e o MySQL, consulte Detalhes da implementação da replicação na documentação do MySQL. Para ter mais informações sobre réplicas de leitura e o MariaDB, consulte Visão geral sobre a replicação na documentação do MariaDB.

É possível reduzir o atraso entre as atualizações em uma instância de banco de dados de origem e as atualizações subsequentes da réplica de leitura fazendo o seguinte:

- Defina a classe da instância de banco de dados da réplica de leitura para que ela tenha um tamanho de armazenamento comparável ao da instância de banco de dados de origem.
- Certifique-se de que as configurações de parâmetros nos grupos de parâmetros do banco de dados utilizados pela instância de banco de dados de origem e pela réplica de leitura sejam compatíveis. Para ter mais informações e um exemplo, consulte a discussão sobre o parâmetro `max_allowed_packet` na próxima seção.
- Desabilite o cache de consulta. Para tabelas que são modificadas com frequência, o uso do cache de consulta pode aumentar o atraso das réplicas, pois o cache é bloqueado e atualizado com frequência. Se esse for o caso, talvez você veja um atraso menor de réplicas se desabilitar o cache de consulta. É possível desabilitar o cache de consultas definindo `query_cache_type parameter` como 0 no grupo de parâmetros de banco de dados da instância de banco de dados. Para ter mais informações sobre o cache de consulta, consulte Configuração do cache de consulta.
- Aqueça o grupo de buffers na réplica de leitura do InnoDB para MySQL ou MariaDB. Por exemplo, suponha que você tenha um pequeno conjunto de tabelas sendo atualizadas com frequência e esteja usando o esquema de tabela InnoDB ou XtraDB. Nesse caso, despeje essas tabelas na réplica de leitura. Isso faz com que o mecanismo de banco de dados explore as linhas dessas tabelas no disco e armazene-as em cache no grupo de buffers, o que pode reduzir o atraso das réplicas. Essa abordagem pode reduzir o atraso da réplica. Por exemplo:

  Para Linux, macOS ou Unix:

```
PROMPT> mysqldump \
    -h <endpoint> \
    --port=<port> \
    -u=<username> \
    -p <password> \
```

```
    database_name table1 table2 > /dev/null
```

Para Windows:

```
PROMPT> mysqldump ^
    -h <endpoint> ^
    --port=<port> ^
    -u=<username> ^
    -p <password> ^
    database_name table1 table2 > /dev/null
```

Diagnosticar e resolver uma falha de replicação de leitura do MySQL ou MariaDB

O Amazon RDS monitora o status de replicação de suas réplicas de leitura. O RDS atualiza o campo Replication State (Estado de replicação) da instância da réplica de leitura para `Error` caso a replicação seja interrompida por qualquer motivo. É possível analisar os detalhes do erro associado lançado pelos mecanismos do MySQL ou MariaDB visualizando o campo Replication Error (Erro de replicação). Os eventos que indicam o status da réplica de leitura também são gerados, incluindo RDS-EVENT-0045, RDS-EVENT-0046 e RDS-EVENT-0057. Para ter mais informações sobre eventos e como se inscrever neles, consulte Trabalhar com a notificação de eventos do Amazon RDS. Se for retornada uma mensagem de erro do MySQL, verifique o erro na documentação de mensagens de erro do MySQL. Se for retornada uma mensagem de erro do MariaDB, verifique o erro na documentação de mensagens de erro do MariaDB.

Situações comuns que podem causar erros de replicação incluem o seguinte:

- O valor do parâmetro `max_allowed_packet` para uma réplica de leitura é menor que o parâmetro `max_allowed_packet` para a instância do banco de dados de origem.

  O parâmetro `max_allowed_packet` é um parâmetro personalizado que você pode definir em um grupo de parâmetros de banco de dados. O parâmetro `max_allowed_packet` é usado para especificar o tamanho máximo de linguagem de manipulação de dados (DML) que pode ser executado no banco de dados. Em alguns casos, o valor `max_allowed_packet` para a instância de banco de dados de origem pode ser maior do que o valor `max_allowed_packet` para a réplica de leitura. Se sim, o processo de replicação poderá lançar um erro e interromper a replicação. O erro mais comum é `packet bigger than 'max_allowed_packet' bytes`.

É possível corrigir o erro fazendo com que a origem e a réplica de leitura usem grupos de parâmetros de banco de dados com os mesmos valores do parâmetro `max_allowed_packet`.

- A gravação em tabelas em uma réplica de leitura. Se você estiver criando índices em uma réplica de leitura, será necessário que o parâmetro `read_only` seja definido como 0 para criar os índices. Se você estiver gravando em tabelas na réplica de leitura, isso poderá interromper a replicação.
- Uso de um mecanismo de armazenamento não transacional, como o MyISAM. As réplicas de leitura exigem um mecanismo de armazenamento transacional. A replicação só é compatível com os seguintes mecanismos de armazenamento: InnoDB para MySQL ou MariaDB.

  Você pode converter uma tabela MyISAM para o InnoDB com o seguinte comando:

  ```
  alter table <schema>.<table_name> engine=innodb;
  ```
- Usando consultas não deterministas inseguras, como `SYSDATE()`. Para ter mais informações, consulte Determinar instruções seguras e não seguras em registros em logs binários na documentação do MySQL.

As seguintes etapas podem ajudar a resolver seu erro de replicação:

- Se você encontrar um erro lógico e puder ignorar o erro com segurança, siga as etapas descritas em Ignorar o erro de replicação atual para o RDS para MySQL. Sua instância de banco de dados MySQL ou MariaDB deve estar executando uma versão que inclua o procedimento `mysql_rds_skip_repl_error`. Para ter mais informações, consulte mysql.rds_skip_repl_error.
- Se encontrar um problema de posição de log binários, você poderá alterar a posição de reprodução da réplica com o comando mysql.rds_next_source_log (RDS para MySQL versões principais 8.4 e posteriores) ou mysql.rds_next_master_log (RDS para MariaDB e RDS para MySQL versões principais 8.0 e anteriores).
- Você pode encontrar um problema temporário de performance devido à alta carga de DML. Se sim, você pode definir o parâmetro `innodb_flush_log_at_trx_commit` como 2 no grupo de parâmetros de banco de dados da réplica de leitura. Fazer isso pode ajudar na recuperação da réplica de leitura, embora isso reduza temporariamente a atomicidade, a consistência, o isolamento e a durabilidade (ACID).
- É possível excluir a réplica de leitura e criar uma instância usando o mesmo identificador de instância de banco de dados. Se você fizer isso, o endpoint permanecerá igual ao da réplica de leitura antiga.

Se um erro de replicação for corrigido, o valor de Replication State (Estado de replicação) mudará para replicating (replicando). Para ter mais informações, consulte Solucionar problemas de uma réplica de leitura do MySQL.

Criar triggers com o registro de logs binários habilitado requer o privilégio SUPER

Ao tentar criar triggers em uma instância de banco de dados RDS para MySQL ou RDS para MariaDB, você pode receber o seguinte erro:

```
"You do not have the SUPER privilege and binary logging is enabled"
```

Para usar triggers quando o registro em logs binários está habilitado, é necessário ter o privilégio SUPER, que é restrito para as instâncias de bancos de dados RDS para MySQL e RDS para MariaDB. Você pode criar triggers quando o registro em logs binários está habilitado sem o privilégio SUPER, definindo o parâmetro `log_bin_trust_function_creators` DevOps Guru para RDS Para definir `log_bin_trust_function_creators` como true, crie um novo grupo de parâmetros de banco de dados ou modifique um grupo de parâmetros de banco de dados existente.

É possível criar um grupo de parâmetros de banco de dados que permita criar triggers em sua instância de banco de dados do RDS para MySQL ou do RDS para MariaDB com o registro em log binário habilitado. Para fazer isso, use os comandos da CLI a seguir. Para modificar um grupo de parâmetros existente, comece com a etapa 2.

Para criar um novo grupo de parâmetros de forma a permitir triggers com o registro em logs binários habilitado usando a CLI

1. Crie um novo grupo de parâmetros.

   Para Linux, macOS ou Unix:

   ```
   aws rds create-db-parameter-group \
       --db-parameter-group-name allow-triggers \
       --db-parameter-group-family mysql8.0 \
       --description "parameter group allowing triggers"
   ```

   Para Windows:

   ```
   aws rds create-db-parameter-group ^
   ```

```
    --db-parameter-group-name allow-triggers ^
    --db-parameter-group-family mysql8.0 ^
    --description "parameter group allowing triggers"
```

2. Modifique o grupo de parâmetros de banco de dados para permitir triggers.

   Para Linux, macOS ou Unix:

```
aws rds modify-db-parameter-group \
    --db-parameter-group-name allow-triggers \
    --parameters "ParameterName=log_bin_trust_function_creators,
 ParameterValue=true, ApplyMethod=pending-reboot"
```

   Para Windows:

```
aws rds modify-db-parameter-group ^
    --db-parameter-group-name allow-triggers ^
    --parameters "ParameterName=log_bin_trust_function_creators,
 ParameterValue=true, ApplyMethod=pending-reboot"
```

3. Modifique sua instância de banco de dados para usar o novo grupo de parâmetros de banco de dados.

   Para Linux, macOS ou Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --db-parameter-group-name allow-triggers \
    --apply-immediately
```

   Para Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --db-parameter-group-name allow-triggers ^
    --apply-immediately
```

4. Para que as alterações entrem em vigor, reinicialize manualmente a instância de banco de dados.

```
aws rds reboot-db-instance --db-instance-identifier mydbinstance
```

Diagnosticar e resolver falhas de restauração pontual

Restaurar uma instância de banco de dados que inclua tabelas temporárias

Ao tentar uma restauração para um ponto específico (PITR) da sua instância de banco de dados MySQL ou MariaDB, você pode encontrar o seguinte erro:

```
Database instance could not be restored because there has been incompatible database
 activity for restore
functionality. Common examples of incompatible activity include using temporary tables,
 in-memory tables,
or using MyISAM tables. In this case, use of Temporary table was detected.
```

A PITR depende do snapshot do backup e dos logs binários (binlogs) do MySQL ou do MariaDB para restaurar sua instância de banco de dados para um momento específico. As informações de tabelas temporárias podem ser pouco confiáveis nos logs binários e podem causar uma falha na PITR. Se você usar tabelas temporárias na sua instância de banco de dados do MySQL ou do MariaDB, poderá reduzir a possibilidade de uma falha de PITR realizando backups mais frequentes. Uma falha de PITR é mais provável no momento entre a criação de uma tabela temporária e o próximo snapshot de backup.

Restaurar uma instância de banco de dados que inclua tabelas na memória

Você pode encontrar um problema ao restaurar um banco de dados que possua tabelas na memória. As tabelas na memória são limpas durante uma reinicialização. Como resultado, suas tabelas na memória podem ficar vazias após uma reinicialização. Recomendamos que, ao usar tabelas na memória, você arquitete sua solução para lidar com tabelas vazias em caso de uma reinicialização. Se você estiver usando tabelas na memória com instâncias de banco de dados replicadas, talvez seja necessário recriar as réplicas de leitura após uma reinicialização. Isso pode ser necessário se uma réplica de leitura for reinicializada e não conseguir restaurar dados de uma tabela vazia na memória.

Para ter mais informações sobre backups e PITR, consulte Introdução aos backups e Restaurar uma instância de banco de dados para um momento especificado no Amazon RDS.

Erro de replicação interrompida

Quando você chama o comando `mysql.rds_skip_repl_error`, você pode receber uma mensagem de erro informando que a replicação está inativa ou desabilitada.

Esta mensagem de erro é exibida porque a replicação parou e não foi possível reiniciá-la.

Se você precisar ignorar um grande número de erros, o atraso de replicação pode aumentar além do período de retenção padrão para arquivos de log binário. Nesse caso, você poderá encontrar um erro fatal, com os arquivos do log binário sendo limpos antes de sua reprodução na réplica. Essa remoção faz com que a replicação pare, e você não consegue chamar o comando `mysql.rds_skip_repl_error` para ignorar erros de replicação.

Você pode mitigar esse problema aumentando o número de horas em que os arquivos de log binário são retidos na origem da replicação. Após aumentar o período de retenção de log binário, você pode reiniciar a replicação e chamar o comando `mysql.rds_skip_repl_error` conforme necessário.

Para definir o tempo de retenção do log binário, use o procedimento mysql.rds_set_configuration. Especifique um parâmetro de configuração de "horas de retenção do log binário" juntamente com o número de horas para reter arquivos de log binário no cluster de banco de dados, até 720 (30 dias). O exemplo a seguir define o período de retenção para arquivos de log binário em 48 horas.

```
CALL mysql.rds_set_configuration('binlog retention hours', 48);
```

Falha na criação da réplica de leitura ou interrupção da replicação com o erro fatal 1236

Depois de alterar os valores de parâmetro padrão para uma instância de banco de dados MySQL ou MariaDB, você pode encontrar um dos seguintes problemas:

- É possível criar uma réplica de leitura para a instância de banco de dados.
- A replicação falha com `fatal error 1236`.

Alguns valores de parâmetro padrão para instâncias de banco de dados do MySQL e MariaDB ajudam a garantir que o banco de dados seja compatível com ACID e que as réplicas de leitura estejam protegidas contra falhas. Eles fazem isso garantindo que cada confirmação seja totalmente sincronizada gravando a transação no log binário antes de ser confirmada. Alterar esses parâmetros de seus valores padrão para melhorar a performance pode fazer com que haja falha na replicação quando uma transação não for gravada no log binário.

Para resolver esse problema, defina os seguintes valores de parâmetros:

- `sync_binlog = 1`

- `innodb_support_xa = 1`
- `innodb_flush_log_at_trx_commit = 1`

A replicação da réplica de leitura falha ao inicializar a estrutura de metadados

Ao tentar iniciar a replicação, você recebeu a seguinte mensagem de erro:

```
Read Replica Replication Error - SQLError: 13124, reason: Replica failed to initialize
 applier metadata structure from the repository
```

Esse erro ocorre quando há um problema com a estrutura de metadados da réplica. Para corrigir a estrutura de metadados, você deve criar outra réplica.

Para evitar que isso aconteça no futuro, execute uma das seguintes ações:

- Se possível, desabilite o multithreading nas réplicas. A partir do MySQL 8.0.27, o multithreading é habilitado por padrão.
- Se você precisar usar multithreading nas réplicas, recomendamos que use a replicação baseada em GTID. Para ter mais informações, consulte Usar a replicação baseada em GTID.

Não é possível definir o período de retenção de backup como 0

Há várias razões pelas quais pode ser necessário definir o período de retenção de backup como 0. Por exemplo, você pode desativar os backups automáticos imediatamente ao configurar o período de retenção como 0.

Em alguns casos, é possível definir o valor como 0 e receber uma mensagem dizendo que o período de retenção deve estar entre 1 e 35. Nesses casos, verifique se você não configurou uma réplica de leitura para a instância. As réplicas de leitura exigem backups para o gerenciamento dos logs de réplica de leitura, portanto, não será possível definir o período de retenção como 0.

Referência da API do Amazon RDS

Além do AWS Management Console e do AWS Command Line Interface (AWS CLI), o Amazon RDS também fornece uma API. Você pode usar a API para automatizar tarefas de gerenciamento de suas instâncias de banco de dados e outros objetos no Amazon RDS.

- Para obter uma lista alfabética de operações da API, consulte Ações.
- Para obter uma lista alfabética de tipos de dados, consulte Tipos de dados.
- Para obter uma lista de parâmetros de consulta comuns, consulte Parâmetros comuns.
- Para obter descrições dos códigos de erro, consulte Erros comuns.

Para obter mais informações sobre a AWS CLI, consulte a Referência da AWS Command Line Interface para o Amazon RDS.

Tópicos

- Uso da API de consulta
- Solução de problemas de aplicações no Amazon RDS

Uso da API de consulta

As seções a seguir discutem brevemente os parâmetros e a autenticação de solicitação usados com a API de consulta.

Para obter informações gerais sobre como a API de consulta funciona, consulte Solicitações de consulta no Amazon EC2 API Reference.

Parâmetros de consulta

As solicitações baseadas em consulta HTTP são solicitações HTTP que usam o verbo HTTP GET ou POST e um parâmetro de consulta chamado `Action`.

Cada solicitação de consulta deve incluir alguns parâmetros comuns para lidar com a autenticação e a seleção de uma ação.

Algumas operações levam listas de parâmetros. Essas listas são especificadas usando a notação `param.`*n*. Os valores de *n* são inteiros a partir de 1.

Para ter informações sobre as regiões e os endpoints do Amazon RDS, acesse Amazon Relational Database Service (RDS) na seção Regiões e endpoints da Referência geral da Amazon Web Services.

Autenticação de solicitação de consulta

Só é possível enviar solicitações de consulta por meio de HTTPS, e é preciso incluir uma assinatura em todas as solicitações de consulta. Você deve usar a versão de assinatura 4 da AWS ou a versão de assinatura 2. Para obter informações, consulte Processo de assinatura do Signature versão 4 e Processo de assinatura do Signature versão 2.

Solução de problemas de aplicações no Amazon RDS

O Amazon RDS fornece erros específicos e descritivos para ajudar você a solucionar problemas enquanto interage com a API do Amazon RDS.

Tópicos

- Recuperação de erros
- Dicas de solução de problemas

Para obter informações sobre solução de problemas para instâncias de banco de dados do Amazon RDS, consulte Solução de problemas para o Amazon RDS.

Recuperação de erros

Normalmente, espera-se que o aplicativo verifique se uma solicitação gerou um erro antes que você precise processar os resultados. A maneira mais fácil de descobrir se ocorreu um erro é procurar por um nó `Error` na resposta da API do Amazon RDS.

A sintaxe XPath apresenta uma maneira simples de procurar pela presença de um nó de `Error`. Ela também fornece uma maneira relativamente fácil de recuperar o código e a mensagem de erro. O snippet de código a seguir usa Perl e o módulo XML::XPath para determinar se ocorreu um erro durante uma solicitação. Caso tenha ocorrido, o código imprimirá o primeiro código de erro e a mensagem na resposta.

```
use XML::XPath;
    my $xp = XML::XPath->new(xml =>$response);
    if ( $xp->find("//Error") )
```

```
    {print "There was an error processing your request:\n", " Error code: ",
    $xp->findvalue("//Error[1]/Code"), "\n", " ",
    $xp->findvalue("//Error[1]/Message"), "\n\n"; }
```

Dicas de solução de problemas

Recomendamos os seguintes processos para diagnosticar e resolver problemas com a API do Amazon RDS:

- Confirme se o Amazon RDS está funcionando normalmente na região da AWS que você está almejando acessando http://status.aws.amazon.com.
- Verificação da estrutura de sua solicitação.

  Cada operação do Amazon RDS tem uma página de referência na Amazon RDS Referência da API. Verifique novamente se você está usando os parâmetros corretamente. Para ter ideias sobre o que pode estar errado, consulte as amostras de solicitações ou cenários de usuários para ver se esses exemplos realizam operações similares.
- Confira o AWS re:Post

  O Amazon RDS conta com um fórum de comunidade de desenvolvimento onde você pode procurar soluções para os problemas que outros enfrentaram ao longo do caminho. Para visualizar os tópicos, acesse AWS re:Post.

Histórico do documento

Versão atual da API: 31-10-2014

A tabela a seguir descreve as alterações importantes em cada versão do Guia do usuário do Amazon RDS depois de maio de 2018. Para receber notificações sobre atualizações dessa documentação, você pode se inscrever em um feed RSS.

> **Note**
>
> Você pode filtrar novos recursos do Amazon RDS na página What's New with Database? (Novidades sobre bancos de dados). Em Products (Produtos), escolha Amazon RDS. Em seguida, pesquise usando palavras-chave como **`RDS Proxy`** ou **`Oracle 2023`**.

Alteração	Descrição	Data
O Amazon RDS é compatível com o MySQL 8.0.39.	Agora é possível criar instâncias de banco de dados do Amazon RDS que executam o MySQL versão 8.0.39. Para ter mais informações, consulte Versões do MySQL no Amazon RDS.	12 de agosto de 2024
Atualizar a política existente	O Amazon RDS removeu a permissão `sns:Publish` da `AmazonRDSPreviewServiceRolePolicy` do perfil vinculado ao serviço `AWSServiceRoleForRDSPreview` . Para ter mais informações, consulte Atualizações do Amazon RDS para políticas gerenciadas pela AWS.	7 de agosto de 2024

Atualizar a política existente	O Amazon RDS removeu a permissão `sns:Publish` da `AmazonRDSBetaServiceRolePolicy` do perfil vinculado ao serviço `AWSServiceRoleForRDSBeta` . Para ter mais informações, consulte Atualizações do Amazon RDS para políticas gerenciadas pela AWS.	7 de agosto de 2024
O Amazon RDS comporta o MySQL 8.4 no Ambiente de Pré-visualização do Banco de Dados.	O MySQL 8.4 já está disponível no Ambiente de Pré-visualização do Banco de Dados na Região da AWS do Leste dos EUA (Ohio). Para ter mais informações, consulte MySQL version 8.4 in the Database Preview environment.	1 de agosto de 2024
Atualizar permissões de função vinculada a serviços do IAM	Agora a política `AmazonRDSCustomServiceRolePolicy` concede permissões adicionais para estabelecer comunicação com os serviços do Amazon RDS em outra Região da AWS e copiar imagens do EC2. Para ter mais informações, consulte Atualizações do Amazon RDS para políticas gerenciadas pela AWS.	18 de julho de 2024

O Amazon RDS comporta o MariaDB 11.4 no Ambiente de Pré-visualização do Banco de Dados	O MariaDB 11.4 já está disponível no Ambiente de Pré-visualização do Banco de Dados na Região da AWS do Leste dos EUA (Ohio). Para ter mais informações, consulte MariaDB version 11.4 in the Database Preview environment.	18 de julho de 2024
Driver ODBC para MySQL da AWS disponível para o público em geral	O Driver ODBC para MySQL da Amazon Web Services (AWS) é um driver de cliente projetado para a alta disponibilidade do RDS para MySQL. Para ter mais informações, consulte Connecting to RDS for MySQL with the Amazon Web Services (AWS) ODBC Driver for MySQL.	18 de julho de 2024
Atualizar a política existente	O Amazon RDS removeu a permissão `sns:Publish` da `AmazonRDSServiceRolePolicy` do perfil vinculado ao serviço `AWSServiceRoleForRDS`. Para ter mais informações, consulte Política gerenciada da AWS: Amazon RDSServiceRolePolicy.	2 de julho de 2024

Oferta privada do AWS Marketplace para Db2	O AWS Marketplace agora comporta ofertas privadas para uma licença do Db2 por meio do AWS Marketplace para Amazon RDS para Db2. Para ter mais informações, consulte Obtaining a private offer.	1º de julho de 2024
Exportar dados de snapshot de cluster de banco de dados multi-AZ para o Amazon S3	Agora é possível exportar dados de snapshot de cluster de banco de dados multi-AZ para o Amazon S3. Para ter mais informações, consulte Exportar dados de snapshot de banco de dados para o Amazon S3.	27 de junho de 2024
O Amazon RDS para Oracle é compatível com as classes de instância otimizada para memória r6i pré-configuradas	As novas classes de instância de banco de dados do Oracle db.r6i são otimizadas para workloads que exigem mais memória, armazenamento e E/S por vCPU. Por exemplo, db.r6i.8xlarge.tpc2.mem4x tem o multithreading ativado e fornece quatro vezes mais memória do que db.r6i.8x large. Para ter mais informaçõ es, consulte Suporte a classes de instância de banco de dados do RDS for Oracle.	21 de junho de 2024

Suporte estendido do Amazon RDS versão 5.7.44-RDS.20240529 para o RDS para MySQL	O Suporte estendido do RDS versão 5.7.44-RDS.20240529 já está disponível para o RDS para MySQL. Para ter mais informações, consulte Amazon RDS Extended Support versions for RDS for MySQL.	20 de junho de 2024
O Amazon RDS é compatível com o MySQL 8.0.37	Agora é possível criar instâncias de banco de dados do Amazon RDS que executam o MySQL versão 8.0.37. Para ter mais informações, consulte Versões do MySQL no Amazon RDS.	18 de junho de 2024
O Amazon RDS é compatível com o MariaDB 10.11.8, 10.6.18, 10.5.25 e 10.4.34	Agora é possível criar instâncias de banco de dados do Amazon RDS que executam o MariaDB versões 10.11.8, 10.6.18, 10.5.25 e 10.4.34. Para ter mais informações, consulte Versões do MariaDB no Amazon RDS.	14 de junho de 2024

O Amazon RDS está encerrando o suporte para as classes de instância de banco de dados db.m4, db.r4 e db.t2	Para os mecanismos de banco de dados do RDS para MariaDB, RDS para MySQL e RDS para PostgreSQL, não é mais possível criar instâncias de banco de dados que usem as classes de instância db.m4, db.r4 e db.t2. O RDS está fazendo upgrade automatic amente das instâncias de banco de dados existentes que usam essas classes para uma geração mais recente. Para ter mais informações, consulte Tipos de classe de instância de banco de dados.	4 de junho de 2024
Clusters de banco de dados multi-AZ estão disponíveis em mais Regiões da AWS	É possível criar clusters de banco de dados multi-AZ em mais Regiões da AWS. Consulte uma tabela que mostra todas as regiões compatíveis em Regiões e mecanismos de banco de dados compatíveis com clusters de banco de dados multi-AZ no Amazon RDS.	29 de maio de 2024

Driver Python da AWS disponível ao público	O driver Python da Amazon Web Services (AWS) foi projetado como um wrapper Python avançado. Esse wrapper é complemen tar e amplia a funcional idade do driver Psycopg de código aberto. Para ter mais informações, consulte Connecting to DB instances with the AWS drivers.	23 de maio de 2024
O RDS Proxy está disponível em mais regiões.	O RDS Proxy agora está disponível nas seguintes regiões: Ásia-Pacífico (Hyderabad), Ásia-Pacífico (Melbourne), Oriente Médio (EAU), Israel (Tel Aviv), Oeste do Canadá (Calgary) e Europa (Zurique). Para ter mais informações sobre o RDS Proxy, consulte o tópico sobre como Utilizar o proxy do Amazon RDS.	21 de maio de 2024
Licença do Db2 por meio do AWS Marketplace	Com a licença do Db2 por meio do AWS Marketplace, agora você paga uma taxa por hora para assinar as licenças do Db2 para o Amazon RDS para Db2. Consulte mais informações em Opções de licenciamento do Amazon RDS para Db2.	21 de maio de 2024

O Amazon RDS comporta acesso refinado para o Insights de Performance	Agora é possível permitir ou negar o acesso a dimensões individuais no Insights de Performance. Esse acesso refinado pode ser usado para as ações `GetResourceMetrics`, `DescribeDimensionKeys` e `GetDimensionKeyDetails`. Para ter mais informações, consulte Granting fine-grained access for Performance Insights.	21 de maio de 2024
Versões do Suporte estendido do Amazon RDS para RDS para MySQL	É possível visualizar todas as versões do Suporte estendido do RDS para RDS para MySQL. Para ter mais informações, consulte Amazon RDS Extended Support versions for RDS for MySQL.	16 de maio de 2024
O Amazon RDS comporta o MySQL 8.3 no Ambiente de Pré-visualização do Banco de Dados.	O MySQL 8.3 já está disponível no Ambiente de Pré-visualização do Banco de Dados na Região da AWS do Leste dos EUA (Ohio). Para ter mais informações, consulte MySQL version 8.3 in the Database Preview environment.	30 de abril de 2024

O Amazon RDS para Db2 comporta fusos horários.	O RDS para Db2 agora aceita configuração de fusos horários locais para instâncias de banco de dados do RDS para Db2. Para ter mais informaçõ es, consulte Local time zones for Amazon RDS for Db2 DB instances.	25 de abril de 2024
Atualizar permissões de função vinculada a serviços do IAM	Agora, a política `AmazonRDS CustomServiceRoleP olicy` concede permissõe s adicionais para associar um perfil de serviço como perfil de instância a uma instância do RDS Custom. Para ter mais informações, consulte Atualizações do Amazon RDS para políticas gerenciadas pela AWS.	19 de abril de 2024
O Amazon RDS para Oracle comporta a transição do Oracle Data Guard em todas as Regiões da AWS.	Agora é possível usar a transição do Oracle Data Guard em todas as regiões aceitas. Para ter mais informações, consulte Visão geral da alternância do Oracle Data Guard.	16 de abril de 2024

O RDS Custom para Oracle é compatível com Oracle Standard Edition 2.	Agora você pode criar instâncias de banco de dados usando o Standard Edition 2 no Oracle Database 12c versão 1 (12.1), 12c versão 2 (12.2), 18c e 19c. É possível criar CDBs e não CDBs. Para ter mais informações, consulte Suporte a edições e licenciamento do RDS Custom para Oracle.	11 de abril de 2024
O Amazon RDS para Oracle é compatível com o Oracle APEX versão 23.2.v1.	É possível usar o APEX 23.2.v1 com o Oracle Database 19c e posterior. Para ter mais informações, consulte Oracle Application Express.	11 de abril de 2024
Atualização para permissões de perfis vinculados a serviços do RDS Custom	O `AmazonRDSCustomServiceRolePolicy` agora concede permissões adicionais para permitir que o RDS Custom para SQL Server tenha informações sobre o tipo de instância do EC2 e modifique o tipo de instância de host de banco de dados. Para ter mais informações, Atualizações para políticas gerenciadas pela AWS.	8 de abril de 2024

O Amazon RDS Custom para Oracle é compatível com a classe de instância de banco de dados db.x2iezn	Agora é possível usar a classe de instância db.x2iezn para instâncias de banco de dados do RDS Custom para Oracle. Para receber mais informaçõ es, consulte Suporte a classes de instância de banco de dados do RDS Custom para Oracle.	26 de março de 2024
O Amazon RDS é compatíve l com classes de instância db.c6gd para clusters de banco de dados multi-AZ	Agora é possível usar as classes de instância db.c6gd para implantações de clusters de banco de dados multi-AZ. Consulte mais informações em Disponibilidade de classes de instâncias.	21 de março de 2024
Suporte estendido do Amazon RDS	Criar ou restaurar um banco de dados do RDS para MySQL 5.7 ou RDS para PostgreSQL 11 agora inscreve automaticamente esse banco de dados no Suporte estendido do Amazon RDS para que suas aplicaçõe s existentes continuem funcionando como estão. Você pode optar por sair do Suporte estendido do RDS a fim de evitar cobranças após a data de término do suporte padrão do RDS. Para obter mais informações, consulte Usar o suporte estendido do Amazon RDS.	21 de março de 2024

Integração do RDS para Db2 com o AWS License Manager	O RDS para Db2 agora está integrado ao AWS License Manager. Se você usar o modelo traga a sua própria licença, a integração do AWS License Manager facilitará o monitoramento do uso da licença Db2 em sua organizaç ão. Para ter mais informações, consulte Integração ao AWS License Manager.	20 de março de 2024
Rotação de certificado CA para clusters de banco de dados multi-AZ	Agora, é possível alternar os certificados CA para clusters de banco de dados multi-AZ. Considere usar um dos novos certificados CA rds-ca-rs a2048-g1, rds-ca-rsa4096-g1 ou rds-ca-ecc384-g1. Consulte mais informações em Alternar o certificado SSL/TLS.	6 de março de 2024
O Amazon RDS é compatíve l com o armazenamento io2 Block Express	Agora, é possível criar instâncias de banco de dados do RDS que usam o tipo de armazenamento io2 Block Express. Consulte mais informações em io2 Block Express storage.	6 de março de 2024

O RDS Custom para SQL Server é compatível com as classes de instância de banco de dados db.r5b e db.x2iedn	Agora é possível usar as classes db.r5b e db.x2iedn para instâncias de banco de dados do RDS Custom para SQL Server. Consulte mais informações em Compatibilidade com classes de instâncias de banco de dados do RDS Custom for SQL Server.	4 de março de 2024
O RDS Custom para Oracle está disponível na região Oriente Médio (EAU)	É possível criar instâncias de banco de dados do RDS Custom para Oracle na região da Oriente Médio (EAU). Para ver uma tabela que mostra todas as Regiões da AWS compatíveis, consulte Supported Regions and DB engines for RDS Custom for Oracle.	4 de março de 2024
Nova política gerenciada pela AWS	O Amazon RDS adicionou uma nova política gerenciada chamada `AmazonRDSCustomInstanceProfileRolePolicy` para permitir que o RDS Custom execute ações de automação e tarefas de gerenciamento de banco de dados por meio de um perfil de instância do EC2. Para ter mais informações, consulte Atualizações do Amazon RDS para políticas gerenciadas pela AWS.	27 de fevereiro de 2024

O Amazon RDS é compatível com o MariaDB 10.11.7, 10.6.17, 10.5.24 e 10.4.33	Agora é possível criar instâncias de banco de dados do Amazon RDS que executam o MariaDB versões 10.11.7, 10.6.17, 10.5.24 e 10.4.33. Para ter mais informações, consulte Versões do MariaDB no Amazon RDS.	26 de fevereiro de 2024
Clusters de banco de dados multi-AZ do Amazon RDS são compatíveis com o volume de armazenamento gp3 do Amazon EBS	Os clusters de banco de dados multi-AZ agora são compatíveis com volumes gp3 do EBS baseados em SSD. Consulte mais informações em Armazenamento gp3.	26 de fevereiro de 2024
Suporte do Amazon RDS para AWS Secrets Manager na região de Israel (Tel Aviv)	O Amazon RDS é compatível com o Secrets Manager na região de Israel (Tel Aviv). Para ter mais informações, consulte Gerenciamento de senhas com o Amazon RDS e o AWS Secrets Manager.	21 de fevereiro de 2024
O Amazon RDS para Db2 é compatível com o registro em log de auditoria	O RDS para Db2 agora é compatível com o registro em log de auditoria no nível de banco de dados. Quando você habilita o registro em log de auditoria para um banco de dados do RDS para Db2, o Amazon RDS registra a atividade do banco de dados e armazena os logs de auditoria no Amazon S3. Consulte mais informações em Db2 audit logging.	15 de fevereiro de 2024

Suporte estendido do Amazon RDS	O Amazon RDS agora habilita automaticamente o Suporte estendido do Amazon RDS quando as versões principais do mecanismo do RDS para MySQL e do RDS para PostgreSQL nas instâncias de banco de dados e clusters de banco de dados multi-AZ atingem a data de término do suporte padrão do RDS. Para obter mais informações, consulte Usar o suporte estendido do Amazon RDS.	15 de fevereiro de 2024
O Amazon RDS é compatível com o MySQL 8.0.36	Agora é possível criar instâncias de banco de dados do Amazon RDS que executam o MySQL versão 8.0.36. Para ter mais informações, consulte Versões do MySQL no Amazon RDS.	12 de fevereiro de 2024
O Amazon RDS é compatível com o agrupamento EBCDIC para RDS para Db2.	Agora é possível criar bancos de dados do Db2 que usam sequências de agrupamento EBCDIC para classificar o conteúdo nos bancos de dados. Para ter mais informações, consulte EBCDIC collation for Db2 databases on Amazon RDS.	29 de janeiro de 2024

Atualização para o certificado CA padrão	O certificado CA padrão está definido como `rds-ca-rsa2048-g1` . Para obter mais informações, consulte Usar SSL/TLS para criptografar uma conexão para uma instância de banco de dados.	26 de janeiro de 2024
O Amazon RDS para PostgreSQL comporta duas novas caixas para PL/Rust, croaring-rs e num-bigint.	É possível usar duas novas caixas no Amazon RDS para PostgreSQL. Para ter mais informações, consulte Usar caixas com PL/Rust.	24 de janeiro de 2024
O Amazon RDS para PostgreSQL é compatível com o TLS versão 1.3.	É possível usar o Transport Layer Security (TLS) versão 1.3 no RDS para PostgreSQL. Para obter mais informações, consulte Usar SSL com uma instância de banco de dados PostgreSQL.	24 de janeiro de 2024
O RDS Custom para SQL Server comporta o Microsoft SQL Server 2022.	Agora é possível criar instâncias de banco de dados do RDS Custom para SQL Server que usam o SQL Server 2022. Para ter mais informações, consulte Trabalhar com o RDS Custom for SQL Server.	22 de janeiro de 2024

Atualização das permissões de políticas gerenciadas pela AWS	A `AmazonRDSServiceRolePolicy` do perfil vinculado ao serviço `AWSServiceRoleForRDS` tem novos IDs de declarações. Para ter mais informações, consulte Atualizações do Amazon RDS para políticas gerenciadas pela AWS.	19 de janeiro de 2024
O RDS Custom para Oracle é compatível com a região da Europa (Paris).	É possível criar instâncias de banco de dados do RDS Custom para Oracle na região da Europa (Paris). Para ter mais informações, consulte Supported Regions and DB engines for RDS Custom for Oracle.	18 de janeiro de 2024
O Amazon RDS para MySQL é compatível com replicação de várias fontes.	Agora é possível usar a replicação de várias fontes em instâncias de banco de dados do RDS para MySQL. Para ter mais informações, consulte Configuring multi-source replication on RDS for MySQL.	16 de janeiro de 2024
O Amazon RDS comporta o MySQL 8.2 no Ambiente de Pré-visualização do Banco de Dados.	O MySQL 8.2 já está disponível no Ambiente de Pré-visualização do Banco de Dados na Região da AWS do Leste dos EUA (Ohio). Para ter mais informações, consulte MySQL version 8.2 in the Database Preview environment.	11 de janeiro de 2024

O RDS Proxy está disponível na região da Europa (Espanha).	O RDS Proxy agora está disponível na região da Europa (Espanha). Para ter mais informações sobre o RDS Proxy, consulte o tópico sobre como Utilizar o proxy do Amazon RDS.	8 de janeiro de 2024
O Amazon RDS está disponível na região do Oeste do Canadá (Calgary).	O Amazon RDS já está disponível na região do Oeste do Canadá (Calgary). Para ter mais informações, consulte Regiões e zonas de disponibilidade.	20 de dezembro de 2023
O Amazon RDS para Db2 aceita 5 mil usuários locais.	Agora é possível adicionar até 5 mil usuários locais a uma lista de autorizações. Para ter mais informações, consulte rdsadmin.add_user.	20 de dezembro de 2023
O Amazon RDS é compatível com a visualização e a resposta às recomendações.	As recomendações do Amazon RDS agora incluem recomendações proativas baseadas em limites e recomendações reativas baseadas em machine learning para RDS para PostgreSQL. Para ter mais informações, consulte Recommendations from Amazon RDS.	19 de dezembro de 2023

O Amazon RDS é compatível com o MariaDB 10.11.6, 10.6.16, 10.5.23 e 10.4.32.	Agora é possível criar instâncias de banco de dados do Amazon RDS que executam as versões 10.11.6, 10.6.16, 10.5.23 e 10.4.32.do MariaDB. Para ter mais informações, consulte Versões do MariaDB no Amazon RDS.	12 de dezembro de 2023
O Amazon RDS apresenta integrações ETL zero com o Amazon Redshift (pré-visualização).	As integrações ETL zero oferecem uma solução totalmente gerenciada para disponibilizar dados transacionais no Amazon Redshift em questão de segundos depois de serem gravados em uma instância de banco de dados do RDS para MySQL. Para ter mais informações, consulte Working with Amazon RDS zero-ETL integrations with Amazon Redshift (preview).	28 de novembro de 2023
O Amazon RDS comporta mecanismos de banco de dados do IBM Db2.	Agora é possível executar mecanismos de banco de dados do IBM Db2 no Amazon RDS. Para ter mais informações, consulte Amazon RDS for Db2.	27 de novembro de 2023

O RDS para PostgreSQL comporta atualizações de versões principais para o PostgreSQL 16.1 e atualizações de versões secundárias para 15.5, 14.10, 13.13, 12.17 e 11.22.	Com o RDS para PostgreSQL, agora é possível atualizar o mecanismo de banco de dados para a versão principal 16.1 e versões secundárias para 15.5, 14.10, 13.13, 12.17 e 11.22. Para ter mais informações, consulte Atualizar o mecanismo de banco de dados PostgreSQL para Amazon RDS.	17 de novembro de 2023
O RDS Custom para Oracle é compatível com grupos de opções.	É possível criar ou modificar um grupo de opções e associá-lo a uma instância de banco de dados do RDS Custom para Oracle. Agora a opção `Timezone` é compatível. Para ter mais informações, consulte Working with option groups in RDS Custom for Oracle.	17 de novembro de 2023
O Amazon RDS para MySQL é compatível com o plug-in Group Replication.	Agora é possível configurar um cluster ativo-ativo com instâncias de banco de dados do RDS para MySQL versão 8.0.35 ou posterior usando o plug-in Group Replication desenvolvido e mantido pela comunidade MySQL. Para ter mais informações, consulte Configuring active-active clusters for RDS for MySQL.	17 de novembro de 2023

O Amazon RDS Proxy é compatível com o RDS para PostgreSQL 16.1.	Agora é possível criar proxies usando o RDS Proxy para RDS para instâncias de banco de dados do PostgreSQL 16.1. Para ter mais informações, consulte o tópico sobre como Usar o proxy do Amazon RDS.	17 de novembro de 2023
O RDS Custom para SQL Server é compatível com o Microsoft SQL Server 2019 Developer Edition.	É possível criar instâncias de banco de dados do RDS Custom para SQL Server que usam o SQL Server 2019 Developer Edition. Para obter mais informações, consulte Traga sua própria mídia com o RDS Custom para SQL Server.	16 de novembro de 2023
Atualizações de versão secundária de clusters de banco de dados multi-AZ com o mínimo de tempo de inatividade.	Ao executar uma atualizaç ão de versão secundária de um cluster de banco de dados multi-AZ, o Amazon RDS agora atualiza as instância s de banco de dados de leitor antes da instância de gravador, reduzindo significa tivamente o tempo de inatividade. É possível reduzir ainda mais o tempo de inatividade para um segundo ou menos usando o RDS Proxy. Para obter mais informações, consulte Atualizar a versão do mecanismo de um cluster de banco de dados multi-AZ.	16 de novembro de 2023

O RDS para SQL Server comporta o Microsoft SQL Server 2022.	Agora, é possível criar instâncias de banco de dados do RDS que usam o SQL Server 2022. Para ter mais informações, consulte Versões do Microsoft SQL Server no Amazon RDS.	15 de novembro de 2023
O RDS para MySQL é compatível com a atualização de snapshots da versão 5.7 para 8.0.	Agora é possível atualizar a versão de mecanismo de um snapshot do RDS para MySQL da versão 5.7 para a versão 8.0. É possível fazer isso usando o AWS Managemen t Console ou a operação `ModifyDBSnapshot` da API do RDS ou a AWS CLI. Para ter mais informações, consulte Upgrading a MySQL DB snapshot engine version.	15 de novembro de 2023
O RDS Custom para SQL Server é compatível com recuperação para um ponto no tempo de 10 mil bancos de dados.	Agora é possível tornar até mil bancos de dados elegíveis para backup completo e recuperação para um ponto no tempo na instância de banco de dados do RDS Custom para SQL Server. Para ter mais informações, consulte Restaurar uma instância do RDS Custom for SQL Server para um ponto anterior no tempo.	15 de novembro de 2023

O RDS Custom para SQL Server comporta o uso de uma chave mestra de serviço.	O RDS Custom para SQL Server agora comporta o uso de uma chave mestra de serviço (SMK). Uma SMK permite criptografar objetos, como credenciais, e usar recursos do SQL Server, como TDE e criptografia de colunas. Para ter mais informações, consulte Using a Service Master Key with RDS Custom for SQL Server.	13 de novembro de 2023
O Amazon RDS oferece suporte ao MySQL 8.1 no ambiente Database Preview	O PostgreSQL versão 14 RC 1 já está disponível no ambiente de visualização de banco de dados na Região da AWS Leste dos EUA (Ohio). Para ter mais informações, consulte PostgreSQL versão 12.0 no ambiente de pré-visualização de banco de dados.	10 de novembro de 2022
RDS oferece suporte ao MySQL 8.0.33 e 5.7.42	Agora é possível criar instâncias de banco de dados do Amazon RDS que executam as versões 8.0.34 e 5.7.43 do MySQL. Para ter mais informações, consulte Versões do MySQL no Amazon RDS.	9 de novembro de 2022

O RDS Proxy é compatível com clusters de banco de dados multi-AZ	O RDS Proxy agora é compatível com a conexão com clusters de banco de dados multi-AZ. Para ter mais informações, consulte o tópico sobre como Trabalhar com o Amazon RDS Custom.	9 de novembro de 2022
O RDS Custom para Oracle está disponível no AWS GovCloud (US) Regions	O já está disponível no Amazon EKS. Para ter mais informações, consulte Supported Regions and DB engines for RDS Custom for Oracle.	9 de novembro de 2022
As gravações otimizadas pelo Amazon RDS comportam a classe de instância de banco de dados db.r5	Agora as gravações otimizadas pelo Amazon RDS comportam a classe de instância de banco de dados db.r5. Para obter mais informações, consulte Melhorar a performance das gravações com gravações otimizadas pelo Amazon RDS para MariaDB e Melhorar a performance das gravações com gravações otimizadas pelo Amazon RDS para MySQL.	9 de novembro de 2022

O RDS para Oracle é compatível com a configuração de locatário único da arquitetura multilocatário do Oracle.	Com o recurso multilocatário do RDS for Oracle, o RDS oferece uma arquitetura e experiência multilocatário Oracle totalmente gerenciadas para seus bancos de dados Oracle. Você pode usar as APIs do RDS para criar vários PDBs, chamados de bancos de dados de locatários, em um CDB. O RDS oferece a configuração multilocatária da arquitetura CDB como alternativa à configuração herdada de inquilino único. Para obter mais informações, consulte Configuração multilocatária da arquitetura CDB.	8 de novembro de 2022

O Amazon RDS publica métricas do contador do Insights de Performance para o Amazon CloudWatch.	O Performance Insights permite exportar os painéis de métricas pré-configurados ou personalizados para o Amazon CloudWatch. Os painéis de métricas exportados estão disponíveis para visualização no console do CloudWatch. Você também pode exportar um widget de métrica do Performance Insights selecionado e visualizar os dados de métricas no console do CloudWatch. Para ter mais informações, consulte Métricas do Performance Insights publicadas no CloudWatch.	8 de novembro de 2022
O Amazon RDS Custom para Oracle permite que você atualize o sistema operacional em uma instância de banco de dados.	Agora você pode atualizar o banco de dados ou o sistema operacional (OS) para uma instância de banco de dados RDS Custom for Oracle usando o comando `modify-db-instance` CLI. Para receber mais informações, consulte Suporte a classes de instância de banco de dados do RDS Custom para Oracle.	7 de novembro de 2022

O RDS Proxy oferece suporte ao Protocolo Estendido para RDS para PostgreSQL	Agora você pode executar protocolos de consulta estendidos em uma instância de banco de dados RDS para PostgreSQL. Para ter mais informações, consulte o tópico sobre como Usar o proxy do Amazon RDS.	6 de novembro de 2019
Suporte ao RDS para PostgreSQL para implantações azul/verde do RDS	Agora você pode criar uma implantação azul/verde a partir de uma instância de banco de dados do RDS para PostgreSQL. Para ter mais informações, consulte Using Amazon RDS Blue/Green Deployments for database updates (Usar implantações azul/verde do Amazon RDS para atualizações de banco de dados).	26 de outubro de 2023
Atualizações do para políticas gerenciadas pela	As `AmazonRDSPerformanceInsightsReadOnly` políticas `AmazonRDSPerformanceInsightsFullAccess` gerenciadas agora incluem `Sid` (ID da instrução) como identificador na declaração de política. Para ter mais informações, consulte Atualizações do Amazon RDS para políticas gerenciadas pela AWS.	23 de outubro de 2023

O RDS Custom para Oracle tem suporte para a Região da Europa (Milão)	Para ter mais informações, consulte Supported Regions and DB engines for RDS Custom for Oracle.	23 de outubro de 2023
Habilitar gravações otimizadas para RDS em bancos de dados existentes	Agora você pode habilitar gravações otimizadas para RDS em uma instância de banco de dados existente, mesmo que ela tenha sido criada com uma versão de mecanismo, classe de instância de banco de dados ou configuração de sistema de arquivos que não ofereça suporte ao recurso. Para obter mais informações, consulte Habilitando gravações otimizadas para RDS em um banco de dados existente para RDS para MySQL e Habilitando gravações otimizadas para RDS em um banco de dados existente para RDS for MariaDB.	19 de outubro de 2023

O Amazon RDS oferece suporte ao uso de um volume de log dedicado (DLV).	Agora você pode usar um volume de registro dedicado (DLV) com o RDS para MariaDB, o RDS para MySQL e o RDS para PostgreSQL. Os DLVs são ideais para bancos de dados com grande armazenamento alocado, altos requisitos de I/O por segundo (IOPS) ou cargas de trabalho sensíveis à latência. Para obter mais informações, consulte Usando um volume de log dedicado (DLV).	17 de outubro de 2023
Amazon RDS para PostgreSQL, MySQL e MariaDB comportam novas classes de instância de banco de dados	É possível criar instâncias de banco de dados do Amazon RDS executando o PostgreSQL, MySQL e MariaDB que usem as classes de instância de banco de dados db.m6.in, db.m6idn, db.r6.in e db.r6.idn. Para ter mais informações, consulte Mecanismos de banco de dados com suporte a todas as classes de instâncias de banco de dados disponíveis.	12 de outubro de 2023
O Amazon RDS para PostgreSQL é compatível com pgactive	A extensão pgactive está disponível no Amazon RDS para PostgreSQL. Para obter mais informações, consulte Usar extensões PostgreSQL com o Amazon RDS para PostgreSQL.	9 de outubro de 2023

O RDS Custom para Oracle está disponível na região Ásia-Pacífico (Jacarta)	É possível criar instâncias de banco de dados do RDS Custom para Oracle na região da Ásia-Pacífico (Jacarta). Para ter mais informações, consulte Supported Regions and DB engines for RDS Custom for Oracle.	5 de outubro de 2023
O RDS Custom para SQL Server comporta novos agrupamentos de servidores	O RDS Custom para SQL Server agora comporta uma ampla variedade de agrupamentos de servidores, tanto na codificação tradicional quanto na codificação UTF-8, para as localidades SQL_Latin , japonês, alemão e árabe. Para obter mais informações, consulte Compatibilidade com agrupamentos e caracteres para instâncias de banco de dados do RDS Custom para SQL Server.	26 de setembro de 2023

Atualização das permissões de políticas gerenciadas pela AWS	A `AmazonRDSCustomServiceRolePolicy` da função vinculada ao serviço `AWSServiceRoleForRDSCustom` tem novas permissões que possibilitam que o RDS Custom crie, modifique e exclua regras gerenciadas do EventBridge. Para ter mais informações, consulte Atualizações do Amazon RDS para políticas gerenciadas pela AWS.	20 de setembro de 2023
O Amazon RDS publica métricas do contador do Insights de Performance para o Amazon CloudWatch.	A função matemática métrica DB_PERF_INSIGHTS no console do CloudWatch permite que você consulte métricas do contador do Amazon RDS do Insights de Performance. Para obter mais informações, consulte Criação de alarmes CloudWatch para monitorar Amazon RDS.	20 de setembro de 2023
O Insights de Performance comporta estatísticas de nível de resumo para o SQL Server	Ao utilizar o Insights de Performance, é possível visualizar estatísticas de SQL em nível de instrução e resumo para o Amazon RDS para SQL Server. Para obter mais informações, consulte Analisar consultas em execução no SQL Server.	18 de setembro de 2023

O Amazon RDS para PostgreSQL, o MySQL e o MariaDB comportam os tipos de classe de instância de banco de dados db.m6.id e db.r6.id	Agora é possível criar instâncias de banco de dados do Amazon RDS que estejam executando PostgreSQL, MySQL e MariaDB e usem os tipos de classe de instância de banco de dados db.m6.id e db.r6.id. Esses tipos oferecem armazenamento SSD local baseado em NVMe. Para ter mais informações, consulte Mecanismos de banco de dados com suporte a todas as classes de instâncias de banco de dados disponíveis.	11 de setembro de 2023
Suporte a atualizações de versão principal para clusters de banco de dados multi-AZ do RDS para PostgreSQL	Agora você pode fazer atualizações de versão principal de clusters de banco de dados multi-AZ do RDS para PostgreSQL. Para obter mais informações, consulte Atualizar a versão do mecanismo de um cluster de banco de dados multi-AZ.	7 de setembro de 2023
O Amazon RDS é compatível com o MariaDB 10.11.5, 10.6.15, 10.5.22 e 10.4.31	Agora é possível criar instâncias de banco de dados do Amazon RDS executando as versões 10.11.5, 10.6.15, 10.5.22 e 10.4.31 do MariaDB. Para ter mais informações, consulte Versões do MariaDB no Amazon RDS.	7 de setembro de 2023

Suporte estendido do Amazon RDS	O Amazon RDS anuncia a possibilidade futura de continuar executando as versões principais do mecanismo RDS para MySQL e RDS para PostgreSQL nas instâncias de banco de dados após a data de término do suporte padrão do RDS. Para obter mais informaçõ es, consulte Usar o suporte estendido do Amazon RDS.	1º de setembro de 2023
O RDS Custom permite iniciar e interromper uma instância de banco de dados do RDS Custom para SQL Server	Agora o RDS Custom permite iniciar e interromper uma instância de banco de dados do RDS Custom para SQL Server. Para obter mais informações, consulte Iniciar e interromper uma instância de banco de dados do RDS Custom para SQL Server.	31 de agosto de 2023

As gravações otimizadas pelo Amazon RDS comportam a classe de instância de banco de dados db.r5	Agora as gravações otimizadas pelo Amazon RDS comportam a classe de instância de banco de dados db.r5. Para obter mais informações, consulte Melhorar a performance das gravações com gravações otimizadas pelo Amazon RDS para MariaDB e Melhorar a performance das gravações com gravações otimizadas pelo Amazon RDS para MySQL.	31 de agosto de 2023
O Amazon RDS para Oracle comporta atualização automática de arquivos de fuso horário para CDBs	Com a opção `TIMEZONE_FILE_AUTOUPGRADE`, você pode atualizar o arquivo de fuso horário atual para a versão mais recente no banco de dados de contêiner (CDB) do RDS para Oracle. Para ter mais informações, consulte Atualização automática do arquivo de fuso horário do Oracle.	29 de agosto de 2023

As gravações otimizadas pelo Amazon RDS são compatíveis com as classes de instância de banco de dados db.m6g e db.m6i	Agora as gravações otimizadas pelo Amazon RDS são compatíveis com as classes de instância de banco de dados db.m6g e db.m6i. Para obter mais informações, consulte Melhorar a performance das gravações com gravações otimizadas pelo Amazon RDS para MariaDB e Melhorar a performance das gravações com gravações otimizadas pelo Amazon RDS para MySQL.	28 de agosto de 2023
O Amazon RDS é compatível com o MariaDB 10.11	Agora é possível criar instâncias de banco de dados do Amazon RDS que executam a versão 10.11 do MariaDB. Para ter mais informações, consulte Versões do MariaDB no Amazon RDS.	21 de agosto de 2023
Atualização das permissões de políticas gerenciadas pela AWS	A `AmazonRDSCustomServiceRolePolicy` do perfil vinculado a serviço `AWSServiceRoleForRDSCustom` tem novas permissões que possibilitam que o RDS Custom crie interfaces de rede. Para ter mais informações, consulte Atualizações do Amazon RDS para políticas gerenciadas pela AWS.	18 de agosto de 2023

Atualização das permissões de políticas gerenciadas pela AWS	A política gerenciada `AmazonRDSFullAccess` tem novas permissões que autorizam gerar, visualizar e excluir o relatório de análise de performance por um período. Para ter mais informações, consulte Atualizações do Amazon RDS para políticas gerenciadas pela AWS.	17 de agosto de 2023
Atualização das permissões de políticas gerenciadas pela AWS	A adição de novas permissões à política gerenciada `AmazonRDSPerformanceInsightsReadOnly` e a adição de uma nova política gerenciada `AmazonRDSPerformanceInsightsFullAccess` permite gerar um relatório de análise de carga do banco de dados por um período. Para ter mais informações, consulte Atualizações do Amazon RDS para políticas gerenciadas pela AWS.	16 de agosto de 2023

O Amazon RDS comporta a análise de performance para determinado período.	O Insights de Performance permite que você crie e visualize relatórios de análise de performance para um período específico. O relatório fornece as recomendações e os insights identificados para resolver os problemas de performance. Para receber mais informações, consulte Analisar a carga do banco de dados por um período.	16 de agosto de 2023
O Amazon RDS Custom para Oracle é compatível com as classes de instância de banco de dados db.r5b e db.x2iedn.	Agora você pode usar as classes db.r5b e db.x2iedn para instâncias de banco de dados do RDS Custom para Oracle. Para receber mais informações, consulte Suporte a classes de instância de banco de dados do RDS Custom para Oracle.	16 de agosto de 2023
O Amazon RDS Custom para Oracle é compatível com as classes de instância de banco de dados db.m6i, db.r6i e db.t3.	Agora você pode usar as classes db.m6i, db.r6i e db.t3 para instâncias de banco de dados do RDS Custom para Oracle. Para receber mais informações, consulte Suporte a classes de instância de banco de dados do RDS Custom para Oracle.	15 de agosto de 2023

Agora o Amazon RDS para PostgreSQL é compatível com o PostgreSQL versão 16 beta 3 no ambiente de visualização de bancos de dados.	O PostgreSQL versão 16 beta 3 já está disponível no ambiente de pré-visualização de banco de dados na Região da AWS Leste dos EUA (Ohio). Para ter mais informações, consulte o tópico sobre como Trabalhar com o ambiente de visualização do banco de dados.	11 de agosto de 2023
O Amazon RDS é compatível com o MySQL 8.0.34 e 5.7.43.	Agora é possível criar instâncias de banco de dados do Amazon RDS que executam as versões 8.0.34 e 5.7.43 do MySQL. Para ter mais informações, consulte Versões do MySQL no Amazon RDS.	9 de agosto de 2023
O RDS para SQL Server é compatível com a visualização de métricas do sistema operacional para a réplica em espera.	Agora é possível visualizar as métricas de sistema operacional da réplica em espera para o RDS para SQL Server. Para receber mais informações, consulte Como visualizar métricas do SO no console do RDS.	3 de agosto de 2023

O RDS para Oracle é compatível com o Oracle Data Guard para CDBs.	O RDS para Oracle é compatível com réplicas de leitura do Data Guard para bancos de dados de contêiner do Oracle Database 19c e 21c (CDBs). Você pode criar, gerenciar e promover réplicas de leitura em um CDB, assim como em um que não seja CDB, usando as APIs do RDS existentes. Para receber mais informações, consulte Réplicas de leitura de multilocação.	1º de agosto de 2023
O Amazon RDS está disponível na região de Israel (Tel Aviv).	O Amazon RDS está disponível na região de Israel (Tel Aviv). Para ter mais informações, consulte Regiões e zonas de disponibilidade.	1º de agosto de 2023
O Amazon RDS é compatível com a versão 23.1.v1 do Oracle APEX.	Você pode usar o APEX 23.1.v1 com o Oracle Database 19c e posterior. Para ter mais informações, consulte Oracle Application Express.	26 de julho de 2023

O Amazon RDS Custom para Oracle oferece suporte a um Oracle SID não padrão	Ao criar uma instância de banco de dados do RDS Custom para Oracle usando o Oracle Database 19c, você pode especificar um identificador de sistema Oracle (Oracle SID) não padrão. Esse valor também é o nome do CDB. Para obter mais informações, consulte Considerações sobre arquitetura multilocatário.	21 de julho de 2023
O RDS para SQL Server oferece suporte a Active Directory autogerenciado	Agora você pode usar Active Directory autogerenciado para associar diretamente suas instâncias de banco de dados do RDS para SQL Server a domínios do Microsoft Active Directory (AD). Os domínios de AD autogerenciados podem estar no ambiente on-premises ou na nuvem. Para obter mais informações, consulte Trabalhar com Active Directory autogerenciado.	7 de julho de 2023

Suporte à replicação lógica do PostgreSQL para clusters de banco de dados multi-AZ	Agora você pode usar a replicação lógica do PostgreSQL com seu cluster de banco de dados multi-AZ para replicar e sincronizar tabelas individuais em vez de toda a instância do banco de dados. Para obter mais informações, consulte Usar a replicação lógica do PostgreSQL com clusters de banco de dados multi-AZ.	6 de julho de 2023
O Amazon RDS para PostgreSQL já oferece suporte ao PostgreSQL versão 16 Beta 2 no ambiente de pré-visualização de bancos de dados	O PostgreSQL versão 16 Beta 2 já está disponível no ambiente de visualização de banco de dados na Região da AWS Leste dos EUA (Ohio). Para ter mais informações, consulte o tópico sobre como Trabalhar com o ambiente de visualização do banco de dados.	6 de julho de 2023

Atualização das permissões de políticas gerenciadas pela AWS	A `AmazonRDSCustomServiceRolePolicy` do perfil vinculado a serviço `AWSServiceRoleForRDSCustom` tem novas permissões que possibilitam que o RDS Custom para Oracle utilize snapshots. Para ter mais informações, consulte Atualizações do Amazon RDS para políticas gerenciadas pela AWS.	23 de junho de 2023
RDS oferece suporte ao MariaDB 10.6.14, 10.5.21 e 10.4.30	Agora é possível criar instâncias de banco de dados do Amazon RDS executando as versões 10.6.14, 10.5.21 e 10.4.30 do MariaDB. Para ter mais informações, consulte Versões do MariaDB no Amazon RDS.	22 de junho de 2023
RDS oferece suporte ao MySQL 8.0.33 e 5.7.42	Agora é possível criar instâncias de banco de dados do Amazon RDS executando as versões 8.0.33 e 5.7.42 do MySQL. Para ter mais informações, consulte Versões do MySQL no Amazon RDS.	15 de junho de 2023

RDS oferece suporte ao MariaDB 10.6.13, 10.5.20, 10.4.29 e 10.3.39	Agora é possível criar instâncias de banco de dados do Amazon RDS executando as versões 10.6.13, 10.5.20, 10.4.29 e 10.3.39 do MariaDB. Para ter mais informações, consulte Versões do MariaDB no Amazon RDS.	15 de junho de 2023
RDS para Oracle oferece suporte a espaços de tabela transportáveis	Você pode migrar dados de um banco de dados Oracle on-premises para uma instância de banco de dados do RDS para Oracle usando espaços de tabela transportáveis. Essa técnica não exige licenciam ento adicional e é a técnica de migração que oferece o menor tempo de inatividade. Para obter mais informaçõ es, consulte Migrar usando espaços de tabela transport áveis da Oracle.	15 de junho de 2023
Amazon RDS oferece suporte ao RDS Proxy com RDS para MariaDB versão 10.6	Agora é possível criar um RDS Proxy com um banco de dados do RDS para MariaDB versão 10.6. Para ter mais informações sobre o RDS Proxy, consulte o tópico sobre como Utilizar o proxy do Amazon RDS.	15 de junho de 2023

RDS Custom para SQL Server oferece suporte ao modelo BYOM (traga sua própria mídia)	Agora você pode criar uma versão de mecanismo personalizada (CEV) usando sua própria mídia do SQL Server. Para obter mais informações, consulte Traga sua própria mídia com o RDS Custom para SQL Server.	8 de junho de 2023
RDS para Oracle consegue converter um não CDB Oracle Database 19c em CDB	Se sua instância de banco de dados executar o Oracle Database 19c com a RU de abril de 2021 ou posterior, você poderá converter um não CDB em um CDB (banco de dados de contêiner). Depois de converter a arquitetura, você poderá fazer upgrade do CDB 19c para CDB 21c. Essa etapa é necessária porque não é possível atualizar um banco de dados e converter a arquitetura usando um único comando. Para obter mais informações, consulte Converter um não CDB RDS para Oracle em um CDB.	31 de maio de 2023

Clusters de banco de dados multi-AZ disponíveis nas regiões da China	Os clusters de banco de dados multi-AZ já estão disponíveis nas Regiões da AWS China (Pequim) e na China (Ningxia). Para ter mais informações, consulte Supported Regions and DB engines for Multi-AZ DB clusters in Amazon RDS.	30 de maio de 2023
Suporte ao recurso Leituras otimizadas pelo Amazon RDS para clusters de banco de dados multi-AZ	O recurso Leituras otimizadas pelo Amazon RDS agora é compatível com clusters de banco de dados multi-AZ. Para obter mais informações, consulte Melhorar a performance das consultas para o RDS para MySQL com o recurso Leituras otimizadas pelo Amazon RDS e Melhorar a performance das consultas para o RDS para PostgreSQL com o recurso Leituras otimizadas pelo Amazon RDS.	30 de maio de 2023
O RDS Custom para Oracle aceita a região da Ásia-Pacífico (Jacarta).	Para ter mais informações, consulte Supported Regions and DB engines for RDS Custom for Oracle.	29 de maio de 2023

Crie uma réplica de leitura de instância de banco de dados com um cluster de banco de dados multi-AZ do RDS para PostgreSQL de origem	Agora é possível criar uma réplica de leitura de instância de banco de dados com um cluster de banco de dados multi-AZ do RDS para PostgreSQL como origem. Anteriormente, somente o RDS para MySQL era compatível. Para ter mais informações, consulte Criar uma réplica de leitura de instância de banco de dados de um cluster de banco de dados multi-AZ.	24 de maio de 2023
Métricas de performance do CloudWatch e do Insights de Performance no painel do Insights de Performance	O Amazon RDS agora fornece uma visão consolidada das métricas do Insights de Performance e do CloudWatch no painel do Insights de Performance. Para ter mais informações, consulte Viewing combined metrics with the Performance Insights dashboard.	24 de maio de 2023

Leituras otimizadas pelo Amazon RDS disponíveis nas regiões da China	As leituras otimizadas pelo Amazon RDS agora estão disponíveis nas regiões China (Pequim) e China (Ningxia) da Regiões da AWS. Para ter mais informações, consulte Melhorar a performance das consultas para o RDS para MariaDB com leituras otimizadas pelo Amazon RDS e Melhorar a performance de consultas para o RDS para MySQL com leituras otimizadas pelo Amazon RDS.	24 de abril de 2023
Suporte do Amazon RDS para AWS Secrets Manager nas regiões da China	O Amazon RDS é compatível com o Secrets Manager nas regiões China (Pequim) e China (Ningxia). Para ter mais informações, consulte Gerenciamento de senhas com o Amazon RDS e o AWS Secrets Manager.	20 de abril de 2023
O RDS Custom para Oracle é compatível com a reutilização de IDs de AMI para novas CEVs	Quando você cria uma versão de mecanismo personalizada (CEV), o RDS Custom para Oracle tem como padrão a imagem de máquina da Amazon (AMI) mais recente disponível. Agora você pode especificar um ID de AMI que foi usado em uma CEV anterior. Para ter mais informações, consulte Criar uma CEV.	19 de abril de 2023

O Amazon RDS é compatível com a publicação de eventos com tags para os assinantes de tópicos	As notificações de eventos do Amazon RDS enviadas ao Amazon Simple Notification Service (Amazon SNS) ou ao Amazon EventBridge agora contêm tags de eventos no corpo da mensagem. Essas tags fornecem dados sobre o recurso que foi afetado pelo evento do serviço. Para ter mais informações, consulte Tags e atributos de notificações de eventos do Amazon RDS.	17 de abril de 2023
Comprar instâncias reservadas para um cluster de banco de dados multi-AZ	Agora é possível comprar instâncias de banco de dados reservadas para um cluster de banco de dados multi-AZ. Para ter mais informações, consulte Instâncias de banco de dados reservadas para um cluster de banco de dados multi-AZ.	12 de abril de 2023

O Amazon RDS é compatível com as classes de instância db.m7g e db.r7g	Agora você pode usar as classes de instância db.m7g e db.r7g para instâncias de banco de dados do RDS para MySQL, do RDS para MariaDB e do RDS para PostgreSQL. Para ter mais informações, consulte Mecanismos de banco de dados com suporte a todas as classes de instâncias de banco de dados disponíveis.	12 de abril de 2023
Atualização para permissões de perfis vinculados a serviços do Amazon RDS Custom	O `AmazonRDSCustomServiceRolePolicy` agora concede permissões adicionais para permitir que o RDS Custom para SQL Server use o Amazon SQS e crie snapshots. Para ter mais informações, consulte Atualizações para políticas gerenciadas pela AWS.	6 de abril de 2023

Migrar para um cluster de banco de dados multi-AZ do RDS para MySQL usando uma réplica de leitura	Agora você pode usar uma réplica de leitura para migrar uma implantação Single-AZ do RDS para MySQL ou uma implantação de instância de banco de dados multi-AZ para uma implantação de cluster de banco de dados multi-AZ do RDS para MySQL com tempo de inatividade reduzido. Para ter mais informações, consulte Migrar para um cluster de banco de dados multi-AZ utilizando uma réplica de leitura.	6 de abril de 2023
Criar uma réplica de instância de banco de dados de um cluster de banco de dados multi-AZ	Agora você pode criar uma réplica de leitura de instância de banco de dados por meio de um cluster de banco de dados multi-AZ para escalar além da capacidad e computacional do cluster de origem. Para ter mais informações, consulte Criar uma réplica de leitura de instância de banco de dados de um cluster de banco de dados multi-AZ.	6 de abril de 2023

O Amazon RDS Custom para SQL Server é compatível com multi-AZ	É possível criar uma implantação multi-AZ do RDS Custom para SQL Server. Para ter mais informações, consulte Gerenciar uma implantação multi-AZ para o RDS Custom para SQL Server.	6 de abril de 2023
Atualização das permissões de políticas gerenciadas pela AWS	As políticas `AmazonRDSFullAccess` e `AmazonRDSReadOnlyAccess` agora concedem permissões adicionais para possibilitar a exibição das descobertas do Amazon DevOps Guru no console do RDS. Para ter mais informações, consulte Atualizações do Amazon RDS para políticas gerenciadas pela AWS.	30 de março de 2023
O Amazon RDS é compatível com a versão 22.2.v1 do Oracle APEX	Você pode usar o APEX 22.2.v1 com todas as versões compatíveis do Oracle Database. Para ter mais informações, consulte Oracle Application Express.	30 de março de 2023

Amazon DevOps Guru disponível para RDS para PostgreSQL	O RDS para PostgreSQL alerta você sobre anomalias recentes detectadas pelo Amazon DevOps Guru. A página de detalhes do banco de dados do console alerta você sobre anomalias atuais e as que ocorreram nas últimas 24 horas. O DevOps Guru publica insights proativos com recomendações para ajudar você a resolver problemas em seus bancos de dados do RDS para PostgreSQL antes que eles aconteçam. Para ter mais informações, consulte Como o DevOps Guru para RDS funciona.	30 de março de 2023
O RDS Custom é compatível com o volume de armazenamento gp3 do Amazon EBS	O RDS Custom para Oracle e o RDS Custom para SQL Server são compatíveis com os volumes do EBS baseados em SSD io1, gp2 e gp3. Para ter mais informações, consulte Requisitos gerais do RDS Custom para Oracle e Requisitos gerais do RDS Custom para SQL Server.	29 de março de 2023

Atualização das permissões de políticas gerenciadas pela AWS	As políticas `AmazonRDSFullAccess` e `AmazonRDSReadOnlyAccess` agora concedem permissões adicionais ao Amazon CloudWatch. Para ter mais informações, consulte Atualizações do Amazon RDS para políticas gerenciadas pela AWS.	16 de março de 2023
O RDS Proxy está disponível nas regiões da China	O RDS Proxy agora está disponível nas regiões China (Pequim) e China (Ningxia). Para ter mais informações sobre o RDS Proxy, consulte o tópico sobre como Utilizar o proxy do Amazon RDS.	15 de março de 2023
O RDS Proxy está disponível na região Ásia-Pacífico (Jacarta)	O RDS Proxy agora está disponível na região Ásia-Pacífico (Jacarta). Para ter mais informações sobre o RDS Proxy, consulte o tópico sobre como Utilizar o proxy do Amazon RDS.	8 de março de 2023

As gravações otimizadas pelo Amazon RDS melhoram a performance das transações de gravação do RDS para MariaDB	Você pode melhorar a performance das transações de gravação para instâncias de banco de dados do RDS para MariaDB com as gravações otimizadas pelo Amazon RDS. Para ter mais informações, consulte Melhorar a performance das gravações com gravações otimizadas pelo Amazon RDS para MariaDB.	7 de março de 2023
Amazon RDS para PostgreSQL versões 15.2	Os novos recursos do Amazon RDS para PostgreSQL 15.2 incluem o comando SQL padrão “MERGE” para consultas SQL condicionais, melhorias de performance para classificação em memória e baseada em disco, e suporte para confirmação em duas fases e filtragem de linha/coluna para replicação lógica.	27 de fevereiro de 2023
RDS Custom para Oracle está disponível nas regiões Canadá (Central) e América do Sul (São Paulo)	Para ver uma tabela que mostra todas as Regiões da AWS compatíveis, consulte Supported Regions and DB engines for RDS Custom for Oracle.	22 de fevereiro de 2023

Amazon RDS oferece suporte a backups automatizados entre regiões para o RDS para MariaDB e o RDS para MySQL	Agora você pode replicar os snapshots de banco de dados e os logs de transaçõe s entre Regiões da AWS para instâncias de banco de dados do RDS para MariaDB e do RDS para MySQL. Para ter mais informações, consulte Replicar backups automatiz ados para outra Região da AWS.	22 de fevereiro de 2023
Amazon RDS para Oracle oferece suporte a aviso prévio de upgrades automáticos de versão secundária	O RDS notificará você com antecedência sobre a data em que uma nova versão secundária do mecanismo RDS para Oracle estará disponível. O RDS começa a programar upgrades automátic os de versão secundária de instâncias de banco de dados do RDS para Oracle na data de disponibilidade. Para ter mais informações, consulte Antes de agendar um upgrade automático de versão secundária.	21 de fevereiro de 2023

Amazon RDS para SQL Server oferece suporte ao recurso Database Activity Streams	Agora você pode monitorar uma instância de banco de dados do SQL Server usando o recurso Database Activity Streams. Uma instância de banco de dados do SQL Server tem a auditoria de servidor que é gerenciad a pelo Amazon RDS. Você pode definir as políticas para registrar eventos do servidor na especificação de auditoria do servidor. Você pode criar uma especificação de auditoria de banco de dados e definir as políticas para registrar eventos do banco de dados. O stream de atividade é coletado e transmitido para o Amazon Kinesis. No Kinesis, é possível monitorar o fluxo de atividade para análise posterior. Para ter mais informações, consulte Monitorar o Amazon RDS com o recurso Database Activity Streams.	15 de fevereiro de 2023
RDS oferece suporte ao MySQL 8.0.32 e 5.7.41	Agora é possível criar instâncias de banco de dados do Amazon RDS executand o o MySQL versões 8.0.32 e 5.7.41. Para ter mais informações, consulte Versões do MySQL no Amazon RDS.	7 de fevereiro de 2023

O Amazon RDS para Oracle oferece suporte a novos pacotes de cifras para SSL	Se você executar o Oracle Database 19c ou 21c, poderá especificar seis novos pacotes de cifras na opção SSL do RDS para Oracle. Esses pacotes oferecem suporte a FIPS e são compatíveis com FedRAMP. Para ter mais informações, consulte Oracle Secure Sockets Layer.	3 de fevereiro de 2023
O Amazon RDS para Oracle oferece suporte a novos pacotes de cifras para Oracle Enterprise Manager	Você pode usar quatro novos pacotes de cifras compatíveis com FedRAMP para a opção OEM. Para ter mais informações, consulte Oracle Management Agent for Enterprise Manager Cloud Control.	3 de fevereiro de 2023
RDS para Oracle oferece suporte ao recurso Database Activity Streams nas regiões Ásia-Pacífico (Hyderabad), Europa (Espanha) e Oriente Médio (EAU)	Para ter informações, consulte Supported Regions and DB engines for database activity streams in Amazon RDS.	27 de janeiro de 2023

Migrar para um cluster de banco de dados multi-AZ do RDS para PostgreSQL usando uma réplica de leitura	Usando uma réplica de leitura, você pode migrar uma implantação single-AZ do RDS para PostgreSQL ou uma implantação de instância de banco de dados multi-AZ para uma implantação de cluster de banco de dados multi-AZ do RDS para PostgreSQL com tempo de inatividade reduzido. Para ter mais informações, consulte Migrar para um cluster de banco de dados multi-AZ utilizando uma réplica de leitura.	23 de janeiro de 2023
Amazon RDS disponível na região Ásia-Pacífico (Melbourne)	O Amazon RDS já está disponível na região Ásia-Pacífico (Melbourne). Para ter mais informações, consulte Regiões e zonas de disponibilidade.	23 de janeiro de 2023
RDS para MariaDB oferece suporte à aplicação de conexões SSL/TLS	O RDS para MariaDB já oferece suporte à aplicação de conexões SSL/TLS definindo parâmetro `require_secure_transport` como `ON`. Para ter mais informações, consulte Exigir SSL/TLS para todas as conexões com uma instância de banco de dados do MariaDB.	19 de janeiro de 2023

As leituras otimizadas do Amazon RDS melhoram a performance das consultas do RDS para MariaDB	É possível acelerar o processamento de consultas para instâncias de banco de dados do RDS para MariaDB com leituras otimizadas do Amazon RDS. Para ter mais informações, consulte Melhorar a performance das consultas para o RDS para MariaDB com leituras otimizadas do Amazon RDS.	11 de janeiro de 2023
Restaurar um snapshot de cluster de banco de dados multi-AZ para uma instância de banco de dados	Agora é possível restaurar um snapshot de cluster de banco de dados multi-AZ para uma implantação single-AZ ou uma implantação de instância de banco de dados multi-AZ. Para ter mais informações, consulte Restauração de um snapshot de cluster de banco de dados multi-AZ para uma instância de banco de dados.	10 de janeiro de 2023
Especificar a autoridade de certificação (CA) durante a criação da instância de banco de dados	Agora você pode especificar qual CA usar para o certificado de servidor de uma instância de banco de dados durante a criação da instância de banco de dados. Para ter mais informações, consulte Certificate authorities (Autoridades de certificação).	5 de janeiro de 2023

O RDS Custom para SQL Server é compatível com versões de mecanismo personalizadas	Uma versão de mecanismo personalizada (CEV) para o RDS Custom para SQL Server é uma imagem de máquina da Amazon (AMI) com o Microsoft SQL Server pré-instalado. Você escolhe uma AMI do Amazon EC2 Windows para usar como imagem base e pode instalar outro software no sistema operacional. Você pode personalizar a configuração do sistema operacional e do SQL Server para atender às necessidades da sua empresa. Para ter mais informações, consulte Working with custom engine versions for RDS Custom for SQL Server (Trabalhar com versões de mecanismo personalizadas para o RDS Custom para SQL Server).	28 de dezembro de 2022
Use as implantações azul/verde do Amazon RDS disponíveis em Regiões da AWS adicionais	O recurso Implantações azul/verde agora está disponível nas regiões China (Pequim) e China (Ningxia). Para ter mais informações, consulte Using Amazon RDS Blue/Green Deployments for database updates (Usar implantações azul/verde do Amazon RDS para atualizações de banco de dados).	22 de dezembro de 2022

Atualizar permissões de função vinculada a serviços do IAM	A política AmazonRDS ServiceRolePolicy agora concede permissões adicionais ao AWS Secrets Manager. Para ter mais informações, consulte Atualizações do Amazon RDS para políticas gerenciadas pela AWS.	22 de dezembro de 2022
O Amazon RDS é compatível com a renomeação de um cluster de banco de dados multi-AZ	Agora é possível renomear um cluster de banco de dados multi-AZ. Para ter mais informações, consulte Renaming a Multi-AZ DB cluster (Renomear um cluster de banco de dados multi-AZ).	22 de dezembro de 2022
O Amazon RDS integra-se ao AWS Secrets Manager para gerenciamento de senhas	O Amazon RDS pode gerenciar a senha de usuário principal para uma instância de banco de dados ou um cluster de banco de dados multi-AZ no Secrets Manager. Para ter mais informações, consulte Gerenciamento de senhas com o Amazon RDS e o AWS Secrets Manager.	22 de dezembro de 2022

As gravações otimizadas pelo Amazon RDS são compatíveis com as classes de instância de banco de dados db.r6g e db.r6gd	As gravações otimizadas pelo Amazon RDS agora são compatíveis com as classes de instâncias de banco de dados db.r6g e db.r6gd. Para ter mais informações, consulte Improving write performance with Amazon RDS Optimized Writes (Melhorar a performance das gravações com gravações otimizadas pelo Amazon RDS).	22 de dezembro de 2022
O Amazon RDS Custom para Oracle é compatível com novas Regiões da AWS	Você pode criar instâncias de banco de dados do RDS Custom para Oracle nas regiões Ásia-Pacífico (Seul) e Ásia-Pacífico (Osaka). Para ter mais informações, consulte Supported Regions and DB engines for RDS Custom for Oracle.	21 de dezembro de 2022
O Amazon RDS no AWS Outposts é compatível com réplicas de leitura	Agora você pode criar uma réplica de leitura a partir de uma instância de banco de dados do RDS no Outposts MySQL ou do PostgreSQL. Para ter mais informações, consulte Criar réplicas de leitura para o Amazon RDS no AWS Outposts.	19 de dezembro de 2022

O RDS Custom para Oracle é compatível com a modificação da classe de instância de banco de dados	Agora você pode alterar a classe de instância da sua instância de banco de dados do RDS Custom para Oracle. Para ter mais informações, consulte Modifying your RDS Custom for Oracle DB instance (Modificar sua instância de banco de dados do RDS Custom para Oracle).	16 de dezembro de 2022
O Amazon RDS para MySQL e o RDS para PostgreSQL são compatíveis com classes de instância de banco de dados db.x2iedn	Agora você pode usar as classes de instância de banco de dados db.x2iedn para o RDS para MySQL e o RDS para PostgreSQL. Para ter mais informações, consulte Mecanismos de banco de dados com suporte a todas as classes de instâncias de banco de dados disponíveis.	14 de dezembro de 2022
As gravações otimizadas pelo Amazon RDS são compatíveis com as classes de instância de banco de dados db.x2iedn	As gravações otimizadas pelo Amazon RDS agora são compatíveis com as classes de instância de banco de dados db.x2iedn. Para ter mais informações, consulte Improving write performance with Amazon RDS Optimized Writes (Melhorar a performance das gravações com gravações otimizadas pelo Amazon RDS).	14 de dezembro de 2022

O Amazon RDS é compatível com a cópia de grupos de opções de banco de dados ao copiar snapshots de banco de dados	Agora você pode copiar um grupo de opções entre Contas da AWS como parte de uma solicitação de cópia de snapshot em bancos de dados do RDS para Oracle. Para ter mais informações, consulte Trabalhar com grupos de opções.	13 de dezembro de 2022
O Amazon RDS é compatível com o RDS Proxy com RDS para PostgreSQL versão 14	Agora, é possível criar um RDS Proxy com um banco de dados do RDS para PostgreSQL versão 14. Para ter mais informações sobre o RDS Proxy, consulte o tópico sobre como Utilizar o proxy do Amazon RDS.	13 de dezembro de 2022
O Amazon RDS para Oracle é compatível com as classes de instância db.x2idn, db.x2iedn e db.x2iezn	Agora você pode usar as classes de instância db.x2idn, db.x2iedn e db.x2iezn de instâncias de banco de dados do Amazon RDS para Oracle. Para ter mais informações, consulte Supported DB engines for DB instance classes (Mecanismos de banco de dados compatíveis para classes de instância de banco de dados) Supported RDS for Oracle instance classes (Classes de instância compatíveis do RDS para Oracle).	12 de dezembro de 2022

As instâncias de banco de dados do RDS para PostgreSQL são compatíveis com Trusted Language Extensions para PostgreSQL	O Trusted Language Extensions para PostgreSQL é um kit de desenvolvimento de código aberto que permite criar extensões PostgreSQL de alta performance e executá-las com segurança em sua instância de banco de dados do RDS para PostgreSQL. Para ter mais informações, consulte Working with Trusted Language Extensions for PostgreSQL (Trabalhar com Trusted Language Extensions para PostgreSQL).	30 de novembro de 2022
Usar implantações azul/verde do Amazon RDS para atualizações de banco de dados	Você pode fazer alterações em uma instância de banco de dados em um ambiente de teste e testar as alterações sem afetar seu cluster de banco de dados de produção. Quando estiver pronto, você poderá promover o ambiente de teste como o novo ambiente de banco de produção, com tempo de inatividade mínimo. Para ter mais informações, consulte Using Amazon RDS Blue/Green Deployments for database updates (Usar implantações azul/verde do Amazon RDS para atualizações de banco de dados).	27 de novembro de 2022

As gravações otimizadas pelo Amazon RDS melhora a performance das transações de gravação para o RDS para MySQL	Você pode melhorar a performance das transações de gravação para instâncias de banco de dados do RDS para MySQL com as gravações otimizadas pelo Amazon RDS. Para ter mais informações, consulte Melhorar a performance das gravações com gravações otimizadas pelo Amazon RDS para MySQL.	27 de novembro de 2022
As leituras otimizadas do Amazon RDS melhoram a performance das consultas do RDS para MySQL	É possível acelerar o processamento de consultas para instâncias de banco de dados do RDS para MySQL com leituras otimizadas do Amazon RDS. Para ter mais informações, consulte Improving query performance with Amazon RDS Optimized Writes (Melhorar a performance das consultas com gravações otimizadas pelo Amazon RDS).	27 de novembro de 2022
O Amazon RDS está disponível na região Ásia-Pacífico (Hyderabad)	O Amazon RDS agora está disponível na região Ásia-Pacífico (Hyderabad). Para ter mais informações, consulte Regiões e zonas de disponibilidade.	22 de novembro de 2022

O RDS é compatível com o MariaDB 10.6.11, 10.5.18, 10.4.27 e 10.3.37	Agora é possível criar instâncias de banco de dados do Amazon RDS executando as versões 10.6.11, 10.5.18, 10.4.27 e 10.3.37 do MariaDB. Para ter mais informações, consulte Versões do MariaDB no Amazon RDS.	18 de novembro de 2022
O RDS Custom para Oracle é compatível com a definição de parâmetros de instalação não padrão em uma versão de mecanismo personalizada (CEV)	Ao criar uma CEV, você pode definir valores não padrão para a base da Oracle, Oracle home, nome de usuário e ID do UNIX e nome e ID do grupo UNIX. Dessa forma, você obtém maior controle sobre a instalação do banco de dados em sua instância de banco de dados do RDS Custom para Oracle. Para ter mais informações, consulte Preparar o manifesto de CEV.	18 de novembro de 2022
O Amazon RDS é compatível com a versão 22.1.v1 do Oracle APEX	Você pode usar o APEX 22.1.v1 com todas as versões compatíveis do Oracle Database. Para ter mais informações, consulte Oracle Application Express.	18 de novembro de 2022

O RDS para SQL Server é compatível com réplicas de leitura entre regiões	Agora você pode criar uma réplica de leitura entre regiões para aprimorar a capacidade de recuperação de desastres , reduzir a latência de leitura da aplicação e descarreg ar workloads de leitura da instância de banco de dados primária. Para ter mais informações, consulte Criar uma réplica de leitura em uma Região da AWS diferente.	16 de novembro de 2022
O Amazon RDS está disponível na região da Europa (Espanha)	O Amazon RDS agora está disponível na região da Europa (Espanha). Para ter mais informações, consulte Regiões e zonas de disponibi lidade.	16 de novembro de 2022
O RDS para SQL Server é compatível com servidores vinculados ao banco de dados Oracle	Agora você pode criar um servidor vinculado para acessar bancos de dados Oracle externos para ler dados e executar comandos SQL. Para ter mais informações, consulte Linked Servers with Oracle OLEDB with RDS for SQL Server (Servidor es vinculados com Oracle OLEDB com RDS para SQL Server).	15 de novembro de 2022

O RDS Custom para Oracle é compatível com Oracle Multitenant	Você pode criar uma instância de banco de dados do RDS Custom para Oracle como um banco de dados de contêiner (CDB). Após a criação, o CDB contém a raiz do CDB, a semente do PDB e um PDB. Você pode adicionar outros PDBs manualmente utilizando o Oracle SQL. Para ter mais informações, consulte Overview of Amazon RDS Custom for Oracle architecture (Visão geral da arquitetura do Amazon RDS Custom para Oracle).	15 de novembro de 2022
O Amazon RDS para Oracle é compatível com a integração com o Amazon EFS	Se você adicionar a opção `EFS_INTEGRATION` ao seu grupo de opções, poderá transferir arquivos entre uma instância de banco de dados do RDS para Oracle e um sistema de arquivos EFS. Para ter mais informações, consulte Amazon EFS.	15 de novembro de 2022
O RDS é compatível com MySQL 8.0.31 e 5.7.40	Agora é possível criar instâncias de banco de dados do Amazon RDS executando o MySQL versão 8.0.31 e 5.7.40. Para ter mais informações, consulte Versões do MySQL no Amazon RDS.	10 de novembro de 2022

O Amazon RDS está disponível na região da Europa (Zurique)	O Amazon RDS agora está disponível na região da Europa (Zurique). Para ter mais informações, consulte Regiões e zonas de disponibilidade.	9 de novembro de 2022
O acesso aos backups de logs de transações agora está disponível para o RDS para SQL Server	Agora é possível visualizar e copiar os backups de logs de transações do banco de dados para um bucket do Amazon S3. Para ter mais informações, consulte Access to transaction log backups (Acesso a backups de logs de transações).	7 de novembro de 2022
Clusters de banco de dados multi-AZ compatíveis em Regiões da AWS adicionais	Clusters de banco de dados multi-AZ agora estão disponíveis em Regiões da AWS adicionais. Para ter mais informações, consulte Supported Regions and DB engines for Multi-AZ DB clusters in Amazon RDS.	4 de novembro de 2022

O Amazon RDS é compatível com armazenamento gp3	Agora você pode criar instâncias de banco de dados do Amazon RDS que usam volumes de armazenamento SSD de uso geral (gp3) do Amazon EBS, que permitem personalizar a performance do armazenamento independentemente da capacidade de armazenamento. Para ter mais informações, consulte General Purpose SSD storage (Armazenamento SSD de uso geral).	4 de novembro de 2022
O Amazon RDS é compatível com um novo evento para atualizações do sistema operacional	O Amazon RDS agora é compatível com um novo evento de instância de banco de dados, RDS-EVENT-0230, na categoria de evento de correção de segurança. Esse novo evento alerta você quando uma atualização do sistema operacional está disponível para sua instância de banco de dados. Para ter mais informações, consulte Monitorar eventos do Amazon RDS e Working with operating system updates (Trabalhar com atualizações do sistema operacional).	28 de outubro de 2022

O Amazon RDS para Oracle é compatível com as classes de instância otimizada para memória r5b pré-configuradas	As novas classes de instância de banco de dados do Oracle db.r5b são otimizadas para workloads que necessitem de mais memória, armazenamento e E/S por vCPU. Por exemplo, db.r5b.4xlarge.tpc2.mem2x tem o multithreading ativado e fornece duas vezes mais memória do que db.r5b.4xlarge. Para ter mais informações, consulte Suporte a classes de instância de banco de dados do RDS for Oracle.	27 de outubro de 2022
O Amazon RDS é compatível com 15 réplicas de leitura do RDS para MariaDB, MySQL e instâncias de banco de dados do PostgreSQL	Agora é possível criar até quinze réplicas de leitura para o RDS para MariaDB, MySQL e instâncias de banco de dados do PostgreSQL. Para ter mais informações sobre réplicas de leitura, consulte Como trabalhar com réplicas de leitura.	20 de outubro de 2022
O Amazon RDS para PostgreSQL agora é compatível com o PostgreSQL versão 15 RC 3 no ambiente de visualização do banco de dados	O PostgreSQL versão 15 Beta 3 já está disponível no ambiente de visualização de banco de dados na Região da AWS Leste dos EUA (Ohio). Para ter mais informações, consulte o tópico sobre como Trabalhar com o ambiente de visualização do banco de dados.	18 de outubro de 2022

O Amazon RDS é compatível com a configuração automática da conectividade entre um banco de dados do RDS e uma instância do EC2	Você pode usar o AWS Management Console para configurar a conectividade entre uma instância de banco de dados existente do RDS ou um cluster de banco de dados multi-AZ e uma instância do EC2. Para ter mais informações, consulte Conectar uma instância do EC2 e um banco de dados do RDS automaticamente.	14 de outubro de 2022
AWS JDBC Driver para PostgreSQL disponível para o público em geral	O AWS JDBC Driver para PostgreSQL é um driver cliente desenvolvido para o RDS para PostgreSQL. O AWS JDBC Driver para PostgreSQL já está disponível para o público em geral. Para ter mais informações, consulte Conectar-se ao AWS JDBC Driver para PostgreSQL.	6 de outubro de 2022
O Amazon RDS para Oracle é compatível com o Oracle APEX versão 21.2.v1	O APEX 21.2 inclui o patch 33420059. Para obter informações, consulte Requisitos da versão do APEX.	3 de outubro de 2022
O RDS é compatível com o MySQL 5.7.39	Agora é possível criar instâncias de banco de dados do Amazon RDS que executam o MySQL versões 5.7.39. Para ter mais informações, consulte Versões do MySQL no Amazon RDS.	29 de setembro de 2022

O RDS é compatível com o MariaDB 10.6.10	Agora é possível criar instâncias de banco de dados do Amazon RDS executando a versão 10.6.10 do MariaDB. Para ter mais informações, consulte Versões do MariaDB no Amazon RDS.	29 de setembro de 2022
O RDS Proxy é compatível com o RDS para SQL Server	Agora, é possível criar um RDS Proxy para uma instância de banco de dados do RDS que execute o Microsoft SQL Server versão 2014 ou posterior. Para ter mais informações sobre o RDS Proxy, consulte o tópico sobre como Utilizar o proxy do Amazon RDS.	19 de setembro de 2022
O RDS é compatível com o MariaDB 10.5.17, 10.4.26 e 10.3.36	Agora é possível criar instâncias de banco de dados do Amazon RDS executando o MariaDB versões 10.5.17, 10.4.26 e 10.3.36. Para ter mais informações, consulte Versões do MariaDB no Amazon RDS.	15 de setembro de 2022

O Amazon RDS para Oracle é compatível com o armazenamento de instância local para dados temporários	Agora você pode iniciar o Amazon RDS para Oracle nos tipos de instância db.r5d e db.m5d do Amazon EC2 com o espaço de tabela temporário e o Cache Flash Inteligente de banco de dados (cache flash) configurados para usar um armazenamento de instância. Ao armazenar dados temporários localmente, você pode obter latências de leitura e gravação mais baixas em comparação com o armazenamento padrão baseado em Amazon EBS. Para ter mais informações, consulte Armazenar dados temporários do Oracle no armazenamento de instância.	14 de setembro de 2022
O Performance Insights mostra as 25 principais consultas SQL	No painel do Performance Insights, a guia Top SQL (SQL principal) mostra as 25 consultas SQL que mais estão contribuindo para a carga do banco de dados. Para ter mais informações, consulte Visão geral da guia Top SQL (SQL principal).	13 de setembro de 2022

O RDS é compatível com o MySQL 8.0.30	Agora é possível criar instâncias de banco de dados do Amazon RDS que executam a versão 8.0.30 do o MySQL. Para ter mais informações, consulte Versões do MySQL no Amazon RDS.	9 de setembro de 2022
O Amazon RDS está disponível na região Oriente Médio (EAU)	O Amazon RDS já está disponível na região Oriente Médio (EAU). Para ter mais informações, consulte Regiões e zonas de disponibilidade.	30 de agosto de 2022
Amazon RDS para SQL Server oferece suporte a assinaturas do SSRS Email	Agora você pode usar a extensão SQL Server Reporting Services (SSRS) Email para enviar relatório s aos usuários e assinar relatórios no servidor de relatórios. Para ter mais informações, consulte Suporte para o SQL Server Reporting Services no RDS para SQL Server.	26 de agosto de 2022
RDS para Oracle oferece suporte a backups de réplicas de leitura	Você pode ativar backups automáticos e criar snapshots manuais de réplicas do RDS para Oracle. Para ter mais informações, consulte Como trabalhar com backups de réplicas do RDS para Oracle.	23 de agosto de 2022

RDS para Oracle oferece suporte à transição do Oracle Data Guard	Uma transição é uma inversão de função entre um banco de dados primário e uma réplica montada ou aberta do Oracle. Durante uma transição, o banco de dados primário original faz a transição para uma função auxiliar, enquanto o banco de dados de espera original faz a transição para a função primária. Para ter mais informações, consulte Executar uma transição do Oracle Data Guard.	23 de agosto de 2022
Amazon RDS oferece suporte à configuração automática da conectividade com uma instância do EC2	Ao criar uma instância de banco de dados ou um cluster de banco de dados multi-AZ, você pode usar o AWS Management Console para configurar a conectividade entre uma instância do Amazon Elastic Compute Cloud e a nova instância ou o novo cluster de banco de dados. Para ter mais informações, consulte Configurar a conectividade automática de rede com uma instância do EC2 para uma nova instância de banco de dados e Configurar a conectividade automática de rede com uma instância do EC2 para um novo cluster de banco de dados.	22 de agosto de 2022

RDS Custom para Oracle oferece suporte à promoção de réplicas do Oracle	Se você usar o RDS Custom para Oracle, poderá promover suas réplicas gerenciadas do Oracle usando o comando `promote-read-replica` da CLI. Além disso, você pode excluir sua instância de banco de dados primária, o que faz com que o RDS Custom para Oracle promova suas réplicas gerenciadas do Oracle para instâncias independentes. Para ter mais informações, consulte Como trabalhar com réplicas do Oracle para o RDS Custom para Oracle.	5 de agosto de 2022
RDS para MySQL oferece suporte à aplicação de conexões SSL/TLS	O RDS para MySQL agora oferece suporte à aplicação de conexões SSL/TLS definindo parâmetro `require_secure_transport` como `ON`. Para ter mais informações, consulte Exigir uma conexão SSL/TLS para uma instância de banco de dados do MySQL.	1º de agosto de 2022

Amazon RDS encerra o suporte para Oracle Database 12c versão 1 (12.1.0.2)	O suporte à versão 12.1.0.2 foi encerrado para os modelos de licenciamento BYOL e LI. Em 1.º de agosto de 2022, o RDS para Oracle inicia atualizações automáticas de instâncias de banco de dados 12c versão 1 (12.1.0.2) e snapshots restaurados da 12.1.0.2 para o Oracle Database 19c. Para obter mais informações, consulte a linha do tempo de fim do suporte em AWS re:Post.	1º de agosto de 2022
RDS Proxy oferece suporte ao RDS para MariaDB	Agora você pode criar um RDS Proxy para uma instância de banco de dados do RDS que executa a versão 10.2, 10.3, 10.4 ou 10.5 do MariaDB. O suporte ao MariaDB está incluído na família de mecanismo s MySQL. Para ter mais informações sobre o RDS Proxy, consulte o tópico sobre como Utilizar o proxy do Amazon RDS.	26 de julho de 2022

RDS para MariaDB oferece suporte às classes de instância de banco de dados db.r5b	Agora você pode criar instâncias de bancos de dados do RDS para MariaDB que usam as classes de instância s de banco de dados db.r5b. Para ter mais informações, consulte Mecanismos de banco de dados compatíveis com as classes de instâncias de banco de dados.	25 de julho de 2022
RDS para Oracle oferece suporte à modificação de fluxos de atividades de banco de dados	Se você usar o RDS para Oracle, poderá alterar o estado da política de auditoria de um fluxo de atividade s do banco de dados para bloqueado (padrão) ou desbloqueado. Em vez de interromper um fluxo de atividades, você pode desbloquear o estado da política, personalizar sua política de auditoria e, depois, rebloquear o estado da política. Para ter mais informações, consulte Modificar um fluxo de atividades do banco de dados.	22 de julho de 2022

Performance Insights compatível com a Região Ásia-Pacífico (Jacarta)	Anteriormente, não era possível usar o Performance Insights na Região Ásia-Pacífico (Jacarta). Essa restrição não existe mais. Para ter mais informações, consulte Supported Regions and DB engines for Performance Insights in Amazon RDS.	21 de julho de 2022
Fim do suporte ao Microsoft SQL Server 2012 no Amazon RDS	Acabou o suporte ao Microsoft SQL Server 2012, o que coincide com o plano da Microsoft de encerrar o suporte estendido a essa versão em 12 de julho de 2022. As instâncias existentes do Microsoft SQL Server 2012 serão atualizados automaticamente para a versão secundária mais recente do Microsoft SQL Server 2014 a partir de 1.º de junho de 2022. Para ter mais informações, consulte Suporte ao Microsoft SQL Server 2012 no Amazon RDS.	12 de julho de 2022

O RDS é compatível com o MariaDB 10.6.8, 10.5.16, 10.4.25, 10.3.35 e 10.2.44	Agora é possível criar instâncias de banco de dados do Amazon RDS que executam o MariaDB 10.6.8, 10.5.16, 10.4.25, 10.3.35 e 10.2.44. Para ter mais informações, consulte Versões do MariaDB compatíveis com o Amazon RDS.	8 de julho de 2022
RDS Performance Insights oferece suporte a períodos de retenção adicionais	Anteriormente, o Performance Insights oferecia apenas dois períodos de retenção: 7 dias (padrão) ou 2 anos (731 dias). Agora, se você precisar reter seus dados de desempenho por mais de 7 dias, especifiq ue de 1 a 24 meses. Para ter mais informações, consulte Preços e retenção de dados para o Performance Insights.	1° de julho de 2022
O RDS Custom é compatíve l com as regiões da Ásia-Pacífico (Mumbai) e Europa (Londres)	Você pode criar instâncias de banco de dados do RDS Custom para Oracle e RDS Custom para SQL Server em duas novas Regiões da AWS: Ásia-Pacífico (Mumbai) e Europa (Londres). Para ter mais informações, consulte Compatibilidade da Região da AWS com o RDS Custom para Oracle e Compatibilidade da Região da AWS com o RDS Custom para SQL Server.	21 de junho de 2022

O RDS Custom para Oracle é compatível com o Oracle Database 18c e 12c versão 2 (12.2)	Agora você pode criar um CEV para o RDS Custom para Oracle usando arquivos de instalação do Oracle Database 18c e 12c versão 2 (12.2). Você pode usar esses CEVs para criar uma instância de banco de dados do RDS Custom para Oracle. Veja mais informações em Trabalhar com versões de mecanismo personalizadas para o Amazon RDS Custom for Oracle.	21 de junho de 2022
Clusters de banco de dados multi-AZ são compatíveis com as classes de instância de banco de dados db.m5d e db.r5d	Agora você pode criar clusters de bancos de dados multi-AZ que usam as classes de instâncias de banco de dados db.m5d e db.r5d. Para ter mais informações, consulte Implantações de clusters de banco de dados multi-AZ e Tipos de classe de instância de banco de dados.	21 de junho de 2022
Clusters de banco de dados multi-AZ disponíveis em mais Regiões da AWS	Agora você pode criar clusters de banco de dados multi-AZ nas seguintes regiões: Europa (Frankfurt) e Europa (Estocolmo). Para ter mais informações, consulte Implantações de clusters de banco de dados multi-AZ.	21 de junho de 2022

O RDS para Microsoft SQL Server é compatível com a migração de bancos de dados que usam a Transparent Data Encryption (TDE)	O RDS para SQL Server agora é compatível com a migração de bancos de dados Microsoft SQL Server com a TDE ativada, usando backup e restauração nativos. Para ter mais informações, consulte Compatibilidade com a Transparent Data Encryption no SQL Server.	14 de junho de 2022
O Amazon RDS é compatível com a publicação de eventos em tópicos criptografados do Amazon SNS	O Amazon RDS agora pode publicar eventos em tópicos do Amazon Simple Notification Service (Amazon SNS) com criptografia do lado do servidor (SSE) ativada, para proteção adicional de eventos que transportam dados confidenciais. Para ter mais informações, consulte Assinar notificações de eventos do Amazon RDS.	1º de junho de 2022
O RDS é compatível com o MySQL 5.7.38	Agora é possível criar instâncias de banco de dados do Amazon RDS que executam o MySQL versão 5.7.38. Para ter mais informações, consulte Versões do MySQL no Amazon RDS.	31 de maio de 2022

O RDS para PostgreSQL é compatível com réplicas de leitura em cascata	Agora é possível usar réplicas de leitura em cascata com o RDS para PostgreSQL versão 14.1 e versões superiores. Para ter mais informações, consulte Trabalhar com réplicas de leitura no Amazon RDS.	4 de maio de 2022
O Amazon RDS em AWS Outposts é compatível com as operações de autoscaling e de armazenamento em escala	Agora é possível alterar os tamanhos de armazenamento de instâncias de banco de dados no Outpost e usar o autoscaling do armazenamento. Para ter mais informações, consulte Suporte ao Amazon RDS on AWS Outposts para recursos do Amazon RDS.	2 de maio de 2022
Clusters de banco de dados multi-AZ disponíveis em mais Regiões da AWS	Agora você pode criar clusters de banco de dados multi-AZ nas seguintes regiões: Ásia-Pacífico (Singapura) e Ásia-Pacífico (Sydney). Para ter mais informações, consulte Implantações de clusters de banco de dados multi-AZ.	29 de abril de 2022

O Amazon RDS é compatível com o modo de pilha dupla	As instâncias de banco de dados agora podem ser executadas no modo de pilha dupla. Nesse modo, os recursos podem se comunicar com o cluster de instância de banco de dados por IPv4, IPv6 ou ambos. Para ter mais informações, consulte Endereçamento IP do Amazon RDS.	29 de abril de 2022
O Amazon RDS publica métricas de uso no Amazon CloudWatch	O namespace `AWS/Usage` no Amazon CloudWatch inclui métricas de uso específico da conta para suas cotas de serviço do Amazon RDS. Para ter mais informações, consulte Métricas de uso do Amazon CloudWatch para o Amazon RDS.	28 de abril de 2022
O Amazon RDS para MySQL é compatível com as classes de instâncias de banco de dados db.m6i e db.r6i	Agora você pode usar as instâncias de banco de dados db.m6i e db.r6i do Amazon RDS que estejam executando o MySQL. Para ter mais informações, consulte Mecanismos de banco de dados com suporte a todas as classes de instâncias de banco de dados disponíveis.	28 de abril de 2022

O Amazon RDS para PostgreSQL é compatível com as classes de instâncias de banco de dados db.m6i e db.r6i	Agora é possível usar as instâncias de banco de dados db.m6i e db.r6i do Amazon RDS que estejam executando o PostgreSQL. Para ter mais informações, consulte Mecanismos de banco de dados com suporte a todas as classes de instâncias de banco de dados disponíveis.	27 de abril de 2022
O Amazon RDS para MariaDB é compatível com as classes de instâncias de banco de dados db.m6i e db.r6i	Agora você pode usar as instâncias de banco de dados db.m6i e db.r6i do Amazon RDS que estejam executando o MariaDB. Para ter mais informações, consulte Mecanismos de banco de dados com suporte a todas as classes de instâncias de banco de dados disponíveis.	26 de abril de 2022
O Amazon RDS on AWS Outposts é compatível com implantações multi-AZ	Agora você pode criar uma instância de banco de dados em espera em outro Outpost. Para ter mais informações, consulte Suporte ao Amazon RDS on AWS Outposts para recursos do Amazon RDS.	19 de abril de 2022

O Amazon RDS para Oracle é compatível com as classes de instâncias de banco de dados db.m6i e db.r6i	Se você executar o Oracle Database 19c, poderá usar as classes de instâncias db.m6i e db.r6i. As classes db.m6i: são de instância de uso geral adequadas para uma grande variedade de workloads. Para ter mais informações, consulte Suporte a classes de instância de banco de dados do RDS for Oracle.	8 de abril de 2022
O Amazon RDS para SQL Server é compatível com a replicação de trabalhos do SQL Server Agent	Quando você ativa esse recurso, os trabalhos do SQL Server Agent criados, modificados ou excluídos no host principal são automatic amente sincronizados com o host secundário em uma configuração multi-AZ. Para ter mais informações, consulte Usar o SQL Server Agent.	7 de abril de 2022
O Amazon RDS é compatíve l com o proxy do RDS com RDS para PostgreSQL versão 13	Agora, é possível criar um proxy do RDS com um banco de dados do RDS para PostgreSQL versão 13. Para ter mais informações sobre o RDS Proxy, consulte o tópico sobre como Utilizar o proxy do Amazon RDS.	4 de abril de 2022

O Amazon RDS pretende desativar o Oracle Database 12c	O Oracle Database 12c está prestes a ser desativado. A Oracle Corporation não fornecerá mais patches para o Oracle Database 12c após as datas de término do suporte. O Amazon RDS pretende começar a atualizar automatic amente as instâncias de banco de dados do Oracle 12c Database para o Oracle Database 19c.	22 de março de 2022
Notas de lançamento do Amazon RDS para PostgreSQL	Agora existe um guia separado para as notas de lançamento do Amazon RDS para PostgreSQL. Para ter mais informações, consulte Notas de lançamento do Amazon RDS para PostgreSQL.	22 de março de 2022
Notas de lançamento do Amazon RDS para Oracle	Agora existe um guia separado para as notas de lançamento do Amazon RDS para Oracle. Para ter mais informações, consulte Notas de lançamento do Amazon RDS para Oracle.	22 de março de 2022

Clusters de banco de dados multi-AZ disponíveis em mais Regiões da AWS	Agora você pode criar clusters de banco de dados multi-AZ nas seguintes regiões: Leste dos EUA (Ohio) e Ásia-Pacífico (Tóquio). Para ter mais informações, consulte Implantações de clusters de banco de dados multi-AZ.	15 de março de 2022
Amazon RDS para PostgreSQL versões 14.2, 13.6, 12.10, 11.15 e 10.20	O RDS para PostgreSQL agora é compatível com as versões 14.2, 13.6, 12.10, 11.15 e 10.20. As versões 14.2 e 13.6 agora são compatíveis com dois novos invólucros de dados externos. A extensão `mysql_fdw` permite que o PostgreSQL trabalhe com dados armazenados em bancos de dados MySQL, MariaDB e Aurora MySQL. A extensão `tds_fdw` permite que o PostgreSQL trabalhe com dados armazenados em bancos de dados SQL Server. Para ter mais informações, consulte Versões compatíveis do banco de dados PostgreSQL.	12 de março de 2022

O RDS é compatível com o MySQL 5.7.37	Agora é possível criar instâncias de banco de dados do Amazon RDS que executam a versão 5.7.37 do MySQL. Para ter mais informações, consulte Versões do MySQL no Amazon RDS.	11 de março de 2022
O Amazon RDS para SQL Server é compatível com novas classes de instâncias de banco de dados	Agora você pode criar instâncias de bancos de dados do Amazon RDS que executam o Microsoft SQL Server e usam as classes de instâncias de banco de dados db.m6i e db.r6i. Para ter mais informações, consulte Suporte a classes de instância de banco de dados para Microsoft SQL Server.	9 de março de 2022
O Amazon RDS para Oracle é compatível com o Oracle Database 21c	Agora é possível criar instâncias de banco de dados do Amazon RDS que executam o Oracle Database 21c (21.0.0.0). Essa é a primeira versão do Oracle Database que é compatíve l somente com a arquitetura multilocatário (CDB). Para ter mais informações, consulte Oracle Database 21c com o Amazon RDS.	7 de março de 2022

O RDS é compatível com o MariaDB 10.6.7, 10.5.15, 10.4.24, 10.3.34 e 10.2.43	Agora é possível criar instâncias de banco de dados do Amazon RDS executando as versões 10.6.7, 10.5.15, 10.4.24, 10.3.34 e 10.2.43 do MariaDB. Para ter mais informações, consulte Versões do MariaDB no Amazon RDS.	3 de março de 2022
JDBC Driver para MySQL da AWS disponível para o público em geral	O JDBC Driver para MySQL da AWS é um driver de cliente projetado para o RDS para MySQL. O JDBC Driver para MySQL da AWS já está disponível para o público em geral. Para ter mais informações, consulte Conectar com o JDBC Driver para MySQL da Amazon Web Services.	2 de março de 2022
Clusters de banco de dados multi-AZ disponíveis ao público	Uma implantação de cluster de banco de dados multi-AZ é um modo de implantação de alta banco de dados primário Amazon RDS com duas instâncias de banco de dados em espera legíveis. Os clusters de banco de dados multi-AZ já estão disponíveis ao público em geral. Para ter mais informações, consulte Implantações de clusters de banco de dados multi-AZ.	1º de março de 2022

O RDS é compatível com o MySQL 8.0.28	Agora é possível criar instâncias de banco de dados do Amazon RDS que executam a versão 8.0.28 do MySQL. Para ter mais informações, consulte Versões do MySQL no Amazon RDS.	28 de fevereiro de 2022
O Amazon RDS para Oracle é compatível com novas configurações para a Native Network Encryption (NNE – Criptografia nativa da rede)	Para controlar se os clientes podem se conectar com métodos de criptografia e soma de verificação não seguros, defina `SQLNET.ALLOW_WEAK_CRYPTO_CLIENTS` e `SQLNET.ALLOW_WEAK_CRYPTO` na opção NNE. Exemplos de métodos inseguros incluem DES, 3DES, RC4 e MD5. Para ter mais informações, consulte Configurações de opção do NNE.	25 de fevereiro de 2022
O Amazon RDS para SQL Server oferece suporte a grupos de disponibilidade Always On para Microsoft SQL Server 2017 Standard Edition	Quando você cria uma instância de banco de dados usando a configuração multi-AZ no SQL Server 2017 Standard Edition 14.00.3401.7 e versões posteriores, o RDS usa automaticamente os grupos de disponibilidade. Para ter mais informações, consulte Implantações multi-AZ para Microsoft SQL Server.	18 de fevereiro de 2022

O RDS for Oracle é compatível com transmissões de atividades de bancos de dados na região Ásia-Pacífico (Jacarta)	Para ter mais informações, consulte Compatibilidade de Regiões da AWS com transmissões de atividades de bancos de dados.	16 de fevereiro de 2022
Amazon RDS Custom para Oracle compatível com Oracle Database 12.1	Agora você pode criar versões de mecanismo personalizadas para o RDS Custom for Oracle que usam o Oracle Database 12.1 Enterprise Edition. Veja mais informações em Trabalhar com versões de mecanismo personalizadas para o Amazon RDS Custom for Oracle.	4 de fevereiro de 2022
O Amazon RDS para MariaDB é compatível com uma nova versão principal	Agora é possível criar instâncias de banco de dados do Amazon RDS que executam a versão 10.6 do MariaDB. Para ter mais informações, consulte Compatibilidade do MariaDB 10.6 com o Amazon RDS.	3 de fevereiro de 2022

O Performance Insights oferece suporte à captura de planos para consultas Oracle	O console do Performance Insights oferece suporte a uma nova dimensão de plano para o melhor SQL. Ao separar por plano, você pode ver quais planos suas principais consultas Oracle estão usando. Se uma consulta usa vários planos, você poderá comparar os planos lado a lado no console e determinar qual plano é mais eficiente. Você também pode detalhar para ver quais etapas em um plano têm o maior custo. Para ter mais informações, consulte Análise dos planos de execução do Oracle usando o painel do Performance Insights.	27 de janeiro de 2022
O Performance Insights oferece suporte a novas APIs	O Performance Insights é compatível com as seguintes APIs: `GetResourceMetadata`, `ListAvailableResourceDimensions`, e `ListAvailableResourceMetrics`. Para ter mais informações, consulte Recuperando métricas com a API do Insights de Performance neste manual e na Referência de API do Insights de Performance do Amazon RDS.	12 de janeiro de 2022

O proxy do RDS oferece suporte a eventos	O proxy do RDS agora gera eventos que você pode assinar e visualizar no CloudWatch Events ou configurar para enviar para o Amazon EventBridge. Para ter mais informações consulte Trabalhando com eventos RDS Proxy.	11 de janeiro de 2022
O Amazon RDS para SQL Server é compatível com o modo SSAS Multidimensional	O Amazon RDS para SQL Server oferece suporte à execução do SQL Server Analysis Services (SSAS) no modo Tabular ou Multidimensional. Para ter mais informações, consulte Suporte para o SQL Server Analysis Services no RDS for SQL Server.	7 de janeiro de 2022
Proxy do RDS disponível em Regiões da AWS adicionais	O proxy do RDS já está disponível nas seguintes regiões: África (Cidade do Cabo), Ásia-Pacífico (Hong Kong), Ásia-Pacífico (Osaka), Europa (Milão), Europa (Paris), Oriente Médio (Bahrein) e América do Sul (São Paulo). Para ter mais informações sobre o RDS Proxy, consulte o tópico sobre como Utilizar o proxy do Amazon RDS.	5 de janeiro de 2022

O RDS é compatível com o MySQL 8.0.27	Agora é possível criar instâncias de banco de dados do Amazon RDS que executam a versão 8.0.27 do MySQL. Para ter mais informações, consulte Versões do MySQL no Amazon RDS.	21 de dezembro de 2021
Amazon RDS disponível na região Ásia-Pacífico (Jacarta)	O Amazon RDS e posterior na região Ásia-Pacífico (Jacarta) . Para ter mais informações, consulte Regiões e zonas de disponibilidade.	13 de dezembro de 2021
O Amazon RDS é compatível com o MariaDB 10.5.13, 10.4.22, 10.3.32 e 10.2.41	Agora é possível criar instâncias de banco de dados do Amazon RDS executando as versões 10.5.13, 10.4.22, 10.3.32 e 10.2.41 do MariaDB. Para ter mais informações, consulte Versões do MariaDB no Amazon RDS.	8 de dezembro de 2021

Amazon RDS Custom for SQL Server	O Amazon RDS Custom é um serviço de banco de dados gerenciado para aplicaçõe s herdadas, personalizadas e em pacote que exigem acesso ao sistema operacion al subjacente e ao ambiente de banco de dados. Com o Amazon RDS Custom, você tem a automação do Amazon RDS e a flexibilidade do Amazon EC2. Para ter mais informações, consulte o tópico sobre como Trabalhar com o Amazon RDS Custom.	1° de dezembro de 2021
Clusters de banco de dados multi-AZ (versão de demonstra ção)	Agora, é possível criar clusters de banco de dados multi-AZ para o RDS para MySQL e o RDS para PostgreSQL. Uma implantação de cluster de banco de dados multi-AZ é um modo de implantação de alta disponibilidade do Amazon RDS com duas instâncias de banco de dados em espera legíveis. Clusters de banco de dados multi-AZ estão em versão de demonstração. Para ter mais informações, consulte Implantações de clusters de banco de dados multi-AZ (versão de demonstração).	23 de novembro de 2021

O Amazon RDS é compatível com o proxy do RDS com RDS para PostgreSQL versão 12	Agora, é possível criar um proxy do RDS com um banco de dados do RDS para PostgreSQL versão 12. Para ter mais informações sobre o RDS Proxy, consulte o tópico sobre como Utilizar o proxy do Amazon RDS.	22 de novembro de 2021
O Amazon RDS on AWS Outposts oferece suporte para backups locais	É possível armazenar backups automatizados e snapshots manuais na sua Região da AWS ou localmente no Outpost. Para ter mais informações, consulte Suporte ao Amazon RDS on AWS Outposts para recursos do Amazon RDS.	22 de novembro de 2021
Suporte do Amazon RDS para AWS KMS keys entre contas	É possível utilizar uma chave do KMS de outra conta da AWS para criptografia ao exportar snapshots de banco de dados para o Amazon S3. Para ter mais informações, consulte Exportar dados de snapshot de banco de dados para o Amazon S3.	3 de novembro de 2021

O Amazon RDS on AWS Outposts agora oferece suporte à publicação de logs de mecanismos de banco de dados no CloudWatch Logs	O RDS on Outposts agora oferece suporte à publicação de logs de mecanismos de banco de dados no CloudWatch Logs. Para ter mais informações, consulte Suporte ao Amazon RDS no AWS Outposts para recursos do Amazon RDS.	2 de novembro de 2021
Amazon RDS Custom for Oracle	O Amazon RDS Custom é um serviço de banco de dados gerenciado para aplicações herdadas, personalizadas e em pacote que exigem acesso ao sistema operacional subjacente e ao ambiente de banco de dados. Com o Amazon RDS Custom, você tem a automação do Amazon RDS e a flexibilidade do Amazon EC2. Para ter mais informações, consulte o tópico sobre como Trabalhar com o Amazon RDS Custom.	26 de outubro de 2021
Suporte para replicação atrasada para RDS para MySQL versão 8.0	Começando com o RDS para MySQL versão 8.0.26, é possível configurar a replicação atrasada para instâncias de banco de dados do RDS para MySQL versão 8.0. Para ter mais informações, consulte Configuração de replicação atrasada com o MySQL.	25 de outubro de 2021

Suporte para MySQL 8.0.26	Agora é possível criar instâncias de banco de dados do Amazon RDS executando o MySQL versão 8.0.26. Para ter mais informações, consulte Versões do MySQL no Amazon RDS.	25 de outubro de 2021
Suporte para replicação baseada em GTID para o RDS para MySQL versão 8.0	Começando com o RDS para MySQL versão 8.0.26, é possível configurar a replicação baseada em GTID para instâncias de banco de dados do RDS para MySQL versão 8.0. Para ter mais informações, consulte Usar a replicação baseada em GTID para o Amazon RDS MySQL	25 de outubro de 2021
O Amazon RDS é compatível com o proxy do RDS com RDS para MySQL 8.0	Agora, é possível criar um proxy do RDS para uma instância de banco de dados do RDS para MySQL 8.0. Para ter mais informações, consulte o tópico sobre como Usar o proxy do Amazon RDS.	21 de outubro de 2021
O Amazon RDS on AWS Outposts é compatível com versões adicionais do RDS para MySQL	O RDS on Outposts agora é compatível com o RDS para MySQL versões 8.0.23 e 8.0.25. Para ter mais informações, consulte Suporte ao Amazon RDS no AWS Outposts para recursos do Amazon RDS.	20 de outubro de 2021

O Amazon RDS para PostgreSQL agora é compatível com o PostgreSQL versão 14 RC 1 no ambiente de visualização do banco de dados	O PostgreSQL versão 14 RC 1 já está disponível no ambiente de visualização de banco de dados na Região da AWS Leste dos EUA (Ohio). Para ter mais informações, consulte o tópico sobre como Trabalhar com o ambiente de visualização do banco de dados.	19 de outubro de 2021
Amazon RDS oferece suporte ao Performance Insights em Regiões da AWS adicionais	O Insights Performance está disponível nas regiões Oriente Médio (Bahrein), África (Cidade do Cabo), Europa (Milão) e Ásia-Pacífico (Osaka). Para ter mais informações, consulte Supported Regions and DB engines for Performance Insights in Amazon RDS.	5 de outubro de 2021
O Performance Insights oferece suporte a estatísticas de nível de resumo para o Oracle	Ao utilizar o Performance Insights, é possível visualizar estatísticas de SQL em nível de instrução e resumo para o Amazon RDS para Oracle. Para ter mais informações, consulte o tópico sobre como Analisar consultas em execução no Oracle.	4 de outubro de 2021

O Amazon RDS on AWS Outposts é compatível com versões adicionais do RDS para PostgreSQL	O RDS on Outposts agora oferece suporte às versões 12.8 e 13.4 do RDS para PostgreSQL. Para ter mais informações, consulte Suporte ao Amazon RDS no AWS Outposts para recursos do Amazon RDS.	1.º de outubro de 2021
O Amazon RDS oferece suporte para a versão 21.1.v1 do Oracle APEX	Você pode usar o APEX 21.1.v1 com todas as versões compatíveis do Oracle Database. Para ter mais informações, consulte Oracle Application Express.	24 de setembro de 2021
O Amazon RDS para Oracle oferece suporte à criptografia do lado do cliente do para a NNE	Ao configurar a NNE, você pode preferir evitar forçar a criptografia do lado do servidor. Por exemplo, talvez você não queira forçar todas as comunicações do cliente a usar criptografia porque o servidor exige isso. Nesse caso, você pode forçar a criptografia no lado do cliente usando as opções do `SQLNET.*CLIENT`. Para ter mais informações, consulte Oracle Native Network Encryption.	24 de setembro de 2021

O Amazon RDS para MySQL e o RDS para PostgreSQL oferecem suporte a novas classes de instância de banco de dados	Agora você pode usar as classes de instâncias db.r5b, db.t4g e db.x2g para criar instâncias de banco de dados do Amazon RDS que estejam executando o MySQL ou PostgreSQL. Para ter mais informações, consulte Mecanismos de banco de dados com suporte a todas as classes de instâncias de banco de dados disponíveis.	15 de setembro de 2021
O Amazon RDS para Microsoft SQL Server é compatível com o Java Database Connectivity (JDBC) com o Microsoft Distributed Transaction Coordinator (MSDTC)	As transações XA do JDBC agora são compatíveis com a versão 14.00.3223.3 e posterior do MSDTC for SQL Server 2017 e o SQL Server 2019. Para ter mais informaçõ es, consulte Suporte para o Microsoft Distributed Transacti on Coordinator no RDS for SQL Server.	7 de setembro de 2021
O Amazon RDS é compatíve l com o MariaDB 10.5.12, 10.4.21, 10.3.31 e 10.2.40	Agora é possível criar instâncias de banco de dados do Amazon RDS executando as versões 10.5.12, 10.4.21, 10.3.31 e 10.2.40 do MariaDB. Para ter mais informações, consulte Versões do MariaDB no Amazon RDS.	2 de setembro de 2021

O Amazon RDS encerrou o suporte ao Oracle Database 18c	É possível criar instâncias de banco de dados somente para o Oracle Database 12c e o Oracle Database 19c. Se você tiver snapshots do Oracle Database 18c, atualize-os para uma versão posterior. Para ter mais informações, consulte Como atualizar para um snapshot de banco de dados Oracle.	17 de agosto de 2021
O Amazon RDS para SQL Server é compatível com atualizações automáticas de versões secundárias	Agora você pode ter suas instâncias de banco de dados do RDS for SQL Server atualizadas automatic amente para a versão secundária mais recente. Para ter mais informaçõ es, consulte Atualização do mecanismo de banco de dados do servidor Microsoft SQL Server.	13 de agosto de 2021

Agora o Amazon RDS para PostgreSQL é compatível com o PostgreSQL versão 14 beta 2 no ambiente de pré-visualização de bancos de dados	Para saber mais sobre o PostgreSQL versão 14 beta 1, consulte as notas de release do PostgreSQL 14 beta 1. Para saber mais sobre o PostgreSQL versão 14 beta 2, consulte as notas de release do PostgreSQL 14 beta 2. Para ter mais informações sobre o Database Preview Environment, consulte Trabalhar com o Database Preview Environment.	9 de agosto de 2021
O Amazon RDS é compatível com o proxy do RDS em uma VPC compartilhada	Agora você pode criar um proxy do RDS em uma VPC compartilhada. Para ter mais informações sobre o RDS Proxy, consulte "Gerenciamento de conexões com o proxy do Amazon RDS" no Guia do usuário do Amazon RDS ou no Guia do usuário do Aurora.	6 de agosto de 2021
O Amazon RDS é compatível com o MariaDB 10.2.39	Agora é possível criar instâncias de banco de dados do Amazon RDS executando a versão 10.2.39 do MariaDB. Para ter mais informações, consulte Versões do MariaDB no Amazon RDS.	4 de agosto de 2021

O Amazon RDS para Oracle inclui a opção `TIMEZONE_FILE_AUTOUPGRADE`	Com essa opção, você pode atualizar o arquivo de fuso horário atual para a versão mais recente em sua instância de banco de dados Oracle. Para ter mais informaçõ es, consulte Atualização automática do arquivo de fuso horário do Oracle.	30 de julho de 2021
O Amazon RDS estende o suporte para backups automatizados entre Regiões	Agora é possível replicar snapshots de banco de dados e logs de transações entre mais Regiões da AWS. Para ter mais informações, consulte Replicating automated backups to anotherAWS Region (Replicar backups automatizados para outra região).	19 de julho de 2021
Suporte para MySQL 5.7.34	Agora é possível criar instâncias de banco de dados do Amazon RDS que executam a versão 5.7.34 do MySQL. Para ter mais informações, consulte Versões do MySQL no Amazon RDS.	8 de julho de 2021
O Amazon RDS on AWS Outposts é compatível com versões adicionais do RDS para PostgreSQL	O RDS on Outposts agora oferece suporte às versões 12.7 e 13.3 do RDS para PostgreSQL. Para ter mais informações, consulte Suporte ao Amazon RDS no AWS Outposts para recursos do Amazon RDS.	8 de julho de 2021

O Amazon RDS para PostgreSQL é compatível com o oracle_fdw	Agora você pode usar a extensão oracle_fdw do PostgreSQL para fornecer um wrapper de dados externo para obter acesso aos bancos de dados Oracle. Para ter mais informações, consulte Accessing external data with the oracle_fdw extension (Acessar dados externos com a extensão oracle_fdw).	8 de julho de 2021
O Amazon RDS é compatível com o Oracle Management Agent (OMA) versão 13.5	Você pode usar o Oracle Management Agent (OMA) versão 13.5 com o Oracle Enterprise Manager (OEM) Cloud Control 13c Release 5 e posteriores. O Amazon RDS para Oracle instala o OMA, que se comunica com o Oracle Management Service (OMS) para fornecer informações sobre o monitoramento. Se você executar o OMS 13.5, poderá gerenciar bancos de dados instalando o OMA 13.5. Para ter mais informações, consulte Oracle Management Agent for Enterprise Manager Cloud Control.	7 de julho de 2021

O Amazon RDS para Oracle é compatível com o download de logs do Amazon S3	Se logs de refazimento arquivados não estiverem em sua instância, mas estiverem protegidos pelo período de retenção de backup, você pode usar `rdsadmin.rdsadmin_archive_log_download` para baixar o Amazon S3. O RDS for Oracle salva os logs no diretório `/rdsdbdata/log/arch` em sua instância de banco de dados. Para ter mais informações, consulte Downloading archived redo logs from Amazon S3 (Como fazer o download de logs de refazimento arquivados do Amazon S3).	2 de julho de 2021
O Amazon RDS é compatível com o MariaDB 10.4.18 e 10.5.9	Agora é possível criar instâncias de banco de dados do Amazon RDS executando as versões 10.4.18 e 10.5.9 do MariaDB. Para ter mais informações, consulte Versões do MariaDB no Amazon RDS.	30 de junho de 2021

O Amazon RDS para Oracle é compatível com fluxos de atividades de banco de dados	Agora você pode monitorar uma instância de banco de dados do Oracle usando o Database Activity Streams. Um banco de dados Oracle grava registros de auditoria para a trilha de auditoria unificada. Quando você inicia um fluxo de atividade de banco de dados em uma instância de banco de dados do Oracle, o Amazon Kinesis transmite todas as atividade s que correspondem às políticas de auditoria do Oracle Database. Para ter mais informações, consulte Monitorar o Amazon RDS com o recurso Database Activity Streams.	23 de junho de 2021
O Amazon RDS para Oracle introduz classes de instâncias otimizadas para memória	As novas classes de instância de banco de dados do Oracle são otimizadas para workloads que necessitem de mais memória, armazenamento e E/S por vCPU. Para ter mais informações, consulte Suporte a classes de instância de banco de dados do RDS for Oracle.	23 de junho de 2021

Suporte para MySQL 8.0.25	Agora, é possível criar instâncias de banco de dados do Amazon RDS que executam a versão 8.0.25 do MySQL. Para ter mais informações, consulte Versões do MySQL no Amazon RDS.	18 de junho de 2021
O Amazon RDS on AWS Outposts é compatível com versões adicionais do RDS para PostgreSQL	O RDS on Outposts agora oferece suporte às versões 12.5, 12.6, 13.1 e 13.2 do RDS para PostgreSQL. Para ter mais informações, consulte Suporte ao Amazon RDS no AWS Outposts para recursos do Amazon RDS.	28 de maio de 2021
O Amazon RDS é compatível com o MariaDB 10.2.37 e 10.3.28	Agora é possível criar instâncias de banco de dados do Amazon RDS executando as versões 10.2.37 e 10.3.28 do MariaDB. Para ter mais informações, consulte Versões do MariaDB no Amazon RDS.	27 de maio de 2021

O Amazon RDS para Oracle é compatível com o banco de dados de contêiner (CDB) multilocatário	Uma arquitetura multilocatário permite que um banco de dados Oracle seja um CDB. No Oracle Database 19c, seu CDB pode incluir um único PDB. A experiência do usuário com um PDB é praticamente idêntica à experiência do usuário com um não CDB. Para ter mais informações, consulte RDS for Oracle architecture (Arquitetura do RDS for Oracle).	25 de maio de 2021
O Amazon RDS on AWS Outposts é compatível com o Amazon RDS para SQL Server	RDS on Outposts agora é compatível com o Amazon RDS para SQL Server. Para ter mais informações, consulte Suporte ao Amazon RDS no AWS Outposts para recursos do Amazon RDS.	11 de maio de 2021
O Amazon RDS estende o suporte para backups automatizados entre Regiões	Agora você pode configurar as instâncias de banco de dados do Amazon RDS que executam o Microsoft SQL Server para replicar snapshots do banco de dados e logs de transações para uma região da AWS diferente. Para ter mais informações, consulte Replicating automated backups to anotherAWS Region (Replicar backups automatizados para outra região).	7 de maio de 2021

O Amazon RDS é compatível com backups automatizados entre Regiões para instâncias de banco de dados criptografadas	Agora você pode replicar os snapshots de bancos de dados e as transações de logs em uma região da AWS para instâncias de banco de dados criptografadas do Amazon RDS que executem o Oracle ou PostgreSQL. Para ter mais informações, consulte Replicating automated backups to anotherAWS Region (Replicar backups automatizados para outra região).	3 de maio de 2021
O Amazon RDS on AWS Outposts é compatível com o monitoramento do Amazon CloudWatch	O RDS on Outposts agora é compatível com o monitoramento do Amazon CloudWatch. Para ter mais informações, consulte Suporte ao Amazon RDS no AWS Outposts para recursos do Amazon RDS.	21 de abril de 2021
O RDS para PostgreSQL é compatível com funções do AWS Lambda	Agora você pode invocar funções do AWS Lambda para suas instâncias de banco de dados do RDS para PostgreSQL. Para ter mais informações, consulte o tópico sobre como Invocar uma função do AWS Lambda de uma instância de banco de dados do RDS para PostgreSQL.	13 de abril de 2021

O RDS for SQL Server é compatível com eventos estendidos	Você pode usar eventos estendidos do SQL Server para capturar informações de depuração e solução de problemas. Para ter mais informações, consulte Como usar eventos estendidos com Amazon RDS para Microsoft SQL Server.	8 de abril de 2021
Suporte para MySQL 8.0.23, 5.7.33 e 5.6.51	Agora é possível criar instâncias de banco de dados do Amazon RDS executando o MySQL versão 8.0.23, 5.7.33 e 5.6.51. Para ter mais informações, consulte Versões do MySQL no Amazon RDS.	31 de março de 2021
Reversão automática em caso de falha na atualização do Amazon RDS para MySQL	Se uma atualização de instância de banco de dados do MySQL versão 5.7 para o MySQL versão 8.0 falhar, o Amazon RDS reverte automaticamente as alterações executadas na atualização. Após a reversão, a instância de banco de dados MySQL executa o MySQL versão 5.7. Para ter mais informações, consulte Reversão após falha ao atualizar do MySQL 5.7 para 8.0.	18 de março de 2021

O Amazon RDS é compatível com réplicas de leitura entre regiões em regiões opcionais	Agora você pode replicar instâncias de banco de dados em regiões opcionais. Para ter mais informações, consulte Criar uma réplica de leitura em uma região da AWS diferente.	18 de março de 2021
O Amazon RDS pretende desativar o Oracle Database 18c	O Oracle Database 18c (18.0.0.0) está em um caminho de depreciação. A Oracle Corporation não fornecerá mais patches para o Oracle Database 18c após a data do fim do suporte. Em 1.º de julho de 2021, o Amazon RDS planeja começar a atualizar automaticamente as instâncias do Oracle 18c Database para o Oracle 19c Database. Antes do início das atualizações automáticas, é altamente recomendável que você atualize manualmente as instâncias existentes do Oracle Database 18c para o Oracle Database 19c. Para ter mais informações, consulte Preparação para a atualização automática do Oracle Database 18c.	11 de março de 2021

O Amazon RDS desativou o suporte ao Oracle Database 11g	Você só pode criar instâncias de banco de dados para o Oracle Database 12c Versão 1 (12.1.0.2) e posterior. Se você tiver snapshots do Oracle Database 11g, atualize-os para uma versão posterior. Para ter mais informações, consulte Como atualizar para um snapshot de banco de dados Oracle.	11 de março de 2021
O Amazon RDS é compatível com backups contínuos de instâncias de banco de dados no AWS Backup	Agora você pode criar backups automatizados no AWS Backup e restaurar instâncias de banco de dados desses backups para um horário especificado. Para ter mais informações, consulte Using AWS Backup to manage automated backups.	10 de março de 2021

O Amazon RDS é compatível com o Oracle Management Agent (OMA) versão 13.4	Você pode usar o Oracle Management Agent (OMA) versão 13.4 com o Oracle Enterprise Manager (OEM) Cloud Control 13c versão 4 atualização 9. O Amazon RDS para Oracle instala o OMA, que se comunica com o Oracle Management Service (OMS) para fornecer informações sobre o monitoramento. Se você executar o OMS 13.4, poderá gerenciar bancos de dados instalando o OMA 13.4. Para ter mais informações, consulte Oracle Management Agent for Enterprise Manager Cloud Control.	10 de março de 2021

Aprimoramentos de endpoint de proxy do RDS	Você pode criar endpoints adicionais associados a cada proxy do RDS . A criação de um endpoint em uma VPC diferente permite o acesso do proxy entre VPCs. Proxies para clusters do Aurora MySQL também podem ter endpoints somente leitura. Esses endpoints de leitor se conectam a instância s de banco de dados de leitores nos clusters e podem melhorar a escalabilidade e a disponibilidade de leitura para aplicações com uso intensivo de consultas. Para ter mais informações sobre o RDS Proxy, consulte "Gerencia r conexões com o proxy do Amazon RDS" no Guia do usuário do Amazon RDS ou no Guia do usuário do Aurora.	8 de março de 2021

O Amazon RDS estende o suporte para backups automatizados entre Regiões	Agora você pode configurar as instâncias de banco de dados do Amazon RDS que executam o PostgreSQL para replicar snapshots do banco de dados e logs de transações para uma região da AWS diferente. Para ter mais informações, consulte Replicating automated backups to anotherAWS Region (Replicar backups automatizados para outra região).	8 de março de 2021
Filtros de replicação para o Amazon RDS para MariaDB e MySQL com suporte nas Regiões China (Pequim) e China (Ningxia)	Agora há suporte para a filtragem de replicação nas regiões China (Pequim) e China (Ningxia). Para ter mais informações, consulte Configuração de filtros de replicação com MariaDB e Configuração de filtros de replicação com MySQL.	5 de março de 2021
O Amazon RDS é compatível com cópia de snapshot de banco de dados entre Regiões em Regiões opcionais	Agora você pode copiar snapshots de banco de dados entre regiões AWS opcionais. Para ter mais informações, consulte o tópico sobre como Copiar snapshots entre regiões da AWS.	4 de março de 2021

O Amazon RDS para SQL Server é compatível com grupos de disponibilidade Always On para a Standard Edition	Quando você cria uma instância de banco de dados usando a configuração multi-AZ no SQL Server 2019 para o mecanismo de banco de dados Standard Edition, o RDS usa automaticamente os grupos de disponibilidade. Para ter mais informações, consulte Implantações multi-AZ para Microsoft SQL Server.	23 de fevereiro de 2021
O Amazon RDS para Oracle introduz procedimentos relacionados ao advisor	O pacote `rdsadmin_util` inclui os procedimentos `advisor_task_set_parameter` , `advisor_task_drop` , e `dbms_stats_init` . Você pode usar esses procedimentos para modificar, interromper e reativar tarefas do advidor, como `AUTO_STATS_ADVISOR_TASK` . Para ter mais informações, consulte Setting parameters for advisor tasks (Configurar parâmetros para tarefas do advisor).	23 de fevereiro de 2021

Amazon RDS fornece motivos de failover para instâncias de banco de dados multi-AZ	Agora você pode ver explicações mais detalhadas quando uma instância de banco de dados multi-AZ executa failover para uma réplica em standby. Para ter mais informações, consulte Processo de failover para Amazon RDS.	18 de fevereiro de 2021
O Amazon RDS amplia o suporte para exportação de snapshots para o Amazon S3	Agora você pode exportar dados de snapshot de banco de dados para Amazon S3 no China. Para ter mais informações, consulte Exportar dados de snapshot de banco de dados para o Amazon S3.	17 de fevereiro de 2021
Filtros de replicação do Amazon RDS para MariaDB e MySQL	Você pode configurar filtros de replicação para instâncias MySQL e MariaDB. Filtros de replicação especificam quais bancos de dados e tabelas são replicados em uma réplica de leitura. Você pode criar listas de bancos de dados e tabelas para inclusão ou exclusão em cada réplica. Para ter mais informações, consulte Configuração de filtros de replicação com MariaDB e Configuração de filtros de replicação com MySQL.	12 de fevereiro de 2021

O RDS for Oracle é compatível com o ambiente de pré-visualização de banco de dados2v1	Você pode usar o ambiente de pré-visualização de banco de dadosv1 com todas as versões compatíveis do Oracle Database. Para ter mais informações, consulte Oracle Application Express.	2 de fevereiro de 2021
O Amazon RDS para SQL Server é compatível com o armazenamento de instância local para o banco de dados tempdb	Agora você pode executar o Amazon RDS para SQL Server nos tipos de instância Amazon EC2 db.r5d e db.m5d com o banco de dados tempdb configurado para usar um armazenamento de instâncias. Ao colocar arquivos de dados temporários e arquivos de log localmente, você pode obter latências de leitura e gravação mais baixas em comparação com as ofertas padrão baseadas em Amazon EBS. Para ter mais informaçõ es, consulte Suporte ao armazenamento de instância s para o banco de dados tempdb no Amazon RDS para SQL Server.	27 de janeiro de 2021

O Amazon RDS para PostgreSQL é compatível com pg_partman e pg_cron	Agora o Amazon RDS para PostgreSQL é compatível com as extensões pg_partman e pg_cron. Para ter mais informações sobre a extensão pg_partman, consulte Gerenciar partições PostgreSQL com a extensão pg_partman. Para ter mais informações sobre a extensão pg_cron, consulte Agendar manutenção com a extensão pg_cron PostgreSQL.	12 de janeiro de 2021
O Amazon RDS é compatível com a publicação do log do Oracle Management Agent no Amazon CloudWatch Logs	O log do Oracle Management Agent consiste em emctl.log, emdctlj.log, gcagent.log, gcagent_errors.log, emagent.nohup e secure.log. O Amazon RDS publica cada um desses logs como um stream de log separado do CloudWatch. Para ter mais informações, consulte Publicação de logs do Oracle no Amazon CloudWatch Logs.	28 de dezembro de 2020
O Amazon RDS on AWS Outposts é compatível com versões adicionais de banco de dados	O RDS no Outposts agora oferece suporte a versões adicionais do MySQL e do PostgreSQL. Para ter mais informações, consulte Suporte ao Amazon RDS no AWS Outposts para recursos do Amazon RDS.	23 de dezembro de 2020

O Amazon RDS on AWS Outposts é compatível com CoIPs	O RDS on Outposts agora agora é compatível com endereços IP de propriedade do cliente (CoIPs). Os CoIPs fornecem conectividade local ou externa aos recursos em suas sub-redes Outpost por meio de sua rede local. Para ter mais informações, consulte Endereços IP de propriedade do cliente para RDS no Outposts.	22 de dezembro de 2020
O Amazon RDS para Oracle planeja a atualização de instâncias 11g BYOL para 19c	Em 4 de janeiro de 2021, planejamos começar a atualizar automaticamente todas as edições de instâncias do Oracle Database 11g no modelo Bring Your Own License (BYOL) para o Oracle Database 19c. Todas as instâncias do Oracle Database 11g, incluindo as instâncias reservadas, serão movidas para a Release Update (RU – Atualização de versão) mais recente disponível. Para ter mais informações, consulte Preparação para a atualização automática do Oracle Database 11g BYOL.	11 de dezembro de 2020

O Amazon RDS é compatível com a replicação de backups automatizados para outra região da AWS	Agora você pode configurar as instâncias de banco de dados do Amazon RDS para replicar snapshots e logs de transações para uma região da AWS de destino que você escolher. Para ter mais informações, consulte Replicating automated backups to anotherAWS Region (Replicar backups automatizados para outra região).	4 de dezembro de 2020
O Amazon RDS para Oracle e o Microsoft SQL Server são compatíveis com uma nova classe de instância de banco de dados	Agora você pode usar a classe de instância db.r5b para criar instâncias de banco de dados do Amazon RDS executando Oracle ou SQL Server. Para ter mais informações, consulte Mecanismos de banco de dados com suporte a todas as classes de instâncias de banco de dados disponíveis.	4 de dezembro de 2020
Suporte ao MariaDB 10.2.32	Agora é possível criar instâncias de banco de dados do Amazon RDS que executam a versão 10.2.32 do MariaDB. Para ter mais informações, consulte Versões do MariaDB no Amazon RDS.	25 de novembro de 2020

O Amazon RDS para SQL Server é compatível com o Microsoft Business Intelligence Suite no SQL Server 2019	Agora você pode executar o SQL Server Analysis Services, o SQL Server Integration Services e o SQL Server Reporting Services em instâncias de banco de dados usando a versão principal mais recente. Para ter mais informações, consulte Opções para o mecanismo de banco de dados do Microsoft SQL Server.	24 de novembro de 2020
Amazon RDS para PostgreSQL versão 13 no ambiente de pré-visualização de banco de dados	O Amazon RDS para PostgreSQL agora é compatível com o PostgreSQL versão 13 no ambiente de visualização de banco de dados. Para ter mais informações, consulte Versões 13 do PostgreSQL.	24 de novembro de 2020

O Amazon RDS Performance Insights introduz novas dimensões	Você pode agrupar a carga do banco de dados de acordo com os grupos de dimensões para banco de dados (PostgreSQL, MySQL e MariaDB), aplicação (PostgreSQL) e tipo de sessão (PostgreSQL). O Amazon RDS também é compatível com as dimensões db.name (PostgreSQL, MySQL e MariaDB), db.application.name (PostgreSQL) e db.session_type.name (PostgreSQL). Para ter mais informações, consulte Tabela de carga superior.	24 de novembro de 2020
O Amazon RDS para MariaDB é compatível com uma nova versão principal	Agora é possível criar instâncias de banco de dados do Amazon RDS que executam a versão 10.5 do MariaDB. Para ter mais informações, consulte Versões do MariaDB no Amazon RDS.	23 de novembro de 2020
Suporte para MySQL 5.6.49	Agora, é possível criar instâncias de banco de dados do Amazon RDS que executam a versão 5.6.49 do MySQL. Para ter mais informações, consulte Versões do MySQL no Amazon RDS.	20 de novembro de 2020

Suporte para MySQL 5.5.62	Agora, é possível criar instâncias de banco de dados do Amazon RDS que executam a versão 5.5.62 do MySQL. Para ter mais informações, consulte Versões do MySQL no Amazon RDS.	20 de novembro de 2020
O Performance Insights é compatível com a análise de estatísticas de consultas PostgreSQL em execução	Agora, você pode analisar estatísticas de consultas em execução com o Performance Insights para instâncias de banco de dados PostgreSQL. Para ter mais informações, consulte Estatísticas para PostgreSQL.	18 de novembro de 2020
O Amazon RDS estende o suporte para autoescalabilidade de armazenamento	Agora você pode habilitar o autoscaling do armazenamento ao criar uma réplica de leitura, restaurar uma instância de banco de dados em um horário especificado ou restaurar uma instância de banco de dados MySQL a partir de um backup do Amazon S3. Para ter mais informações, consulte Gerenciar a capacidade automaticamente com o dimensionamento automático de armazenamento do Amazon RDS.	18 de novembro de 2020

O Amazon RDS para SQL Server é compatível com o Database Mail	Com o Database Mail, você pode enviar mensagens de e-mail de sua instância de banco de dados do Amazon RDS para SQL Server. Depois de especificar os destinatários do e-mail, você poderá adicionar arquivos ou resultados da consulta à mensagem enviada. Para ter mais informações, consulte Usar o Database Mail no Amazon RDS para SQL Server.	4 de novembro de 2020
Suporte para MySQL 8.0.21	Agora, é possível criar instâncias de banco de dados do Amazon RDS que executam a versão 8.0.21 do MySQL. Para ter mais informações, consulte Versões do MySQL no Amazon RDS.	22 de outubro de 2020
O Amazon RDS amplia o suporte para exportação de snapshots para o Amazon S3	Agora, é possível exportar dados de snapshots do banco de dados para o Amazon S3 em todas as regiões comerciais da AWS. Para ter mais informações, consulte Exportar dados de snapshot de banco de dados para o Amazon S3.	22 de outubro de 2020

O Amazon RDS para PostgreSQL é compatível com atualizações de réplica de leitura	Com o Amazon RDS para PostgreSQL, ao fazer uma atualização de versão principal da instância de banco de dados primária, as réplicas de leitura também são atualizadas automaticamente. Para ter mais informações, consulte Atualizar versões do mecanismo de banco de dados PostgreSQL.	15 de outubro de 2020
O Amazon RDS para MariaDB, MySQL e PostgreSQL são compatíveis com as classes de instância de banco de dados Graviton2	Agora é possível usar as classes de instância de banco de dados Graviton2 db.m6g.x e db.r6g.x para criar instâncias de banco de dados executando o Amazon RDS MariaDB, MySQL ou PostgreSQL. Para ter mais informações, consulte Mecanismos de banco de dados compatíveis para classes de instância de banco de dados.	15 de outubro de 2020
O Amazon RDS para SQL Server é compatível com atualizações para o SQL Server 2019	Você pode atualizar suas instâncias de banco de dados do SQL Server para o SQL Server 2019. Para ter mais informações, consulte Atualização do mecanismo de banco de dados do servidor Microsoft SQL Server.	6 de outubro de 2020

O Amazon RDS para Oracle é compatível com a especificação do conjunto de caracteres nacionais	O conjunto de caracteres nacionais, também chamado de conjunto de caracteres NCHAR, é usado nos tipos de dados NCHAR, NVARCHAR2 e NCL0B. Ao criar um banco de dados, você pode especificar AL16UTF16 (padrão) ou UTF8 como o conjunto de caracteres NCHAR. Para ter mais informações, consulte Conjuntos de caracteres Oracle compatíveis no Amazon RDS.	2 de outubro de 2020
Suporte para MySQL 5.7.31	Agora é possível criar instâncias de banco de dados do Amazon RDS que executam a versão 5.7.31 do MySQL. Para ter mais informações, consulte Versões do MySQL no Amazon RDS.	1º de outubro de 2020
O Amazon RDS para PostgreSQL é compatível com a exportação de dados para o Amazon S3	É possível consultar dados de uma instância de banco de dados PostgreSQL e exportá-los diretamente para arquivos armazenados em um bucket do Amazon S3. Para ter mais informações, consulte Exportar dados de uma instância de banco de dados do RDS para PostgreSQL para o Amazon S3.	24 de setembro de 2020

O Amazon RDS para MySQL 8.0 é compatível com o Percona XtraBackup	Agora você pode usar o Percona XtraBackup para restaurar um backup em uma instância de banco de dados do Amazon RDS para MySQL 8.0. Para ter mais informaçõ es, consulte Restaurar um backup em uma instância de banco de dados MySQL do Amazon RDS.	17 de setembro de 2020
O Amazon RDS para SQL Server é compatível com o backup e restauração nativos em instâncias de banco de dados com réplicas de leitura	Você pode restaurar um backup nativo do SQL Server em uma instância de banco de dados que tenha réplicas de leitura configuradas. Para ter mais informações, consulte Importação e exportação de bancos de dados do SQL Server.	16 de setembro de 2020
O Amazon RDS para SQL Server é compatível com fusos horários adicionais	É possível combinar o fuso horário da instância de banco de dados com o fuso horário escolhido. Para ter mais informações, consulte Fuso horário local para instância s de banco de dados do Microsoft SQL Server.	11 de setembro de 2020

Versão 13 beta 3 do Amazon RDS para PostgreSQL no ambiente de pré-visualização de banco de dados	Agora o Amazon RDS para PostgreSQL é compatível com o PostgreSQL versão 13 beta 3 no ambiente de pré-visualização de banco de dados Para ter mais informaçõ es, consulte Versões 13 do PostgreSQL.	9 de setembro de 2020
O Amazon RDS para SQL Server é compatível com o sinalizador de rastreamento 692	Agora, é possível usar o sinalizador de rastreamento 692 como um parâmetro de inicialização usando grupos de parâmetros de banco de dados. Habilitar esse sinalizad or de rastreamento desabilit ará as inserções rápidas durante o carregamento em massa de dados no heap ou índices em cluster. Para ter mais informações, consulte Desabilitar inserções rápidas durante o carregamento em massa.	27 de agosto de 2020
O Amazon RDS para SQL Server é compatível com o Microsoft SQL Server 2019	Agora, é possível criar instâncias de banco de dados do RDS que usam o SQL Server 2019. Para ter mais informações, consulte Versões do Microsoft SQL Server no Amazon RDS.	26 de agosto de 2020

O RDS for Oracle é compatível com o banco de dados de réplica montado	Ao criar ou modificar uma réplica do Oracle, você pode colocá-la no modo montado. Como o banco de dados de réplica não aceita conexões de usuário, ele não pode atender a uma workload somente leitura. A réplica montada exclui os arquivos de log redo arquivados depois de aplicá-los. O principal uso para réplicas montadas é a recuperação de desastres entre regiões. Para ter mais informações, consulte Visão geral das réplicas do Oracle.	13 de agosto de 2020
O RDS for Oracle planeja a atualização de instâncias 11g SE1 LI	Em 1º de novembro de 2020, planejamos começar a atualizar automaticamente as instâncias do Oracle Database 11g SE1 Licença inclusa (LI) para o Oracle 19c Database para o Amazon RDS para Oracle. Todas as instâncias do 11g, incluindo as instâncias reservadas, serão movidas para o atualização da versão do Oracle mais recente disponível. Para ter mais informações, consulte Preparação para a atualização automática do Oracle Database 11g SE1.	31 de julho de 2020

O Amazon RDS é compatível com novas classes de instância de banco de dados Graviton2 na versão de pré-visualização para PostgreSQL e MySQL	Agora é possível criar instâncias de banco de dados do Amazon RDS que estejam executando PostgreSQL ou MySQL que usam as classes de instância de banco de dados db.m6g.x e db.r6g.x. Para ter mais informações, consulte Mecanismos de banco de dados com suporte a todas as classes de instâncias de banco de dados disponíveis.	30 de julho de 2020
O RDS for Oracle é compatível com o APEX 20.1v1	Você pode usar o ambiente de pré-visualização de banco de dados1v1 com todas as versões compatíveis do Oracle Database. Para ter mais informações, consulte Oracle Application Express.	28 de julho de 2020
Suporte para MySQL 8.0.20	Agora, é possível criar instâncias de banco de dados do Amazon RDS que executam a versão 8.0.20 do MySQL. Para ter mais informações, consulte Versões do MySQL no Amazon RDS.	23 de julho de 2020

O Amazon RDS para MariaDB e MySQL são compatíveis com novas classes de instância de banco de dados	Agora, é possível criar instâncias de banco de dados do Amazon RDS executando o MariaDB e o MySQL que usem as classes de instância de banco de dados db.m5.16xlarge, db.m5.8xlarge, db.r5.16xlarge e db.r5.8xlarge. Para ter mais informações, consulte Mecanismos de banco de dados com suporte a todas as classes de instâncias de banco de dados disponíveis.	23 de julho de 2020
O RDS for SQL Server é compatível com a desativação de versões antigas do TLS e de cifras	É possível ativar e desativar determinadas cifras e protocolos de segurança. Para ter mais informações, consulte Configurar cifras e protocolos segurança.	21 de julho de 2020
O RDS é compatível com o Oracle Spatial no SE2	É possível usar o Oracle Spatial no Standard Edition 2 (SE2) para todas as versões de 12.2, 18c e 19c. Para ter mais informações, consulte Oracle Spatial.	9 de julho de 2020

O Amazon RDS é compatível com AWS PrivateLink	O Amazon RDS agora oferece suporte à criação de endpoints da Amazon VPC para chamadas de API do Amazon RDS a fim de manter o tráfego entre aplicações e o Amazon RDS na rede da AWS. Para ter mais informaçõ es, consulte o tópico sobre o Amazon RDS e endpoints de VPC de interface (AWS PrivateLink).	9 de julho de 2020
Acabou o suporte às versões 9.4.x do Amazon RDS para PostgreSQL.	O Amazon RDS para PostgreSQL não oferece mais suporte às versões 9.4.x. Para obter versões com suporte, consulte Versões de banco de dados PostgreSQL com suporte.	8 de julho de 2020
Suporte para MariaDB 10.3.23 e 10.4.13	Agora é possível criar instâncias de banco de dados do Amazon RDS executand o o MariaDB versão 10.3.23 e 10.4.13. Para ter mais informações, consulte Versões do MariaDB no Amazon RDS.	6 de julho de 2020
Amazon RDS em AWS Outposts	Você pode criar instâncias de banco de dados do Amazon RDS on AWS Outposts. Para ter mais informações, consulte Trabalhar com o Amazon RDS no AWS Outposts.	6 de julho de 2020

Amazon RDS para Oracle cria arquivos de inventário automaticamente	Para abrir solicitações de serviço para clientes de BYOL, o Oracle Support solicita arquivos de inventário gerados pelo Opatch. O Amazon RDS para Oracle cria arquivos de inventário automaticamente de hora em hora do diretório BDUMP. Para ter mais informações, consulte Acessar arquivos do Opatch.	6 de julho de 2020
Suporte para MySQL 5.7.30 e 5.6.48	Agora é possível criar instâncias de banco de dados do Amazon RDS executando o MySQL versão 5.7.30 e 5.6.48. Para ter mais informações, consulte Versões do MySQL no Amazon RDS.	25 de junho de 2020

O Amazon RDS para Oracle é compatível com ADRCI	O utilitário ADRCI (Automatic Diagnostic Repository Command Interpreter) é uma ferramenta da linha de comando Oracle usada para gerenciar dados de diagnóstico. Usando as funções no pacote do Amazon RDS `rdsadmin_adrci_util` , é possível listar e empacotar problemas e incidentes, além de mostrar arquivos de rastreamento. Para ter mais informações, consulte Tarefas de diagnóstico de DBA comuns para instâncias de banco de dados Oracle.	17 de junho de 2020
Suporte para MySQL 8.0.19	Agora é possível criar instâncias de banco de dados do Amazon RDS que executam a versão 8.0.19 do MySQL. Para ter mais informações, consulte Versões do MySQL no Amazon RDS.	2 de junho de 2020

O MySQL 8.0 é compatível com nomes de tabelas em letras minúsculas	Agora é possível definir o parâmetro `lower_case_table_names` como 1 para instâncias de banco de dados do Amazon RDS que executam o MySQL versão 8.0.19 e versões 8.0 posteriores. Para ter mais informações, consulte, Exceções de parâmetros do MySQL para instâncias de bancos de dados do Amazon RDS.	2 de junho de 2020
O Amazon RDS para Microsoft SQL Server é compatível com o SQL Server Integration Services (SSIS)	O SSIS é uma plataforma para integração de dados e aplicativos de fluxo de trabalho. É possível habilitar o SSIS em instâncias de banco de dados novas ou existentes. Ele é instalado na mesma instância de banco de dados que o mecanismo de banco de dados. Para ter mais informações, consulte Suporte para o SQL Server Integration Services no SQL Server.	19 de maio de 2020

O Amazon RDS para Microsoft SQL Server é compatível com o SQL Server Reporting Services (SSRS)	O SSRS é um aplicativo baseado em servidor usado para geração e distribuição de relatórios. É possível habilitar o SSRS em instâncias de banco de dados novas ou existentes. Ele é instalado na mesma instância de banco de dados que o mecanismo de banco de dados. Para ter mais informações, consulte Suporte para o SQL Server Reporting Services no SQL Server.	15 de maio de 2020
O Amazon RDS para Microsoft SQL Server é compatível com a integração do S3 em instâncias multi-AZ	Agora é possível usar o Amazon S3 com recursos do SQL Server, como inserção em massa em instâncias de banco de dados multi-AZ. Para ter mais informaçõ es, consulteIntegrar uma instância de banco de dados do Amazon RDS para SQL Server com o Amazon S3.	15 de maio de 2020
O Amazon RDS para Oracle é compatível com a limpeza da lixeira	O procedimento `rdsadmin.rdsadmin_util.purge_dba_recyclebin` limpa a lixeira. Para ter mais informações, consulte Limpar a lixeira.	13 de maio de 2020

O Amazon RDS para Oracle melhora a capacidade de gerenciamento do AWR (Automatic Workload Repository)	Os procedimentos `rdsadmin.rdsadmin_diagnostic_util` geram relatórios do AWR e extraem dados do AWR em arquivos de despejo. Para ter mais informações, consulte Gerar relatórios de performance com o AWR (Automatic Workload Repository).	13 de maio de 2020
O Amazon RDS para Microsoft SQL Server é compatível com o MSDTC (Microsoft Distributed Transaction Coordinator)	O Amazon RDS para SQL Server oferece suporte a transações distribuídas entre hosts. Para ter mais informações, consulte Suporte para o Microsoft Distributed Transaction Coordinator no SQL Server.	4 de maio de 2020
O Amazon RDS para Microsoft SQL Server é compatível com novas versões	Agora é possível criar instâncias de banco de dados do Amazon RDS executando versões do SQL Server 2017 CU19 14.00.3281.6, 2016 SP2 CU11 13.00.5598.27, 2014 SP3 CU4 12.00.6329.1 e 2012 SP4 GDR 11.0.7493.4 para todas as edições. Para ter mais informações, consulte Versões do Microsoft SQL Server no Amazon RDS.	28 de abril de 2020

O Amazon RDS está disponível na Região da Europa (Milão)	O Amazon RDS e posterior na Região da Europa (Milão). Para ter mais informações, consulte Regiões e zonas de disponibilidade.	28 de abril de 2020
Suporte do Amazon RDS a zonas locais	Agora é possível executar instâncias de banco de dados em uma sub-rede de zona local. Para ter mais informaçõ es, consulte Regiões, zonas de disponibilidade e zonas locais.	23 de abril de 2020
O Amazon RDS está disponível na Região da África (Cidade do Cabo)	O Amazon RDS e posterior na Região da África (Cidade do Cabo). Para ter mais informações, consulte Regiões e zonas de disponibilidade.	22 de abril de 2020
O Amazon RDS para Microsoft SQL Server é compatível com o SQL Server Analysis Services (SSAS)	O SSAS é um processamento analítico on-line (OLAP) e uma ferramenta de mineração de dados que está instalado no SQL Server. É possível habilitar o SSAS em instância s de banco de dados novas ou existentes. Ele é instalado na mesma instância de banco de dados que o mecanismo de banco de dados. Para ter mais informações, consulte Suporte para o SQL Server Analysis Services no SQL Server.	17 de abril de 2020

Proxy do Amazon RDS para PostgreSQL	Agora o proxy do Amazon RDS está disponível para o PostgreSQL. É possível usar o proxy do RDS para reduzir a sobrecarga do gerenciamento de conexões na instância de banco de dados e também a chance de erros de "muitas conexões". No momento, o proxy do RDS está em visualização pública para o PostgreSQL. Para ter mais informações, consulte Gerenciar conexões com o Amazon RDS Proxy (visualização) .	8 de abril de 2020
O Amazon RDS para Oracle é compatível com o Oracle APEX versão 19.2.v1	Agora o Amazon RDS para Oracle agora é compatível com o Oracle Application Express (APEX) versão 19.2.v1. Para ter mais informações, consulte Oracle Application Express.	8 de abril de 2020
O Amazon RDS para MariaDB é compatível com uma nova versão principal	Agora é possível criar instâncias de banco de dados do Amazon RDS que executam a versão 10.4 do MariaDB. Para ter mais informações, consulte Versões do MariaDB no Amazon RDS.	6 de abril de 2020

O Amazon RDS Performance Insights está disponível para o Amazon RDS para MariaDB 10.4	O Amazon RDS Performance Insights e posterior para o Amazon RDS para MariaDB versão 10.4. Para ter mais informações, consulte Uso de Insights de Performance do Amazon RDS.	6 de abril de 2020
Fim do suporte às versões 9.3.x do Amazon RDS para PostgreSQL	O Amazon RDS para PostgreSQL não oferece mais suporte às versões 9.3.x. Para obter versões com suporte, consulte Versões de banco de dados PostgreSQL com suporte.	3 de abril de 2020
O Amazon RDS para Microsoft SQL Server é compatível com réplicas de leitura	Agora é possível criar réplicas de leitura para instâncias de banco de dados do SQL Server. Para ter mais informações, consulte Trabalhar com réplicas de leitura.	3 de abril de 2020
O Amazon RDS para Microsoft SQL Server é compatível com backups de diversos arquivos	Agora é possível fazer backup de bancos de dados para diversos arquivos usando backup e restauração nativos do SQL Server. Para ter mais informações, consulte Fazer backup de um banco de dados.	2 de abril de 2020

Integração do Amazon RDS para Oracle com o AWS License Manager	O Amazon RDS para Oracle agora está integrado ao AWS License Manager. Se você usar o modelo Bring-Your-Own-License, a integração do AWS License Manager facilitará o monitoramento do uso da licença Oracle em sua organização. Para ter mais informações, consulte Integração ao AWS License Manager.	23 de março de 2020
Suporte para 64 TiB em instâncias db.r5 no Amazon RDS para MariaDB e MySQL	Agora é possível criar instâncias de banco de dados do Amazon RDS para MariaDB e MySQL que usam a classe de instância de banco de dados db.r5 com até 64 TiB de armazenamento. Para ter mais informações, consulte Fatores que afetam a performance do armazenamento.	18 de março de 2020
Suporte para MySQL 8.0.17	Agora é possível criar instâncias de banco de dados do Amazon RDS que executam a versão 8.0.17 do MySQL. Para ter mais informações, consulte Versões do MySQL no Amazon RDS.	10 de março de 2020

O Amazon RDS Performance Insights está disponível para o Amazon RDS para MySQL 8.0	O Insights de Performance do Amazon RDS e posterior para o Amazon RDS para MySQL versão 8.0.17 e versões 8.0 posteriores. Para ter mais informações, consulte Uso de Insights de Performance do Amazon RDS.	10 de março de 2020
Suporte para MySQL 5.6.46	Agora é possível criar instâncias de banco de dados do Amazon RDS que executam a versão 5.6.46 do MySQL. Para ter mais informações, consulte Versões do MySQL no Amazon RDS.	28 de fevereiro de 2020
O Amazon RDS Performance Insights está disponível para o Amazon RDS para MariaDB 10.3	O Amazon RDS Performance Insights e posterior para o Amazon RDS para MariaDB versão 10.3.13 e versões 10.3 posteriores. Para ter mais informações, consulte Uso de Insights de Performance do Amazon RDS.	26 de fevereiro de 2020
Suporte para MySQL 5.7.28	Agora é possível criar instâncias de banco de dados do Amazon RDS que executam a versão 5.7.28 do MySQL. Para ter mais informações, consulte Versões do MySQL no Amazon RDS.	20 de fevereiro de 2020

Suporte ao MariaDB 10.3.20	Agora é possível criar instâncias de banco de dados do Amazon RDS que executam a versão 10.3.20 do MariaDB. Para ter mais informações, consulte Versões do MariaDB no Amazon RDS.	20 de fevereiro de 2020
O Amazon RDS para Microsoft SQL Server é compatível com uma nova classe de instância de banco de dados	Agora é possível criar instâncias de bancos de dados do Amazon RDS executando o SQL Server com a classe de instância de banco de dados db.z1d. Para ter mais informações, consulte Suporte a classes de instância de banco de dados para Microsoft SQL Server.	19 de fevereiro de 2020
Suporte para domínios do Diretório Ativo entre contas e entre VPCs no Amazon RDS para SQL Server	O Amazon RDS para Microsoft SQL Server agora oferece suporte à associação de instâncias de banco de dados com domínios do Active Directory pertencentes a diferentes contas e VPCs. Para ter mais informações, consulte Usar a autenticação do Windows com uma instância de banco de dados do Microsoft SQL Server.	13 de fevereiro de 2020

Opção Oracle OLAP	O Amazon RDS para Oracle agora oferece suporte à opção On-line Analytical Processing (OLAP) para instâncias de banco de dados Oracle. É possível usar o Oracle OLAP para analisar grandes quantidades de dados criando cubos e objetos dimensionais de acordo com o padrão OLAP. Para ter mais informações, consulte Oracle OLAP.	13 de fevereiro de 2020
Suporte ao FIPS 140-2 do Oracle	O Amazon RDS para Oracle oferece suporte à publicação do Padrão federal de processamento de informações 140-2 (FIPS 140-2) para conexões SSL/TLS. Para ter mais informações, consulte Suporte ao FIPS.	11 de fevereiro de 2020
O Amazon RDS para PostgreSQL é compatível com novas classes de instância de banco de dados	Agora é possível criar instâncias de banco de dados do Amazon RDS executando o PostgreSQL que usem as classes de instância de banco de dados db.m5.16xlarge, db.m5.8xlarge, db.r5.16xlarge e db.r5.8xlarge. Para ter mais informações, consulte Mecanismos de banco de dados com suporte a todas as classes de instâncias de banco de dados disponíveis.	11 de fevereiro de 2020

O Performance Insights é compatível com a análise de estatísticas de consultas do MariaDB e do MySQL	Agora, você pode analisar estatísticas de consultas em execução com o Performance Insights para instâncias de banco de dados MariaDB e MySQL. Para ter mais informações, consulte Analisar estatísticas de consultas em execução.	4 de fevereiro de 2020
Suporte à exportação de dados de snapshot de banco de dados para o Amazon S3 para MariaDB, MySQL e PostgreSQL	O Amazon RDS é compatível com a exportação de dados de snapshot de banco de dados para o Amazon S3 para MariaDB, MySQL e PostgreSQL. Para ter mais informações, consulte Exportar dados de snapshot de banco de dados para o Amazon S3.	23 de janeiro de 2020
O Amazon RDS para MySQL é compatível com a autenticação Kerberos	Agora é possível usar a autenticação Kerberos para autenticar usuários quando se conectam a suas instâncias de banco de dados do Amazon RDS para MySQL. Para ter mais informações, consulte Usar a autenticação Kerberos para MySQL.	21 de janeiro de 2020

O Amazon RDS Performance Insights é compatível com a visualização de mais textos em SQL do Amazon RDS para Microsoft SQL Server	Agora o Insights de Performance do Amazon RDS é compatível com a visualização de mais textos em SQL no painel do Insights de Performance para instâncias de banco de dados do Amazon RDS para Microsoft SQL Server. Para ter mais informações, consulte Visualizar mais textos em SQL no painel do Performance Insights.	17 de dezembro de 2019

Proxy do Amazon RDS	Você pode reduzir a sobrecarga do gerenciamento de conexões em seu cluster e reduzir a possibilidade de erros de “conexões em excesso” usando o proxy do Amazon RDS. Associe cada proxy a uma instância de banco de dados do RDS ou cluster de bancos de dados Aurora. Então, você usa o endpoint do proxy na string de conexão de seu aplicativo. O Proxy do Amazon RDS, atualmente, está em um estado de demonstração pública. Ele é compatível com o mecanismo de banco de dados do RDS para MySQL. Para ter mais informações, consulte Gerenciar conexões com o Amazon RDS Proxy (visualização) .	3 de dezembro de 2019
Amazon RDS on AWS Outposts (pré-visualização)	Com o Amazon RDS on AWS Outposts, você pode criar bancos de dados relacionais gerenciados pela AWS em seus datacenters locais. O RDS on Outposts permite que você execute bancos de dados do RDS on AWS Outposts. Para ter mais informações, consulte Amazon RDS on AWS Outposts (pré-visualização).	3 de dezembro de 2019

O Amazon RDS para Oracle é compatível com réplicas de leitura entre regiões	O Amazon RDS para Oracle agora é compatível com réplicas de leitura entre regiões com o Active Data Guard. Para ter mais informações, consulte Trabalhar com réplicas de leitura e Trabalhar com réplicas de leitura da Oracle.	26 de novembro de 2019
O Performance Insights é compatível com a análise de estatísticas de consultas Oracle em execução	Agora, você pode analisar estatísticas de consultas em execução com o Performance Insights para instâncias de banco de dados Oracle. Para ter mais informações, consulte Analisar estatísticas de consultas em execução.	25 de novembro de 2019
O Amazon RDS para Microsoft SQL Server é compatível com a publicação de logs no CloudWatch Logs	Você pode configurar sua instância de banco de dados do Amazon RDS para SQL Server para publicar logs de eventos diretamente no Amazon CloudWatch Logs. Para ter mais informações, consulte Publicar logs do SQL Server no Amazon CloudWatch Logs.	25 de novembro de 2019

O Amazon RDS para Microsoft SQL Server é compatível com novas classes de instâncias de banco de dados	Agora, você pode criar instâncias de bancos de dados do Amazon RDS executando o SQL Server usando as classes de instâncias de banco de dados db.x1e e db.x1. Para ter mais informações, consulte Suporte a classes de instância de banco de dados para Microsoft SQL Server.	25 de novembro de 2019
O Amazon RDS para Microsoft SQL Server é compatível com restaurações de log e diferenciais	Você pode restaurar logs e backups diferenciais usando o processo nativo de backup e restauração do SQL Server. Para ter mais informações, consulte Usar o backup e a restauração nativos.	25 de novembro de 2019
multi-AZ com suporte no Amazon RDS para Microsoft SQL Server em novas regiões	A multi-AZ no SQL Server já está disponível nas regiõesChina, Oriente Médio (Bahrein) e Europa (Estocolmo). Para ter mais informações, consulte Implantações multi-AZ para Microsoft SQL Server.	22 de novembro de 2019

O Amazon RDS para Microsoft SQL Server agora oferece suporte para inserção em massa e integração com o S3	Você pode transferir arquivos entre uma instância de banco de dados do SQL Server e um bucket do Amazon S3. Dessa maneira, você pode usar recursos do Amazon S3 com o SQL Server com a inserção em massa. Para ter mais informações, consulteIntegrar uma instância de banco de dados do Amazon RDS para SQL Server com o Amazon S3.	21 de novembro de 2019
Contadores do Performance Insights para o Amazon RDS para Microsoft SQL Server	Agora, você pode adicionar contadores de performance aos gráficos do Performance Insights para instâncias de banco de dados do Microsoft SQL. Para ter mais informaçõ es, consulte Contadores do Performance Insights para o Amazon RDS para Microsoft SQL Server.	12 de novembro de 2019

O Amazon RDS para Microsoft SQL Server é compatível com novos tamanhos de classes de instâncias de banco de dados	Agora é possível criar instâncias de banco de dados do Amazon RDS executando o SQL Server que usam tamanhos de instâncias 8xlarge e 16xlarge para as classes de instâncias db.m5 e db.r5. Tamanhos de instância que variam de pequenos a 2xlarge agora estão disponíveis para a classe de instância db.t3. Para ter mais informações, consulte Suporte a classes de instância de banco de dados para Microsoft SQL Server.	11 de novembro de 2019
Suporte para atualizações de snapshot do PostgreSQL	Se você tiver snapshots de banco de dados manuais existentes de instâncias de banco de dados PostgreSQL do Amazon RDS, agora poderá atualizá-las para uma versão posterior do mecanismo de banco de dados PostgreSQL. Para ter mais informações, consulte Atualizar um snapshot de banco de dados PostgreSQL.	7 de novembro de 2019

O Amazon RDS para Oracle é compatível com uma nova versão principal	Agora é possível criar instâncias de banco de dados do Amazon RDS executando o Oracle Database 19c (19.0). Para ter mais informações, consulte Oracle Database 19c com o Amazon RDS.	7 de novembro de 2019
Amazon RDS para PostgreSQL versão 12.0 no ambiente de pré-visualização de banco de dados	O Amazon RDS para PostgreSQL agora é compatível com o PostgreSQL versão 12.0 no Ambiente de prévia de banco de dados. Para ter mais informações, consulte PostgreSQL versão 12.0 no ambiente de pré-visualização de banco de dados.	1º de novembro de 2019
O Amazon RDS para PostgreSQL é compatível com a autenticação Kerberos	Agora é possível usar a autenticação Kerberos para autenticar usuários quando se conectam à instância de banco de dados do Amazon RDS executando o PostgreSQL. Para ter mais informações, consulte Como usar a autenticação Kerberos com o Amazon RDS para PostgreSQL.	28 de outubro de 2019

Tarefas de banco de dados do OEM Management Agent para instâncias de banco de dados Oracle	Agora, as instâncias de banco de dados do Amazon RDS para Oracle oferecem suporte a procedimentos para chamar determina dos comandos EMCTL no Management Agent. Para ter mais informações, consulte Tarefas do banco de dados do OEM Agent.	24 de outubro de 2019
O Amazon RDS para PostgreSQL é compatível com bancos de dados PostgreSQL transportáveis	Os bancos de dados PostgreSQL transport áveis fornecem um método extremamente rápido de migrar um banco de dados do RDS PostgreSQL entre duas instâncias de banco de dados. Para ter mais informações, consulte Como transportar bancos de dados PostgreSQ L entre instâncias de banco de dados.	8 de outubro de 2019
O Amazon RDS para Oracle é compatível com a autentica ção Kerberos	Agora é possível usar a autenticação Kerberos para autenticar usuários quando se conectam à instância de banco de dados do Amazon RDS executando o Oracle. Para ter mais informações, consulte Usar a autenticação Kerberos com o Amazon RDS para Oracle.	30 de setembro de 2019

Versão 12 beta 3 do Amazon RDS para PostgreSQL no ambiente de pré-visualização de banco de dados	Agora o Amazon RDS para PostgreSQL é compatível com a versão 12 beta 3 do PostgreSQL no ambiente de pré-visualização de banco de dados Para ter mais informaçõ es, consulte Versão 12 beta 3 do PostgreSQL no Amazon RDS no ambiente de pré-visua lização de banco de dados.	28 de agosto de 2019
Suporte para MySQL 8.0.36	Agora é possível criar instâncias de banco de dados do Amazon RDS que executam a versão 8.0 do MySQL. Para ter mais informações, consulte Versões do MySQL no Amazon RDS.	19 de agosto de 2019
O Amazon RDS para Oracle é compatível com uma nova versão principal	Agora é possível criar instâncias de banco de dados do Amazon RDS executando o Oracle Database 18c (18.0). Para ter mais informações, consulte Oracle Database 18c com Amazon RDS.	15 de agosto de 2019

Versão 3 do Management Agent para OEM 13c	As instâncias de banco de dados do Amazon RDS para Oracle agora são compatíveis com o Management Agent para Oracle Enterprise Manager (OEM) Cloud Control 13c Release 3. Para ter mais informações, consulte Oracle Management Agent for Enterprise Manager Cloud Control.	7 de agosto de 2019
Versão 12 beta 2 do Amazon RDS para PostgreSQL no ambiente de pré-visualização de banco de dados	Agora o Amazon RDS para PostgreSQL é compatível com a versão 12 beta 2 do PostgreSQL no ambiente de pré-visualização de banco de dados Para ter mais informações, consulte Versão 12 beta 2 do PostgreSQL no Amazon RDS no ambiente de pré-visualização de banco de dados.	6 de agosto de 2019
O Amazon RDS é compatível com agrupamentos de servidores para o SQL Server	O Amazon RDS para SQL Server oferece suporte a uma seleção de agrupamentos para novas instâncias de banco de dados. Para ter mais informações, consulte Agrupamentos e conjuntos de caracteres do Microsoft SQL Server.	29 de julho de 2019

O Amazon RDS para Oracle é compatível com o Oracle APEX versão 19.1.v1	Agora o Amazon RDS para Oracle é compatível com o Oracle Application Express (APEX) versão 19.1.v1. Para ter mais informações, consulte Oracle Application Express.	28 de junho de 2019
Versão 13 beta 1 do Amazon RDS para PostgreSQL no ambiente de pré-visualização de banco de dados	Agora o Amazon RDS para PostgreSQL é compatível com o PostgreSQL versão 13 beta 1 no ambiente de pré-visualização de banco de dados Para ter mais informaçõ es, consulte Versões 13 do PostgreSQL.	22 de junho de 2019
Escalabilidade automática de armazenamento do Amazon RDS	A escalabilidade automátic a de armazenamento para instâncias de banco de dados do Amazon RDS permite que o Amazon RDS expanda automaticamente o armazenamento associado a uma instância de banco de dados para reduzir a possibili dade de condições de falta de espaço. Para obter informaçõ es sobre a escalabilidade automática de armazenam ento, consulte Trabalhar com armazenamento para instâncias de banco de dados do Amazon RDS.	20 de junho de 2019

O Amazon RDS para Oracle é compatível com as classes de instância de banco de dados db.z1d	Agora, você pode criar instâncias de bancos de dados do Amazon RDS executando o Oracle com as classes de instâncias de banco de dados db.z1d. Para ter mais informações, consulte Classe de instância de banco de dados.	13 de junho de 2019
O Amazon RDS Performance Insights é compatível com a visualização de mais textos em SQL para o Amazon RDS para Oracle	Agora o Insights de Performance do Amazon RDS é compatível com a visualização de mais textos em SQL no painel do Insights de Performance para instâncias de banco de dados do Amazon RDS para Oracle. Para ter mais informações, consulte Visualizar mais textos em SQL no painel do Performance Insights.	10 de junho de 2019
O Amazon RDS adiciona suporte a restaurações nativas de bancos de dados do SQL Server de até 16 TB	Agora você pode fazer restaurações nativas de até 16 TB do SQL Server para o Amazon RDS. Para ter mais informações, consulte Amazon RDS para SQL Server: limitações e recomendações.	4 de junho de 2019

O Amazon RDS adiciona suporte à auditoria do Microsoft SQL Server	Usando o Amazon RDS para Microsoft SQL Server, você pode auditar eventos no nível do servidor e do banco de dados usando o SQL Server Audit e visualizar os resultados em sua instância de banco de dados ou enviar os arquivos de log de auditoria diretamente para o Amazon S3. Para ter mais informaçõ es, consulte Auditoria do SQL Server.	23 de maio de 2019
Melhorias às recomendações do Amazon RDS	O Amazon RDS melhorou suas recomendações automatizadas para recursos de banco de dados. Por exemplo, o Amazon RDS agora fornece recomenda ções para Parâmetros de banco de dados. Para ter mais informações, consulte Usar recomendações do Amazon RDS.	22 de maio de 2019
Suporte a mais bancos de dados por instância de banco de dados no Amazon RDS para SQL Server	Você pode criar até 30 bancos de dados em cada uma das instâncias de banco de dados que executam o Microsoft SQL Server. Para ter mais informações, consulte Limites para instâncias de banco de dados do Microsoft SQL Server.	21 de maio de 2019

Suporte a 64 TiB e 80.000 IOPS de armazenamento para o Amazon RDS para MariaDB, MySQL e PostgreSQL	Agora é possível criar instâncias de banco de dados do Amazon RDS para MariaDB, MySQL e PostgreSQL com até 64 TiB de armazenamento e até 80.000 IOPS provisionadas. Para ter mais informações, consulte Armazenamento de instâncias de banco de dados.	20 de maio de 2019
O Amazon RDS para MySQL é compatível com pré-verificações de atualizações	Quando você atualiza uma instância de banco de dados do MySQL 5.7 para o MySQL 8.0, o Amazon RDS executa pré-verificações para incompatibilidades. Para ter mais informações, consulte Pré-verificações para atualizações do MySQL 5.7 para o 8.0.	17 de maio de 2019
Suporte ao plug-in de validação de senhas do MySQL	Agora você pode usar o plug-in `validate_password` do MySQL para segurança aprimorada de instâncias de banco de dados do Amazon RDS para MySQL. Para ter mais informações, consulte Usar o plug-in de validação de senhas.	16 de maio de 2019

Contadores do Performance Insights para o Amazon RDS para Oracle	Já é possível adicionar contadores de performance aos gráficos do Performance Insights para instâncias de banco de dados Oracle. Para ter mais informações, consulte Contadores do Performance Insights para o Amazon RDS para Oracle.	8 de maio de 2019
Suporte à cobrança por segundo	O Amazon RDS agora é cobrado em incrementos de um segundo em todas as regiões da AWS, exceto AWS GovCloud (EUA) para instâncias sob demanda. Para ter mais informaçõ es, consulte Cobrança da instância de banco de dados para o Amazon RDS.	25 de abril de 2019
Compatibilidade com importação de dados do Amazon S3 para o Amazon RDS para PostgreSQL	Agora você pode importar dados do arquivo do Amazon S3 para uma tabela em uma instância de banco de dados PostgreSQL do RDS. Para ter mais informações, consulte Importar dados do Amazon S3 para uma instância de banco de dados PostgreSQL do RDS.	24 de abril de 2019

Suporte à restauração de backups do 5.7 do Amazon S3	Agora você pode criar um backup de seu banco de dados do MySQL versão 5.7, armazená-lo no Amazon S3 e, em seguida, restaurar o arquivo de backup em uma nova instância de banco de dados do Amazon RDS executando o MySQL. Para ter mais informações, consulte Restaurar um backup em uma instância de banco de dados MySQL do Amazon RDS.	17 de abril de 2019
Suporte a várias atualizações da versão principal do Amazon RDS para PostgreSQL	Com o Amazon RDS para PostgreSQL, agora você pode várias atualizações da versão principal ao atualizar o mecanismo de banco de dados. Esse recurso permite que você avance para uma versão principal mais nova ao atualizar versões seleciona das do mecanismo de banco de dados PostgreSQL. Para ter mais informações, consulte Atualizar versões do mecanismo de banco de dados PostgreSQL.	16 de abril de 2019

Suporte a 64 TiB de armazenamento para o Amazon RDS para Oracle	Agora é possível criar instâncias de banco de dados do Amazon RDS para Oracle com até 64 TiB de armazenamento e até 80.000 IOPS provisionadas. Para ter mais informações, consulte Armazenamento de instâncias de banco de dados.	4 de abril de 2019
Suporte para MySQL 8.0.15	Agora é possível criar instâncias de banco de dados do Amazon RDS executando o MySQL versão 8.0.15. Para ter mais informações, consulte Versões do MySQL no Amazon RDS.	3 de abril de 2019
Suporte ao MariaDB 10.3.13	Agora é possível criar instâncias de banco de dados do Amazon RDS executando o MariaDB versão 10.3.13. Para ter mais informações, consulte Versões do MariaDB no Amazon RDS.	3 de abril de 2019

Fim do suporte ao Microsoft SQL Server 2008 R2 no Amazon RDS	Acabou o suporte ao Microsoft SQL Server 2008 R2, o que coincide com o plano da Microsoft de encerrar o suporte estendido a essa versão em 9 de julho de 2019. Os snapshots existentes do Microsoft SQL Server 2008 R2 deverão ser atualizados para a versão secundária mais recente do Microsoft SQL Server 2012 a partir de 1º de junho de 2019. Para ter mais informações, consulte Suporte ao Microsoft SQL Server 2008 R2 no Amazon RDS.	2 de abril de 2019
Grupos de Disponibilidade Always On com suporte no Microsoft SQL Server 2017	Agora você pode usar Grupos de Disponibilidade Always On no Microsoft SQL Server 2017 Enterprise Edition 14.00.3049.1 ou posterior. Para ter mais informações, consulte Implantações multi-AZ para Microsoft SQL Server.	29 de março de 2019
Visualizar métricas de volume	Agora você pode visualizar métricas de volumes do Amazon Elastic Block Store (Amazon EBS), que são os dispositivos físicos usados para armazenamento de banco de dados e logs. Para ter mais informações, consulte Visualizar o monitoramento avançado.	20 de março de 2019

Suporte para MySQL 5.7.25	Agora é possível criar instâncias de banco de dados do Amazon RDS executando o MySQL versão 5.7.25. Para ter mais informações, consulte Versões do MySQL no Amazon RDS.	19 de março de 2019
O Amazon RDS para Oracle é compatível com tarefas de DBA do RMAN	O Amazon RDS para Oracle agora oferece suporte a tarefas de DBA do Oracle Recovery Manager (RMAN), incluindo backups do RMAN. Para ter mais informações, consulte Tarefas comuns de DBA do Oracle Recovery Manager (RMAN) para instâncias de banco de dados Oracle.	14 de março de 2019
O Amazon RDS para PostgreSQL é compatível com a versão 11.1.	Agora é possível criar instâncias de banco de dados do Amazon RDS executando o PostgreSQL versão 11.1. Para ter mais informações, consulte PostgreSQL versão 11.1 no Amazon RDS.	12 de março de 2019
A restauração de vários arquivos está disponível no Amazon RDS para SQL Server	Agora, você pode restaurar a partir de vários arquivos com o Amazon RDS para SQL Server. Para ter mais informações, consulte Restaurar um banco de dados.	11 de março de 2019

MariaDB 10.2.21	Agora é possível criar instâncias de banco de dados do Amazon RDS executando o MariaDB versão 10.2.21. Para ter mais informações, consulte Versões do MariaDB no Amazon RDS.	11 de março de 2019
O Amazon RDS para Oracle é compatível com réplicas de leitura	O Amazon RDS para Oracle agora é compatível com réplicas de leitura com o Active Data Guard. Para ter mais informações, consulte Trabalhar com réplicas de leitura e Trabalhar com réplicas de leitura da Oracle.	11 de março de 2019
O Amazon RDS Performance Insights está disponível para o Amazon RDS para MariaDB	O Amazon RDS Performance Insights e posterior para o Amazon RDS para MariaDB. Para ter mais informações, consulte Usar o Amazon RDS Performance Insights.	11 de março de 2019
MySQL 8.0.13 e 5.7.24	Agora é possível criar instâncias de banco de dados do Amazon RDS executando o MySQL versões 8.0.13 e 5.7.24. Para ter mais informações, consulte Versões do MySQL no Amazon RDS.	8 de março de 2019

O Amazon RDS Performance Insights está disponível para o Amazon RDS para SQL Server	O Amazon RDS Performance Insights e posterior para o Amazon RDS para SQL Server. Para ter mais informações, consulte Usar o Amazon RDS Performance Insights.	4 de março de 2019
O Amazon RDS para Oracle oferece suporte à integração com Amazon S3	Agora você pode transferir arquivos entre uma instância de banco de dados do Amazon RDS para Oracle e um bucket do Amazon S3. Para ter mais informações, consulte Integrar o Amazon RDS para Oracle e o Amazon S3.	26 de fevereiro de 2019
O Amazon RDS para MySQL e o Amazon RDS para MariaDB são compatíveis com classes de instâncias de banco de dados db.t3	Agora você pode criar instâncias de bancos de dados do Amazon RDS executando o MySQL ou MariaDB que usam as classes de instâncias de banco de dados db.t3. Para ter mais informações, consulte Classe de instância de banco de dados.	20 de fevereiro de 2019
O Amazon RDS para MySQL e o Amazon RDS para MariaDB são compatíveis com classes de instâncias de banco de dados db.r5	Agora você pode criar instâncias de bancos de dados do Amazon RDS executando o MySQL ou MariaDB que usam as classes de instâncias de banco de dados db.r5. Para ter mais informações, consulte Classe de instância de banco de dados.	20 de fevereiro de 2019

Contadores do Performance Insights para o RDS para MySQL e PostgreSQL	Agora é possível adicionar contadores de performance aos gráficos do Performance Insights para instâncias de banco de dados MySQL e PostgreSQL. Para ter mais informações, consulte Componentes do painel do Performance Insights.	19 de fevereiro de 2019
O Amazon RDS para PostgreSQL agora é compatível com o ajuste do parâmetro autovacuum adaptável	O ajuste do parâmetro autovacuum adaptável com o Amazon RDS para PostgreSQL ajuda a evitar a conclusão do ID da transação ajustando os valores do parâmetro autovacuum automaticamente. Para ter mais informações, consulte Reduzir a probabilidade de conclusão do ID da transação.	12 de fevereiro de 2019
O Amazon RDS para Oracle é compatível com o Oracle APEX versões 18.1.v1 e 18.2.v1	O Amazon RDS para Oracle agora é compatível com o Oracle Application Express (APEX) versões 18.1.v1 e 18.2.v1. Para ter mais informações, consulte Oracle Application Express.	11 de fevereiro de 2019

O Amazon RDS Performance Insights é compatível com a visualização de mais textos em SQL para o RDS para MySQL	O Amazon RDS Performance Insights agora é compatível com a visualização de mais textos em SQL no painel do Performance Insights para instâncias de banco de dados MySQL. Para ter mais informações, consulte Visualizar mais textos em SQL no painel do Performance Insights.	6 de fevereiro de 2019
O Amazon RDS para PostgreSQL é compatível com as classesde instância de banco de dados db.t3	Agora você pode criar instâncias de bancos de dados do Amazon RDS que estejam executando o PostgreSQL que usam as classes de instância de banco de dados db.t3. Para ter mais informações, consulte Classe de instância de banco de dados.	25 de janeiro de 2019
O Amazon RDS para Oracle é compatível com as classes de instância de banco de dados db.t3	Agora, você pode criar instâncias de bancos de dados do Amazon RDS executando o Oracle que usam as classes de instâncias de banco de dados db.t3. Para ter mais informações, consulte Classe de instância de banco de dados.	25 de janeiro de 2019

O Amazon RDS Performance Insights é compatível com a visualização de mais textos em SQL para o Amazon RDS para PostgreSQL	O Amazon RDS Performance Insights agora é compatível com a visualização de mais textos em SQL no painel do Performance Insights para instâncias de banco de dados do Amazon RDS para PostgreSQL. Para ter mais informações, consulte Visualizar mais textos em SQL no painel do Performance Insights.	24 de janeiro de 2019
O Amazon RDS para Oracle é compatível com uma nova versão do SQLT	Amazon RDS para Oracle agora é compatível com o SQLT versão 12.2.180725. Para ter mais informações, consulte Oracle SQLT.	22 de janeiro de 2019
O Amazon RDS para PostgreSQL é compatível com as classesde instância de banco de dados db.r5	Agora crie instâncias de bancos de dados do Amazon RDS que estejam executando o PostgreSQL que usam as classes de instância de banco de dados db.r5. Para ter mais informações, consulte Classe de instância de banco de dados.	19 de dezembro de 2018

O Amazon RDS para PostgreSQL já é compatível com o gerenciamento de senhas restringido	O Amazon RDS para PostgreSQL permite restringir quem pode gerenciar senhas de usuário e alterações na expiração da senha usando o parâmetro `rds.restrict_password_commands` e a função `rds_password`. Para ter mais informações, consulte Restringir gerenciamento de senhas.	19 de dezembro de 2018
O Amazon RDS para PostgreSQL é compatível com o carregamento de logs de banco de dados no Amazon CloudWatch Logs	O Amazon RDS para PostgreSQL é compatível com o carregamento de logs de banco de dados no CloudWatch Logs. Para ter mais informações, consulte Publicar logs do PostgreSQL no CloudWatch Logs.	10 de dezembro de 2018
O Amazon RDS para Oracle é compatível com as classesde instância de banco de dados db.r5	Agora crie instâncias de bancos de dados do Amazon RDS que estejam executando o Oracle que usam as classes de instâncias de banco de dados db.r5. Para ter mais informações, consulte Classe de instância de banco de dados.	20 de novembro de 2018

Reter backups ao excluir uma instância de banco de dados	O Amazon RDS oferece suporte à retenção de backups automáticos quando você exclui uma instância de banco de dados. Para ter mais informações, consulte Trabalhar com backups.	15 de novembro de 2018
O Amazon RDS para PostgreSQL é compatível com as classesde instância de banco de dados db.m5	Agora crie instâncias de bancos de dados do Amazon RDS que estejam executando o PostgreSQL que usam as classes de instância de banco de dados db.m5. Para ter mais informações, consulte Classe de instância de banco de dados.	15 de novembro de 2018
O Amazon RDS para Oracle é compatível com uma nova versão principal	Agora crie instâncias de banco de dados do Amazon RDS que estejam executando o Oracle versão 12.2.	13 de novembro de 2018
O Amazon RDS para SQL Server é compatível com Always On	O Amazon RDS para SQL Server agora oferece suporte a Grupos de Disponibilidade Always On. Para ter mais informações, consulte Implantações multi-AZ para Microsoft SQL Server.	8 de novembro de 2018

O Amazon RDS para PostgreSQL é compatível com o acesso de rede de saída usando servidores DNS personalizados	O Amazon RDS para PostgreSQL é compatível com o acesso de rede de saída usando servidores DNS personalizados. Para ter mais informações, consulte Usar um servidor DNS personalizado para acesso à rede de saída.	8 de novembro de 2018
O Amazon RDS para MariaDB, MySQL e PostgreSQL são compatíveis com 32 TiB de armazenamento	Agora crie instâncias de banco de dados do Amazon RDS com até 32 TiB de armazenamento para MySQL, MariaDB e PostgreSQL. Para ter mais informações, consulte Armazenamento de instâncias de banco de dados.	7 de novembro de 2018
O Amazon RDS para Oracle é compatível com tipos de dados estendidos	Você agora pode habilitar os tipos de dados estendidos em instâncias de banco de dados do Amazon RDS que executam o Oracle. Com tipos de dados estendidos, o tamanho máximo é 32.767 bytes para os tipos de dados VARCHAR2, NVARCHAR2 e RAW. Para ter mais informações, consulte Como usar tipos de dados estendidos.	6 de novembro de 2018

O Amazon RDS para Oracle é compatível com as classes de instância de banco de dados db.m5	Agora, você pode criar instâncias de bancos de dados do Amazon RDS executando o Oracle com as classes de instâncias de banco de dados db.m5. Para ter mais informações, consulte Classe de instância de banco de dados.	2 de novembro de 2018
Migração do Amazon RDS para Oracle de SE, SE1 ou SE2 para EE	Agora, você pode migrar de qualquer Oracle Database Standard Edition (SE, SE1 ou SE2) para o Oracle Database Enterprise Edition (EE). Para ter mais informações, consulte Migração entre as edições do Oracle.	31 de outubro de 2018
O Amazon RDS já pode parar instâncias multi-AZ	O Amazon RDS já pode parar uma instância de banco de dados que faça parte de uma implantação multi-AZ. Anteriormente, o recurso de interrupção de instância tinha uma limitação para instâncias multi-AZ. Para ter mais informações, consulte Interrupção temporária de uma instância de banco de dados do Amazon RDS.	29 de outubro de 2018

O Amazon RDS Performance Insights está disponível para o Amazon RDS para Oracle	O Amazon RDS Performance Insights e posterior para o Amazon RDS para Oracle. Para ter mais informações, consulte Usar o Amazon RDS Performance Insights.	29 de outubro de 2018
O Amazon RDS para PostgreSQL é compatível com o PostgreSQL versão 11 no ambiente de pré-visualização de banco de dados	O Amazon RDS para PostgreSQL agora é compatível com o PostgreSQL versão 11 no ambiente de pré-visualização de banco de dados. Para ter mais informações, consulte PostgreSQL versão 11 no Amazon RDS no ambiente de pré-visualização de banco de dados.	25 de outubro de 2018
O MySQL é compatível com uma nova versão principal	Agora é possível criar instâncias de banco de dados do Amazon RDS que executam a versão 8.0 do MySQL. Para ter mais informações, consulte Versões do MySQL no Amazon RDS.	23 de outubro de 2018
O MariaDB é compatível com uma nova versão principal	Agora é possível criar instâncias de banco de dados do Amazon RDS que executam a versão 10.3 do MariaDB. Para ter mais informações, consulte Versões do MariaDB no Amazon RDS.	23 de outubro de 2018

Amazon RDS para Oracle é compatível com o Oracle JVM	Amazon RDS para Oracle agora oferece suporte à opção Java Virtual Machine (JVM). Para ter mais informações, consulte Oracle Java Virtual Machine.	16 de outubro de 2018
Grupo de parâmetros personalizado para restauração e recuperação em um ponto anterior no tempo	Agora você pode especificar um grupo de parâmetros personalizado ao restaurar um snapshot ou executar uma operação de recuperação em um ponto anterior no tempo. Para ter mais informações, consulte Restauração a partir de um snapshot de banco de dados e Restauração de uma instância de banco de dados em um horário especificado.	15 de outubro de 2018
O Amazon RDS para Oracle é compatível com o armazenamento de 32 TiB	Agora é possível criar instâncias de banco de dados doOracle RDS com até 32 TiB de armazenamento. Para ter mais informações, consulte Armazenamento de instâncias de banco de dados.	15 de outubro de 2018

O Amazon RDS para MySQL é compatível com GTIDs	O Amazon RDS para MySQL agora oferece suporte a identificadores de transaçõe s globais (GTIDs), que são exclusivos em todas as instâncias de banco de dados e em uma configuração de replicação. Para ter mais informações, consulte Usar a replicação baseada em GTID para o Amazon RDS MySQL	10 de outubro de 2018
MySQL 5.7.23, 5.6.41 e 5.5.61	Agora é possível criar instâncias de banco de dados do Amazon RDS executand o as versões 5.7.23, 5.6.41 e 5.5.61 do MySQL. Para ter mais informações, consulte Versões do MySQL no Amazon RDS.	8 de outubro de 2018
O Amazon RDS para Oracle é compatível com uma nova versão do SQLT	Amazon RDS para Oracle agora é compatível com o SQLT versão 12.2.180331. Para ter mais informações, consulte Oracle SQLT.	4 de outubro de 2018
O Amazon RDS para PostgreSQL agora é compatível com a autentica ção do IAM	O Amazon RDS para PostgreSQL agora é compatível com a autentica ção do IAM. Para ter mais informações, consulte Autenticação de banco de dados do IAM para MySQL e PostgreSQL.	27 de setembro de 2018

Você pode habilitar a proteção contra exclusão para as instâncias de banco de dados do Amazon RDS	Quando você habilita a proteção contra exclusão para uma instância de banco de dados, o banco de dados não pode ser excluído por nenhum usuário. Para ter mais informações, consulte Excluir uma instância de banco de dados.	26 de setembro de 2018
O Amazon RDS para MySQL e o Amazon RDS para MariaDB são compatíveis com as classes db.m5 de instâncias de banco de dados	Agora você pode criar instâncias de bancos de dados do Amazon RDS executando o MySQL ou MariaDB usando as classes de instâncias de banco de dados db.m5. Para ter mais informações, consulte Classe de instância de banco de dados.	18 de setembro de 2018
O Amazon RDS agora é compatível com atualizações do SQL Server 2017	Atualize sua instância de banco de dados existente para o SQL Server 2017 a partir de qualquer versão, exceto SQL Server 2008. Para atualizar a partir do SQL Server 2008, primeiro atualize para uma outra versão. Para obter informações, consulte Atualização do mecanismo de banco de dados do servidor Microsoft SQL Server.	11 de setembro de 2018

Agora o Amazon RDS para PostgreSQL é compatível com a versão 11 beta 3 do PostgreSQL no ambiente de pré-visualização de banco de dados	Nesta versão, o tamanho de segmento do log prévio de gravação (WAL) (wal_segment_size) agora está definido como 64 MB. Para saber mais sobre a versão 11 beta 3 do PostgreSQL, consulte PostgreSQL 11 beta 3 lançado. Para ter mais informações sobre o Database Preview Environment, consulte Trabalhar com o Database Preview Environment.	7 de setembro de 2018
Amazon Aurora Guia do usuário	O Guia do usuário do Amazon Aurora descreve todos os conceitos do Amazon Aurora e fornece instruções sobre como usar os vários recursos com o console e com a interface da linha de comando. O Guia do usuário do Amazon RDS agora abrange mecanismos de banco de dados não Aurora.	31 de agosto de 2018
O Amazon RDS Performance Insights está disponível para RDS para MySQL	O Amazon RDS Performance Insights e posterior para RDS para MySQL. Para ter mais informações, consulte Usar o Amazon RDS Performance Insights.	28 de agosto de 2018

O Aurora edição compatível com PostgreSQL agora é compatível com o Auto Scaling do Aurora	Réplicas de Auto Scaling do Aurora agora estão disponíveis para Aurora Edição compatível com PostgreSQL. Para ter mais informações, consulte Usar o Amazon Aurora Auto Scaling com réplicas do Aurora.	16 de agosto de 2018
Aurora Serverless for Aurora MySQL	Aurora ServerlessO é uma configuração sob demanda e de escalabilidade automática do Amazon Aurora. Para ter mais informações, consulte Usar o Amazon Aurora Serverless.	9 de agosto de 2018
MySQL 5.7.22 e 5.6.40	Agora é possível criar instâncias de banco de dados do Amazon RDS executando as versões 5.7.22 e 5.6.40 do MySQL. Para ter mais informações, consulte Versões do MySQL no Amazon RDS.	6 de agosto de 2018
Agora o Aurora está disponível na Região da China (Ningxia)	O Aurora MySQL e o Aurora PostgreSQL agora estão disponíveis na região da China (Ningxia). Para ter mais informações, consulte Disponibilidade para Amazon Aurora MySQL e Disponibilidade para Amazon Aurora PostgreSQL.	6 de agosto de 2018

O Amazon RDS para MySQL é compatível com replicação atrasada	O Amazon RDS para MySQL agora é compatível com a replicação atrasada como uma estratégia para a recuperação de desastres. Para ter mais informações, consulte Configuração de replicação atrasada com o MySQL.	6 de agosto de 2018
O Amazon RDS Performance Insights está disponível para Aurora MySQL	O Amazon RDS Performance Insights e posterior para Aurora MySQL. Para ter mais informações, consulte Usar o Amazon RDS Performance Insights.	6 de agosto de 2018
Integração do Amazon RDS Performance Insights com o Amazon CloudWatch	O Amazon RDS Performance Insights publica métricas automaticamente para o Amazon CloudWatch. Para ter mais informações, consulte Métricas do Performance Insights publicadas no CloudWatch.	6 de agosto de 2018
Recomendações do Amazon RDS	Agora o Amazon RDS fornece recomendações automatizadas para recursos de banco de dados. Para ter mais informações, consulte Usar recomendações do Amazon RDS.	25 de julho de 2018

Cópias de snapshots incrementais entre regiões da AWS	O Amazon RDS oferece suporte a cópias de snapshots incrementais entre regiões da AWS para instâncias criptografadas e não criptografadas. Para ter mais informações, consulte o tópico sobre como Copiar snapshots entre regiões da AWS.	24 de julho de 2018
O Amazon RDS Performance Insights está disponível para Amazon RDS para PostgreSQL	O Amazon RDS Performance Insights e posterior para Amazon RDS para PostgreSQL. Para ter mais informações, consulte Usar o Amazon RDS Performance Insights.	18 de julho de 2018
O Amazon RDS para Oracle é compatível com o Oracle APEX versão 5.1.4.v1	O Amazon RDS para Oracle agora é compatível com o Oracle Application Express (APEX) versão 5.1.4.v1. Para ter mais informações, consulte Oracle Application Express.	10 de julho de 2018
O Amazon RDS para Oracle é compatível com a publicação de logs no Amazon CloudWatch Logs	O Amazon RDS para Oracle agora oferece suporte à publicação de dados de logs de alerta, auditoria e ouvinte em um grupo de logs no CloudWatch Logs. Para ter mais informações, consulte Publicação de logs do Oracle no Amazon CloudWatch Logs.	9 de julho de 2018

MariaDB 10.2.15, 10.1.34 e 10.0.35	Agora é possível criar instâncias de banco de dados do Amazon RDS executando as versões 10.2.15, 10.1.34 e 10.0.35 do MariaDB. Para ter mais informações, consulte Versões do MariaDB no Amazon RDS.	5 de julho de 2018
O Aurora PostgreSQL 1.2 está disponível e é compatível com o PostgreSQL 9.6.8	O Aurora PostgreSQL 1.2 e posterior e é compatível com o PostgreSQL 9.6.8. Para ter mais informações, consulte Versão 1.2.	27 de junho de 2018
As réplicas de leitura do Amazon RDS PostgreSQL são compatíveis com implantações de multi-AZ	As réplicas de leitura do RDS no Amazon RDS PostgreSQL agora são compatíveis com várias zonas de disponibilidade. Para ter mais informações, consulte Trabalhar com réplicas de leitura do PostgreSQL.	25 de junho de 2018
Performance Insights disponível para Aurora PostgreSQL	O Performance Insights está disponível geralmente para o Aurora PostgreSQL, com suporte para retenção estendida de dados de performance. Para ter mais informações, consulte Uso de Insights de Performance do Amazon RDS.	21 de junho de 2018

O Aurora PostgreSQL está disponível na região Oeste dos EUA (Norte da Califórnia	O Aurora PostgreSQL e posterior na região Oeste dos EUA (Norte da Califórnia). Para ter mais informações, consulte Disponibilidade para o Amazon Aurora PostgreSQL.	11 de junho de 2018
O Amazon RDS para Oracle agora é compatível com a configuração de CPU	O Amazon RDS para Oracle oferece suporte à configuração do número de núcleos de CPU e do número de threads para cada núcleo do processador de uma classe de instância do banco de dados. Para ter mais informações, consulte Configuração do processador da classe de instância de banco de dados.	5 de junho de 2018

Atualizações anteriores

A tabela a seguir descreve as alterações importantes em cada versão do Guia do usuário do Amazon RDS antes de junho de 2018.

Alteração	Descrição	Alterado em
Agora o Amazon RDS para PostgreSQL é compatível com a versão 11 beta 1 do PostgreSQL no ambiente de pré-visualização de banco de dados	A versão 11 beta 1 do PostgreSQL contém várias melhorias descritas em PostgreSQL 11 beta 1 lançado! Para obter informações sobre o ambiente de visualização de banco de dados, consulte Trabalhar com o ambiente de visualização de banco de dados.	31 de maio de 2018

Alteração	Descrição	Alterado em
Agora o Amazon RDS para Oracle oferece suporte às versões 1.0 e 1.2 do TLS	O Amazon RDS para Oracle é compatível com o Transport Layer Security (TLS) versões 1.0 e 1.2. Para ter mais informações, consulte Versões do TLS para a opção Oracle SSL.	30 de maio de 2018
O Aurora MySQL oferece suporte à publicação de logs no Amazon CloudWatch Logs	Agora, o Aurora MySQL oferece suporte à publicaçã o de dados de log geral, lento, de auditoria e de erros em um grupo de logs no CloudWatch Logs. Para ter mais informações, consulte Publicação de Aurora MySQL no CloudWatch Logs.	23 de maio de 2018
Database Preview Environment para o PostgreSQL do Amazon RDS	Agora você pode iniciar uma nova instância do PostgreSQL do Amazon RDS em um modo de visualização. Para ter mais informações sobre o Database Preview Environment, consult, Trabalhar com o ambiente de visualização de banco de dados.	22 de maio de 2018
As instância s de banco de dados do Amazon RDS para Oracle oferecem suporte a novas classes de instância de banco de dados	As instâncias de banco de dados do Oracle agora oferecem suporte às classes de instâncias de banco de dados db.x1e e db.x1. Para ter mais informações, consulte Classes de instâncias de banco de dados e Classes de instância de banco de dados do RDS para Oracle.	22 de maio de 2018
O Amazon RDS PostgreSQL agora é compatível com o postgres_fdw em uma réplica de leitura.	Agora é possível usar o postgres_fdw para se conectar a um servidor remoto a partir de uma réplica de leitura. Para ter mais informações, consulte Usar a extensão postgres_fdw para acessar dados externos.	17 de maio de 2018

Alteração	Descrição	Alterado em
O Amazon RDS para Oracle agora oferece suporte à configuração de parâmetros do sqlnet.ora	Agora você pode definir parâmetros sqlnet.ora com o Amazon RDS para Oracle. Para ter mais informações, consulte Modificação das propriedades de conexão usando parâmetros sqlnet.ora.	10 de maio de 2018
Aurora PostgreSQL disponível na região Ásia-Pacífico (Seul).	Agora o Aurora PostgreSQL está disponível na região da Ásia-Pacífico (Seul). Para ter mais informações, consulte Disponibilidade para o Amazon Aurora PostgreSQL.	9 de maio de 2018
O Aurora MySQL é compatível com o retrocesso	O Aurora MySQL agora é compatível com o retrocesso de um cluster de banco de dados para um período específico, sem restaurar os dados de um backup. Para ter mais informações, consulte Retrocesso de um cluster de bancos de dados Aurora.	9 de maio de 2018
Aurora MySQL oferece migração criptografada e replicação de MySQL externo	Aurora MySQL agora oferece migração criptografada e replicação de um banco de dados MySQL externo. Para ter mais informações, consulte Migração de dados de um banco de dados MySQL externo para um cluster de banco de dados Amazon Aurora MySQL e Replicação entre Aurora e MySQL ou entre Aurora e outro cluster de banco de dados Aurora.	25 de abril de 2018
Suporte do Aurora Edição compatível com PostgreSQL para o protocolo Copy-on-Write (Cópia-em-gravação).	Agora você pode clonar bases de dados em um cluster de bancos de dados Aurora PostgreSQL. Para ter mais informações, consulte Clonagem de bancos de dados em um cluster de bancos de dados Aurora.	10 de abril de 2018

Alteração	Descrição	Alterado em
MariaDB 10.2.12, 10.1.31 e 10.0.34	Agora é possível criar instâncias de banco de dados do Amazon RDS executando as versões 10.2.12, 10.1.31 e 10.0.34 do MariaDB. Para ter mais informações, consulte MariaDB em versões do Amazon RDS.	21 de março de 2018
Suporte do Aurora PostgreSQL a novas regiões	O Aurora PostgreSQL e posterior nas regiões da UE (Londres) e da Ásia-Pacífico (Singapura). Para ter mais informações, consulte Disponibilidade para o Amazon Aurora PostgreSQL.	13 de março de 2018
MySQL 5.7.21, 5.6.39 e 5.5.59	Agora é possível criar instâncias de banco de dados do Amazon RDS executando as versões 5.7.21, 5.6.39 e 5.5.59 do MySQL. Para ter mais informações, consulte MySQL em versões do Amazon RDS.	9 de março de 2018
O Amazon RDS para Oracle agora é compatível com o Oracle REST Data Services	O Amazon RDS para Oracle é compatível com o Oracle REST Data Services como parte da opção do APEX. Para ter mais informações, consulte Oracle Application Express (APEX).	9 de março de 2018
Amazon Aurora Edição compatível com MySQL disponível na nova região da AWS	Agora, o Aurora MySQL está disponível na região Ásia-Pacífico (Singapura). Para obter a lista completa de regiões da AWS para o Aurora MySQL, consulte a disponibilidade para o Amazon Aurora MySQL.	6 de março de 2018

Alteração	Descrição	Alterado em
As instâncias de banco de dados do Amazon RDS que executam o Microsoft SQL Server comportam captura de dados de alteração (CDC).	As instâncias de banco de dados que executam o Amazon RDS para Microsoft SQL Server agora comportam captura de dados de alteração (CDC). Para ter mais informações, consulte Compatibilidade entre captura de dados de alteração e instâncias de banco de dados do Microsoft SQL Server.	6 de fevereiro de 2018
O Aurora MySQL é compatível com uma nova versão principal	Agora, você pode criar clusters de bancos de dados Aurora MySQL executando a versão 5.7 do MySQL. Para ter mais informações, consulte Atualizações do mecanismo de banco de dados MySQL do Amazon Aurora 06/02/2018.	6 de fevereiro de 2018
Publicar logs MySQL e MariaDB no Amazon CloudWatch Logs	Agora você pode publicar dados de log MySQL e MariaDB no CloudWatch Logs. Para ter mais informações, consulte Publicação de logs do MySQL no Amazon CloudWatch Logs e Publicar logs do MariaDB no Amazon CloudWatch Logs.	17 de janeiro de 2018
Suporte a multi-AZ para réplicas de leitura	Agora é possível criar uma réplica de leitura como uma instância de banco de dados multi-AZ. O Amazon RDS cria um em modo de espera de sua réplica em outra zona de disponibilidade para suporte a failover da réplica. Você pode criar a réplica de leitura como uma instância de banco de dados multi-AZ independe ntemente de o banco de dados de origem ser ou não uma instância de banco de dados multi-AZ. Para ter mais informações, consulte Trabalhar com réplicas de leitura de instância de banco de dados.	11 de janeiro de 2018

Alteração	Descrição	Alterado em
O Amazon RDS para MariaDB é compatível com uma nova versão principal	Agora é possível criar instâncias de banco de dados do Amazon RDS que executam a versão 10.2 do MariaDB. Para ter mais informações, consulte Suporte para MariaDB 10.2 no Amazon RDS.	3 de janeiro de 2018
Amazon Aurora Edição compatível com PostgreSQL disponível na nova região da AWS	Agora, o Aurora PostgreSQL está disponível na região UE (Paris). Para obter a lista completa de regiões da AWS para o Aurora PostgreSQL, consulte Disponibilidade para o Amazon Aurora PostgreSQL.	22 de dezembro de 2017
O Aurora PostgreSQL oferece suporte a novos tipos de instâncias	Agora, o Aurora PostgreSQL oferece suporte a novos tipos de instâncias. Para obter a lista completa de tipos de instância, consulte Escolha da classe de instância de banco de dados.	20 de dezembro de 2017
Amazon Aurora Edição compatível com MySQL disponível na nova região da AWS	Agora, o Aurora MySQL está disponível na região UE (Paris). Para obter a lista completa de regiões da AWS para o Aurora MySQL, consulte a disponibilidade para o Amazon Aurora MySQL.	18 de dezembro de 2017
O Aurora MySQL suporta junções de hash	Esse recurso pode melhorar a performance da consulta quando você precisa unir uma grande quantidade de dados usando um equijoin. Para ter mais informações, consulte Trabalhar com hash joins no Aurora MySQL.	11 de dezembro de 2017
O Aurora MySQL oferece suporte a funções nativas para invocar funções do AWS Lambda	Você pode chamar as funções nativas `lambda_sync` e `lambda_async` ao usar o Aurora MySQL. Para ter mais informações, consulte Invocação de uma função do Lambda a partir de um cluster de banco de dados Amazon Aurora MySQL.	11 de dezembro de 2017

Alteração	Descrição	Alterado em
Adição da qualificação para HIPAA do Aurora PostgreSQL	O Aurora PostgreSQL agora é compatível com a criação de aplicativos compatíveis com HIPAA. Para ter mais informações, consulte Trabalhar com Amazon Aurora PostgreSQL.	6 de dezembro de 2017
Regiões da AWS adicionais disponíveis para Amazon Aurora compatível com o PostgreSQL	O Amazon Aurora compatível com o PostgreSQL e posterior em quatro novas regiões da AWS. Para ter mais informações, consulte Disponibilidade para o Amazon Aurora PostgreSQL.	22 de novembro de 2017
Modificar armazenamento para instâncias de banco de dados do Amazon RDS executando o Microsoft SQL Server	Agora, você pode modificar o armazenamento de suas instâncias de bancos de dados do Amazon RDS que executam o SQL Server. Para ter mais informações, consulte Modificar uma instância de banco de dados do Amazon RDS.	21 de novembro de 2017
O Amazon RDS suporta 16 TiB de armazenamento para mecanismos baseados em Linux	Agora é possível criar instâncias de banco de dadosMySQL, MariaDB, PostgreSQL e Oracle RDS com até 16 TiB de armazenamento. Para ter mais informações, consulte Armazenamento de instâncias de banco de dados do Amazon RDS.	21 de novembro de 2017
O Amazon RDS suporta uma rápida expansão do armazenamento	Agora você pode adicionar armazenamento às instâncias de banco de dados MySQL, MariaDB, PostgreSQL e Oracle RDS em poucos minutos. Para ter mais informações, consulte Armazenamento de instâncias de banco de dados do Amazon RDS.	21 de novembro de 2017

Alteração	Descrição	Alterado em
O Amazon RDS suporta o MariaDB nas versões 10.1.26 e 10.0.32	Agora é possível criar instâncias de banco de dados do Amazon RDS executando as versões 10.1.26 e 10.0.32 do MariaDB. Para ter mais informações, consulte MariaDB em versões do Amazon RDS.	20 de novembro de 2017
O Amazon RDS para Microsoft SQL Server agora suporta novas classes de instâncias de banco de dados	Agora você pode criar instâncias de bancos de dados do Amazon RDS executando o SQL Server usando as classes de instâncias de banco de dados db.r4 e db.m4.16xlarge. Para ter mais informações, consulte Suporte a classes de instâncias de banco de dados para o Microsoft SQL Server.	20 de novembro de 2017
O Amazon RDS para MySQL e MariaDB agora suporta novas classes de instâncias de banco de dados	Agora é possível criar instâncias de banco de dados do Amazon RDS executando MySQL e MariaDB usando as classes de instância de banco de dados db.r4, db.m4.16xlarge, db.t2.xlarge e db.t2.2xlarge. Para ter mais informações, consulte Classes de instâncias de banco de dados .	20 de novembro de 2017
SQL Server 2017	Agora é possível criar instâncias de banco de dados do Amazon RDS executando o Microsoft SQL Server 2017. Você também pode criar instâncias de banco de dados executando o SQL Server 2016 SP1 CU5. Para ter mais informações, consulte Amazon RDS para Microsoft SQL Server.	17 de novembro de 2017
Restaurar backups do MySQL usando o Amazon S3	Agora você pode criar um backup de seu banco de dados no local, armazená-lo no Amazon S3 e, em seguida, restaurar o arquivo de backup em uma nova instância de banco de dados do Amazon RDS executando o MySQL. Para ter mais informações, consulte Restauração de um backup em uma instância de banco de dados MySQL.	17 de novembro de 2017

Alteração	Descrição	Alterado em
Auto Scaling com réplicas do Aurora	Agora o Amazon Aurora MySQL é compatível com o Auto Scaling do Aurora. O Auto Scaling do Aurora ajusta dinamicamente o número de réplicas do Aurora com base em aumentos ou diminuições de conectividade ou de workload. Para ter mais informações, consulte Usar o Amazon Aurora Auto Scaling com réplicas do Aurora.	17 de novembro de 2017
Suporte à edição padrão da Oracle	As instâncias de banco de dados do Amazon RDS para Oracle agora oferecem suporte à edição padrão para a instância de banco de dados. Para ter mais informações, consulte Definir a edição padrão para uma instância de banco de dados.	3 de novembro de 2017
Validação de arquivos da instância de banco de dados Oracle	As instâncias de banco de dados do Amazon RDS para Oracle agora oferecem suporte à validação de arquivos de instância de banco de dados com o utilitário de validação lógica Oracle Recovery Manager (RMAN). Para ter mais informações, consulte Validar arquivos de banco de dados no RDS para Oracle .	3 de novembro de 2017
Management Agent para OEM 13c	As instâncias de banco de dados do Amazon RDS para Oracle agora são compatíveis com o Management Agent para Oracle Enterprise Manager (OEM) Cloud Control 13c. Para ter mais informações, consulte Oracle Management Agent para Enterprise Manager Cloud Control.	1 de novembro de 2017
Reconfiguração de armazenamento para snapshots do Microsoft SQL Server	Agora, é possível reconfigurar o armazenamento ao restaurar um snapshot para uma instância de banco de dados do Amazon RDS com Microsoft SQL Server. Para ter mais informações, consulte Restaurar uma instância de banco de dados.	26 de outubro de 2017

Alteração	Descrição	Alterado em
Pré-busca de chave assíncrona para Aurora Edição compatível com MySQL	A pré-busca de chave assíncrona (AKP) melhora a performance de junções de índice não armazenadas em cache executando a pré-busca de chaves na memória antes de elas serem necessárias. Para ter mais informações, consulte Trabalhar com pré-busca de chave assíncrona no Amazon Aurora.	26 de outubro de 2017
MySQL 5.7.19, 5.6.37 e 5.5.57	Agora é possível criar instâncias de banco de dados do Amazon RDS executando as versões 5.7.19, 5.6.37 e 5.5.57 do MySQL. Para ter mais informações, consulte MySQL em versões do Amazon RDS.	25 de outubro de 2017
Disponibilidade geral do Amazon Aurora com compatibilidade PostgreSQL	Com o Amazon Aurora com compatibilidade PostgreSQL, configurar, operar e dimensionar suas implantações do PostgreSQL novas e existentes se torna algo simples e econômico, permitindo que você se concentre em seus negócios e aplicativos. Para ter mais informações, consulte Trabalhar com Amazon Aurora PostgreSQL.	24 de outubro de 2017
As instâncias de banco de dados Amazon RDS para Oracle são compatíveis com novas classes de instância de banco de dados	As instâncias de banco de dados Amazon RDS para Oracle agora são compatíveis com classes de instância de próxima geração otimizadas para memória (db.r4). As instâncias de banco de dados Amazon RDS para Oracle agora também oferecem suporte às seguintes novas classes atuais de instância de geração: db.m4.16xlarge, db.t2.xlarge e db.t2.2xlarge. Para ter mais informações, consulte Classes de instâncias de banco de dados e Classes de instância de banco de dados do RDS para Oracle.	23 de outubro de 2017

Alteração	Descrição	Alterado em
Novo recurso	Suas instâncias reservadas novas e existentes agora podem cobrir vários tamanhos na mesma classe de instância de banco de dados. As instâncias reservadas de tamanho flexível estão disponíveis para instâncias de banco de dados com a mesma região da AWS, mecanismo de banco de dados e família de instância, e em toda a configuração AZ. As instâncias reservadas de tamanho flexível estão disponíveis para os seguintes mecanismos de banco de dados: Amazon Aurora, MariaDB, MySQL, Oracle (Bring-Your-Own-License), PostgreSQL. Para ter mais informações, consulte Instâncias de banco de dados reservadas de tamanho flexível.	11 de outubro de 2017
Novo recurso	Agora você pode usar a opção Oracle SQLT para ajustar uma instrução SQL para a performance ideal. Para ter mais informações, consulte Oracle SQLT.	22 de setembro de 2017
Novo recurso	Se você tiver snapshots de banco de dados manuais existentes das suas instâncias de banco de dados do Amazon RDS para Oracle, agora é possível atualizá-las para uma versão posterior do mecanismo de banco de dados Oracle. Para ter mais informações, consulte Como atualizar para um snapshot de banco de dados Oracle.	20 de setembro de 2017
Novo recurso	Agora você pode usar o Oracle Spatial para armazenar, recuperar, atualizar e consultar dados espaciais em suas instâncias de banco de dados do Amazon RDS com Oracle. Para ter mais informações, consulte Oracle Spatial.	15 de setembro de 2017

Alteração	Descrição	Alterado em
Novo recurso	Agora você pode usar o Oracle Locator para oferecer suporte a aplicativos baseados em serviços wireless e da Internet, e soluções GIS baseadas no parceiro com suas instâncias de banco de dados do Amazon RDS com Oracle. Para ter mais informações, consulte Oracle Locator.	15 de setembro de 2017
Novo recurso	Agora você pode usar o Oracle Multimedia para armazenar, gerenciar e recuperar imagens, áudio, vídeo e outros dados heterogêneos de mídia em suas instâncias de banco de dados do Amazon RDS com Oracle.	15 de setembro de 2017
Novo recurso	Agora você pode exportar logs de auditoria dos clusters de banco de dados MySQL do Amazon Aurora para o Amazon CloudWatch Logs. Para ter mais informações, consulte Publicação de logs do Aurora MySQL no Amazon CloudWatch Logs.	14 de setembro de 2017
Novo recurso	O Amazon RDS agora é compatível com várias versões do Oracle Application Express (APEX) para suas instâncias de banco de dados com Oracle. Para ter mais informações, consulte Oracle Application Express (APEX).	13 de setembro de 2017
Novo recurso	Agora você pode usar o Amazon Aurora para migrar um snapshot de banco de dados criptografado ou não, ou uma instância de banco de dados MySQL, para um cluster de banco de dados criptografado do Aurora. Para ter mais informações, consulte Migrar de um snapshot MySQL do RDS para o Aurora e Migrar de uma instância de banco de dados MySQL para um cluster de bancos de dados Amazon Aurora MySQL usando uma réplica de leitura do Aurora.	5 de setembro de 2017

Alteração	Descrição	Alterado em
Novo recurso	Você pode usar os bancos de dados do Amazon RDS para Microsoft SQL Server para criar aplicativos em conformidade com a norma HIPAA. Para ter mais informações, consulte Suporte ao programa de conformidade de instâncias de banco de dados do Microsoft SQL Server.	31 de agosto de 2017
Novo recurso	Agora você pode usar bancos de dados Amazon RDS para MariaDB para criar aplicações em conformidade com a norma HIPAA. Para ter mais informações, consulte Amazon RDS para MariaDB.	31 de agosto de 2017
Novo recurso	Agora é possível criar instâncias de banco de dados do Amazon RDS com Microsoft SQL Server com armazenamento alocado de até 16 TiB e IOPS provisionadas para intervalos de armazenamento de 1:1 a 50:1. Para ter mais informações, consulte Armazenamento de instâncias de banco de dados do Amazon RDS.	22 de agosto de 2017
Novo recurso	Agora você pode usar implantações multi-AZ para instâncias de banco de dados com Microsoft SQL Server na região UE (Frankfurt). Para ter mais informações, consulte Implantações multi-AZ para o Amazon RDS for Microsoft SQL Server.	3 de agosto de 2017
Novo recurso	Agora é possível criar instâncias de banco de dados do Amazon RDS executando as versões 10.1.23 e 10.0.31 do MariaDB. Para ter mais informações, consulte MariaDB em versões do Amazon RDS.	17 de julho de 2017
Novo recurso	Agora o Amazon RDS oferece suporte para o Microsoft SQL Server Enterprise Edition com o modelo de licença inclusa em todas as regiões da AWS. Para ter mais informações, consulte Licenciar o Microsoft SQL Server no Amazon RDS.	13 de julho de 2017

Alteração	Descrição	Alterado em
Novo recurso	Agora o Amazon RDS para Oracle oferece suporte a páginas imensas do kernel do Linux para aumentar a escalabilidade do banco de dados. O uso de HugePages resulta em tabelas de página menores e menos tempo de CPU gasto no gerenciamento de memória, aumentando a performance de instâncias de banco de dados grandes. Você pode usar páginas enormes com suas instâncias de bancos de dados do Amazon RDS que executam todas as edições do Oracle versões 12.1.0.2 e 11.2.0.4. Para ter mais informações, consulte Ativar o HugePages para uma instância do RDS para Oracle.	7 de julho de 2017
Novo recurso	Atualizado para dar suporte à criptografia em repouso (EAR) para classes de instância de banco de dados db.t2.small e db.t2.medium para todos os mecanismos de banco de dados não Aurora. Para ter mais informações, consulte Disponibilidade da criptografia do Amazon RDS.	27 de junho de 2017
Novo recurso	Atualizado para dar suporte ao Amazon Aurora na região Europa (Frankfurt). Para ter mais informações, consulte Disponibilidade para o Amazon Aurora MySQL.	16 de junho de 2017
Novo recurso	Agora, você pode especificar um grupo de opções ao copiar um snapshot de banco de dados entre regiões AWS. Para ter mais informações, consulte Considerações sobre grupos de opções.	12 de junho de 2017

Alteração	Descrição	Alterado em
Novo recurso	Agora, você pode copiar snapshots de banco de dados criados das instâncias de banco de dados especializadas entre regiões AWS. Você pode copiar snapshots de instâncias de banco de dados que usam o Oracle TDE, o Microsoft SQL Server TDE e o Microsoft SQL Server multi-AZ com espelhamento. Para ter mais informações, consulte Copiar um snapshot de banco de dados.	12 de junho de 2017
Novo recurso	Agora, o Amazon Aurora permite copiar rapidamente e de forma econômica todos os seus bancos de dados em um cluster de banco de dados Amazon Aurora. Para ter mais informações, consulte Clonagem de bancos de dados em um cluster de bancos de dados Aurora.	12 de junho de 2017
Novo recurso	Agora o Amazon RDS oferece suporte para o Microsoft SQL Server 2016 SP1 CU2. Para ter mais informações, consulte Amazon RDS para Microsoft SQL Server.	7 de junho de 2017
Demonstração	Prévia pública do Amazon Aurora com compatibilidade com PostgreSQL. Para ter mais informações, consulte Trabalhar com Amazon Aurora PostgreSQL.	19 de abril de 2017
Novo recurso	Agora, o Amazon Aurora permite executar uma operação ALTER TABLE tbl_name ADD COLUMN col_name column_definition quase instantaneamente. A operação é concluída sem exigir que a tabela seja copiada e sem causar impacto material em outras instruções DML. Para ter mais informações, consulte Alteração de tabelas no Amazon Aurora usando DDL rápido.	5 de abril de 2017

Alteração	Descrição	Alterado em
Novo recurso	Adicionamos um novo comando de monitoramento, SHOW VOLUME STATUS, para exibir o número de nós e discos em um volume. Para ter mais informaçõ es, consulte Exibição do status de volume para um cluster de bancos de dados do Aurora.	5 de abril de 2017
Novo atributo	Agora, você pode usar sua própria lógica personali zada em suas funções de verificação de senha personalizadas para o Oracle no Amazon RDS. Para ter mais informações, consulte Criação de funções personalizadas para verificar senhas.	21 de março de 2017
Novo recurso	Agora, você pode acessar seus arquivos de log redo on-line e arquivados nas suas instâncias de banco de dados Oracle no Amazon RDS. Para ter mais informações, consulte Acessando logs de refazimento online e arquivados .	21 de março de 2017
Novo recurso	Agora, você pode copiar snapshots de cluster de banco de dados criptografados e não criptogra fados entre contas na mesma região. Para ter mais informações, consulte Cópia de um snapshot do cluster de banco de dados entre contas.	7 de março de 2017
Novo recurso	Agora, você pode compartilhar snapshots de cluster de banco de dados entre contas na mesma região. Para ter mais informações, consulte Compartilhamento de um snapshot do cluster de banco de dados.	7 de março de 2017
Novo recurso	Agora você pode replicar clusters de banco de dados criptografados do Amazon Aurora MySQL para criar réplicas do Aurora entre regiões. Para ter mais informações, consulte Como replicar clusters de bancos de dados Amazon Aurora MySQL entre regiões da AWS.	7 de março de 2017

Alteração	Descrição	Alterado em
Novo recurso	Agora, você pode exigir que todas as conexões com a sua instância de banco de dados que executa o Microsoft SQL Server usem a Secure Sockets Layer (SSL). Para ter mais informações, consulte ???.	27 de fevereiro de 2017
Novo recurso	Agora, você pode definir seu fuso horário local em um dos 15 fusos horários adicionais. Para ter mais informações, consulte Fusos horários compatíveis.	27 de fevereiro de 2017
Novo recurso	Agora você pode usar o procedimento do `msdb.dbo.rds_shrink_tempdbfile` do Amazon RDS para reduzir o banco de dados tempdb em suas instâncias de banco de dados executando o Microsoft SQL Server. Para ter mais informações, consulte Recolher o banco de dados tempdb.	17 de fevereiro de 2017
Novo recurso	Agora você pode compactar seu arquivo de backup ao exportar seu banco de dados Microsoft SQL Server Enterprise e Standard Editions de uma instância de banco de dados do Amazon RDS para o Amazon S3. Para ter mais informações, consulte Compactar arquivos de backup.	17 de fevereiro de 2017
Novo recurso	Agora o Amazon RDS oferece suporte para servidores DNS personalizados para resolver nomes DNS usados no acesso à rede de saída em suas instâncias de bancos de dados executando o Oracle. Para ter mais informações, consulte Configuração de um servidor DNS personalizado.	26 de janeiro de 2017
Novo recurso	Agora o Amazon RDS oferece suporte à criação de uma réplica de leitura criptografada em outra região. Para ter mais informações, consulte Criar uma réplica de leitura em uma Região da AWS diferente e CreateDBInstanceReadReplica.	23 de janeiro de 2017

Alteração	Descrição	Alterado em
Novo recurso	Agora o Amazon RDS oferece suporte à atualizaç ão de um snapshot de banco de dados MySQL do MySQL 5.1 para o MySQL 5.5.	20 de janeiro de 2017
Novo recurso	Agora o Amazon RDS oferece suporte para copiar um snapshot de banco de dados para outra região para os mecanismos de banco de dados MariaDB, MySQL, Oracle, PostgreSQL e Microsoft SQL Server. Para ter mais informações, consulte Copiar um snapshot de banco de dados e CopyDBSnapshot.	20 de dezembro de 2016
Novo recurso	O Amazon Aurora MySQL agora é compatível com indexação espacial. A indexação espacial melhora a performance das consultas em conjuntos de dados grandes para consultas que utilizam dados espaciais. Para ter mais informações, consulte Amazon Aurora MySQL e dados espaciais.	14 de dezembro de 2016
Novo recurso	O Amazon RDS agora oferece suporte para acesso à rede de saída nas suas instâncias de bancos de dados que executam o Oracle. Você pode usar utl_http, utl_tcp e utl_smtp para se conectar da sua instância de banco de dados à rede. Para ter mais informaçõ es, consulte Configurar o acesso UTL_HTTP usando certificados e uma carteira Oracle.	5 de dezembro de 2016
Novo recurso	O Amazon RDS removeu o suporte para o MySQL versão 5.1. No entanto, você pode restaurar snapshots MySQL 5.1 existentes para uma instância MySQL 5.5. Para ter mais informações, consulte Suporte aos mecanismos de armazenamento no RDS para MySQL.	15 de novembro de 2016

Alteração	Descrição	Alterado em
Novo recurso	Agora o Amazon RDS oferece suporte para o Microsoft SQL Server 2016 RTM CU2. Para ter mais informações, consulte Amazon RDS para Microsoft SQL Server.	4 de novembro de 2016
Novo recurso	Agora o Amazon RDS oferece suporte para atualizaç ões de versões principais para instâncias de banco de dados executando o Oracle. Agora, você pode atualizar suas instâncias de bancos de dados Oracle do 11g para o 12c. Para ter mais informações, consulte Fazer upgrade do mecanismo de banco de dados RDS para Oracle.	2 de novembro de 2016
Novo recurso	Agora é possível criar instância de banco de dadosexecutando o Microsoft SQL Server 2014 Enterprise Edition. O Amazon RDS é compatível com o SQL Server 2014 SP2 para todas as edições e todas as regiões. Para ter mais informações, consulte Amazon RDS para Microsoft SQL Server.	25 de outubro de 2016
Novo recurso	Agora o Amazon Aurora MySQL se integra com outros produtos da AWS: você pode carregar dados de texto ou XML para a tabela de um bucket do Amazon S3 ou invocar uma função do AWS Lambda sando o código do banco de dados. Para ter mais informações, consulte Integração do Amazon Aurora MySQL com outros serviços da AWS.	18 de outubro de 2016

Alteração	Descrição	Alterado em
Novo recurso	Agora, você pode acessar o banco de dados tempdb nas suas instâncias de bancos de dados do Amazon RDS que executam o Microsoft SQL Server. Você pode acessar o banco de dados tempdb usando Transact-SQL por meio do Microsoft SQL Server Management Studio (SSMS) ou de qualquer outro aplicativo cliente SQL padrão. Para ter mais informaçõ es, consulte Acessar o banco de dados tempdb em instâncias de banco de dados Microsoft SQL Server no Amazon RDS.	29 de setembro de 2016
Novo recurso	Agora, você pode usar o pacote UTL_MAIL com suas instâncias de bancos de dados do Amazon RDS que executam o Oracle. Para ter mais informações, consulte Oracle UTL_MAIL.	20 de setembro de 2016
Novos recursos	Agora, você pode definir o fuso horário das suas novas instâncias de bancos de dados Microsoft SQL Server como um fuso horário local, para correspon der ao fuso horário dos seus aplicativos. Para ter mais informações, consulte Fuso horário local para instâncias de banco de dados do Microsoft SQL Server.	19 de setembro de 2016
Novo recurso	Agora, você pode usar a opção Oracle Label Security para controlar o acesso a linhas de tabela individuais nas suas instâncias de bancos de dados do Amazon RDS que executam o Oracle Database 12c. Com o Oracle Label Security, você pode impor a conformid ade regulamentar com um modelo de administração baseado em políticas e garantir que um acesso a dados confidenciais seja restrito a apenas usuários com o nível de autorização apropriado. Para ter mais informações, consulte Oracle Label Security.	8 de setembro de 2016

Alteração	Descrição	Alterado em
Novo recurso	Agora, você pode se conectar a um cluster de bancos de dados Amazon Aurora usando o endpoint do leitor, que balanceia a carga de conexões entre as Réplicas do Aurora disponíveis no cluster de banco de dados. À medida que os clientes solicitam novas conexões ao endpoint do leitor, o Aurora distribui as solicitações de conexão entre as réplicas do Aurora no cluster de banco de dados. Essa funcionalidade pode ajudar a equilibrar sua workload de leitura entre várias réplicas do Aurora em seu cluster de banco de dados. Para ter mais informações, consulte Endpoints do Amazon Aurora.	8 de setembro de 2016
Novo recurso	Agora, você pode dar suporte ao Oracle Enterprise Manager Cloud Control nas suas instâncias de bancos de dados do Amazon RDS que executam o Oracle. É possível habilitar o Management Agent nas suas instâncias de banco de dados e compartilhar dados com o Oracle Management Service (OMS). Para ter mais informações, consulte Oracle Management Agent para Enterprise Manager Cloud Control.	1 de setembro de 2016
Novo recurso	Essa versão adiciona suporte para obter um ARN para um recurso. Para ter mais informações, consulte Obter um ARN existente para o Amazon RDS.	23 de agosto de 2016
Novo recurso	Agora, você pode atribuir até 50 tags para cada recurso do Amazon RDS, para gerenciar seus recursos e os custos de rastreamento. Para ter mais informações, consulte Marcar recursos do Amazon RDS.	19 de agosto de 2016

Alteração	Descrição	Alterado em
Novo recurso	Agora o Amazon RDS oferece suporte para o modelo de Licença inclusa para o Oracle Standard Edition Two. Para ter mais informações, consulte Criar uma instância de banco de dados do Amazon RDS. Agora, você pode alterar o modelo de licença das suas instâncias de bancos de dados do Amazon RDS que executam o Microsoft SQL Server e Oracle. Para ter mais informações, consulte Licenciar o Microsoft SQL Server no Amazon RDS e Opções de licenciamento do RDS para Oracle.	5 de agosto de 2016
Novo recurso	Agora o Amazon RDS suporta backup e restauração nativos para bancos de dados do Microsoft SQL Server usando arquivos de backup completo (arquivos .bak). Agora você pode migrar facilmente bancos de dados SQL Server para o Amazon RDS e importar e exportar bancos de dados em um único arquivo facilmente portável, usando o Amazon S3 para armazenamento e o AWS KMS para criptografia. Para ter mais informações, consulte Importar e exportar bancos de dados do SQL Server usando backup e restauração nativos.	27 de julho de 2016
Novo recurso	Agora você pode copiar os arquivos de origem de um banco de dados MySQL para um bucket do Amazon Simple Storage Service (Amazon S3) e restaurar um cluster de bancos de dados Amazon Aurora a partir desses arquivos. Essa opção pode ser considerada mais rápida do que migrar dados usando `mysqldump`. Para ter mais informações, consulte Migração de dados de um banco de dados MySQL externo para um cluster de bancos de dados Aurora MySQL.	20 de julho de 2016

Alteração	Descrição	Alterado em
Novo recurso	Agora você pode restaurar um snapshot do cluster de banco de dados Amazon Aurora não criptografado para criar um cluster de banco de dados Amazon Aurora criptografado, incluindo uma chave de criptografia do AWS Key Management Service (AWS KMS) durante a operação de restauração. Para ter mais informações, consulte Criptografar recursos do Amazon RDS.	30 de junho de 2016
Novo recurso	Você pode usar o Oracle Repository Creation Utility (RCU) para criar um repositório no Amazon RDS para Oracle. Para ter mais informações, consulte Usar o Oracle Repository Creation Utility no RDS for Oracle .	17 de junho de 2016
Novo recurso	Adiciona suporte a réplicas de leitura entre regiões PostgreSQL. Para ter mais informações, consulte Criar uma réplica de leitura em uma Região da AWS diferente.	16 de junho de 2016
Novo recurso	Agora, você pode usar o AWS Management Console para adicionar facilmente o multi-AZ com espelhamento a uma instância de banco de dados Microsoft SQL Server. Para ter mais informações, consulte Adicionar Multi-AZ a uma instância de banco de dados do Microsoft SQL Server.	9 de junho de 2016
Novo recurso	Agora, você pode usar implantações multi-AZ com espelhamento do SQL Server nas seguintes regiões adicionais: Ásia-Pacífico (Sydney), Ásia-Pacífico (Tóquio) e América do Sul (São Paulo). Para ter mais informações, consulteImplantações multi-AZ para o Amazon RDS for Microsoft SQL Server	9 de junho de 2016
Novo recurso	Atualizado para dar suporte ao MariaDB versão 10.1. Para ter mais informações, consulte Amazon RDS para MariaDB.	1 de junho de 2016

Alteração	Descrição	Alterado em
Novo recurso	Atualizado para dar suporte a clusters de banco de dados entre regiões do Amazon Aurora que são réplicas de leitura. Para ter mais informações, consulte Como replicar clusters de bancos de dados Amazon Aurora MySQL entre regiões da AWS.	1 de junho de 2016
Novo recurso	O Monitoramento avançado e posterior para instâncias de banco de dados Oracle. Para ter mais informações, consulte Monitorar métricas do SO com o monitoramento avançado e Modificar uma instância de banco de dados do Amazon RDS.	27 de maio de 2016
Novo recurso	Atualizado para dar suporte ao compartilhamento manual de snapshots para snapshots de cluster de banco de dados Amazon Aurora. Para ter mais informações, consulte Compartilhamento de um snapshot do cluster de banco de dados.	18 de maio de 2016
Novo recurso	Agora, você pode usar o MariaDB Audit Plugin para registrar a atividade do banco de dados em instâncias de banco de dados MariaDB e MySQL. Para ter mais informações, consulte Opções para o mecanismo de banco de dados do MariaDB e Opções para instâncias de bancos de dados MySQL.	27 de abril de 2016
Novo recurso	Atualizações in-loco de versões principais agora estão disponíveis para atualização do MySQL versão 5.6 para a versão 5.7. Para ter mais informações, consulte Atualizações do mecanismo de banco de dados do RDS para MySQL.	26 de abril de 2016
Novo recurso	O Monitoramento avançado e posterior para instâncias de banco de dados Microsoft SQL Server. Para ter mais informações, consulte Monitorar métricas do SO com o monitoramento avançado.	22 de abril de 2016

Alteração	Descrição	Alterado em
Novo recurso	Atualizado para fornecer uma visualização Clusters do Amazon Aurora no console do Amazon RDS. Para ter mais informações, consulte Exibição de um cluster de bancos de dados Aurora.	1.º de abril de 2016
Novo recurso	Atualizado para dar suporte ao SQL Server multi-AZ com espelhamento na região Ásia-Pacífico (Seul). Para ter mais informações, consulte Implantações multi-AZ para o Amazon RDS for Microsoft SQL Server.	31 de março de 2016
Novo recurso	Atualizado para dar suporte ao Amazon Aurora multi-AZ com espelhamento na região Ásia-Pacífico (Seul). Para ter mais informações, consulte Disponibilidade para o Amazon Aurora MySQL.	31 de março de 2016
Novo recurso	As instâncias de bancos de dados PostgreSQL têm a capacidade de exigir que as conexões usem SSL. Para ter mais informações, consulte Usar o SSL com uma instância de banco de dados PostgreSQL.	25 de março de 2016
Novo recurso	O Monitoramento avançado agora está disponível para instâncias de banco de dados PostgreSQL. Para ter mais informações, consulte Monitorar métricas do SO com o monitoramento avançado.	25 de março de 2016
Novo recurso	Agora, instâncias de bancos de dados Microsoft SQL podem usar a Autenticação do Windows para a autenticação dos usuários. Para ter mais informaçõ es, consulte Trabalhar com o AWS Managed Active Directory com o RDS para SQL Server.	23 de março de 2016
Novo recurso	Agora, o Monitoramento avançado está disponíve l na região da Ásia-Pacífico (Seul). Para ter mais informações, consulte Monitorar métricas do SO com o monitoramento avançado.	16 de março de 2016

Alteração	Descrição	Alterado em
Novo recurso	Agora, você pode personalizar a ordem em que as Réplicas do Aurora são promovidas para instância primária durante um failover. Para ter mais informaçõ es, consulte Tolerância a falhas para um cluster de bancos de dados Aurora.	14 de março de 2016
Novo recurso	Atualizado para dar suporte à criptografia ao migrar para um cluster de banco de dados Aurora. Para ter mais informações, consulte Migração de dados para um cluster de bancos de dados Aurora.	2 de março de 2016
Novo recurso	Atualizado para dar suporte ao fuso horário local para clusters de banco de dados Aurora. Para ter mais informações, consulte Fuso horário local para clusters de bancos de dados Aurora.	1 de março de 2016
Novo recurso	Atualizado para adicionar suporte ao MySQL versão 5.7 para classes de instância de banco de dados do Amazon RDS de geração atual.	22 de fevereiro de 2016
Novo recurso	Atualizado para dar suporte às classes de instância de banco de dados db.r3 e db.t2 na região AWS GovCloud (Oeste dos EUA).	11 de fevereiro de 2016
Novo recurso	Atualizado para dar suporte à criptografia de cópias de snapshot de banco de dadoss e ao compartilhamento de snapshot de banco de dadoss criptografados. Para ter mais informações, consulte Copiar um snapshot de banco de dados para o Amazon RDS e Compartilhar um snapshot de banco de dados para o Amazon RDS.	11 de fevereiro de 2016
Novo recurso	Atualizado para dar suporte ao Amazon Aurora na região Ásia-Pacífico (Sydney). Para ter mais informaçõ es, consulte Disponibilidade para o Amazon Aurora MySQL.	11 de fevereiro de 2016

Alteração	Descrição	Alterado em
Novo recurso	Atualizado para dar suporte ao SSL para instâncias de banco de dados Oracle. Para ter mais informaçõ es, consulte Usar SSL com uma instância de banco de dados Oracle.	9 de fevereiro de 2016
Novo recurso	Atualizado para dar suporte ao fuso horário local para instâncias de bancos de dados MySQL e MariaDB. Para ter mais informações, consulte Fuso horário local para instâncias de banco de dados MySQL e Fuso horário local para as instâncias de banco de dados do MariaDB.	21 de dezembro de 2015
Novo recurso	Atualizado para dar suporte ao Monitoramento avançado de métricas do sistema operacional para instâncias MySQL e MariaDB e clusters de banco de dados Aurora. Para ter mais informações, consulte Visualizar métricas no console do Amazon RDS.	18 de dezembro de 2015
Novo recurso	Atualizado para dar suporte a classes de instância de banco de dados db.t2, db.r3 e db.m4 para o MySQL versão 5.5. Para ter mais informações, consulte Classes de instâncias de banco de dados .	4 de dezembro de 2015
Novo recurso	Atualizado para dar suporte à modificação da porta do banco de dados para uma instância de banco de dados existente.	3 de dezembro de 2015
Novo recurso	Atualizado para dar suporte a atualizações de versões principais do mecanismo de banco de dados para instâncias PostgreSQL. Para ter mais informações, consulte Atualizações do mecanismo de banco de dados do RDS para PostgreSQL.	19 de novembro de 2015

Alteração	Descrição	Alterado em
Novo recurso	Atualizado para dar suporte à modificação da acessibilidade pública de uma instância de banco de dados existente. Atualizado para dar suporte a classes de instância de banco de dados db.m4 padrão.	11 de novembro de 2015
Novo recurso	Atualizado para dar suporte ao compartilhamento manual de snapshot de banco de dadoss. Para ter mais informações, consulte Compartilhar um snapshot de banco de dados para o Amazon RDS.	28 de outubro de 2015
Novo recurso	Atualizado para dar suporte ao Microsoft SQL Server 2014 para as edições Web, Express e Standard.	26 de outubro de 2015
Novo recurso	Atualizado para dar suporte ao mecanismo de banco de dados MariaDB baseado em MySQL. Para ter mais informações, consulte Amazon RDS para MariaDB.	7 de outubro de 2015
Novo recurso	Atualizado para dar suporte ao Amazon Aurora na região Ásia-Pacífico (Tóquio). Para ter mais informações, consulte Disponibilidade para o Amazon Aurora MySQL.	7 de outubro de 2015
Novo recurso	Atualizado para dar suporte a classes de instâncias de bancos de dados db.t2 com capacidade de intermitência para todos os mecanismos de bancos de dados e à adição da classe de instância de banco de dados db.t2.large. Para ter mais informações, consulte Classes de instâncias de banco de dados .	25 de setembro de 2015
Novo recurso	Atualizado para dar suporte a instância de banco de dados Oracle nas classes de instâncias de bancos de dados R3 e T2. Para ter mais informações, consulte Classes de instâncias de banco de dados .	5 de agosto de 2015

Alteração	Descrição	Alterado em
Novo recurso	O Microsoft SQL Server Enterprise Edition agora está disponível com o modelo de serviço de licença inclusa. Para ter mais informações, consulte Licenciar o Microsoft SQL Server no Amazon RDS.	29 de julho de 2015
Novo recurso	O Amazon Aurora foi oficialmente lançado. O mecanismo de banco de dados do Amazon Aurora oferece suporte a várias instâncias de bancos de dados em um cluster de banco de dados. Para obter informações detalhadas, consulte O que é o Amazon Aurora?.	27 de julho de 2015
Novo recurso	Atualizado para dar suporte à cópia de tags para snapshot de banco de dadoss.	20 de julho de 2015
Novo recurso	Atualizado para dar suporte a aumentos no tamanho de armazenamento para todos os mecanismos de banco de dados e a um aumento nas IOPS provision adas para o SQL Server.	18 de junho de 2015
Novo recurso	Opções atualizadas para instâncias de bancos de dados reservadas.	15 de junho de 2015
Novo recurso	Atualizado para dar suporte ao uso do Amazon CloudHSM com instâncias de bancos de dados Oracle usando o TDE.	8 de janeiro de 2015
Novo recurso	Atualizado para dar suporte à criptografia de dados em repouso e à nova versão da API 2014-10-31.	6 de janeiro de 2015

Alteração	Descrição	Alterado em
Novo recurso	Atualizado para incluir o novo mecanismo de banco de dados Amazon: o Aurora. O mecanismo de banco de dados do Amazon Aurora oferece suporte a várias instâncias de bancos de dados em um cluster de banco de dados. O Amazon Aurora atualmente está na versão de release de pré-visualização e sujeito a alterações. Para obter informações detalhadas, consulte O que é o Amazon Aurora?.	12 de novembro de 2014
Novo recurso	Atualizado para dar suporte a réplicas de leitura PostgreSQL.	10 de novembro de 2014
Nova API e recursos	Atualizado para dar suporte ao tipo de armazenam ento GP2 e à nova versão da API 2014-09-01. Atualizado para dar suporte à capacidade de copiar uma opção ou um grupo de parâmetros existente para criar uma nova opção ou grupo de parâmetros.	7 de outubro de 2014
Novo recurso	Atualizado para dar suporte ao Aquecimento de Cache do InnoDB para instâncias de bancos de dados que executam o MySQL versão 5.6.19 e posterior.	3 de setembro de 2014
Novo recurso	Atualizado para dar suporte à verificação de certifica dos SSL ao conectar-se aos mecanismos de banco de dados MySQL versão 5.6, SQL Server e PostgreSQL.	5 de agosto de 2014
Novo recurso	Atualizado para oferecer suporte às classes de instâncias de banco de dados expansíveis db.t2.	4 de agosto de 2014
Novo recurso	Atualizado para oferecer suporte às classes de instância de banco de dados com otimização de memória db.r3 para uso com os mecanismos de banco de dados MySQL (versão 5.6), SQL Server e PostgreSQL.	28 de maio de 2014

Alteração	Descrição	Alterado em
Novo recurso	Atualizado para dar suporte a implantações multi-AZ do SQL Server usando o Espelhamento do SQL Server.	19 de maio de 2014
Novo recurso	Atualizado para dar suporte a atualizações do MySQL versão 5.5 para a versão 5.6.	23 de abril de 2014
Novo recurso	Atualizado para dar suporte ao Oracle GoldenGate.	3 de abril de 2014
Novo recurso	Atualizado para dar suporte a classes de instância de banco de dados M3.	20 de fevereiro de 2014
Novo recurso	Atualizado para dar suporte à opção de fuso horário Oracle.	13 de janeiro de 2014
Novo recurso	Atualizado para dar suporte à replicação entre instâncias de bancos de dados MySQL em diferentes regiões.	26 de novembro de 2013
Novo recurso	Atualizado para dar suporte ao mecanismo de banco de dados do PostgreSQL.	14 de novembro de 2013
Novo recurso	Atualizado para dar suporte ao SQL Server Transparent Data Encryption (TDE).	7 de novembro de 2013
Nova API e novo recurso	Atualizado para dar suporte a cópias de snapshot de banco de dados entre regiões; nova versão da API, 2013-09-09.	31 de outubro de 2013
Novos recursos	Atualizado para dar suporte ao Oracle Statspack.	26 de setembro de 2013
Novos recursos	Atualizado para dar suporte ao uso da replicação para importar ou exportar dados entre instâncias do MySQL em execução no Amazon RDS e instâncias do MySQL em execução no local ou no Amazon EC2.	5 de setembro de 2013

Alteração	Descrição	Alterado em
Novos recursos	Atualizado para dar suporte à classe de instância de banco de dados db.cr1.8xlarge para o MySQL 5.6.	4 de setembro de 2013
Novo recurso	Atualizado para dar suporte à replicação de réplicas de leitura.	28 de agosto de 2013
Novo recurso	Atualizado para dar suporte à criação de réplica de leitura em paralelo.	22 de julho de 2013
Novo recurso	Atualizado para dar suporte a permissões granulares e à marcação para todos os recursos do Amazon RDS.	8 de julho de 2013
Novo recurso	Atualizado para dar suporte ao MySQL 5.6 para novas instâncias, incluindo suporte para a interface MySQL 5.6 memcached e acesso a logs binários.	1 de julho de 2013
Novo recurso	Atualizado para dar suporte a atualizações de versões principais do MySQL 5.1 para o MySQL 5.5.	20 de junho de 2013
Novo recurso	Grupos de parâmetros de banco de dados atualizados para permitir expressões para valores de parâmetros.	20 de junho de 2013
Nova API e novo recurso	Atualizado para dar suporte ao status de réplica de leitura; nova versão da API, 2013-05-15.	23 de maio de 2013
Novos recursos	Atualizado para oferecer suporte aos recursos do Oracle Advanced Security para criptografia de rede nativa e ao Oracle Transparent Data Encryption.	18 de abril de 2013
Novos recursos	Atualizado para dar suporte a atualizações de versões principais para o SQL Server e funcionalidades adicionais para IOPS provisionadas.	13 de março de 2013
Novo recurso	Atualizado para dar suporte à VPC por padrão para o RDS.	11 de março de 2013

Alteração	Descrição	Alterado em
Nova API e recurso	Atualizado para dar suporte ao acesso a logs; nova versão da API 2013-02-12	4 de março de 2013
Novo recurso	Atualizado para dar suporte a assinaturas de notificação de eventos do RDS.	4 de fevereiro de 2013
Nova API e recurso	Atualizado para dar suporte à renomeação de instâncias de banco de dados e à migração de membros do grupo de segurança de banco de dados em uma VPC para um grupo de segurança de VPC.	14 de janeiro de 2013
Novo recurso	Atualizado para suporte à região AWS GovCloud (Oeste dos EUA).	17 de dezembro de 2012
Novo recurso	Atualizado para dar suporte às classes de instância de banco de dados m1.medium e m1.xlarge.	6 de novembro de 2012
Novo recurso	Atualizado para dar suporte à promoção de réplicas de leitura.	11 de outubro de 2012
Novo recurso	Atualizado para dar suporte ao SSL em instâncias de bancos de dados do Microsoft SQL Server.	10 de outubro de 2012
Novo recurso	Atualizado para dar suporte a instâncias de bancos de dados Oracle.	27 de setembro de 2012
Novo recurso	Atualizado para dar suporte ao SQL Server 2012.	26 de setembro de 2012
Nova API e recurso	Atualizado para oferecer suporte a IOPS provisionadas. Versão da API 2012-09-17.	25 de setembro de 2012
Novos recursos	Atualizado para suporte ao SQL Server para instâncias de bancos de dados na VPC e para suporte ao Oracle para Data Pump.	13 de setembro de 2012

Alteração	Descrição	Alterado em
Novo recurso	Atualizado para suporte ao SQL Server Agent.	22 de agosto de 2012
Novo recurso	Atualizado para suporte à marcação de instâncias de bancos de dados.	21 de agosto de 2012
Novos recursos	Atualizado para suporte ao Oracle APEX e ao XML DB, a fusos horários Oracle e a instâncias de banco de dados Oracle em uma VPC.	16 de agosto de 2012
Novos recursos	Atualizado para suporte ao Orientador de Otimização do Mecanismo de Banco de Dados e a instâncias de banco de dados Oracle na VPC.	18 de julho de 2012
Novo recurso	Atualizado para suporte a grupos de opções e à primeira opção, o Oracle Enterprise Manager Database Control.	29 de maio de 2012
Novo recurso	Atualizado para suporte a réplicas de leitura na Amazon Virtual Private Cloud.	17 de maio de 2012
Novo recurso	Atualizado para suporte ao Microsoft SQL Server.	8 de maio de 2012
Novos recursos	Atualizado para suporte para failover forçado, implantação multi-AZ de instâncias de banco de dados Oracle e conjuntos de caracteres não padrão para instâncias de banco de dados Oracle.	2 de maio de 2012
Novo recurso	Atualizado para suporte à Amazon Virtual Private Cloud (VPC).	13 de fevereiro de 2012
Conteúdo atualizado	Atualizado para novos tipos de instâncias reservadas.	19 de dezembro de 2011
Novo recurso	Atualizado para suporte ao mecanismo Oracle.	23 de maio de 2011

Alteração	Descrição	Alterado em
Conteúdo atualizado	Atualizações do console.	13 de maio de 2011
Conteúdo atualizado	Conteúdo editado para janelas de backup e manutenção mais curtas.	28 de fevereiro de 2011
Novo recurso	Suporte adicionado para MySQL 5.5.	31 de janeiro de 2011
Novo recurso	Suporte adicionado para réplicas de leitura.	4 de outubro de 2010
Novo recurso	Adicionado suporte para AWS Identity and Access Management (IAM).	2 de setembro de 2010
Novo recurso	Adição do gerenciamento de versões de mecanismos de banco de dados.	16 de agosto de 2010
Novo recurso	Adição de instâncias de bancos de dados reservadas	16 de agosto de 2010
Novo recurso	Agora o Amazon RDS comporta conexões SSL com instâncias de banco de dados.	28 de junho de 2010
Novo guia	Esta é a primeira versão do Guia do usuário da Amazon RDS.	7 de junho de 2010

Glossário da AWS

Para obter a terminologia mais recente da AWS, consulte o glossário da AWS na Referência do Glossário da AWS.